Aurona Gerber
Jacques Maritz
Anban W. Pillay (Eds.)

Communications in Computer and Information Science 2326

Artificial Intelligence Research

5th Southern African Conference, SACAIR 2024
Bloemfontein, South Africa, December 2–6, 2024
Proceedings

Springer

Communications in Computer and Information Science 2326

Series Editors

Gang Li, *School of Information Technology, Deakin University, Burwood, VIC, Australia*
Joaquim Filipe, *Polytechnic Institute of Setúbal, Setúbal, Portugal*
Ashish Ghosh, *Indian Statistical Institute, Kolkata, West Bengal, India*
Zhiwei Xu, *Chinese Academy of Sciences, Beijing, China*

Rationale
The CCIS series is devoted to the publication of proceedings of computer science conferences. Its aim is to efficiently disseminate original research results in informatics in printed and electronic form. While the focus is on publication of peer-reviewed full papers presenting mature work, inclusion of reviewed short papers reporting on work in progress is welcome, too. Besides globally relevant meetings with internationally representative program committees guaranteeing a strict peer-reviewing and paper selection process, conferences run by societies or of high regional or national relevance are also considered for publication.

Topics
The topical scope of CCIS spans the entire spectrum of informatics ranging from foundational topics in the theory of computing to information and communications science and technology and a broad variety of interdisciplinary application fields.

Information for Volume Editors and Authors
Publication in CCIS is free of charge. No royalties are paid, however, we offer registered conference participants temporary free access to the online version of the conference proceedings on SpringerLink (http://link.springer.com) by means of an http referrer from the conference website and/or a number of complimentary printed copies, as specified in the official acceptance email of the event.

CCIS proceedings can be published in time for distribution at conferences or as post-proceedings, and delivered in the form of printed books and/or electronically as USBs and/or e-content licenses for accessing proceedings at SpringerLink. Furthermore, CCIS proceedings are included in the CCIS electronic book series hosted in the SpringerLink digital library at http://link.springer.com/bookseries/7899. Conferences publishing in CCIS are allowed to use Online Conference Service (OCS) for managing the whole proceedings lifecycle (from submission and reviewing to preparing for publication) free of charge.

Publication process
The language of publication is exclusively English. Authors publishing in CCIS have to sign the Springer CCIS copyright transfer form, however, they are free to use their material published in CCIS for substantially changed, more elaborate subsequent publications elsewhere. For the preparation of the camera-ready papers/files, authors have to strictly adhere to the Springer CCIS Authors' Instructions and are strongly encouraged to use the CCIS LaTeX style files or templates.

Abstracting/Indexing
CCIS is abstracted/indexed in DBLP, Google Scholar, EI-Compendex, Mathematical Reviews, SCImago, Scopus. CCIS volumes are also submitted for the inclusion in ISI Proceedings.

How to start
To start the evaluation of your proposal for inclusion in the CCIS series, please send an e-mail to ccis@springer.com.

Aurona Gerber · Jacques Maritz ·
Anban W. Pillay
Editors

Artificial Intelligence Research

5th Southern African Conference, SACAIR 2024
Bloemfontein, South Africa, December 2–6, 2024
Proceedings

Springer

Editors
Aurona Gerber
University of the Western Cape
Cape Town, South Africa

Jacques Maritz
University of the Free State
Bloemfontein, South Africa

Anban W. Pillay
University of KwaZulu-Natal
Durban, South Africa

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-3-031-78254-1 ISBN 978-3-031-78255-8 (eBook)
https://doi.org/10.1007/978-3-031-78255-8

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2025

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

If disposing of this product, please recycle the paper.

Preface

This volume of Springer CCIS (CCIS 2326) contains the revised accepted papers of SACAIR 2024, the *5th Southern African Conference for Artificial Intelligence Research*[1].

Foreword from the Conference Chairs

Dear authors and readers,

It is with great pleasure that we write this foreword to the proceedings of the Fifth Southern African Conference for Artificial Intelligence Research (SACAIR 2024). The conference was hosted by University of the Free State, Bloemfontein, South Africa from 2 to 6 December 2024. The programme included an unconference for students on 2 December 2024 (a student-driven event for students to interact with each other, as well as with sponsors and other possible employers), a day of tutorials on 3 December, and the main conference from 4-6 December 2021.

SACAIR 2024 was the fifth international conference focussed on Artificial Intelligence organized under the auspices of the SACAIR Steering Committee, an affiliate of the Centre for AI Research (CAIR), South Africa. The Centre for AI Research (CAIR)[2] is a South African distributed research network that was established in 2011 with the aim of building world-class Artificial Intelligence research capacity in Southern Africa.

Although still a young conference, SACAIR is quickly establishing itself as a premier artificial intelligence conference in the Southern African region. The fifth conference built on the success of previous conferences. The inaugural CAIR conference, the Forum for AI Research (FAIR 2019), was held in Cape Town, South Africa, in December 2019. SACAIR 2020 was held in February 2021 after being postponed due to the Covid pandemic and SACAIR 2021 was an online event hosted by the University of KwaZulu-Natal in December 2021. The 2022 conference edition was held in Stellenbosch, Western Cape, and SACAIR 2023 was held in Gauteng, South Africa.

The conference, held under the theme of *Artificial Intelligence for Societal Impact*, brought together a diverse group of researchers and practitioners. The role of AI in shaping future work and societies cannot be overstated. While significant strides have been made in offering targeted AI solutions, it is increasingly presenting significant challenges to civil society. The depth and reach of AI-driven technologies has surprised many and necessitates a conversation between many disciplines. SACAIR has always been, and will continue to be, an inter- and trans-disciplinary conference and invites research from diverse disciplines that include Law and Philosophy, Logic, Information Systems and Information Technology, Computer Science and Engineering. It is hoped that this will lead researchers to consider their work in the wider academic community and lead

[1] https://sacair.org.za/.

[2] https://www.cair.org.za/.

to advancements of responsible AI research, development of tangible outcomes in our immediate contexts, and sensitivity to societal hardships that persist due to the complex interplay of socio-political, economic, historical, and environmental realities. The conference aims to explore the actual societal impact delivered by artificial intelligence beyond its potential promises, and from all scientific perspectives.

We are pleased that SACAIR 2024 continued to enjoy the support of the South African artificial intelligence research community. The conference attracted support from both authors, who submitted high-quality research papers, as well as researchers who supported the conference by serving on the international program committee. SACAIR 2024 brought together a diverse group of researchers and practitioners in the fields of Computer Science, Information Systems, Knowledge Representation and Reasoning, Law, and Philosophy of AI. The conference was organized as a multi-track conference that covered broad areas of Artificial Intelligence, namely:

- Algorithmic and Data-Driven AI (Computer Science).
- Socio-technical and Human-Centred AI (Information Systems).
- Responsible and Ethical AI (Philosophy, Law and Humanities).
- Knowledge Representation and Reasoning (Symbolic AI).

The accepted papers include contributions from a diverse and thriving artificial intelligence research community within Southern Africa. The conference has a central development agenda and is also focused on cultivating and establishing a network of talented students and emerging researchers and academics working in AI from across Africa.

A conference of this nature is not possible without the contributions of many stakeholders. We extend our sincere gratitude to our sponsors: the Artificial Intelligence Journal (AIJ), the National Institute of Computational Sciences (NiTHeCS), the Centre for Artificial Intelligence Research (CAIR) and the University of the Free State. These sponsors have made it possible to offer generous scholarships to students and emerging academics to participate in the conference. We sincerely thank the technical chairs for their work in overseeing the technical aspects of the conference and the publication of the two volumes of the proceedings, the international panel of reviewers, our keynotes, authors, and participants for their contributions. Finally, we extend our gratitude to the track chairs, the local organising committee, the student organisers, and the conference organiser for their substantive contributions to the success of SACAIR 2024.

December 2024

Katinka de Wet
Herkulaas Combrink

Message from the Technical Chairs

Dear readers,

This volume, volume 2326 of the Springer Communications in Computer and Information Science (CCIS) series, contains the revised accepted papers of the South African Conference for Artificial Intelligence, 2024 (SACAIR 2024). The second SACAIR 2024 proceedings volume is published as an online proceedings that will be available on the SACAIR 2024 website.

The inter- and trans-disciplinary nature of the SACAIR series of conferences in Artificial Intelligence is unique in that it provides a venue for researchers from a diverse set of disciplines that include Computer Science, Engineering, Information Systems, Law, Philosophy, and the Humanities. The organization of such a conference has to carefully consider the differing research methods, interests, publication standards, and cultures of these disciplines. The conference was thus organized around the following tracks:

– Algorithmic and Data-Driven AI (Computer Science):

- Deep Learning and Machine Learning.
- Computer Vision and Image Processing.
- Pattern Recognition.
- Evolutionary Computing.
- Biometrics and Cybersecurity in AI.
- Explainable AI.
- Multi-agent Systems.
- Agent-Based Modelling.
- Quantum Artificial Intelligence.

– Socio-technical and Human-Centred AI (Information Systems):

- AI Information Systems.
- AI Socio-technical Systems.
- Human-Centred Artificial Intelligence.
- AI for and in Business, including AI Adoption.
- AI Supporting Sustainable Development and the Circular Economy.
- AI for Development and Social Good.

– Responsible and Ethical AI (Philosophy, Law and Humanities):

- Data Ethics.
- Machine Ethics.
- Ethics of Socio-robotics.
- Neuro Ethics.
- AI and the Law.
- Responsible AI Governance.

– Knowledge Representation and Reasoning (Symbolic AI):

- Knowledge Representation and Reasoning.
- Ontologies.
- Computational Logic.

The program committee comprised 156 members (representing some 73 research institutions), 29 of whom were from outside Southern Africa. Each paper was sent for review to at least three members of the program committee in a rigorous, double-blind peer-review process. Most papers received at least three reviews, often followed by a meta-review by the respective track chairs. Great care was taken to ensure the conference's integrity, including careful attention to avoid conflicts of interest. The following criteria were used to rate submissions and to guide decisions: relevance to SACAIR, significance, technical quality, scholarship, and presentation (including quality and clarity of writing).

We received 153 abstracts, and after submission and a first round of evaluation, 101 submissions were sent to our SACAIR program committee for review. The papers consisted of 70 in the CS track, 12 in the IS track, 5 in the KRR track and 14 in the P&L track. Twenty-nine full research papers were selected for publication in this Springer CCIS volume (which translates to an acceptance rate of 29%), whilst a further 21 papers were accepted for inclusion in the online volume (21%). The total acceptance rate for publication in the two SACAIR 2024 proceedings volumes was 50% for reviewed submissions. In total, two papers from the Responsible and Ethical AI track, three papers from the Knowledge Representation and Reasoning track, four papers from the Socio-technical and Human-Centred AI track, and 20 papers from the Algorithmic and Data-Driven AI track were accepted for publication in this CCIS volume.

Thank you to all the authors who submitted work of an exceptional standard to the conference and congratulations to the authors whose work was accepted for publication. We place on record our gratitude to the Program Committee members, whose thoughtful and constructive comments were well received by the authors.

December 2024

Aurona Gerber
Jacques Maritz
Anban W. Pillay

Organization



Local Organising Committee

Katinka de Wet (General Co-chair)	University of the Free State, South Africa
Herkulaas Combrink (General Co-chair)	University of the Free State, South Africa
Renee le Roux (Conference Organiser)	Mongoose Communications & Design, South Africa

Technical Committee

Susan Brokensha	University of the Free State, South Africa
Aurona Gerber	University of the Western Cape, South Africa
Anban Pillay	University of KwaZulu-Natal, South Africa
Jacques Maritz	University of the Free State, South Africa
Wynand Nel	University of the Free State, South Africa

Programme Committee

Algorithmic and Data-Driven AI (Computer Science)

Track Chairs

Akinyelu, Andronicus	University of the Free State, South Africa
Davel, Marelie	North-West University, South Africa
Gerber, Aurona	University of the Western Cape, South Africa
Jembere, Edgar	University of KwaZulu-Natal, South Africa
Nyirenda, Clement	University of the Western Cape, South Africa
Pillay, Anban	University of KwaZulu-Natal, South Africa
Van Zyl, Terence	University of Johannesburg, South Africa

PC Members

Abate, Demmelash	Wolaita Sodo University, Ethiopia
Achary, Thimershen	Independent, South Africa
Agbobli, Gabriel	University of Ghana, Ghana
Amankwatia, Manuel	Kwame Nkrumah University of Science and Technology, Ghana
Amuda, Kamorudeen	University of Ibadan, Nigeria
Amukwatse, Jean	Uganda Technology and Management University, Uganda
Anderson, George	University of Botswana, Botswana
Asaju, Christine	Thswane University of Technology, South Africa
Atemkeng, Marcellin	Rhodes University, South Africa
Barnard, Etienne	North-West University, South Africa
Batani, John	University of KwaZulu-Natal, South Africa
Bopape, Lebogang	Durban University of Technology, South Africa
Bosman, Anna Sergeevna	University of Pretoria, South Africa
Brink, Willie	Stellenbosch University, South Africa
Brown, Dane	Rhodes University, South Africa
Buys, Jan	University of Cape Town, South Africa
Chilipa, Richard	Malawi University of Science and Technology, Malawi
Daramola, Olawande	University of Pretoria, South Africa
De Freitas, Allan	University of Pretoria, South Africa
Derks, Iena	University of Pretoria, South Africa
Dlamini, Johnson	University of Pretoria, South Africa
Dlamini, Sibonelo	University of KwaZulu-Natal, South Africa
Du Toit, Tiny	North-West University, South Africa
Eiselen, Roald	North-West University, South Africa
Engelbrecht, Emile	Stellenbosch University, South Africa
Ezeji, Ijeoma	University of Zululand, South Africa
Ezugwu, Absalom	North-West University, South Africa
Fabris-Rotelli, Inger	University of Pretoria, South Africa
Fashoto, Stephen	University of Swaziland, Eswatini
Fendji, Louis	University of Ngaoundere, Cameroon
Formanek, Claude	University of Cape Town, South Africa
Harley, Charis	University of Johannesburg, South Africa
Haskins, Bertram	Nelson Mandela University, South Africa
Helberg, Albert	North-West University, South Africa
Hounkanrin, Anicet	University of Cape Town, South Africa
Isingizwe, Frederic	University of the Western Cape, South Africa
James, Steven	University of the Witwatersrand, South Africa

Jeewa, Asad	University of KwaZulu-Natal, South Africa
Kankuzi, Bennett	Malawi University of Science and Technology, Malawi
Karim, Zainoolabadien	University of the Witwatersrand, South Africa
Khangamwa, Gift	University of the Witwatersrand, South Africa
Klein, Richard	University of the Witwatersrand, South Africa
Kogeda, Paul Okuthe	University of KwaZulu-Natal, South Africa
Kotze, Michelle	University of Pretoria, South Africa
Kotzé, Eduan	University of the Free State, South Africa
Laidi, Amel	Abou Bekr Belkaid University, Algeria
Louw, Aby	Council for Scientific and Industrial Research, South Africa
Makhoere, Lucky	Sol Plaatje University, South Africa
Malan, Katherine	University of South Africa, South Africa
Marais, Patrick	University of Cape Town, South Africa
Masethe, Dan	Tshwane University of Technology, South Africa
Mathebula, Miehleketo	University of Pretoria, South Africa
Mathonsi, Topside	Tshwane University of Technology, South Africa
Mauwa, Hope	University of Mpumalanga, South Africa
Melatagia Yonta, Paulin	University of Yaounde, Cameroon
Modipa, Thipe	University of Limpopo, South Africa
Modupe, Abiodun	University of Pretoria, South Africa
Mogale, Hope	North-West University, South Africa
Moodley, Deshendran	University of Cape Town, South Africa
Moore, Stephen	University of Cape Coast, Ghana
Motsoehli, Moseli	University of Hawaii at Manoa, USA
Mthethwa, Sthembile	Council for Scientific and Industrial Research, South Africa
Muite, Benson	University of Tartu, Estonia
Mulaudzi, Rudzani	University of the Witwatersrand, South Africa
Mungloo-Dilmohamud, Zahra	University of Mauritius, Mauritius
Mutanga, Murimo Bethel	Mangosuthu University of Technology, South Africa
Mutembei, Leonard	University of Johannesburg, South Africa
Nandutu, Irene	Uganda Technology and Management University, Uganda
Nemakhavhani, Ndamulelo	University of Pretoria, South Africa
Ngxande, Mkhuseli	Stellenbosch University, South Africa
Nicolls, Fred	University of Cape Town, South Africa
Nogwina, Mnoneleli	Sol Plaatje University, South Africa
Nzomo, Mbithe	University of Cape Town, South Africa
Okuboyejo, Damilola	Durban University of Technology, South Africa

Olusanya, Michael	Sol Plaatje University, South Africa
Oluyide, Oluwakorede Monica	University of the Western Cape, South Africa
Onuegbu, Joseph	Ariel University, Israel
Otoo, Joseph	C. K. Tandem University of Technology and Applied Sciences, Ghana
Rananga, Seani	University of Pretoria, South Africa
Schlippe, Tim	IU International University of Applied Sciences, Germany
Sefara, Tshephisho Joseph	University of Limpopo, South Africa
Semwayo, Thembinkosi	University of the Witwatersrand, South Africa
Senekane, Makhamisa	University of Johannesburg, South Africa
Shiabaw, Dawit	University of Gondar, Ethiopia
Shock, Jonathan	University of Cape Town, South Africa
Singh, Yuvika	University of KwaZulu-Natal, South Africa
Tachiona, Terence	Midlands State University, Zimbabwe
Tapamo, Jules-Raymond	Univesity of KwaZulu-Natal, South Africa
Tom, Sheethal	Cape Peninsula University of Technology, South Africa
Tuyishimire, Emmanuel	North-West University, South Africa
Vadapalli, Hima Bindu	University of Johannesburg, South Africa
Wang, Hairong	University of the Witwatersrand, South Africa
Watson, Bruce	National Security Centre of Excellence, Canada
Wegner, Jan Dirk	University of Switzerland, Switzerland
Yibre, Abdulkerim Mohammed	Bahir Dar Institute of Technology, Bahir Dar University, Ethiopia

Socio-technical and Human-Centred AI (Information Systems)

Track Chairs

Eybers, Sunet	University of South Africa, South Africa
Hinkelmann, Knut	FHNW University of Applied Sciences and Arts, Northwestern Switzerland

PC Members

Abdullah, Hanifa	University of South Africa, South Africa
Botha, Benjamin	University of the Free State, South Africa
Callisto De Donato, Massimo	University of Camerino, Italy
Fornari, Fabrizio	University of Camerino, Italy
Hanslo, Ridewaan	University of Johannesburg, South Africa

Howard, Grant Royd	University of South Africa, South Africa
Härer, Felix	University of Fribourg, Switzerland
Laurenzi, Emanuele	FHNW University of Applied Sciences and Arts, Northwestern Switzerland
Mahlaza, Zola	University of Cape Town, South Africa
Martin, Andreas	FHNW University of Applied Sciences and Arts, Northwestern Switzerland
Matthee, Machdel	University of Pretoria, South Africa
Parry, Douglas	Stellenbosch University, South Africa
Piangerelli, Marco	University of Camerino, Italy
Pretorius, Henk	University of Pretoria, South Africa
Price, Catherine S.	University of KwaZulu-Natal, South Africa
Smit, Danie	University of Pretoria, South Africa
Smuts, Hanlie	University of Pretoria, South Africa
Van Belle, Jean-Paul	University of Cape Town, South Africa
Van Staden, Corné	University of South Africa, South Africa

Responsible and Ethical AI (Philosophy, Law and Humanities)

Track Chairs

Ruttkamp-Bloem, Emma	University of Pretoria, South Africa
Tollon, Fabio	University of Edinburgh, UK

PC Members

Basu, Kaberi	Trinity College Dublin, Ireland
Benhammou, Morraya	The Hague University of Applied Sciences, Netherlands
Benton, Paige	University of Pretoria, South Africa
Bock, Ivan	Stellenbosch University, South Africa
Claassen, Kristy	University of Twente, Netherlands
Formánek, Arzu	University of Vienna, Austria
Friedman, Cindy	Utrecht University, Netherlands
Garforth, James	University of Edinburgh, UK
Robertson, Helen Sarah	University of the Witwatersrand, South Africa
Schulz, Maria	Tilburg University, Germany
Vorster, Ethan	University of Johannesburg, South Africa
Xivuri, Khensani	University of Johannesburg, South Africa
de Villiers-Botha, Tanya	Stellenbosch University, South Africa

Symbolic AI (Knowledge Representation and Reasoning)

Track Chairs

Casini, Giovanni	Institute of Information Science and Technologies, National Research Council of Italy, Italy
Heyninck, Jesse	Open Universiteit, Netherlands

PC Members

Booth, Richard	Cardiff University, UK
Haldimann, Jonas Philipp	Vienna University of Technology, Austria
Giordano, Laura	Università del Piemonte Orientale, Italy
Leenen, Louise	University of the Western Cape, South Africa
Meyer, Thomas	University of Cape Town, South Africa
Pino Perez, Ramon	Université d'Artois, France
Rens, Gavin	Stellenbosch University, South Africa

SACAIR 2024 Sponsors

The sponsors of SACAIR 2024, *The Journal of Artificial Intelligence*, the *Centre for AI Research (CAIR)*, the *National Institute for Theoretical and Computational Sciences (NITheCS)* and the *University of the Free State* are herewith gratefully acknowledged. Their generous contributions allowed us to support many students and emerging researchers.









Contents

Algorithmic and Data Driven AI

Parameter-Efficient Fine-Tuning of Pre-trained Large Language Models for Financial Text Analysis 3
Kelly Langa, Hairong Wang, and Olaperi Okuboyejo

Optimally Traversing Explainability in Bayesian Networks via the Graphical Lasso 21
Iena Petronella Derks, Alta de Waal, Jarod Smith, Theodor Loots, and Jean-Pierre Stander

Assessing Multilinguality of Topic Models on a Short-Text South African Languages Dataset 38
Darren Craig Roos and Katherine Mary Malan

Cascaded RFM-Based Fuzzy Clustering Model for Dynamic Customer Segmentation in the Retail Sector 53
Sive Sobantu and Omowunmi E. Isafiade

Automatic Assessment of Speech Impediment for South African Early Literacy Readers 74
Jaco Badenhorst

Automated Enhancement of isiZulu Data Collection for the African Health Research Institute 91
Jaco Badenhorst and Avashna Govender

AI in Education: An Analysis of Large Language Models for Twi Automatic Short Answer Grading 107
Alex Agyemang and Tim Schlippe

Enhancing Credit Risk Assessment Through Transformer-Based Machine Learning Models 124
Elekanyani Siphuma and Terence van Zyl

Predicting and Discovering Weather Patterns in South Africa Using Spatial-Temporal Graph Neural Networks 144
Adeeb Gaibie, Hamza Amir, Irene Nandutu, and Deshendran Moodley

Benchmarking Political Bias Classification with In-Context Learning: Insights from GPT-3.5, GPT-4o, LLaMA-3, and Gemma-2 161
Eduan Kotzé and Burgert A. Senekal

Uncovering the ANPR Performance Gap: A Commercial Systems Analysis 176
Frank Zandamela, Rofhiwa Seletani, Patrick Malatjie, Dumisani Kunene, Muhammad Ahmed Nana, and Teboho Sekopa

Neural Network-Based Vehicular Channel Estimation Performance: Effect of Noise in the Training Set 192
Simbarashe Aldrin Ngorima, Albert S. J. Helberg, and Marelie H. Davel

Deep Learning-Based Network Intrusion Detection Systems: A Systematic Literature Review 207
Leonard L. Mutembei, Makhamisa C. Senekane, and Terence van Zyl

Impact of Batch Normalization on Convolutional Network Representations 235
Hermanus L. Potgieter, Coenraad Mouton, and Marelie H. Davel

Does Simple Trump Complex? Comparing Strategies for Adversarial Robustness in DNNs 253
William Brooks, Marelie H. Davel, and Coenraad Mouton

A Novel Approach To Lion Re-Identification Using Vision Transformers 270
Boitumelo Matlala, Dustin van der Haar, and Hima Vandapalli

Assessing Data-Driven of Discriminative Deep Learning Models in Classification Task Using Synthetic Pandemic Dataset 282
Sunday Adeola Ajagbe, Pragasen Mudali, and Matthew O. Adigun

Deep Neural Network Compression for Lightweight and Accurate Fish Classification 300
Daanyaal Salie, Dane Brown, and Kenneth Chieza

Pre-training a Transformer-Based Generative Model Using a Small Sepedi Dataset 319
Simon Phetole Ramalepe, Thipe I. Modipa, and Marelie H. Davel

Automated Fish Detection in Underwater Environments: Performance Analysis of YOLOv8 and YOLO-NAS 334
Kenneth Chieza, Dane Brown, James Connan, and Daanyaal Salie

Socio-Technical and Human-Centred AI (Information Systems)

The Effect of Generative Artificial Intelligence on Cognitive Thinking Skills in Higher Education Institutions: A Systematic Literature Review 355
Harry Moongela, Machdel Matthee, Marita Turpin, and Alta van der Merwe

Large Language Models for Democratizing Business Process Modeling: BPMN Model Generation and Style Guide Adherence 372
Seline Wenger, Maja Spahic-Bogdanovic, and Andreas Martin

The Role of AI in Modern Language Translation and Its Societal Applications: A Systematic Literature Review 390
Samuel Ssemugabi

Deep Learning Forecasting of Photovoltaics Output Using Digital Twin Data 405
Nomfundo Vilakazi and Terence van Zyl

Responsible and Ethical AI (Philosophy, Law and Humanities)

Requirements Engineering (RE) in Artificial Intelligence (AI) Systems Implementation: The Need to Emphasize Non-Functional Requirements (NFRs) for Ethical AI 423
Eddie Liywalii

Democratic AI: Justification for a Broad View of Public Reason 440
Paige Benton

Symbolic AI and Knowledge Representation and Reasoning

Towards Propositional KLM-Style Defeasible Standpoint Logics 459
Nicholas Leisegang, Thomas Meyer, and Sebastian Rudolph

Non-monotonic Extensions to Formal Concept Analysis via Object Preferences 476
Lucas Carr, Nicholas Leisegang, Thomas Meyer, and Sebastian Rudolph

Knowledge Compilation for KLM-Style Defeasible Reasoning 493
Luke Slater, Thomas Meyer, and Jesse Heyninck

Author Index 509

Algorithmic and Data Driven AI



Parameter-Efficient Fine-Tuning of Pre-trained Large Language Models for Financial Text Analysis

Kelly Langa(✉), Hairong Wang, and Olaperi Okuboyejo

University of the Witwatersrand, Johannesburg 2017, South Africa
kellylanga2@gmail.com

Abstract. Recent advancements in natural language processing (NLP) have been driven by large language models (LLMs) that excel in understanding the complexities of natural language. These models have transformed NLP tasks through transfer learning, where pre-trained LLMs are fine-tuned on domain-specific datasets. Financial sentiment analysis is particularly challenging due to the complexity of financial language, requiring more advanced methods than traditional sentiment analysis approaches. Fine-tuning LLMs can enhance performance in the financial domain, but the high computational cost of standard full fine-tuning is a barrier. This study explores the effectiveness of four (4) parameter-efficient fine-tuning (PEFT) methods, namely, Low-Rank Adaptation (LoRA), prompt tuning, prefix tuning, and adapters, for financial sentiment analysis. The findings show that PEFT methods can match or exceed the performance of full fine-tuning while significantly reducing computational requirements. Specifically, adapting the Open Pre-trained Transformers (OPT) model with LoRA achieved the highest accuracy of 89% using only 0.19% of the model's parameters. PEFT methods also resulted in substantial graphics processing unit (GPU) memory savings of up to 80%. Small-scale fine-tuned LLMs outperformed cutting-edge large-scale general-purpose models like ChatGPT, highlighting the value of domain-specific fine-tuning. LLMs demonstrated superiority over conventional long short-term memory (LSTM) models by achieving a 18% increase in accuracy, thereby validating their higher implementation costs.

Keywords: Natural language processing · Large language models · Parameter efficient fine tuning · Transfer learning · Financial sentiment analysis

1 Introduction

The field of natural language processing (NLP) has experienced remarkable advancements in recent years, largely attributed to the emergence of large language models (LLMs) that have attained cutting-edge performance on a wide range of language-related tasks. LLMs possess the ability to capture the intricate semantic and syntactic complexities of natural language, making them highly suitable

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 3–20, 2025.
https://doi.org/10.1007/978-3-031-78255-8_1

for tasks that require language understanding and text generation [4]. Given their centrality in language tasks, there has been a fundamental shift in NLP methodologies, moving away from specialized custom models trained from scratch towards building on top of general-purpose base models. With the increasing availability of open-source pre-trained general-purpose base LLMs, transfer learning has emerged as a dominant paradigm to specialize (fine-tune) pre-trained models for specific down-stream tasks using a domain specific dataset.

The current state-of-the-art LLMs such as GPT-4 and Gemini Ultra are generalists in that they have been pre-trained on a diverse large corpus of text. These LLMs perform poorly out-of-the-box (zero-shot inference) on knowledge-intensive tasks as these tasks require domain specific expertise and go beyond syntax analysis or simple pattern recognition [17]. One such task is financial sentiment analysis, which involves predicting the sentiment polarity of finance textual data, such as earnings calls and social media posts. Financial sentiment analysis poses unique challenges due to the complex and domain-specific nature of financial data, characterized by technical jargon and acronyms [21]. Traditional approaches to sentiment analysis using rule-based or machine learning techniques have limitations in accurately capturing the nuances of financial language, resulting in suboptimal performance [6].

Fine-tuning base LLMs for financial sentiment analysis holds promise for improving modelling accuracy in this domain. Fine-tuned LLMs have shown promising results in financial sentiment analysis due to their ability to learn comprehensive language representations that capture the semantic and syntactic relationships within words and phrases [28]. Full parameter fine-tuning, which encompasses fine-tuning all the layers of a pre-trained LLM, remains the most effective method for adapting a model to a new domain and task in terms of modelling performance [14]. However, full fine-tuning is computationally expensive and time-consuming for training large-scale language models, making it almost infeasible in academia and industry. In recent years, researchers have been actively exploring light-weight adaptation methods that allow fine-tuning using a small subset of parameters while maintaining high modelling performance [8]. Parameter-efficient fine-tuning (PEFT) techniques are lightweight adaptation methods which involve adding a small set of weights to a pre-trained LLM and only fine-tuning the newly added weights, while keeping the pre-trained LLM parameters fixed.

In this paper, we explore the potential of adapting LLMs, considering both modelling performance and training efficiency, for sentiment analysis of financial data using PEFT methods. By harnessing the capabilities of LLMs and optimizing the fine-tuning process, the study seeks to improve the efficiency and accuracy of sentiment analysis in the financial domain, enabling the democratization of artificial intelligence and better data-driven decision-making. As such, the main contributions of this paper can be summarised as follows:

1. We developed resource-efficient models leveraging LLMs and PEFT methods achieving state-of-the-art performance in financial sentiment analysis.

2. We extend on existing work by conducting a comprehensive comparative study into the fine-tuning of different open-source LLMs employing different PEFT methods and their various configurations.
3. We show that PEFT methods lead to significant savings in peak GPU memory consumption, conserving up to 80% of GPU memory.
4. We show that PEFT methods demonstrate high modularity by utilizing merely a fraction of the model's total parameters, allowing for the creation of compact checkpoints compared to full fine-tuning.

The rest of the paper is organized into five sections. Section 2 provides a background of the relevant topics, including large language models, the transformer model architecture, transfer learning and model adaptation. Section 3 outlines the experimental procedure. Section 4 presents and discusses the study's findings. Section 5 concludes the paper and suggests future research directions.

2 Background

Transfer learning is among the most impactful concepts in deep learning and refers to the process of leveraging knowledge acquired from pre-training on a generic large-scale dataset to initialise a model for a target task [28]. In the realm of NLP, pre-trained LLMs capture general language patterns and semantic representation that possess transferability across diverse downstream tasks [19]. Leveraging a pre-trained model weights as a starting point to building a customized model has proven to be superior in terms of modelling performance and modelling efficiency as compared to building a model from scratch [4,19].

Fine-tuning is often employed to facilitate efficient transfer learning from pre-trained models to specific tasks. It involves taking a pre-trained model and adapting it to a specific task by further training it on a task-specific dataset. Full fine-tuning of a pre-trained LLM is the gold standard approach for adapting a model to a new target task [13,14]. With full parameter fine-tuning, all layers of the model are updated. However, it is computationally expensive and time consuming to update all weights of LLMs.

2.1 Parameter-Efficient Fine-Tuning

In recent times, substantial advancements have been made by researchers in the development of techniques for fine-tuning LLMs in a computationally efficient manner without sacrificing on modelling performance. These techniques are known as PEFT. PEFT methods enable efficient adaption of pre-trained LLMs to downstream tasks by adding a small set of weights to the pretrained LLM and only fine-tune the newly added weights while the weights of the pre-trained LLM remain frozen [8,13]. This approach results in major savings in computational and storage costs, while being able to maintain performance levels that are comparable to those achieved through full fine-tuning [3,8].

PEFT techniques were developed based on the observation that earlier layers in pre-trained LLMs learn generic linguistic patterns that are applicable across

various tasks while later layers tend to capture more task-specific patterns [18]. The later layers therefore undergo significant changes while earlier layers remain relatively stable during full fine-tuning [25]. The most notable PEFT methods include prompt tuning, prefix tuning, adapter tuning and low-rank adaptation (LoRA). Further elaboration on these techniques is provided in the subsequent subsections.

Prompt Tuning. Prompt tuning represents an innovative approach for fine-tuning LLMs for specific tasks, leveraging the concept of soft prompts which are learnable tensors that are concatenated with the input embeddings. Specifically, prompt tuning inserts trainable vectors at the beginning of the input sequence only at the embedding layer, without altering the model's deeper layers. Unlike traditional prompts that involve prepending text with specific instructions or examples, soft prompts are not directly interpretable by humans since they do not correspond to real-world tokens but rather serve as virtual tokens optimized during training [15]. This method preserves the original model parameters, focusing updates exclusively on the soft prompt embeddings through backpropagation, which makes it a more parameter-efficient alternative to full model fine-tuning.

Despite its efficiency prompt tuning is not without challenges. It tends to converge slower than other fine-tuning methods, which can make optimization more difficult in practice [15]. However, its ability to improve language model performance on natural language understanding tasks, combined with the computational and memory efficiencies it offers, makes it a promising tool in the evolving landscape of model adaptation.

Prefix Tuning. Introduced by Li and Liang in 2021 [16], this technique innovates by appending task-specific trainable vectors, referred to as prefixes, not only to the input layer but across all layers of the transformer model [16]. These prefixes are integrated into the key and value matrices of the model's self-attention mechanism, providing a task-relevant context that guides the model towards generating desired outputs [16]. Distinctively, prefix tuning diverges from direct manipulation of soft prompts. It employs a separate bottleneck feed-forward neural network (FFN) to optimize these prefix parameters, thereby circumventing the training instabilities and performance degradation associated with direct training through the reparameterizion of the prefix vectors [8]. After the optimization process, this FFN can be discarded, leaving the optimized prefixes to influence the model's behavior.

The length of the prefix plays a critical role in the tuning process, with empirical evidence suggesting that there exists an optimal prefix length beyond which performance may plateau or slightly decline [16]. This balance underscores the nuanced relationship between the amount of introduced task-specific information and model performance. A drawback of prefix tuning, however, is that its optimization process can be complex, necessitating experimentation with various parameters to achieve the best results [23]. Moreover, the introduction of

prefixes reduces the available sequence length for actual input data, potentially limiting the model's capacity to handle longer inputs.

Adapter Tuning. Adapter tuning has emerged as an innovative and lightweight approach to customizing pre-trained networks for specific downstream tasks without the need for comprehensive retraining of the entire model. By integrating small, fully connected networks known as adapters between the sub-layers of transformer models as shown in Fig. 1, this method achieves notable performance enhancements while tuning a small percentage of the total model weights [12]. The essence of adapter tuning lies in its ability to introduce architectural modifications that repurpose a pre-trained network, focusing solely on the new parameters introduced by the adapters while keeping the original LLM parameters frozen [10]. This strategy not only preserves the model's acquired knowledge but also ensures a minimal addition of parameters, making it a highly memory-efficient solution.

The core architecture of these adapters incorporates a bottleneck design, which significantly reduces the dimensionality of the input features before applying a non-linearity and projecting them back to their original dimension. This process, coupled with an internal skip-connection within each adapter module, allows for an effective balance between performance and parameter efficiency [23]. The skip-connection plays a crucial role, ensuring that the adapters can function as an approximate identity when their parameters are near-zero at initialization, thus preserving the integrity of the original input while introducing task-specific adaptation [12]. Originating from the pioneering work of Houlsby et al. [10], the adapter tuning method has been refined to include not just the bottleneck structure but also specific layer normalization parameters trained for each task. The Houlsby adapter configuration places two adapters in the sublayers of the transformer as shown in Fig. 1.

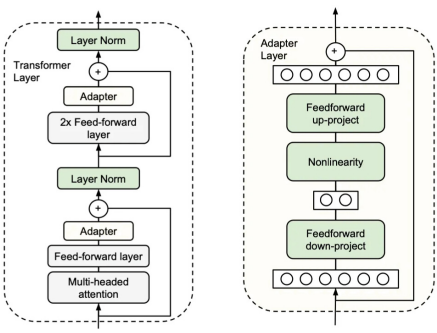


Fig. 1. Adapters incorporated in the transformer architecture using the Houlsby adapters configuration [10]

Low-Rank Adaptation. Developed by Hu et al. [11], LoRA fundamentally changes the approach to model adaptation by injecting trainable low-rank decomposition matrices into the layers of a pre-trained model [1]. Figure 2 illustrates a schematic representation of LoRA, showcasing the self-attention layer LoRA configuration which adds LoRA matrices to the self-attention sub-layer. This method is predicated on the insight that pre-trained models, despite their size, operate within a low intrinsic dimension, meaning that their complexity can be effectively captured using matrices of much lower rank than initially presumed [1]. At the heart of LoRA is the concept of matrix rank, which signifies the maximum number of linearly independent rows or columns in a matrix. This concept is crucial because it informs us about the amount of unique information or the dimensionality of the space spanned by the matrix [30]. LoRA leverages this by approximating the weight adjustments needed during fine-tuning as the product of two smaller matrices (W_A and W_B), significantly reducing the number of parameters required for effective fine-tuning. The rank r becomes a critical hyperparameter in this setup, dictating the degree of approximation and the balance between model complexity and training efficiency [7].

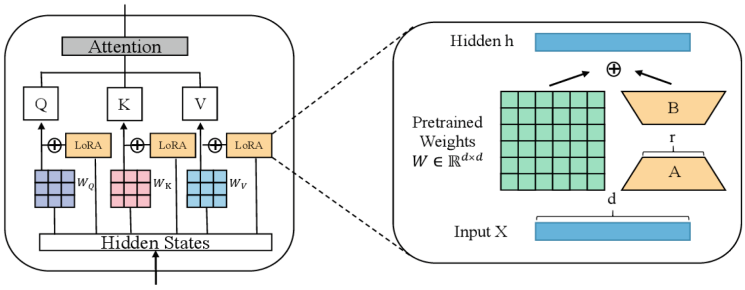


Fig. 2. Low-rank adaptation incorporated in the transformer architecture using the self-attention layer LoRA configuration [27]

3 The Proposed Method

This section outlines the methodologies employed in our study to fine-tune LLMs using PEFT techniques. We detail the steps from dataset preprocessing and splitting, through the integration of PEFT methods in model adaptation, to the benchmarking against established adaptation methods. The section also describes the evaluation metrics used to assess the efficiency and performance of the fine-tuned models, and the iterative process of analysis and hyperparameter tuning to optimize results.

3.1 Data and Data Preprocessing

Data Description. Large language models were fine-tuned using the financial phrasebank dataset [22] for a sentiment analysis task, aiming to thoroughly assess the encoded knowledge within the LLMs. This dataset consists of sentences from financial news categorised by sentiment. The dataset serves as a benchmark to assess the effectiveness of sentiment models specifically designed for economic texts. The dataset consists of 4846 sentences labelled as either negative, positive or neutral sentiment. The class distribution of the neutral, negative and positive labels are 59%, 28% and 13% respectively.

Tokenization and Encoding. The text in the dataset was tokenized and encoded into numerical representations that can be understood by the model. Pre-trained LLMs have their own built-in-tokenizers specifically designed to work with the model's vocabulary and architecture [9]. The tokenizers of the employed LLMs are based on the byte-pair encoding (BPE) and SentencePiece tokenizers. These tokens were then encoded into unique integers based on the model's vocabulary.

Batch Processing and Padding. The sequence length of the sentences in the dataset varies, necessitating uniform input sequences to enable parallel processing. To optimize memory usage and computational efficiency, batch processing and padding were employed. A batch size of 8 was utilized along with dynamic padding. With dynamic padding the sequence length of a batch is determined by the longest sequence in the batch. Sequences shorter than the designated length will be padded with special tokens.

Attention Mask. Attention masks were added to the padded sequences to guide the model's attention mechanism, helping to identify the actual content of each sequence while distinguishing it from the padding tokens used to equalize the sequence lengths.

Sentiment Label Encoding. The financial phrasebank dataset has categorical sentiment labels. The categorical labels were converted into unique integers in line with the vocabulary of the model.

3.2 Methods

Training, Validation and Test Split. The dataset was split into 70% training set, 15% validation set, and 15% testing set.

Pre-trained Models. We conducted a comparative analysis of LLMs built on three distinct transformer architectures: encoder-only, encoder-decoder, and decoder-only, due to the limited existing literature on their parameter-efficient adaptation strategies. The selection of LLMs was based on several criteria, including performance on benchmarks such as the Massive Multitask Language Understanding (MMLU) and General Language Understanding Evaluation (GLUE), open-source licensing allowing commercial use, fine-tunability, model size, and out-of-the-box quality. The LLMs employed are Robustly Optimized BERT approach (RoBERTa) [20], Fine-tuned LAnguage Net Text-To-Text Transfer Transformer (FLAN-T5) [5] and Open Pre-trained Transformers (OPT) [29]. We used smaller versions of the LLMs taking into account our limited computational resources. Table 1 gives a summary of the LLMs that were employed, showing their model size, architecture, training objective, vocabulary size, context length and dataset size used for pre-training. The capacity of an LLM is a dynamic interplay between these factors. The models fall within a similar scale range, and their pre-training utilised a diverse generic dataset consisting of billions of tokens making them general-purpose models. The pre-training strategies used are the Masked Language Modelling (MLM) and Autoregressive Language Modelling (ALM).

Table 1. Comparative overview of hyperparameters utilized in the pre-training of RoBERTa-large, FLAN-T5-base and OPT-350 models

	RoBERTa-large	FLAN-T5-base	OPT-350
Model size	355M	248M	350M
Model Architecture	encoder-only	encoder-decoder	decoder-only
Pre-training strategy	MLM	MLM	ALM
Pre-training data size	160B tokens	34B tokens	180B tokens
Vocabulary size	50K	32K	50K
Context length	512	512	2048

Incorporating PEFT Techniques in Fine-Tuning LLMs. To date, there still exists a void in comprehensive investigation regarding the impact of various efficient tuning methods on LLMs pre-trained under diverse settings. We employed four PEFT methods, each introducing distinct modifications to the transformer architecture. The PEFT methods are prompt tuning, prefix tuning, adapter tuning and low-rank adaptation (LoRA). We further explored various configurations of these methods as summarized in Table 2.

Model Training. The aforementioned parameter-efficient fine-tuning techniques were employed to fine-tune the LLMs. The models were fine-tuned by adjusting the extra added PEFT parameters using backpropagation and gradient decent while the base model was kept frozen. Table 3 summarises the

Table 2. PEFT methods and their configurational variants

PEFT Method	Configuration
Prompt Tuning	Initializing prompt with own text
	Random prompt initialization
	Initializing prompt with LLM tokens
Prefix Tuning	No reparameterization
	Reparameterization
Adapter	Pfeiffer adapter [26]
	Houlsby adapter [10]
	Parallel adapter [12]
LoRA	Self-attention layer LoRA
	Feed-forward layer intermediate weights LoRA
	Feed-forward layer output weights LoRA

hyperparameters that were utilized to fine-tune the OPT model with LoRA. Prompt tuning and prefix tuning utilize the number of virtual tokens hyperparameter, adapters employ the bottleneck dimension hyperparameter, and LoRA utilizes the LoRA rank, LoRA alpha, and LoRA dropout hyperparameters. The LoRA alpha and dropout were set at 32 and 0.1, respectively, in line with the recommendations from the original LoRA paper [11]. To identify the optimal hyperparameter configurations for training the LLMs, we utilized the HyperOpt Tree of Parzen Estimators (TPE) search algorithm in conjunction with an Asynchronous Successive Halving (ASHA) trial scheduler. Fractional GPU utilization was employed to concurrently execute multiple trials on a single GPU, thereby conserving resources.

Table 3. Hyperparameters used for fine-tuning OPT model with LoRA

Hyperparameter	OPT model		
	Low-rank adaptation		
	Attention weights	FFN intermediate weights	FFN output weights
LoRA rank	18	5	41
LoRA dropout	0.1	0.1	0.1
LoRA alpha	32	32	32
Batch size	8	8	8
Epochs	3	3	4
Learning rate (LR)	0.0001227	0.00008069	0.0001569
LR Scheduler	warmup LR	warmup LR	warmup LR
Warmup ratio	0.03560	0.03778	0.08630
Optimizer	AdamW	AdamW	AdamW
Loss	cross-entropy	cross-entropy	cross-entropy
Dropout	0.1	0.1	0.1

Hardware, Libraries and Software. Model training and analysis were conducted in Python 3 on Google Colab, utilizing an Intel Xeon CPU (13 GB RAM) and NVIDIA Tesla K80 GPU (12 GB VRAM). Open-source libraries, transformers and PyTorch were used for model implementation, with scikit-learn for performance evaluation and pandas for data manipulation. Visualizations were created with seaborn and matplotlib. PEFT methods were implemented using Peft, opendelta, and adapter-transformers libraries. Hyperparameter tuning was performed with Ray Tune, and artifacts were logged using the weights and biases library.

3.3 Analysis

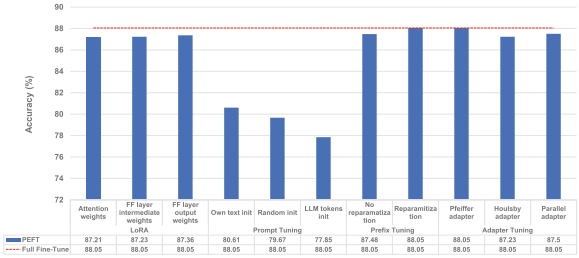
Benchmark. To evaluate the effectiveness of fine-tuning LLMs using PEFT techniques, we established a benchmark and conducted tests using established fine-tuning methods and traditional NLP algorithms, as well as presenting outcomes obtained from previous studies. The methods included full fine-tuning (baseline), model head fine-tuning (adapting only the last layer), and zero-shot inference without fine-tuning. Additionally, we referenced results from prior studies employing traditional NLP algorithms (LSTM) and state-of-the-art general-purpose LLMs (GPT-4 and ChatGPT) to justify the use of LLMs and the associated fine-tuning costs.

Evaluation Metric. In evaluating the modelling performance of the models, accuracy scores were computed. In evaluating the computational efficiency of training the models, the following evaluation metrics were used: number of trainable parameters, storage required to save the adaptation parameters, training time, inference latency and peak GPU memory usage.

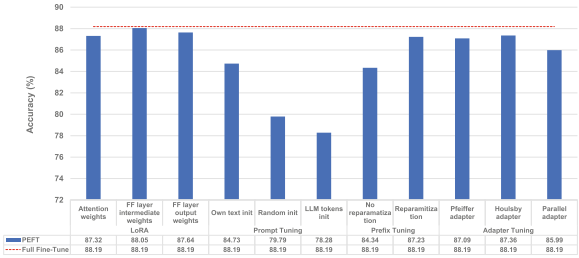
4 Results and Discussion

4.1 Knowledge Transfer Efficacy of LLMs Through PEFT Techniques

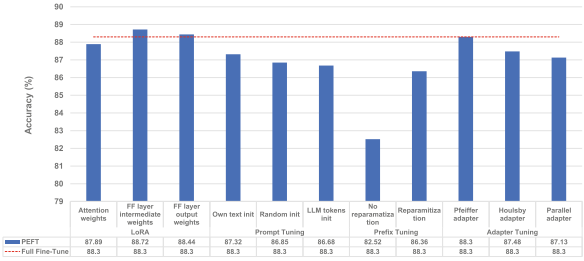
Figure 3 illustrates the modeling performance of the employed PEFT methods and their various configurations in terms of accuracy across three different models, with full fine-tuning serving as the baseline for comparison. Evaluation of each PEFT method's performance on RoBERTa, FLAN-T5 and OPT models revealed consistent performance, ranging from a minimum accuracy of 78% to a maximum of 89%. This suggests the effectiveness of PEFT methods in transferring knowledge during the fine-tuning process of LLMs pre-trained using different strategies. Overall, the OPT model showed the highest average accuracy when fine-tuned using the different PEFT methods, followed by RoBERTa and then FLAN-T5. The observed trend suggests that the LLM's capacity and its ability to transfer knowledge are primarily determined by its model size and the scale of the pre-training data, with the pre-training strategy having a less significant impact in terms of architecture.



(a) RoBERTa model



(b) FLAN-T5 model

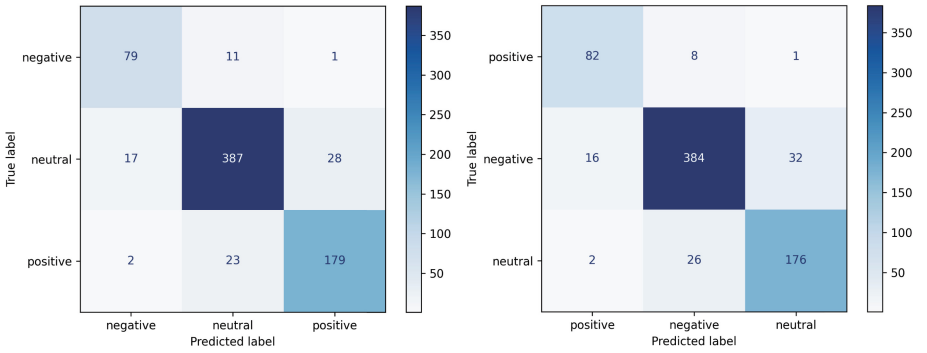


(c) OPT model

Fig. 3. Modelling performance of PEFT methods and their various configurations with full fine-tuning as the baseline

4.2 Comparing the Modeling Performance of PEFT Methods

LoRA emerged as the top-performing PEFT method in terms of modeling accuracy. It surpassed full fine-tuning when applied to the OPT model, specifically in configurations using feed-forward layer intermediate weights and feed-forward layer output weights. The feed-forward layer intermediate weights LoRA refers to the application of the LoRA technique to the intermediate weights within the feed-forward layer, while the feed-forward layer output weights LoRA applies LoRA to the output weights of the feed-forward layer. Drawing from our review of existing literature, LoRA is almost always applied using the self-attention weights LoRA configuration which was proposed in the original LoRA paper [11]. However, our findings strongly advocate for its utilization using the feed-forward layer intermediate weights. This deviation from the conventional approach aligns



(a) LoRA (b) Adapter tuning

Fig. 4. Confusion matrix generated for LoRA and adapter tuning when applied to the OPT model

with the common understanding in deep learning that early layers capture generic linguistic patterns, while later layers specialize in task-specific features. Indeed, both feed-forward layer intermediate weights and feed-forward layer output weights LoRA configurations outperform the self-attention weights configuration. Both these methods inject LoRA modules in the later layers closer to the model head.

Adapter tuning emerged as the second-best performing method, matching the performance of full fine-tuning when utilizing the Pfeiffer adapter configuration on both the RoBERTa and OPT models. Notably, the Pfeiffer adapter configuration outperforms the pioneering Houlsby adapter configuration, which originally introduced the concept of adapters. Unlike the Houlsby adapter, which inserts an adapter module after both the self-attention layer and the feed-forward layer, the Pfeiffer adapter inserts the module solely after the feed-forward layer. Injecting adapter modules solely at the higher layers yields superior results compared to distributing them across both early and top layers, which can potentially lead to diminished performance. The results further emphasize the intuition that lower layers capture lower general features that are shared among tasks, while the later layers develop task-specific features. The parallel adapter configuration falls short of both the Houlsby and Pfeiffer adapter configurations in terms of modelling performance, highlighting the advantage of applying adapters in a serial manner.

Prefix tuning secured the third position, achieving the same performance as full fine-tuning when employing the reparameterization prefix tuning configuration on the RoBERTa model. Prefix-tuning prepends task-specific trainable prefix vectors to the key matrix and value matrix in the self-attention layer. Considering that the self-attention layer is the foundational element of the transformer architecture, which provides it with the capability to capture intricate relationships and nuances within data, the incorporation of prefix modules in the self-attention layer's matrix weights proves to be an efficient strategy for

enhancing LLMs. It is more effective to carry out the optimization of prefix parameters through an independent feed-forward network (reparameterization), rather than directly adjusting the soft prompts, to avoid instability and performance degradation.

The performance of prompt tuning fell behind that of the other PEFT methods. Notably, initializing the prompt with a custom-designed prompt yielded better results compared to other prompt tuning configurations. The subpar performance of prompt tuning was expected, as this method introduces no alterations to the internal structure of the transformer but merely appends tokens to the input sequence. These additional tokens solely guide the model in generating the output. Consequently, prompt tuning with random prompt initialization and initialization using tokens from the model's vocabulary proved ineffective in steering the model towards generating the desired output. Both approaches seemed to cause confusion within the model, resulting in poorer performance.

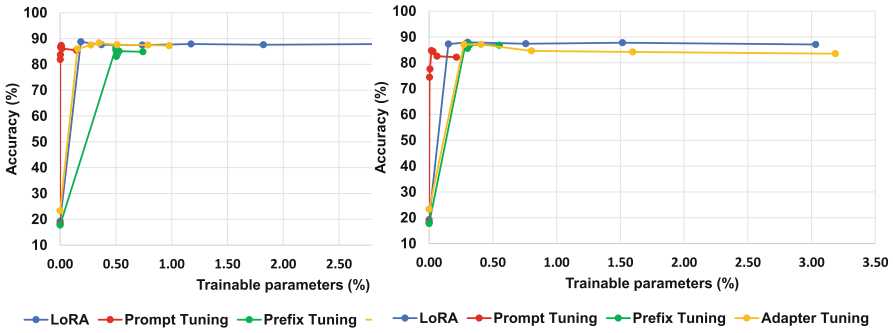
Notably, in handling imbalance inherent in our dataset, our models demonstrate remarkable capability in classifying each class effectively without exhibiting bias towards any specific class. This equitable performance is further elucidated through the detailed examination provided by the confusion matrix showcased in Fig. 4. In the realm of finance, where precision and recall play pivotal roles in decision making processes concerning investments, striking a balance between these two metrics is paramount.

4.3 Number of Trainable Parameters

Table 4 illustrates the characteristics of each PEFT method, including trainable parameters, storage size, and training time, focusing on the best performing configuration for each PEFT method, with full fine-tuning serving as the baseline. Across the board, the trainable parameters introduced by our PEFT methods represent a fraction ranging from 0.01% to 1.06% of the model's total parameters. Notably, LoRA applied to the OPT model excels, outperforming full fine-tuning while utilizing 0.19% of the model's total parameters. Interestingly, the relationship between the performance of PEFT methods and the number of trainable parameters is not linear. Figure 5 elucidates how performance tends to improve with an increase in trainable parameters up to a certain threshold, after which a slight decline is observed. This phenomenon underscores the critical role of the PEFT method's architecture and the placement of PEFT modules in determining its efficacy. Notably, some PEFT methods outperform full fine-tuning due to their ability to mitigate catastrophic forgetting, a phenomenon observed during full fine-tuning of LLMs, where the model inadvertently forgets previously acquired knowledge from pre-training, leading to overfitting on small fine-tuning datasets. Given the minimal number of trainable parameters, PEFT methods offer notable benefits in terms of portability and modularity. They enable the creation of compact checkpoints, typically just a few megabytes in size, compared to the bulky checkpoints generated through full fine-tuning.

Table 4. Comparison of trainable parameters and training time for different fine-tuning methods

Model	Method	Trainable parameters	Trainable parameters size (MB)	Training time (s)
RoBERTa	Full fine-tune	355.36M (100%)	1421.45 (100%)	458
	LoRA (FFN intermediate weights)	3.33M (0.93%)	13.34 (0.94%)	303
	Prompt tuning (Own text init)	2.11M (0.59%)	8.44 (0.59%)	779
	Prefix tuning (Reparamitization)	3.78M (1.06%)	15.12 (1.06%)	363
	Adapter tuning (Pfeiffer)	3.59M (1.00%)	14.34 (1.01%)	229
Flan-T5	Full fine-tune	296.93M (100%)	990.31 (100%)	289
	LoRA (FFN intermediate weights)	67.58M (0.30%)	101.4 (10.24%)	470
	Prompt tuning (Own text init)	46.08K (0.01%)	0.18 (0.02%)	207
	Prefix tuning (Reparamitization)	0.64M (0.002%)	101.24 (10.22%)	340
	Adapter tuning (Pfeiffer)	0.90M (0.41%)	102.31 (10.33%)	389
OPT	Full fine-tune	331.20M (100%)	1324.79 (100%)	477
	LoRA (FFN intermediate weights)	0.62K (0.19%)	2.46 (0.19%)	206
	Prompt tuning (Own text init)	32.26K (0.01%)	0.13 (0.01%)	1886
	Prefix tuning (Reparamitization)	1.66M (0.50%)	6.65 (0.50%)	600
	Adapter tuning (Pfeiffer)	1.16M (0.35%)	4.63 (0.35%)	291



(a) OPT model (b) FLAN-T5 model

Fig. 5. Performance of PEFT methods with number of trainable parameters

4.4 Computational Efficiency

Figure 6 depicts the peak GPU memory consumption during the fine-tuning of LLMs using both PEFT methods and full fine-tuning. Notably, PEFT methods demonstrate significant savings, conserving up to 80% of GPU memory when fine-tuning the RoBERTa model. These savings are primarily achieved by reducing the need for gradient computation for most parameters, as they are held fixed during fine-tuning.

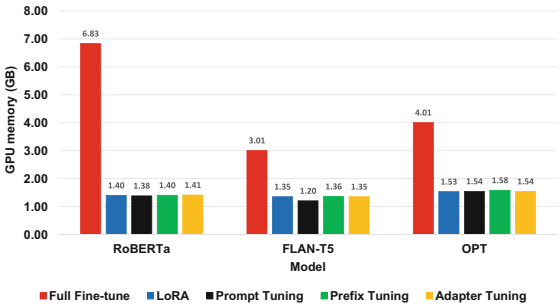


Fig. 6. Peak GPU memory consumed during fine-tuning

4.5 Inference Latency

Network latency is a crucial consideration when deploying deep networks in real-world scenarios, where rapid inference times are paramount, often ranging from milliseconds to seconds. In assessing network latency, the focus is primarily on measuring the feed-forward operation of the network. Figure 7 illustrates the inference latency of the PEFT methods for the utilized LLMs, with full fine-tuning serving as the baseline. PEFT methods introduce additional trainable parameters to the base model, potentially resulting in inference latency due to increased data flow paths. However, the results indicate that there is no statistically significant difference in the inference latency between the various techniques. Given their negligible impact on inference latency, PEFT methods present an appealing option for deployment in sectors like finance.

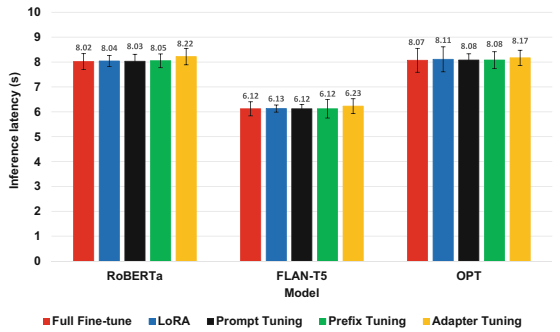


Fig. 7. Inference latency using various fine-tuning methods

4.6 Comparison with Traditional NLP Algorithms and Cutting-Edge Models

The extent to which LLMs, fine-tuned for financial text using lightweight adaptation methods, outperform traditional NLP models in finance-related tasks remains uncertain. Table 5 presents sentiment analysis results using LSTM models, a conventional choice for NLP tasks. By employing the OPT model fine-tuned with LoRA, we observed up to 18% increase in accuracy over the LSTM model. This superior performance suggests that self-supervised pre-training of LLMs, although costly, offers substantial benefits. Contrary to the common belief that fine-tuning LLMs requires a large dataset, our findings indicate that LSTMs need more data to achieve optimal performance compared to domain-adapted LLMs.

This study was limited to small-scale LLMs due to computing power constraints. Based on LLM scaling laws, we expect performance to improve with large-scale LLMs. Considering the emergent zero-shot learning abilities of large-scale LLMs, a pertinent question arises: are specialized models crafted through fine-tuning necessary, or can superior performance be achieved with general-purpose large-scale LLMs like ChatGPT? Comparing our fine-tuned models using PEFT methods to state-of-the-art large-scale LLMs, we observe that small-scale fine-tuned LLMs outperform GPT-4 and ChatGPT. Therefore, in specialized domains, it is preferable to fine-tune a small-scale domain-specific LLM rather than rely on a large-scale general-purpose LLM out-of-the-box.

Table 5. The modelling performance of LSTM models, state-of-the-art LLMs and OPT model fine-tuned using LoRA

Model	Accuracy (%)	F1-score
LSTM [2]	71	0.64
LSTM [24]	72	0.66
ChatGPT [17]	78	–
GPT-4 [17]	76	–
OPT+LoRA	89	0.89

5 Conclusion

This paper examines the effectiveness of parameter-efficient fine-tuning methods for adapting LLMs to sentiment analysis of financial data, focusing on both performance and training efficiency. Through a thorough analysis of methods such as LoRA, prompt tuning, prefix tuning, and adapter tuning, alongside traditional full fine-tuning, key insights emerged. Adapting the OPT model with LoRA achieved the highest performance with 89% accuracy, using less than 1.06% of the model's total parameters. The storage size for trainable parameters was minimal (0.1 MB to 7 MB), and PEFT methods added negligible inference

overhead while reducing peak GPU memory consumption by up to 80%. Additionally, small-scale fine-tuned LLMs outperformed state-of-the-art large-scale LLMs in specialized domains and demonstrated an 18% accuracy improvement over traditional LSTM models, justifying their higher implementation costs. The advancements from this paper can make AI more accessible, especially for those with limited computational resources, and contribute to environmental sustainability by reducing the energy and carbon footprint of training large-scale AI models.

Future Work. These findings emphasize the potential of PEFT methods for practical applications in various domains and highlight the need for further research to refine these techniques and explore their broader applicability in NLP. The following are possible future research avenues to build on this work. Firstly, exploring the composability of PEFT methods by stacking, fusing, or mixing different PEFT modules to harness their combined knowledge effectively. Additionally, leveraging lower-precision floating point representations (e.g., FP16) during training with PEFT methods has the potential to accelerate training speed.

Acknowledgement. The first author, K. Langa, wishes to acknowledge the DSI-CSIR NEPTTP for their generous financial support, which has made this work possible.

References

1. Aghajanyan, A., Zettlemoyer, L., Gupta, S.: Intrinsic dimensionality explains the effectiveness of language model fine-tuning. arXiv preprint arXiv:2012.13255 (2020)
2. Araci, D.: FinBERT: financial sentiment analysis with pre-trained language models (2019)
3. Chen, L., Zaharia, M., Zou, J.: FrugalGPT: how to use large language models while reducing cost and improving performance. arXiv preprint arXiv:2305.05176 (2023)
4. Chronopoulou, A., Baziotis, C., Potamianos, A.: An embarrassingly simple approach for transfer learning from pretrained language models. arXiv preprint arXiv:1902.10547 (2019)
5. Chung, H.W., et al.: Scaling instruction-finetuned language models. arXiv preprint arXiv:2210.11416 (2022)
6. Dalika, N.K., Seetharam, Y.: Sentiment and returns: analysis of investor sentiment in the South African market. Ph.D. thesis, University of the Witwatersrand, Faculty of Commerce, Law and Management (2014)
7. Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L.: QLORA: efficient finetuning of quantized LLMs. arXiv preprint arXiv:2305.14314 (2023)
8. Ding, N., et al.: Parameter-efficient fine-tuning of large-scale pre-trained language models. Nat. Mach. Intell. **5**(3), 220–235 (2023)
9. He, P., Liu, X., Gao, J., Chen, W.: DeBERTa: decoding-enhanced BERT with disentangled attention. In: International Conference on Learning Representations (2021). https://openreview.net/forum?id=XPZIaotutsD
10. Houlsby, N., et al.: Parameter-efficient transfer learning for NLP. In: International Conference on Machine Learning, pp. 2790–2799. PMLR (2019)

11. Hu, E.J., et al.: LoRA: low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
12. Hu, Z., et al.: LLM-adapters: an adapter family for parameter-efficient fine-tuning of large language models (2023)
13. Kumar, A., Raghunathan, A., Jones, R., Ma, T., Liang, P.: Fine-tuning can distort pretrained features and underperform out-of-distribution. arXiv preprint arXiv:2202.10054 (2022)
14. Lee, Y., et al.: Surgical fine-tuning improves adaptation to distribution shifts. arXiv preprint arXiv:2210.11466 (2022)
15. Lester, B., Al-Rfou, R., Constant, N.: The power of scale for parameter-efficient prompt tuning. arXiv preprint arXiv:2104.08691 (2021)
16. Li, X.L., Liang, P.: Prefix-tuning: optimizing continuous prompts for generation. arXiv preprint arXiv:2101.00190 (2021)
17. Li, X., Zhu, X., Ma, Z., Liu, X., Shah, S.: Are ChatGPT and GPT-4 general-purpose solvers for financial text analytics? An examination on several typical tasks. arXiv preprint arXiv:2305.05862 (2023)
18. Lialin, V., Deshpande, V., Rumshisky, A.: Scaling down to scale up: a guide to parameter-efficient fine-tuning. arXiv preprint arXiv:2303.15647 (2023)
19. Liu, R., Shi, Y., Ji, C., Jia, M.: A survey of sentiment analysis based on transfer learning. IEEE Access **7**, 85401–85412 (2019)
20. Liu, Y., et al.: RoBERTa: a robustly optimized BERT pretraining approach (2019)
21. Loughran, T., McDonald, B.: Textual analysis in finance. Annu. Rev. Financ. Econ. **12**, 357–375 (2020)
22. Malo, P., Sinha, A., Korhonen, P., Wallenius, J., Takala, P.: Good debt or bad debt: detecting semantic orientations in economic texts. J. Assoc. Inf. Sci. Technol. **65** (2014)
23. Mo, Y., Yoo, J., Kang, S.: Parameter-efficient fine-tuning method for task-oriented dialogue systems. Mathematics **11**(14), 3048 (2023)
24. Omarkhan, M., Kissymova, G., Akhmetov, I.: Handling data imbalance using CNN and LSTM in financial news sentiment analysis. In: 2021 16th International Conference on Electronics Computer and Computation (ICECCO), pp. 1–8. IEEE (2021)
25. Peters, M.E., Ruder, S., Smith, N.A.: To tune or not to tune? Adapting pretrained representations to diverse tasks. arXiv preprint arXiv:1903.05987 (2019)
26. Pfeiffer, J., et al.: AdapterHub: a framework for adapting transformers. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pp. 46–54 (2020)
27. Wang, C., Yi, J., Zhang, X., Tao, J., Xu, L., Fu, R.: Low-rank adaptation method for Wav2Vec2-based fake audio detection (2023)
28. You, K., Liu, Y., Wang, J., Long, M.: LogME: practical assessment of pre-trained models for transfer learning. In: International Conference on Machine Learning, pp. 12133–12143. PMLR (2021)
29. Zhang, S., et al.: OPT: open pre-trained transformer language models (2022)
30. Zi, B., Qi, X., Wang, L., Wang, J., Wong, K.F., Zhang, L.: Delta-LoRA: fine-tuning high-rank parameters with the delta of low-rank matrices. arXiv preprint arXiv:2309.02411 (2023)

Optimally Traversing Explainability in Bayesian Networks via the Graphical Lasso

Iena Petronella Derks[1(✉)], Alta de Waal[1,2], Jarod Smith[1,2], Theodor Loots[1,2], and Jean-Pierre Stander[1,2]

[1] Department of Statistics, University of Pretoria, Pretoria, South Africa
inekederks1@gmail.com
[2] Bayerische Motoren Werke AG, Munich, Germany

Abstract. Abductive inference plays a critical role in solving problems involving uncertain information and multiple competing hypotheses. In the Bayesian network framework, abductive inference involves generating the configuration of target variables that best explains the observed evidence. Furthermore, research has shown that these configurations should be precise and concise. However, the search space for potential explanations can be vast, particularly in more complex networks, requiring efficient search algorithms to traverse this space and find optimal configurations. This aligns with the challenges in explainable artificial intelligence, where researchers aim to provide relevant and meaningful explanations in a computationally efficient manner. Previous work has focused on developing several search algorithms, such as forward, backward, tabu, and reverse jump MCMC algorithms. Algorithms like the forward search algorithm still involve exhaustive searching as it does not prune the neighbourhood of potential solutions. To address this, we propose incorporating a statistical neighbourhood selection method, such as the graphical Lasso (Least Absolute Shrinkage and Selection Operator), with the forward search algorithm to prune the search space. The graphical Lasso's penalty term encourages sparsity, helping to prune less relevant variables, thereby improving computational efficiency. By reducing the set of instantiations, the forward-gLasso algorithm provides explanations focusing on the most relevant dependencies. We evaluate the performance and computational efficiency of the forward-gLasso algorithm on a set of benchmark Bayesian networks and compare it to the forward search algorithm.

Keywords: Explainable artificial intelligence · Bayesian networks · most relevant explanation · graphical Lasso

1 Introduction

Consider a detective investigating a homicide case. At the crime scene, they find a victim with multiple stab wounds and formulate several hypotheses. For example, the victim was attacked by a stranger; they knew the attacker, or they were involved in illegal activities. As evidence is gathered, the detective must weigh each hypothesis against the available information, such as the victim's relationships, witness statements, and forensics. To solve the case, the detective would use *abductive inference* to identify the most

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 21–37, 2025.
https://doi.org/10.1007/978-3-031-78255-8_2

likely explanation for the crime given the evidence, acknowledging some uncertainty may exist.

Abductive inference involves reasoning and forming explanations based on uncertain information, making it useful when multiple hypotheses can explain observed evidence. The level of uncertainty depends on the evidence and the plausibility of alternative hypotheses [28]. To reduce uncertainty, it is important to consider evidence and carefully evaluate competing explanations to reduce uncertainty. Probabilistic methods can help quantify and manage uncertainties associated with abductive inference.

Explainable artificial intelligence (XAI) and abductive inference are closely related. XAI techniques enhance transparency in AI decision-making by explaining the predictions and identifying key factors that influenced those predictions. This research utilises abductive inference to generate explanations for observed evidence in Bayesian networks. Here, abductive inference refers to finding the configuration of variables most likely to explain the observed evidence [11]. Methodologies developed for this purpose include Maximum a posterior assignment, most probable explanation, most inforbable explanation [21], and most relevant explanation [38].

Much like how the number of possible structures increases exponentially as more variables are added in the structure learning problem [15], the search space encompassing all configurations capable of explaining the evidence also expands exponentially. Notably, increasing the number of states of at least one variable increases the complexity of the search space. Therefore, an exhaustive enumeration of all possible configurations becomes impractical. Traversing this vast search space to identify the optimal configuration requires efficient search algorithms. The demand for efficient search algorithms aligns with the challenges faced in XAI, where researchers aim to develop computationally efficient methods for providing meaningful explanations.

This research presents a novel forward-gLasso search algorithm that provides the most relevant explanation for the observed evidence. It combines the search strategy of the forward search with the graphical Lasso (gLasso) [10] to shrink the set of neighbouring solutions. We evaluate the algorithm's performance by investigating two research questions: 1) Is the forward-gLasso as accurate as the benchmark forward search? 2) Is it more computationally efficient? Experimental results show that the forward-gLasso efficiently reduces the neighbourhood size while maintaining accuracy comparable to the forward search, depending on the regularisation parameter, λ. Moreover, all variants of the forward-gLasso solve test cases more efficiently than the forward search.

The paper is organised as follows. A brief overview of Bayesian networks, explainability, and existing explanation methods is presented in Sect. 2, while Sect. 3 expands on the concepts related to the most relevant explanation. The proposed forward-gLasso is discussed in Sect. 4 followed by the experimental results in Sect. 5. Lastly, we provide our concluding remarks in Sect. 6.

2 Literature Study

2.1 Bayesian Networks

Introduced by [27], Bayesian networks are probabilistic graphical models that can be used to model probabilistic relationships among variables within a domain of uncer-

tainty. Variables are connected by *directed* arcs. These connections form a directed acyclic graph representing the dependencies among variables in the domain [15]. The probabilistic relationships among the nodes are quantified by defining a conditional probability distribution over each node as conditional probability tables [20]. The framework facilitates reasoning about the domain through both *belief updating* and *belief revision*. For example, when a variable in the network is observed, one might be interested in conditioning upon this new evidence. This conditioning is performed according to the 'flow of information' through the network. Probabilistically, this is done by computing the posterior probabilities for a set of query nodes based on the observed evidence nodes [20]. On the other hand, one might be interested in explaining *why* the evidence was observed. This can be done by revising the belief commitment to obtain a set of variable instantiations, given the evidence [27].

2.2 Explainability

The last decade has seen significant expansion in artificial intelligence (AI). Although there is no universally agreed-upon definition, explanations in AI involve providing clear and relevant insights into a model's predictions and behaviour. An explanation should go beyond providing output and reveal the underlying factors or logic contributing to a system's prediction. It should enable users to comprehend, trust, and potentially dispute the results obtained, promoting human-centred AI and increasing the acceptance and adoption of AI in various domains. Two main factors influence the need for explainability. Firstly, the degree of explainability varies depending on the application domain [12], with higher demand in areas where predictions influence human lives [1]. Secondly, *transparency*; most AI systems operate as "black boxes", offering no explicit insight into their decision-making process. In contrast, explainable-by-design systems provide clear reasoning or an easily understandable process, reducing the necessity for additional explanations.

2.3 Explanation in Bayesian Networks

Although Bayesian networks are considered transparent, they become more difficult to understand as their complexity increases. Dependencies and probabilistic priors are considered during the model-building phase, which can lead to counter-intuitive results [17]. Finally, interdependencies among nodes may lead to complex and indirect relationships [13]. As such, Bayesian networks can benefit from explanations. Lacave and Díez [22] propose three categories of explanation, with the categories differing based on the explanation's focus: 1) explanation of the model, presenting the knowledge base in simple formats; 2) explanation of reasoning, justifying the observed evidence and reasoning processes [19]; and 3) explanation of evidence, explaining why evidence variables take their observed states [39]. A fourth category, focused on decision-readiness, was proposed by [9]: explanation of decisions. This research focuses on the explanation of evidence category.

Bayesian networks offer a principled approach to finding explanations for observed evidence by revising beliefs and making inferences based on the observed data. These techniques range from providing singleton explanations to multivariate explanations.

However, given that several variables may explain observed evidence, explanations comprised of single nodes may be *underspecified*. In other words, the explanation may be too simple, especially in scenarios where the observed evidence results from multiple causes.

Maximum a posteriori assignment (MAP) is often used to find the most probable configuration of variables for the observed evidence. The MAP-derived explanation comprises a subset or partial instantiation of unobserved variables that maximise the joint posterior probability. Kwisthout [21] proposes an extension to MAP, named the *most inforbable explanation*, which integrates two fundamental properties of abduction: the selection of candidate hypotheses along with the inference to the best explanation. A special case of MAP, namely the *Most Probable Explanation* (MPE), finds instantiations consisting of all unobserved variables in the domain. Several local and approximate algorithms have been developed to solve MAP and MPE [26,30]. Since the explanations obtained from MPE consist of the full instantiation of target variables, the explanation may be *overspecified*. Yuan and Lu [38] presented a technique named the *Most Relevant Explanation* (MRE) to produce explanations based on the maximisation of the generalised Bayes factor.

3 Most Relevant Explanation

In essence, the most relevant explanation [38] searches and enumerates all possible partial instantiations of a subset of identified unobserved variables and finds the instantiation that maximises some relevance measure. Bayesian confirmation theory is a framework within Bayesian statistics and Philosophy of Science that addresses the problem of updating hypotheses when new evidence is presented. It offers a probabilistic manner to represent the degree of evidential support. One such probabilistic tool is the *Bayes factor* [16], which measures the strength of evidence among two competing hypotheses, i.e., two competing instantiations of target variables. The generalisation of the Bayes factor allows for comparing multiple competing hypotheses [36]. The generalised Bayes factor (GBF), for observed evidence e and an explanation x, is given by

$$GBF(x;e) = \frac{P(e|x)}{P(e|\bar{x})}, \tag{1}$$

where $\bar{x}$ denotes the set of all alternative hypotheses of x. The GBF penalises more complex explanations by considering the relative magnitude of variables, retaining only relevant variables in the explanation [38]. Ranking the explanations by the GBF yields the most relevant explanation for the observed evidence. Therefore, the explanation x that maximises the GBF for the observed evidence e is considered the most relevant explanation. Mathematically, MRE is defined as

$$MRE(M;e) \equiv \arg\max_{x,\varnothing \subset X \subseteq M} GBF(x;e), \tag{2}$$

where M is the set of target nodes.

3.1 Multiple Explanations: k-MRE

Bayesian confirmation theory further recognises the importance of considering alternative explanations and comparing these explanations with regard to the observed evidence. By providing multiple explanations, end-users are encouraged to explore different possible explanations that offer a broader perspective and allow for a comprehensive evaluation of evidence [36]. This exploration allows end-users to compare various explanations in terms of each explanation's Bayes factor and allows them to assess which explanation is more strongly supported by the available evidence. Anderson *et al.* [2] study the benefit of reasoning with multiple explanations.

The solution space for the most relevant explanation contains all partial instantiations; two neighbouring explanations are connected if they have a local difference. In other words, both explanations have the same variable-state combinations except for one explanation having one less variable or the same variable with a single variable in a different state [37]. Invoking Occam's Razor [33], the set of explanations should be as simple as possible [24]. Yuan, Lim and Lu [36] propose filtering out explanations based on dominance relations such that the final explanation set is *minimal*. An explanation is considered minimal when there is no other explanation that either *strongly* or *weakly* dominates it. The remaining explanation set consists of k explanations that are diverse and representative.

We use terminology from diagnostic Bayesian networks to differentiate between the various functions of nodes [35]. *Observation* nodes represent the observed evidence, this might be a test, a symptom, or even an error message displayed by the system. *Target* nodes are variables of interest, in other words, variables we would like to investigate that could provide a deeper understanding of the observed evidence, such as a patient's health status. *Auxiliary* nodes are those nodes in the network that are neither an observation nor a target variable.

3.2 Running Example

To illustrate the concepts discussed in this paper, we use the Insurance network developed by [5]. The network consists of 27 variables, with three designated output variables: *MedCost*, *ILiCost*, and *PropCost*, illustrated in Fig. 1. Where *MedCost* refers to the cost of medical treatment, *ILiCost* the inspection cost, and *PropCost* the ratio of vehicle costs. The remaining variables are used to estimate the expected insurance claim cost for a policyholder. The Insurance network is considered one of the benchmark Bayesian networks and has been used previously to evaluate feature selection algorithms [6] and structure learning algorithms [34] that can be used for decision support and classification.

In the context of explainability, the Insurance network is appropriate as it allows for multiple scenarios, such as observing high medical expenses, accidents, or instances of theft. By generating explanations, we can gain a better understanding of the attributes that contribute to these observations. Although one may be inclined to use explainability in the Insurance network to better understand the factors that influence premium calculations and claim approvals, it can also be used to determine factors that led to the accident. For the purpose of this paper, consider the scenario where a policyholder

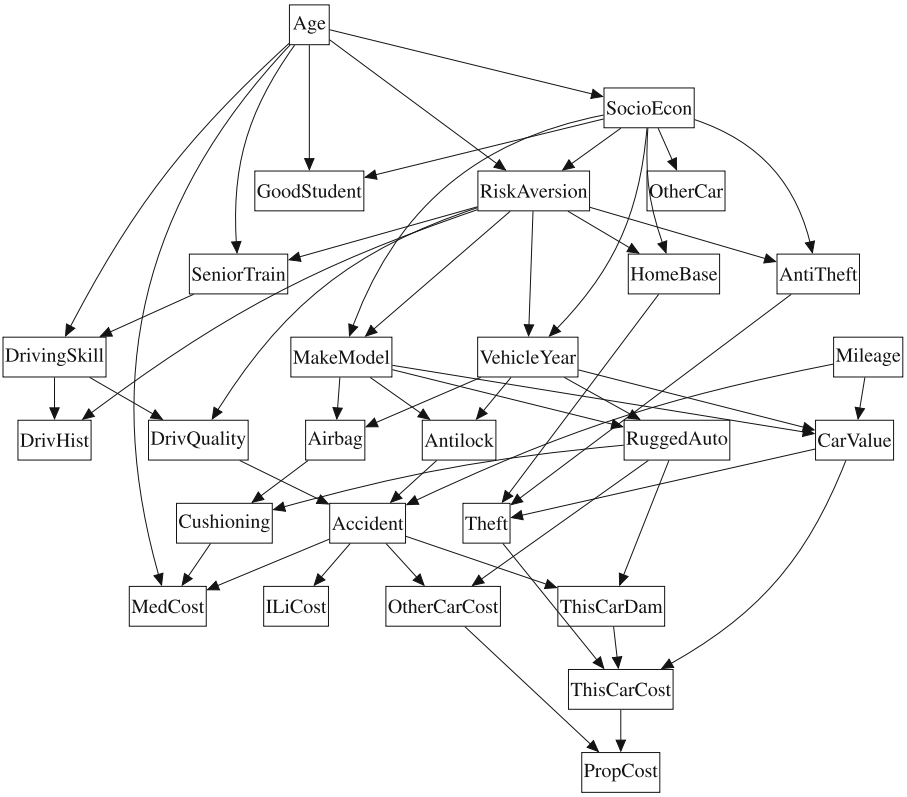
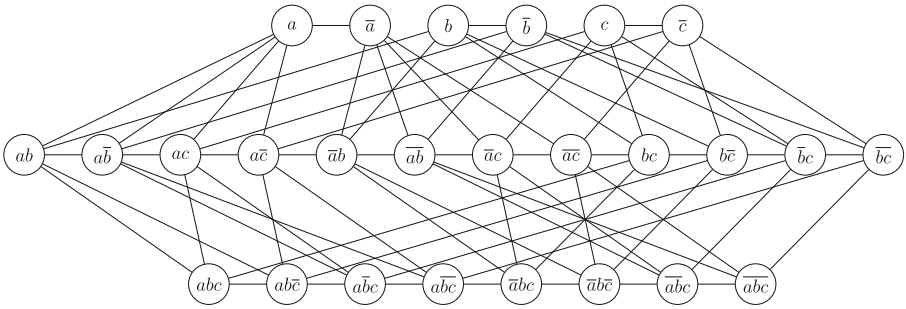


Fig. 1. Illustration of the Insurance Bayesian network from [5].

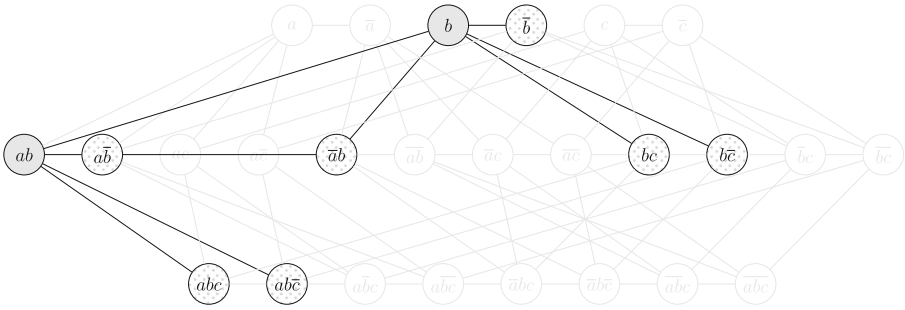
submitted a claim after their vehicle was involved in a *mild Accident*. The observation variables are $\{RuggedAuto = Tank, Accident = Mild\}$. To access the claim, the insurance company investigates three binary variables, namely *AntiLock*, *OtherCar*, and *Airbag*.

3.3 Forward Search to Solve Most Relevant Explanation

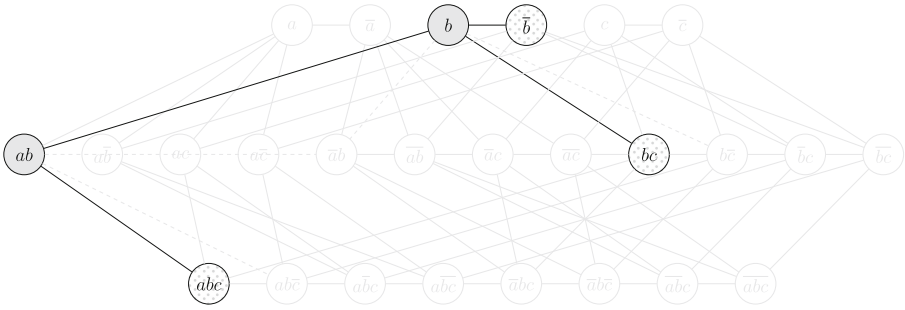
The solution space for MRE may be large since MRE is computed using all partial instantiations of a subset of unobserved variables. The solution space for the insurance running example is illustrated in Fig. 2a. Several local and approximation methods have been developed for solving MRE in Bayesian networks [35]. The inspiration behind these search methods stems from the similarity between MRE and feature selection, whose objective is to eliminate redundant and irrelevant characteristics from the set of features. The resulting subset is comprised of only relevant features [7]. However, MRE involves selecting not only the most relevant features but also their states to maximise GBF, resulting in a larger solution space. Yuan, Lim and Littman [35] adapt feature selection techniques to solve MRE, comparing local search methods such as



(a) *Antilock* (**A**), *OtherCar* (**B**), and *Airbag* (**C**).



(b) Path for *OtherCar* (**B**) through forward search.



(c) Path for *OtherCar* (**B**) through forward-gLasso search.

Fig. 2. Solution space based on three binary target variables for the most relevant explanation in the Insurance network [5], given a vehicle with *Tank* ruggedness involved in a *Mild* accident. The target variables of interest are *Antilock* (**A**), *OtherCar* (**B**), and *Airbag* (**C**). The target variables take states $\{True, False\}$, where $False$ is indicated as $\overline{a}, \overline{b}, \overline{c}$. 2b illustrates the path for *OtherCar* (**B**) through forward search while 2c illustrates the path for the same variable through forward-gLasso search.

forward, backward, tabu, and a reverse jump MCMC approximation algorithm, which shows a tradeoff between accuracy and efficiency. Similarly, [40] developed a hierarchical beam search that, while more accurate than the forward and tabu search, is slower. This positions the forward search as a promising starting point for our method, which aims to improve computational performance while maintaining accuracy.

In essence, one or more starting solutions are invoked to initiate the forward search. For each initial solution, the solution is improved by either *adding* an additional feature or by *changing* the state of an existing feature in the solution [35], we define the former as *add-one* neighbours and the latter as *change-one* neighbours. Although there are two ways to initialise a starting solution; *empty initialisation* and *best pivot*, we will focus on the best pivot starting solution, following the authors of [35], who demonstrated that the best pivot yields better performance. Here, we set the target features to their most likely state as a starting point. Figure 2b illustrates the forward search for three target variables in the Insurance Bayesian network [5]. In particular, it shows the search path for *OtherCar*. The lattice structure divides the nodes into layers. Nodes in layer 1 consist of singular instantiations, whereas nodes in the bottom layer consist of full instantiations. The algorithm is given in Algorithm 1[1].

Algorithm 1. Forward-Search Algorithm

Input: Bayesian network B, set of observation variables E, and a set of target variables X.
Output: k-MRE solution.

1: Initialise the starting solution set I with the best pivot initialisation rule.
2: Initialise the current best solution, $y_{best} = \emptyset$ for each starting solution s
3: **for** each starting solution s in I do **do**
4: $y = s$
5: **repeat**
6: Find the neighbouring solution set N of y by either *changing* the state of a single variable or by *adding* an additional target variable with *any* state.
7: Compute the GBF score for each solution in N.
8: Filter the neighbouring solution set N based on dominance relations.
9: Update y if the best solution in N has a *higher* GBF score.
10: **until** y stops updating
11: **if** $GBF(y) > GBF(y_{best})$ **then**
12: $y_{best} = y$
13: **end if**
14: **end for**
15: **return** y_{best}

Exhaustive search algorithms are computationally feasible only for low-dimensional models. While the forward search algorithm is a helpful tool for exploring and identifying relevant features, it faces a notable challenge when applied to high-dimensional data [25], particularly in the Bayesian network domain. As the complexity increases, the number of potential solutions that could explain the observed evidence grows rapidly. As a result, finding the optimal solution can be challenging, especially if the algorithm is not able to efficiently prune the solution space. We propose incorporating the Graphical Least Absolute Shrinkage and Selection Operator (graphical Lasso or gLasso) [10], a statistical neighbourhood selection method, to prune the solution space.

[1] Algorithm adapted from [35].

4 A Novel Forward-gLasso Search for Solving MRE

Given that the solution space for the most relevant explanation can consist of both full and partial instantiations, it becomes computationally infeasible for moderate to larger Bayesian networks. As such, it makes sense to prune the solution space to a subset of instantiations that could explain the observed evidence. By doing so, we can ensure that the current best solution is compared to a smaller, more concise, set of neighbours. To achieve this goal, we propose leveraging the gLasso algorithm. The integration of gLasso into the forward search algorithm is a symbiotic approach that combines the strengths of both methods. The forward search algorithm provides a systematic way to explore the solution space, while gLasso helps identify the most relevant dependencies. Moreover, the sparsity induced by gLasso allows us to identify and select a reduced set of relevant instantiations that exhibit dependencies, thus reducing the computational burden and enhancing the explainability of the results.

4.1 Graphical Lasso

Several researchers have explored using L_1 regularisation in estimating sparse undirected graphical models [3,8,10,25]. These models employ undirected graphs to define the conditional independence relationships between the variables. The default model assumes a multivariate Gaussian distribution. Consider the matrix $X_{n\times p}$, with n observations from p features with mean μ and covariance matrix Σ. Note that a zero element in Σ^{-1} indicates conditional independence between the two variables. Accordingly, sparser graphs are yielded when zero off-diagonal elements in Σ^{-1} increases. Define the precision matrix as $\Theta = \Sigma^{-1}$ and let S denote the empirical covariance matrix, with $S = \frac{1}{n}X^T X$. The gLasso problem is then defined as the maximisation of the penalised log-likelihood over non-negative definite matrix Θ,

$$\log \det\Theta - \text{tr}(S\Theta) - \lambda||\Theta||_1, \tag{3}$$

where $||\Theta||_1$ is the L_1 norm, λ is a tuning parameter controlling matrix sparsity, and tr represents the trace of a matrix.

4.2 Search Strategy

The forward search algorithm serves as the foundational framework for the proposed forward-gLasso algorithm. The forward search, introduced in Sect. 3.3, provides the essential architecture that guides the proposed algorithm through the solution space. We can think of the forward search as a puzzle where we need to assemble many pieces to obtain the full picture. We start with one piece and gradually add one at a time to find the best-fitting combination. Referring to Fig. 2b, we initiate the search process by assigning the most likely state to each target variable. In this case, *OtherCar* is set to *True*, i.e., b – node filled with a solid grey in layer 1. We attempt to gradually improve the solution through the addition of one variable to the solution or by altering the state of one variable. These paths are indicated by solid lines. The addition of one variable is

shown in layer 2 of the lattice structure. We evaluate the solution against the neighbouring solutions at each iteration according to the GBF. We stop the process if the current solution has a higher GBF than the best neighbour. In contrast, if the best neighbour has a higher GBF than the current best solution, we update the current best solution and repeat the process. Eventually, we reach a point where adding more variables (or changing the state of a single variable) does not improve the explanatory power of the explanation. Here, the algorithm visits all possible instantiations of variables, which can be computationally inefficient.

Neighbourhood Selection. To avoid visiting all possible instantiations of variables and the possibility of *overspecified* explanations, we apply the gLasso at each iteration of the forward search to prune the neighbourhood, N. In general, gLasso adds variables to the solution with the strongest connections to the already included variables. In other words, gLasso helps us identify which variables (and their states) are closely related. The gLasso can contribute to variable selection through two approaches. Firstly, it facilitates the precision matrix estimation, which contains information about the dependencies among the variables. Variables that display strong connections in the precision matrix will likely hold significance in the model. Secondly, variables can be selected with the graphical lasso by using the penalised log-likelihood score. The penalised log-likelihood measures how well the model fits the data. The variables with the highest penalised log-likelihood score are the most likely to be important for the explanation. We will focus on estimating the precision matrix to select the neighbours. Using gLasso to select the neighbourhood ensures that the variables we include are not only individually relevant but also contribute meaningfully to the explanatory power of the explanation.

Once the neighbourhood, N, has been pruned, we can continue the forward search by comparing the current best solution with the updated neighbouring set. Figure 2c illustrates the path for *OtherCar* using the proposed forward-gLasso algorithm. At initialization, *OtherCar* is set to *True*, i.e., b. Notice here that the path to $b\overline{c}$ and $\overline{a}b$ are dashed. During the forward search, these paths were solid. However, applying gLasso highlighted that these variables are conditionally independent of the current best solution and can be eliminated from the neighbouring set. We see here that the neighbouring set for the first iteration was reduced from the initial 5 to 3.

Estimating the Precision Matrix. We will use a GBF score matrix to approximate the precision matrix used as input for the gLasso algorithm. Each cell in the score matrix corresponds to a neighbour's GBF score in the neighbouring solution set N. The GBF scores are calculated using Eq. 1. We split each neighbour into two components based on neighbour type to get off-diagonal scores. For *add-one* neighbours, the first component consists of the current best solution, and the second component consists of the additional variable. Then, for *change-one* neighbours, the first component consists of all variables in that neighbour for which the state did not change, whereas the second component consists of the variable for which the state changed. To illustrate this, let's consider the current best solution, ab, in Fig. 2c. Our *add-one* neighbours are $\{abc, ab\overline{c}\}$ and the *change-one* neighbours are $\{a\overline{b}, \overline{a}b\}$. We can split the neighbours as follows:

$ab\overline{c} = \{ab, \overline{c}\}$, $abc = \{ab, c\}$, $\overline{a}b = \{\overline{a}, b\}$, and $a\overline{b} = \{\overline{b}, a\}$. The resulting score matrix is symmetric with the following row and column names: ab, $\overline{c}$, c, $\overline{a}$, b, $\overline{b}$, a:

$$\Theta = \begin{array}{c} \\ ab \\ \overline{c} \\ c \\ \overline{a} \\ b \\ \overline{b} \\ a \end{array} \begin{array}{c} \begin{array}{ccccccc} ab & \overline{c} & c & \overline{a} & b & \overline{b} & a \end{array} \\ \begin{pmatrix} 1.61 & \mathbf{0.65} & \mathbf{1.66} & 0 & 1.61 & 0 & 1.61 \\ \mathbf{0.65} & 0.67 & 0 & 0.68 & 0.91 & 0.56 & 0.54 \\ \mathbf{1.66} & 0 & 1.49 & 1.05 & 1.54 & 0.97 & 1.66 \\ 0 & 0.68 & 1.05 & 0.63 & \mathbf{1.01} & 0.60 & 0 \\ 1.61 & 0.91 & 1.54 & \mathbf{1.01} & 1.53 & 0 & 1.61 \\ 0 & 0.56 & 0.97 & 0.60 & 0 & 0.65 & \mathbf{1.25} \\ 1.61 & 0.54 & 1.66 & 0 & 1.61 & \mathbf{1.25} & 1.60 \end{pmatrix} \end{array},$$

where boldface values represent the generalised Bayes factor scores for the neighbours of ab.

5 Experiments

5.1 Experimental Design

We evaluate the performance of the forward-gLasso algorithm on a set of benchmark Bayesian networks: Asia [23], Alarm [4], Circuit [29], Hepar2 [18], and Insurance [5]. Asia, Alarm, Hepar2, and Insurance are included in the Bayesian network data repository [31]. We used the target and observation nodes described in the relevant literature for each network. If the diagnostic node groupings were not available for a particular network, we utilised alternative groupings. Table 1 provides an overview of the benchmark Bayesian networks. Following a similar approach as [35], we will use each benchmark network as a generative model to generate test cases for evaluation. We limit the generated test cases to only include unique observations, as these observation nodes will serve as evidence. In smaller networks, like Asia and Circuit, where there is only one binary observation, we have a total of two distinct test cases. Since the Asia network allows for two possible diagnostics, running experiments on both results in four distinct test cases.

We first evaluate the proposed forward-gLasso according to two performance indicators, namely *cases solved exactly* (CSE) and *average execution time* (AET). Since the graphical Lasso includes a non-negative tuning parameter to encourage sparsity, we implemented the forward-gLasso algorithm with three different tuning values: $\lambda = 0.01$, $\lambda = 0.001$, $\lambda = 0.0001$. The results obtained from a brute-force approach implemented in `R` were used as ground truth. Thereafter, we report on the proposed forward-gLasso search algorithm's ability to prune the neighbourhood size.

Table 1. Summary of the benchmark networks used in the experiments.

Network	Type	Nodes	Arcs	Targets	Observations
Asia	Small	8	8	3	1
Circuit	Small	10	11	4	1
Insurance	Medium	27	52	6	5
Alarm	Medium	37	46	8	16
Hepar2	Large	70	123	7	63

Lastly, we provide the explanations obtained through the forward-gLasso for the running example introduced in Sect. 3.2. The experiments were performed in a `R` environment with R version 4.3.1 for programming. The Bayesian networks were loaded through the `bnlearn` [31] and `gRain` [14] packages. Graphical Lasso was performed through the `glassoFast` [32] package. The test cases were generated using the `rbn` function from `bnlearn` , with the loaded network structure and the number of samples to generate as input. We restricted the generated cases to only include unique observations, as previously described. Where applicable, we sampled 200 of these test cases without replacement, with a seed of 13.

5.2 Experimental Results: Computational Efficiency

Table 2 provides a detailed breakdown of the computational results. The table includes the Bayesian network, the total test cases solved exactly (CSE), and the average execution time in seconds (AET). Boldface entries highlight the forward-gLasso variant that solved the same number of test cases exactly as the forward search but achieved greater computational efficiency with a lower average execution time. Overall, the forward-gLasso$_{\lambda=0.0001}$ stands out as it successfully solves the same number of test cases as the forward search, but in less time. This finding underscores the computational advantages of including gLasso in the forward search. However, it is worthwhile to note that while the average execution time for the forward-gLasso implementations is generally quicker than the forward search, the forward-gLasso$_{\lambda=0.01}$ and forward-gLasso$_{\lambda=0.001}$ variants tend to solve fewer cases exactly, particularly in larger networks. This observation sheds light on the trade-off between computational efficiency and accuracy. When using a smaller value of the regularisation parameter λ, the algorithms achieve higher computational efficiency but sacrifice some degree of accuracy in the results.

5.3 Experimental Results: Neighborhood Reduction

Table 3 displays the average number of neighbours visited for each algorithm. Boldface entries indicate the smallest average neighbourhood size among the algorithms. Overall, the results demonstrate the efficiency of the proposed forward-gLasso algorithm in reducing neighbourhood sizes compared to the forward search algorithm. This is particularly significant in scenarios where computational efficiency and resource utilisation are important. By reducing the number of neighbours to explore, the forward-gLasso algorithms simplify the process of finding the most relevant explanation within

Table 2. Comparison of test cases solved exactly (CSE) and the average execution time (AET) of each algorithm in seconds.

	Forward		F-gLasso$_{\lambda_{0.01}}$		F-gLasso$_{\lambda_{0.001}}$		F-gLasso$_{\lambda_{0.0001}}$	
	CSE	AET	CSE	AET	CSE	AET	CSE	AET
Asia	4	0.236	4	0.159	**4**	**0.155**	4	0.166
Circuit	2	0.678	**2**	**0.469**	2	0.494	2	0.511
Insurance	199	14.887	195	9.663	192	10.375	**199**	**11.873**
Alarm	200	16.943	193	7.307	199	9.952	**200**	**12.051**
Hepar2	199	10.719	195	4.882	195	6.779	**199**	**7.985**

Bayesian networks. It is important to acknowledge the exception found in the Asia network, which is considered a small network. In this specific instance, the forward-gLasso$_{\lambda_{0.001}}$ variant displayed a larger average neighbourhood size in contrast to other versions of the algorithm as well as the forward search algorithm. This highlights the sensitivity of the choice of regularisation parameter, λ, as it determines the level of sparsity in the estimated matrix, thereby influencing the inclusion or exclusion of neighbours.

Table 3. Comparison of the average neighbourhood size of each algorithm.

	Forward	F-gLasso$_{\lambda_{0.01}}$	F-gLasso$_{\lambda_{0.001}}$	F-gLasso$_{\lambda_{0.0001}}$
Asia	13.75	**13.25**	14.25	13.75
Circuit	25.50	**24**	25.50	25.50
Insurance	120.76	**92.27**	112.05	118.83
Alarm	123.14	**88.02**	106.23	118.28
Hepar2	105.48	**74.79**	92.94	102.79

Let's take a closer look at the CSE and AET for the Alarm network presented in Table 2 and the average neighbourhood size presented in Table 3. The forward-gLasso$_{\lambda=0.0001}$ solves cases efficiently while maintaining high accuracy. In contrast, the forward-gLasso$_{\lambda=0.01}$ and forward-gLasso$_{\lambda=0.001}$ variants, while faster, do not attain the same level of accuracy. This echoes our observation in Table 3 when comparing the neighbourhood pruning capabilities of each variant. Despite a reduction in average neighborhood size from 118.28 to 106.225, the forward-gLasso$_{\lambda=0.001}$ solves one less case than the forward-gLasso$_{\lambda=0.0001}$.

5.4 Experimental Results: Most Relevant Explanation According to Forward-gLasso

We also report the explanations obtained through the forward-gLasso search algorithm on our running example and a second scenario based on the Insurance network with 5 targets and 5 observation variables.

Scenario 1: Running Example. In this example, a policyholder submitted a claim after their vehicle, with tank-level ruggedness, was involved in a mild accident. The three target variables are *Antilock*, *OtherCar*, and *Airbag*. The minimal explanations are given in Table 4. In this scenario, the most relevant explanation for the observed evidence is the presence of an anti-lock braking system and an airbag with a GBF score of 1.645, followed by the presence of an anti-lock braking system and the involvement of a second vehicle, and lastly the presence of an airbag and the involvement of a second vehicle.

Table 4. Set of minimal explanations for the Insurance network using the forward-gLasso algorithm.

AntiLock	OtherCar	Airbag	Generalised Bayes Factor
True		True	1.657
True	True		1.613
	True	True	1.539

Note that in the running example, we only include binary target variables. We perform the experiment on a larger set of observation and target nodes from the Insurance network to further investigate the most relevant explanations obtained with the forward-gLasso algorithm.

Scenario 2: Including Socio-Economic and Demographic Factors. Suppose we observe a *moderate* accident and include additional socio-economic and demographic factors related to the driver as evidence, such as *Age*, *SocioEcon*, *HomeBase*, and *DrivHist*. In this scenario, the policyholder submitting a claim is a middle-class adult based in a rural neighbourhood with zero prior accidents. We are interested in attributes of the vehicle, such as *Antilock*, *Airbag*, *VehicleYear*, *MakeModel*, and *RuggedAuto* to explain the observed evidence. The minimal explanations, using $\lambda = 0.001$, obtained through the proposed forward-gLasso are given in Table 5. The first explanation consists of a singular instantiation, *MakeModel=FamilySedan* with a GBF of 3.074. The remaining explanations are partial instantiations of the target variables and no explanation consists of a full instantiation, i.e., all five target variables. In this scenario, the simplest observation carries the highest explanatory power and the more complex explanations have lower explanatory power. This observation agrees with [24] who argue that explanations should be simple since this is a requirement of comprehensibility. Accordingly, the set of explanations provided are considered *minimal* explanations, since neither of the explanations are either strongly or weakly dominated by another explanation. As such, the explanations obtained through the forward-gLasso algorithm retain the properties of a good explanation, namely *preciseness* and *conciseness*.

Table 5. Set of minimal explanations for Scenario 2 of the Insurance network using the forward-gLasso algorithm.

AntiLock	Airbag	VehicleYear	MakeModel	RuggedAuto	Generalised Bayes Factor
			FamilySedan		3.074
False		Older		Tank	2.173
False	False			Tank	2.167
		Older		Football	1.480
	False			Football	1.449

6 Conclusion

The primary contribution of this paper is the introduction of the proposed forward-gLasso search algorithm, which forms the foundation of an `R` package[2] designed to facilitate explanation methods for Bayesian networks. The availability of such an `R` package will prove highly advantageous as it will enhance the accessibility of explanation methods in Bayesian networks. Currently, these methods are predominantly accessible through proprietary software.

Using the gLasso, we can identify and select a reduced set of neighbours based on the most relevant dependencies, thus pruning the neighbourhood and decreasing the complexity of the most relevant explanation. Previous work shows that the most relevant explanation exhibits the conditions of a *good* explanation, i.e., preciseness and conciseness. Integrating the graphical Lasso into the forward search algorithm retains these attributes and enhances the explanations by focusing on key relationships. Bayesian networks inherently handle uncertainty, and the gLasso can capture uncertainty associated with conditional dependencies. The zero entries in the sparse inverse covariance matrix obtained from the gLasso represent conditional independence between variables, indicating the level of uncertainty in their relationships given the observed evidence.

We implemented the proposed forward-gLasso using three different regularisation parameters: $\lambda = 0.01$, $\lambda = 0.001$, and $\lambda = 0.0001$. The computational efficiency of the three implementations is compared to the benchmark forward search algorithm while using the explanations obtained through a brute-force search as ground truth. The experimental results on five benchmark Bayesian networks show an improvement in execution time compared to the benchmark forward search algorithm. At the same time, the accuracy is preserved, particularly in the case of the forward-gLasso$_{\lambda_{0.0001}}$. Furthermore, our results emphasise the potential benefits of harnessing the neighbourhood pruning capabilities of the gLasso. This leads to the exploration of other neighbourhood reduction techniques for future work. Other future work includes investigating the trade-off between accuracy and efficiency.

[2] The `R` package developed alongside this work can be downloaded using `devtools::install_github('iEna101/XBN')`.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)
2. Anderson, A., et al.: Mental models of mere mortals with explanations of reinforcement learning. ACM Trans. Interact. Intell. Syst. (TiiS) **10**(2), 1–37 (2020)
3. Banerjee, O., El Ghaoui, L., d'Aspremont, A.: Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. J. Mach. Learn. Res. **9**, 485–516 (2008)
4. Beinlich, I.A., Suermondt, H.J., Chavez, R.M., Cooper, G.F.: The ALARM monitoring system: a case study with two probabilistic inference techniques for belief networks. In: Hunter, J., Cookson, J., Wyatt, J. (eds.) AIME 1989. LNMED, vol. 38, pp. 247–256. Springer, Heidelberg (1989). https://doi.org/10.1007/978-3-642-93437-7_28
5. Binder, J., Koller, D., Russell, S., Kanazawa, K.: Adaptive probabilistic networks with hidden variables. Mach. Learn. **29**, 213–244 (1997)
6. Broom, B.M., Do, K.A., Subramanian, D.: Model averaging strategies for structure learning in Bayesian networks with limited data. BMC Bioinform. **13**, 1–18 (2012)
7. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. Comput. Electr. Eng. **40**(1), 16–28 (2014)
8. Dahl, J., Vandenberghe, L., Roychowdhury, V.: Covariance selection for nonchordal graphs via chordal embedding. Optimiz. Methods Softw. **23**(4), 501–520 (2008)
9. Derks, I.P., de Waal, A.: A taxonomy of explainable Bayesian networks. In: Gerber, A. (eds.) SACAIR 2020. CCIS, vol. 1342, pp. 220–235. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-66151-9_14
10. Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical Lasso. Biostatistics **9**(3), 432–441 (2008)
11. Gámez, J.A.: Abductive inference in Bayesian networks: a review. In: Gámez, J.A., Moral, S., Salmerón, A. (eds.) Advances in Bayesian Networks. STUDFUZZ, vol. 146, pp. 101–120. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-39879-0_6
12. Han, H., Liu, X.: The challenges of explainable AI in biomedical data science (2021)
13. Hennessy, C., Diz, A.B., Reiter, E.: Explaining Bayesian networks in natural language: state of the art and challenges. In: 2nd Workshop on Interactive Natural Language Technology for Explainable Artificial Intelligence, pp. 28–33 (2020)
14. Højsgaard, S.: Graphical independence networks with the gRain package for R. J. Stat. Softw. **46**(10), 1–26 (2012). https://doi.org/10.18637/jss.v046.i10
15. Jensen, F.V., Nielsen, T.D.: Bayesian Networks and Decision Graphs. ISS, vol. 2. Springer, New York (2007). https://doi.org/10.1007/978-0-387-68282-2
16. Kass, R.E., Raftery, A.E.: Bayes factors. J. Am. Stat. Assoc. **90**(430), 773–795 (1995)
17. Keppens, J.: Explainable Bayesian network query results via natural language generation systems. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law, pp. 42–51 (2019)
18. Oniśko, A., Druzdzel, M.J., Wasyluk, H.: Extension of the HEPAR II model to multiple-disorder diagnosis. In: Kłopotek, M., Michalewicz, M., Wierzchoń, S.T. (eds.) Intelligent Information Systems. AINSC, vol. 4, pp. 303–313. Springer, Heidelberg (2000). https://doi.org/10.1007/978-3-7908-1846-8_27

19. Koopman, T., Renooij, S.: Persuasive contrastive explanations for Bayesian networks. In: Vejnarová, J., Wilson, N. (eds.) ECSQARU 2021. LNAI, vol. 12897, pp. 229–242. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86772-0_17
20. Korb, K.B., Nicholson, A.E.: Bayesian Artificial Intelligence. CRC Press (2010)
21. Kwisthout, J.: Most inforbable explanations: finding explanations in Bayesian networks that are both probable and informative. In: van der Gaag, L.C. (eds.) ECSQARU 2013. LNAI, vol. 7958, pp. 328–339. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39091-3_28
22. Lacave, C., Díez, F.J.: A review of explanation methods for Bayesian networks. Knowl. Eng. Rev. **17**(2), 107–127 (2002)
23. Lauritzen, S.L., Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems. J. Roy. Stat. Soc.: Ser. B (Methodol.) **50**(2), 157–194 (1988)
24. Lötsch, J., Kringel, D., Ultsch, A.: Explainable artificial intelligence (XAI) in biomedicine: making AI decisions trustworthy for physicians and patients. BioMedInformatics **2**(1), 1–17 (2022)
25. Meinshausen, N., Bühlmann, P.: High-dimensional graphs and variable selection with the Lasso. Ann. Stat. **34**(3), 1436–1462 (2006)
26. Park, J.D.: MAP complexity results and approximation methods. In: Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence, UAI 2002, pp. 388–396. Morgan Kaufmann (2002)
27. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1988)
28. Peng, Y., Reggia, J.A.: Abductive Inference Models for Diagnostic Problem-Solving. Springer, New York (2012)
29. Poole, D.L., Provan, G.M.: What is an optimal diagnosis? In: Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence, UAI 1990, pp. 46–53. Elsevier (1990)
30. Santos Jr, E.: On the generation of alternative explanations with implications for belief revision. In: Uncertainty Proceedings 1991, pp. 339–347. Elsevier (1991)
31. Scutari, M.: Learning Bayesian networks with the bnlearn R package. J. Stat. Softw. **35**(3), 1–22 (2010). https://doi.org/10.18637/jss.v035.i03
32. Sustik, M.A., Calderhead, B., Clavel, J.: glassoFast: fast graphical LASSO (2023). https://CRAN.R-project.org/package=glassoFast. R Package Version 1.0.1
33. Thorburn, W.M.: The myth of Occam's razor. Mind **27**(107), 345–353 (1918)
34. Tsamardinos, I., Brown, L.E., Aliferis, C.F.: The max-min hill-climbing Bayesian network structure learning algorithm. Mach. Learn. **65**, 31–78 (2006)
35. Yuan, C., Lim, H., Littman, M.L.: Most relevant explanation: computational complexity and approximation methods. Ann. Math. Artif. Intell. **61**(3), 159–183 (2011)
36. Yuan, C., Lim, H., Lu, T.C.: Most relevant explanation in Bayesian networks. J. Artif. Intell. Res. **42**, 309–352 (2011)
37. Yuan, C., Liu, X., Lu, T.C., Lim, H.: Most relevant explanation: properties, algorithms, and evaluations. In: Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence, UAI 2009, pp. 631–638 (2009)
38. Yuan, C., Lu, T.C.: A general framework for generating multivariate explanations in Bayesian networks. In: AAAI, pp. 1119–1124 (2008)
39. Zhu, X., Yuan, C.: Exact algorithms for MRE inference. J. Artif. Intell. Res. **55**, 653–683 (2016)
40. Zhu, X., Yuan, C.: Hierarchical beam search for solving most relevant explanation in Bayesian networks. J. Appl. Logic **22**, 3–13 (2017)



Assessing Multilinguality of Topic Models on a Short-Text South African Languages Dataset

Darren Craig Roos(✉) and Katherine Mary Malan

Department of Decision Sciences, University of South Africa, Pretoria, South Africa
21001154@mylife.unisa.ac.za

Abstract. Topic modelling is an important technique in natural language processing, enabling the extraction of underlying themes from large textual corpora. Measuring the performance of topic models is largely an open problem. In addition, the increasing number of multilingual corpora makes it important to be able to measure the multilingual performance of topic models. Current coherence metrics either fail to take into account multilinguality, or they rely on expensive parallel corpora. To address these gaps, we introduce a novel performance metric called Multilingual Topic Coherence (MTC). MTC measures the extent to which topics represent multilingual ideas without using a parallel corpus. It combines word embedding-based coherence with a multilinguality score. To test MTC's effectiveness, we constructed a short-text dataset comprising texts from nine of South Africa's official languages. We compared the MTC performance of three widely used topic models: Latent Dirichlet Allocation (LDA), ZeroShotTM, and BERTopic. Our findings indicate that contextualized topic models, such as ZeroShotTM and BERTopic, outperform traditional models like LDA in terms of both coherence and multilinguality. The MTC metric provides a more nuanced evaluation, highlighting the strengths and weaknesses of each model in handling multilingual topics. This research advances the understanding and application of topic modelling in multilingual and low-resource environments, offering a robust metric for future evaluations. Code used to conduct this study is provided (https://github.com/AlgorithmicAmoeba/multilingual_topic_coherence).

Keywords: Natural language processing · Topic modelling · South African languages · topic model performance · Multilingual topic coherence

1 Introduction

Topic modelling is a technique in natural language processing that aims to uncover underlying themes within a large textual corpus. By identifying which thematic "topics" are present in each document, it utilizes words from the corpus to describe each topic. For example, the words "apple", "banana", and

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 38–52, 2025.
https://doi.org/10.1007/978-3-031-78255-8_3

"pear" might collectively represent a topic related to fruit. As an unsupervised machine learning technique, topic modelling is particularly valuable for mining information from extensive collections of unlabelled documents. Its applications span various fields, including document retrieval [23,41] and text summarization [9,13]. Furthermore, the insights derived from topic modelling can assist in identifying trends and patterns within specific domains such as law [28] and education [39].

Topic models can also be applied to multilingual corpora, which are becoming increasingly more common [40]. However, naïvely applying traditional topic modelling methods to multilingual corpora often fails. This is because traditional methods have no way of connecting similar words/concepts across languages. To address this limitation, multilingual topic models have been developed to capture topics that remain consistent across different languages. This enables a comprehensive overview of document structures within multilingual datasets by learning multilingual topics. These characteristics make multilingual topic models useful for various cross-lingual tasks, such as translation [18] and cultural analysis [33].

Measuring the performance of topic models remains an important area of research. Most methods rely on evaluating the coherence of topics based on the frequency of co-occurrences of words in a reference corpus [16]. However, these methods become inaccurate when dealing with multilingual corpora, because they have no way to translate words/concepts. This limitation makes it challenging to accurately assess the coherence of topics in a multilingual context. Metrics that do measure multilinguality make use of parallel corpora—a corpus where the same documents are translated across multiple languages. These corpora are costly to construct [24] or do not reflect modern language usage, such as those based on translations of the Bible [14].

To address these concerns we make the following contributions: (i) we develop a novel performance metric called Multilingual Topic Coherence (MTC) that measures the extent to which topics represent multilingual ideas without making use of a parallel reference corpus; (ii) we construct a short-text dataset to test the effectiveness of our MTC metric comprising texts from nine of the official South African languages; (iii) we compare the MTC performance of three widely used topic models on the constructed multilingual dataset. Our contributions aim to advance the understanding and application of topic modelling in multilingual and low-resource environments.

The remainder of this paper is set out as follows: Sect. 2 gives background on the subject matter used in the paper and related work; Sect. 3 details how we constructed the short-text South African languages dataset; Sect. 4 introduces our novel performance metric; Sect. 5 describes the experimental work done and presents a discussion on the results. Section 6 concludes the paper and gives directions for future areas of research.

2 Background

This section outlines necessary background information on topic models and topic model performance metrics.

2.1 Topic Models

Traditional topic models, such as probabilistic latent semantic indexing [15] and Latent Dirichlet Allocation (LDA) [3,4], assume that a document is generated by drawing words from a set of underlying probability distributions. Subsequently, inference is done to find the parameters of these distributions that best match the observed data. LDA models latent topics using a Dirichlet distribution. It also models documents as multinomial distributions over topics and topics as multinomial distributions over words.

LDA represents documents as a bag of words, a method that disregards the order and contextual information of words within a document and focuses solely on the frequency of each word. Creating a bag of words representation begins with the selection of a vocabulary, a list of words derived by identifying and selecting the most common terms from a collection of documents. The document's bag of words representation is then constructed by counting how many times each word from the vocabulary appears within it. This vector representation serves as a fundamental step in making textual data amenable to machine learning.

Modern topic models represent documents using embeddings from a Pre-trained Language Model (PLM). PLMs, often taking the form of deep neural networks, are trained on extensive datasets that encompass a wide array of text sources, including books, articles, and internet forums [29,38]. This gives them the ability to capture semantic and contextual information [30,32]. This knowledge in language models is accessible by probing the network's activations [26]. These activations, also known as embeddings, are representations of the model's understanding of the input data. These embeddings allow the model to relate words and concepts based on their contextual usage, facilitating tasks such as word analogy and semantic similarity assessments [27].

Topic models that harness PLM embeddings are often referred to as contextualized topic models [1]. These models take advantage of the deep contextual knowledge of language inherent in PLMs, enabling them to generate topics that are more semantically coherent and contextually relevant. Contextual embeddings allow contextualized topic models to operate on multilingual corpora due to the ability of the PLMs to create embeddings for multiple languages. Two popular contextualized topic models are ZeroShotTM [2] and BERTopic [12].

ZeroShotTM [2] is a neural topic model that uses a variational autoencoder (VAE) [17,22] for inference. The model uses Sentence BERT (SBERT) [32] embeddings to represent the documents as the input to the VAE. SBERT is a BERT-based model fine-tuned to generate semantically relevant embeddings. The autoencoder then aims to output the document's bag of words rep-

resentation. The topic representations are derived using weights in the decoder network.

BERTopic [12] similarly embeds documents using SBERT. It then performs a dimensionality reduction on these embeddings before passing them through a clustering step. The clusters are taken to represent topics. Topic representations are built using a custom Term Frequency–Inverse Document Frequency (TF–IDF) metric, which serves as a measure to gauge the significance of individual words within a given corpus relative to the entire collection of documents.

Several works have been published comparing traditional and contextualized topic models on various datasets. Omizo [25] compares LDA and BERTopic on a dataset composed of forum posts for cities in the United States. Defersha et al. [7] compare the performance of several traditional topic models, including LDA, and BERTopic on a dataset comprised of Ethiopian language Facebook posts. Similarly, Masethe et al. [20] apply these topic models to extract insights from a corpus of Northern Sotho data. To the best of the authors' knowledge this is the first study performing a comparison on a multilingual South African languages dataset.

2.2 Performance Metrics

Measuring the performance of a topic model is an important step in determining its effectiveness in extracting meaningful insights from text data. This evaluation process is essential for researchers and practitioners who aim to quantitatively assess the model's capability in revealing coherent and distinct topics. By doing so, they can ensure that the model aligns with their intended analytical goals and produces valuable results. However, evaluating the performance of topic models is challenging due to the unsupervised nature of the tasks, which differ from supervised learning tasks where model accuracy can be readily measured.

Topic coherence is a metric for assessing the interpretability of topics generated by a topic model. It involves measuring the similarity of the words within a topic. For instance, a set of words like "hospital", "doctor", and "scrub" is considered more coherent than the set "nurse", "car", and "leaf".

Normalized Pointwise Mutual Information (NPMI) [5] is a commonly used metric. NPMI computes the likelihood of word pairs occurring together within a topic by referencing a larger corpus, for example Wikipedia documents. However, this approach is computationally expensive, since it requires the pairwise co-occurrence to be calculated for every pair of words in the reference corpus. It also cannot be applied to multilingual corpora, and there is debate as to whether NPMI scores align with human judgments of topic coherence [19], or not [16].

In the same vein as with topic models, word embedding-based similarity measures [35] overcome the computational burden faced by NPMI. They make use of word embeddings to capture semantic relationships between words. In word embedding spaces, vector representations of the words appearing in similar contexts tend to be close to each other. This, along with further structure, allows semantic information to be encoded, for example: "king" - "man" + "woman" $\approx$ "queen" [21].

One such metric is the Embedding Coherence (EC) [36]. The measure involves taking the mean pairwise similarity between word embeddings of the words in a topic. While these embeddings are capable of operating in a multilingual context (if multilingual embeddings are used), the metric itself does not account for multilinguality. For N topics $(t_1, \ldots, t_N)$, each consisting of T words $(w_1, \ldots, w_T)$, the metric can be calculated using a similarity measure $(\text{sim}(\cdot))$ (usually, the cosine similarity) that measures how similar two word embeddings $E(w_i)$ are. The calculation is shown in Eq. 1. Note, that if there are T words in a topic, there are $M = \frac{(T)(T-1)}{2}$ pairs of words.

$$\begin{aligned} \text{EC} &= \frac{1}{N} \sum_{i=1}^{N} C_{t_i} \\ C_{t_i} &= \frac{1}{M} \sum_{j=1}^{T-1} \sum_{k=j+1}^{T} \text{sim}(E(w_{t_i,j}), E(w_{t_i,k})) \end{aligned} \tag{1}$$

Terragni et al. [35] introduce several other embedding-based similarity metrics, however, they mainly measure the inverse topic diversity, rather than topic coherence. Again, while these metrics might be useful in multilingual contexts, they have, to the best of the authors' knowledge, yet to be applied in such a context.

Hao et al. [14] introduce a multilingual topic coherence metric called crosslingual NPMI. Like the NPMI, the crosslingual NPMI also relies on a large reference corpus. Their reference corpus is required to be a multilingual parallel corpus, that is each document in the corpus, must have a translation in every one of the languages that the metric will be applied to. This is problematic since such a large parallel corpus is difficult to construct. Their solution is to use the Christian Bible, which has been translated into many languages. However, this has the issue that those texts often use language that is different from everyday language.

The metric we introduce in Sect. 4 aims to apply word embedding-based coherence metrics in a multilingual environment to overcome the above challenges.

3 Dataset Construction

Constructing a multilingual corpus of South African text is not easy due to the limited availability of good quality data [6]. Because of this, we made use of resources provided by the Leipzig Corpora Collection project[1]. The project presents corpora in over 250 different languages using the same format and comparable sources. The datasets contain randomly selected sentences in the language of the corpus. Depending on availability, datasets can be downloaded in sizes ranging from 10,000 sentences up to 1 million sentences and are derived from

[1] https://wortschatz.uni-leipzig.de/en.

newspaper texts or texts randomly collected from the web. With the exception of South African Sign Language, the project has data for all of South Africa's official languages.

For this project, we use 10,000 sentences from each language from web sources scraped in 2020. We do this because this is the maximum amount of information available for Swati, Sesotho, Tswana, Tsonga, and Venda. Our corpus also includes Afrikaans, Pedi, Xhosa and Zulu. Ndebele is not included because it only has data for 318 sentences. We also exclude English sentences since it is already a well-studied language in the literature. Thus, our corpus consists of 90,000 sentences from nine of South Africa's official languages.

Bag of Words Embeddings. Pre-processing to transform the text into a bag of words representation is required for ZeroShot and LDA. We convert the text to lowercase, tokenize on spaces, and remove inter-word punctuation as well as words that are only numbers. We then choose the 600 most frequent words in each language. This gives us a set of 5400 words (600×9). However, there are duplicate words between the sets. For example "umami" ("my mother") is used in five of the languages. This is due to the languages having words in common. After removing duplicates, our vocabulary consists of 4412 unique words.

Contextual Embeddings. For this project, we considered various embeddings, including LaBSE [11] and embeddings available from OpenAI. LaBSE is an attempt to create multilingual embeddings using a BERT [8] model. However, while LaBSE supports 109 languages, it does not offer much support for South African languages. Embeddings from OpenAI were rejected since there is no way to know which languages the model was trained on. Ultimately, we settled on SONAR [10], which is trained using Flores [24]. Flores is a machine translation evaluation benchmark that consists of roughly 2000 sentences that have been translated into over 200 languages. The data is considered to be very high quality. SONAR is a multilingual and multimodal sentence embedding model. It also has decoders that allow it to produce translated text or speech. SONAR substantially outperforms existing sentence embeddings such as LaBSE.

4 Multilingual Topic Coherence

We propose a new multilingual topic coherence metric. It makes use of word embeddings instead of a parallel reference corpus. This makes it more computationally efficient, cost effective, and allows the metric to incorporate semantic understanding of topic coherence [37]. MTC provides a tempering of pure coherence with multilinguality.

As part of the MTC calculation, a multilinguality score μ_{t_k} is calculated for each topic t_k. The multilinguality score is scaled from 0 to 1, where 1 indicates that a topic is more multilingual. A topic with words from a single language from the set of languages in the corpus will receive a score of 0 for multilinguality, while a topic that contains words from each language will receive a score of 1. For N topics $(t_1, \ldots, t_N)$, each consisting of T words $(w_{t_k,1}, \ldots, w_{t_k,T})$, as well

as a function $L(w)$ that returns the set of languages that a word w appears in according to the corpus, the multilinguality for a topic t_k can be calculated as in Eq. 2.

$$\mu_{t_k} = \frac{\left|\bigcup_{i=1}^{T} L(w_{t_k,i})\right| - 1}{\left|\bigcup_{j=1}^{N}\bigcup_{i=1}^{T} L(w_{t_j,i})\right| - 1} \tag{2}$$

The MTC is a combination of the multilinguality score for a topic and a word embedding-based coherence score. We make use of embedding-based coherence mentioned in Sect. 2.2. As seen in the full calculation in Algorithm 1, the harmonic mean is used to combine the scores. We chose to use the harmonic mean over the arithmetic mean, the same as F1 scores used in classification performance measurement. This is because it encourages good performance in both coherence and multilinguality. As recommended by Terragni et al. [34], we make use of normalized embeddings in the similarity score calculation.

Algorithm 1. Multilingual topic coherence

Require: N topics (t_i) each with T words (w_j); an embedding function E

```
C ← 0
for i := 1, ..., N do
    simScore_{t_i} ← 0
    for j := 1, ..., T − 1 do
        for k := j + 1, ..., T do
            s ← (E(w_{t_i,j}) · E(w_{t_i,k})) / (|E(w_{t_i,j})||E(w_{t_i,k})|)   ▷ Cosine similarity of normalized embeddings
            simScore_{t_i} ← simScore_{t_i} + s
        end for
        multilingualityScore_{t_i} ← from Eq. 2
        C_{t_i} ← (simScore_{t_i}^{-1} + multilingualityScore_{t_i}^{-1})^{-1}   ▷ Harmonic mean
        C ← C + C_{t_i}
    end for
end for
M ← (T)(T−1) / 2
C ← C / MN
```

5 Performance Comparison

This section details the method used to compare different topic models on our multilingual South African languages corpus. We use this to show the usefulness of our MTC metric. Section 5.1 details the experimental setup used to collect the results, and Sect. 5.2 shows and discusses the results.

5.1 Experimental Setup

For the experiments, we use three base models: LDA, ZeroShotTM [2], and BERTopic [12]. We run the models on the South African languages dataset using a range of 6 topic numbers: 50, 100, 150, 200, 250, 300. We replicate each of those runs 5 times, resulting in 30 data points for each model. Performance is measured using NPMI [5], Embedding Coherence [37], as well as out novel MTC metric.

The LDA implementation used is taken from the `gensim` [31] Python library. We use the default hyperparameters $\alpha = \eta = \frac{1}{N}$, where N is the number of topics. The ZeroShotTM implementation is taken from OCTIS [34]. Again the default hyperparameters are used: hidden layer size = 100; number of hidden layers = 2; dropout = 0.2; number of epochs = 100. For BERTopic, we make use of the implementation in the Python library `bertopic`. We again use the default hyperparameters: n-gram range = $(1, 1)$; minimum topic size = 10; dimensionality reducer = UMAP; and clusterer = HDBSCAN.

For LDA and ZeroShotTM, which both make use of a fixed vocabulary, we investigate using two representations for the models. The first is the standard Bag of Words (BoW) representation. The second is the Term Frequency Inverse Document Frequency (TF–IDF). TF–IDF transforms a BoW representation. For a given word, it increases as the word is used more frequently (term frequency) but is offset by the frequency of the word in the entire corpus (inverse document frequency), which helps to filter out common words that contain less information. Essentially, TF–IDF helps highlight words that are unique and significant to specific documents within a larger set of documents. For example, words with high frequencies like common stop words ("the", "is" "an", etc.), but which appear in many documents will receive a lower TF–IDF score than words that appear frequently in only a couple of documents in the corpus.

We include the TF–IDF representation to serve as a means to imitate stop word removal. Stop words are a commonly used in dataset preparation. However, stop word lists for the languages used in the corpus (see Sect. 3) are not easily available due to the low resource nature of the languages. Thus, using TF–IDF representations should allow the models to focus less on stop words, since they will have their frequency values adjusted.

5.2 Results and Discussion

Averages for the performance metrics over all runs are shown in Table 1. The results show that the contextualized topic models—ZeroShotTM and BERTopic—outperform LDA in terms of NPMI score. This confirms results found by the authors of those models [2,12]. MTC values also give preference to the contextualized models. Both NPMI and MTC give values between 0 and 1, however, NPMI scores for the models are all clustered below 0.3, while MTC uses more of the range, thus giving more nuanced information.

Comparing the LDA BoW and LDA TF–IDF performance values shows that the change had little effect for the LDA model. However, ZeroShotTM shows better MTC values for the TF–IDF representation over the BoW representation.

Table 1. Average performance results for models. The values are shown as mean ± standard deviation. The best value for each metric is bolded (confirmed to be statistically significant using a two-sided Mann–Whitney U test $P < 0.05$)

	NPMI	MTC
ZeroShotTM BoW	**0.2842 ± 0.0232**	0.8564 ± 0.0200
ZeroShotTM TF–IDF	0.2751 ± 0.0253	**0.9053 ± 0.0151**
BERTopic	0.2269 ± 0.0102	0.7310 ± 0.0375
LDA TF–IDF	0.0595 ± 0.0368	0.5379 ± 0.1820
LDA BoW	0.0586 ± 0.0354	0.5402 ± 0.1835

As discussed in Sect. 4, MTC is made up of an embedding coherence score and a multilinguality score. It is possible to calculate these separately for each topic. Figure 1 shows the separate multilinguality and Embedding Coherence (EC) scores for the models as the number of topics vary. Additionally, the average results are shown in Table 2. This allows us to see a rough breakdown of why a model receives a particular score, and help in understanding the strengths and weaknesses of each model in handling multilingual topics. ZeroShotTM and BERTopic receive similar scores for both the coherence and multilinguality scores across the full range.

Table 2. Embedding coherence (EC), multilinguality (μ) and MTC performance. The values are shown as mean ± standard deviation. The best value for each metric is bolded (confirmed to be statistically significant using a two-sided Mann–Whitney U test $P < 0.05$).

	EC	μ	MTC
ZeroShotTM TF–IDF	**0.9041 ± 0.0148**	**0.9181 ± 0.0213**	**0.9053 ± 0.0151**
ZeroShotTM BoW	0.8765 ± 0.0330	0.8638 ± 0.0322	0.8564 ± 0.0200
BERTopic	0.6860 ± 0.0298	0.8445 ± 0.0413	0.7310 ± 0.0375
LDA BoW	0.8179 ± 0.0041	0.4641 ± 0.2524	0.5402 ± 0.1835
LDA TF–IDF	0.8174 ± 0.0036	0.4608 ± 0.2505	0.5379 ± 0.1820

We can see how MTC balances performance in both coherence and multilinguality. ZeroShotTM, attains a good MTC score because it has good scores in both embedding coherence and multilinguality. However, BERTopic and LDA show better performance in one score over another (see Table 2). Unsurprisingly, LDA performs poorly in the multilinguality score, since it can only use words shared between languages to gain a cross language understanding. Its multilinguality score also drops as the number of topics increases (see Fig. 1). This is explained in Table 3, where we see that as the number of topics increases, the number of multilanguage words and the number of languages represented per topic decreases. This decline in multilinguality highlights the limitations of LDA in handling multilingual datasets effectively.

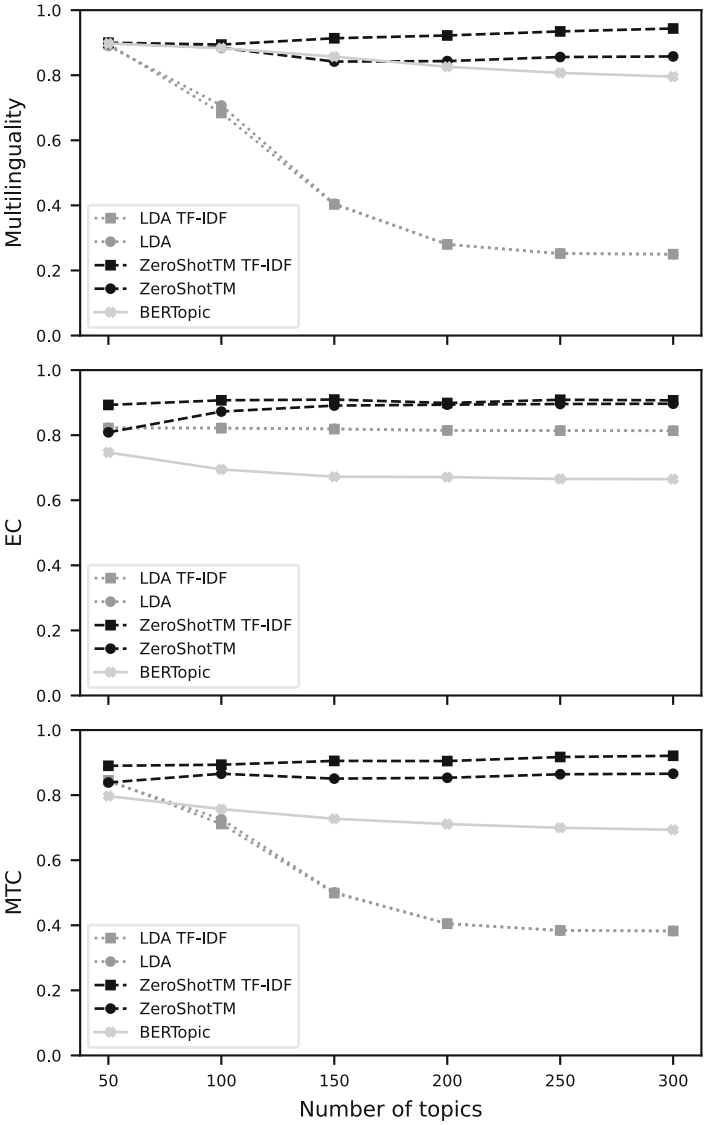


Fig. 1. Multilinguality, EC and MTC vs number of topics for different models.

Table 3. Number of multilanguage words (i.e. words that are used across multiple languages in the corpus) as well as the average number of languages in a topic for the LDA BoW model.

# Topics	# Multilanguage words	# Languages per topic
50	9.932 ± 0.292	3.773 ± 0.810
100	8.368 ± 1.598	2.851 ± 0.753
150	5.468 ± 2.242	1.889 ± 0.784
200	4.273 ± 1.188	1.494 ± 0.418
250	4.019 ± 0.258	1.406 ± 0.082
300	4.000 ± 0.000	1.400 ± 0.000

From Table 2, we can see that BERTopic performs well for the multilinguality score, but poorer for the coherence score. A manual investigation of the topics showed that a number of topic words suggested by the model were numbers (e.g. "00", "112", etc.). There were a total of 1767 instances of this across all the runs. Table 4 shows the average number of times a number appears as a topic word. Also, since BERTopic is not limited to using a set vocabulary, we also show the average number of non-vocabulary words BERTopic uses per topic. We can see that BERTopic becomes more likely to use a number as a topic word as the number of topics increases. This could explain why its coherence scores decrease slowly.

Table 4. BERTopic vocabulary: number of topic word used that are not part of the vocabulary used by ZeroShotTM and LDA. As well as the number of topic words that are pure numbers.

# topics	# numbers per topic	# non-vocabulary words per topic
50	0.196 ± 0.125	3.828 ± 0.683
100	0.320 ± 0.155	5.030 ± 0.643
150	0.383 ± 0.182	5.805 ± 0.636
200	0.332 ± 0.163	6.226 ± 0.605
250	0.344 ± 0.173	6.506 ± 0.589
300	0.339 ± 0.176	6.638 ± 0.575

Overall, the MTC metric shows that the contextualized topic models outperform LDA, and it allows models to be evaluated on both coherence and multilinguality. These findings also shed light on the strengths and weaknesses of each model in handling multilingual topics.

6 Conclusion

In this paper, we address the limitations of evaluating topic models when applied to multilingual corpora by introducing a novel performance metric called Multi-

lingual Topic Coherence (MTC). Our contributions include the development of the MTC metric, the construction of a short-text dataset comprising texts from nine of South Africa's official languages, and a comparative analysis of three widely used topic models: LDA, ZeroShotTM, and BERTopic.

The MTC metric combines word embedding-based coherence with a multilinguality score, providing a comprehensive evaluation of a model's ability to handle multilingual topics. It improves on previous metrics by incorporating multilingual understanding without making use of parallel reference corpora. Our experimental results demonstrate that contextualized topic models, such as ZeroShotTM and BERTopic, outperform traditional models like LDA in terms of both coherence and multilinguality. Specifically, ZeroShotTM achieved the highest MTC score, indicating its superior performance in generating coherent and multilingual topics.

Our MTC metric offered a more nuanced evaluation compared to traditional metrics like NPMI. The breakdown of MTC into its constituent parts—embedding coherence and multilinguality—provides valuable insights into the strengths and weaknesses of each model. For instance, while LDA struggled with multilinguality, BERTopic showed good multilingual performance but was hampered by the inclusion of non-vocabulary words and numbers as topic words which reduced its coherence score.

Our findings impact the field of natural language processing, particularly in multilingual and low-resource environments. The MTC metric offers a reliable and comprehensive tool for evaluating topic models, ensuring that they can effectively handle the complexities of multilingual corpora.

Future research could explore the application of MTC to other low-resource languages and further refinement of the metric to account for additional nuances in multilingual text. Additionally, the metric could be applied to multilingual corpora where there is less overlap in shared words. Lastly, our results indicate that there is potential to improve the way BERTopic selects topic words, perhaps by restricting the words it can use to a vocabulary.

Acknowledgement. The authors would like to acknowledge the contributions and assistance of George Davie.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bianchi, F., Terragni, S., Hovy, D.: Pre-training is a hot topic: contextualized document embeddings improve topic coherence. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), June 2021

2. Bianchi, F., Terragni, S., Hovy, D., Nozza, D., Fersini, E.: Cross-lingual contextualized topic models with zero-shot learning. In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, pp. 1676–1683. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.eacl-main.143
3. Blei, D., Ng, A., Jordan, M.: Latent Dirichlet allocation. Adv. Neural Inf. Process. Syst. **14** (2001)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. J. Mach. Learn. Res. **3**(Jan), 993–1022 (2003)
5. Bouma, G.: Normalized (pointwise) mutual information in collocation extraction. In: Proceedings of the Biennial GSCL Conference, pp. 31–40 (2009)
6. De Wet, F., Eiselen, R., Schillack, E., Puttkammer, M.: Investigating the extent and usability of webtext available in South Africa's official languages. In: Pillay, A., Jembere, E., Gerber, J.A. (eds.) SACAIR 2023. CCIS, vol. 1976, pp. 120–135. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-49002-6_9
7. Defersha, N.B., Tune, K.K., Abate, S.T.: Adapting outperformer from topic modeling methods for topic extraction and analysis: the case of Afaan Oromo, Amharic, and Tigrigna Facebook text comments. Int. J. Adv. Comput. Sci. Appl. **15**(3) (2024). https://doi.org/10.14569/IJACSA.2024.0150391
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805, May 2019
9. Dieu, D.T., Dinh, D.: Using topic in summarization for Vietnamese paragraph. Int. J. Adv. Comput. Sci. Appl. **14**(10) (2023). https://doi.org/10.14569/IJACSA.2023.0141078
10. Duquenne, P.A., Schwenk, H., Sagot, B.: SONAR: Sentence-Level Multimodal and Language-Agnostic Representations. arXiv:2308.11466, August 2023
11. Feng, F., Yang, Y., Cer, D., Arivazhagan, N., Wang, W.: Language-agnostic BERT Sentence Embedding. arXiv:2007.01852, March 2022
12. Grootendorst, M.: BERTopic: neural topic modeling with a class-based TF-IDF procedure. arXiv:2203.05794, March 2022
13. Gurusamy, B.M., Rengarajan, P.K., Srinivasan, P.: A hybrid approach for text summarization using semantic latent Dirichlet allocation and sentence concept mapping with transformer. Int. J. Electr. Comput. Eng. (IJECE) **13**(6), 6663 (2023). https://doi.org/10.11591/ijece.v13i6.pp6663-6672
14. Hao, S., Boyd-Graber, J., Paul, M.J.: Lessons from the Bible on modern topics: low-resource multilingual topic model evaluation. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 1090–1100. Association for Computational Linguistics, New Orleans (2018). https://doi.org/10.18653/v1/N18-1099
15. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 50–57 (1999)
16. Hoyle, A., Goel, P., Peskov, D., Hian-Cheong, A., Boyd-Graber, J., Resnik, P.: Is automated topic model evaluation broken?: The incoherence of coherence. In: Neural Information Processing System (2021)
17. Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. In: The International Conference on Learning Representations (ICLR), Banff (2014). https://doi.org/10.48550/arXiv.1312.6114

18. Krstovski, K., Smith, D., Kurtz, M.J.: Online multilingual topic models with multi-level hyperpriors. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 454–459. Association for Computational Linguistics, San Diego (2016). https://doi.org/10.18653/v1/N16-1053
19. Lau, J.H., Newman, D., Baldwin, T.: Machine reading tea leaves: automatically evaluating topic coherence and topic model quality. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pp. 530–539. Association for Computational Linguistics, Gothenburg (2014). https://doi.org/10.3115/v1/E14-1056
20. Masethe, H.D., Masethe, M.A., Ojo, S.O., Owolawi, P.A.: Using topic modeling to find hidden structures in the language of Sesotho sa Leboa. In: 2024 Conference on Information Communications Technology and Society (ICTAS), pp. 75–81. IEEE, Durban, March 2024. https://doi.org/10.1109/ICTAS59620.2024.10507132
21. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient Estimation of Word Representations in Vector Space. arXiv:1301.3781, September 2013
22. Mnih, A., Gregor, K.: Neural variational inference and learning in belief networks. In: International Conference on Machine Learning (2014)
23. Niţu, M., Dascălu, M., Dascălu, M.I.: Romanian topic modeling – an evaluation of probabilistic versus transformer-based topic modeling for domain categorization. Revue roumaine des sciences techniques — série électrotechnique et énergétique **68**(3), 295–300 (2023). https://doi.org/10.59277/RRST-EE.2023.3.8
24. Costa-Jussà, M.R., et al.: No language left behind: scaling human-centered machine translation (2022)
25. Omizo, R.M.: A comparison of topic modeling approaches using networked discussion forum posts from the city-data.com corpus. J. Open Humanit. Data **10**(1) (2024)
26. Peters, M.E., et al.: Deep contextualized word representations. arXiv:1802.05365, March 2018
27. Petroni, F., et al.: Language models as knowledge bases? In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 2463–2473. Association for Computational Linguistics, Hong Kong (2019). https://doi.org/10.18653/v1/D19-1250
28. Pylov, P., Maitak, R., Protodyakonov, A.: The Latent Dirichlet Allocation (LDA) generative model for automating process of rendering judicial decisions. E3S Web of Conf. **431**, 05005 (2023). https://doi.org/10.1051/e3sconf/202343105005
29. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving Language Understanding by Generative Pre-Training (2018). https://openai.com/research/language-unsupervised
30. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language Models are Unsupervised Multitask Learners (2018). https://api.semanticscholar.org/CorpusID:160025533
31. Řehůřek, R., Sojka, P.: Software framework for topic modelling with large corpora. In: Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pp. 45–50. ELRA, Valletta, May 2010
32. Reimers, N., Gurevych, I.: Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084, August 2019
33. Shutova, E., Sun, L., Gutiérrez, E.D., Lichtenstein, P., Narayanan, S.: Multilingual metaphor processing: experiments with semi-supervised and unsupervised learning. Comput. Linguist. **43**(1), 71–123 (2017). https://doi.org/10.1162/COLI_a_00275

34. Terragni, S., Fersini, E., Galuzzi, B.G., Tropeano, P., Candelieri, A.: OCTIS: comparing and optimizing topic models is simple! In: Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations, pp. 263–270. Association for Computational Linguistics (2021, online). https://doi.org/10.18653/v1/2021.eacl-demos.31
35. Terragni, S., Fersini, E., Messina, E.: Word embedding-based topic similarity measures. In: Métais, E., Meziane, F., Horacek, H., Kapetanios, E. (eds.) NLDB 2021. LNISA, vol. 12801, pp. 33–45. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-80599-9_4
36. Thielmann, A., Reuter, A., Seifert, Q., Bergherr, E., Säfken, B.: Topics in the haystack: enhancing topic quality through corpus expansion. Comput. Linguist. 1–36 (2024). https://doi.org/10.1162/coli_a_00506
37. Thurnbauer, M., Reisinger, J., Goller, C., Fischer, A.: Towards Resolving Word Ambiguity with Word Embeddings. arXiv:2307.13417, July 2023
38. Touvron, H., et al.: LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971, February 2023
39. Wheeler, J.M., Cohen, A.S., Wang, S.: A comparison of latent semantic analysis and latent Dirichlet allocation in educational measurement. J. Educ. Behav. Stat. (2023). https://doi.org/10.3102/10769986231209446
40. Xu, Y., Hu, L., Zhao, J., Qiu, Z., Ye, Y., Gu, H.: A Survey on Multilingual Large Language Models: Corpora, Alignment, and Bias. arXiv:2404.00929, April 2024
41. Yuan, M., Lin, P., Rashidi, L., Zobel, J.: Assessment of the quality of topic models for information retrieval applications. In: Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval, pp. 265–274. ACM, Taipei, August 2023. https://doi.org/10.1145/3578337.3605118



Cascaded RFM-Based Fuzzy Clustering Model for Dynamic Customer Segmentation in the Retail Sector

Sive Sobantu and Omowunmi E. Isafiade(✉)

Department of Computer Science, Faculty of Natural Science, University of the Western Cape, Cape Town, South Africa
4109102@myuwc.ac.za, oisafiade@uwc.ac.za

Abstract. The retail sector increasingly uses data mining approaches to identify customer purchasing behaviors and improve service quality. The Recency, Frequency, and Monetary (RFM) model, when integrated with machine learning, effectively segments customers and offers profound insights into their purchase behavior. While customer segmentation can be efficiently executed using the RFM model, the model is constrained to three factors and does not disclose segments based on demographic characteristics and purchase history. However, the influence of customer purchase attributes on marketing strategies is highly significant. This study implemented a cascaded solution integrating the RFM model with behavioral variables and conducted a comparative analysis of the Modified Fuzzy C-Means (MDFCM) and Crespo's Dynamic Fuzzy C-Means (CDFCM) models. An experiment was conducted using a retail dataset and the results revealed the presence of two to four clusters exhibiting similar characteristics over a specified duration. The efficacy of the models is assessed using standard metrics such as Silhouette scores and the Xie-Beni index (XB-index). The evaluation indicated that the implemented cascaded RFM model yielded stable results with six clusters; however, it was observed to perform optimally with two clusters. Conversely, the MDFCM exhibited improved performance as the number of clusters increased, while the CDFCM demonstrated superior performance with two to three clusters, subsequently declining thereafter. Businesses can utilize this model to strategically analyze customers' behavior through their purchasing history.

Keywords: Recency · Frequency · Monetary · Silhouette coefficient · Dynamic Customer Segmentation · Fuzzy clustering · Cascaded RFM

1 Introduction

Effective marketing strategies are developed by businesses to boost sales and promote overall company expansion [1]. Big data and predictive analysis are technologies that are used by marketers to increase the effectiveness of product sales and usage [2, 3]; by processing the data they have acquired for decision support, helping marketers achieve efficient and effective marketing. The gathered information through dynamic customer segmentation (DCS) helps in locating potential customers and formulating successful marketing strategies [4].

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 53–73, 2025.
https://doi.org/10.1007/978-3-031-78255-8_4

While several research have explored the application of clustering techniques such as RFM in customer segmentation [5–7], there is still a need to improve on automated and dynamic approach for DCS, to eliminate manual intervention in building of the RFM model, and the limitations of RFM Model which include its reliance on three attributes, which could over simplify customer behavior.

This work considered three separate customer clustering approaches namely, the proposed cascaded RFM (Recency Frequency Monetary) method, Modified Dynamic Fuzzy C-Means (MDFCM) clustering, and Crespo's Dynamic Fuzzy C-Means (CDFCM) models. The RFM model was cascaded into three components namely Recency vs. Frequency, Recency vs. Monetary and Frequency vs. Monetary. The RFM was then used as a base model for MDCFM and CDFCM. The models were evaluated using standard metrics such as the Silhouette scores and XB-index respectively [8, 9].

1.1 Research Aim and Objectives

This paper aims to overcome the shortcomings of the traditional RFM model for solving the Dynamic Customer Segmentation (DCS) problem by effectively implementing dynamic updates of customer purchasing data history into the RFM model over an unlimited number of time cycles (measured in months). Furthermore, to overcome the shortcomings of user-defined 'β' (The threshold used to identify changes in the cluster structure) and 'T' threshold parameters (where T is the total number of cycles measured in months).

The objective of this study is to implement strategic attribute selection from an initial customer retail dataset to build a Cascaded RFM model from those attributes and dynamically update the initial customer data with new information updates from customer purchasing records. This enables the building of new clusters, moving clusters accordingly or eliminating existing clusters that are no longer beneficial. The updating of clusters with new data occurs across different timescales. This automatically adds new customers based on buying patterns, triggering immediate strategic marketing opportunities.

1.2 Research Question

This research aims to perform a comparison of segmentation techniques and clustering algorithms. The clustering algorithms include K-means for initially building the base RFM model and determining the value of K for the amount of clusters to be created using the elbow method technique. Fuzzy C-Means forms the base approach of Crespo's Dynamic Fuzzy C-Means (CDFCM) and Modified Dynamic Fuzzy C-means (MDFCM) clustering algorithms. In light of the research aim, the following research questions are considered:

1. Which combination of the segmentation techniques and clustering models performs best for the dynamic customer segmentation task?
2. How can the derived customer cluster segments be profiled to assist with decision support to enhance customer experience in retail sector?

2 Related Work

The RFM model was originally introduced by Hughes of the American Database Institute in 1994 [10], and several research [11, 12] have subsequently employed this approach. A preliminary investigation established that clients with elevated RFM ratings exhibit a greater propensity to make subsequent purchases [13]. Consequently, many attempts have been made to classify the client base according to RFM values. Aggelis and Christodoulakis [6] employed the K-Means clustering approach to segment their bank client base and computed the corresponding RFM values, facilitating an analysis of customer profitability and enhancing customer segmentation to tailor marketing efforts. Chiang [14] performed an RFM analysis of clients for a Taipei outfitter utilizing the K-means, Fuzzy C-means, and Bagged Clustering methods.

Authors in [15] introduced a novel customer segmentation technique utilizing RFM and demographic data. They utilized a self-organizing map (SOM) to get demographic characteristics and employed the K-Means clustering algorithm to segment customers of an Iranian bank, with customers likewise clustered based on demographic data. A customer profile was then created utilizing lifetime value, which yielded valuable management metrics and recommendations. Furthermore, Dursun and Caber [12] examined the characteristics of profitable customers at a five-star hotel in Antalya within the framework of Customer Relationship Management (CRM), employing RFM analysis and the K-Means algorithm. The findings revealed that 369 profitable hotel customers were categorized into eight segments: Loyal Customers, Loyal Summer Season Customers, Collective Buying Customers, Winter Season Customers, Lost Customers, High Potential Customers, New Customers, and Winter Season High Potential Customers. A significant portion of the clientele (36%) was categorized as 'Lost Customers,' characterized by brief stays, lower expenditure compared to other segments, and a tendency to visit the hotels during the summer period. The results demonstrated that RFM successfully categorizes clients, potentially enabling hotel executives to formulate novel approaches for enhancing their CRM capabilities.

Munusamy and Murugesan [16] introduced the MDFCM clustering model to enhance the efficacy of DCS. The MDFCM algorithm was developed by modifying the previous CDFCM algorithm, integrating the XB index to effectively detect alterations in the cluster structure. The MDFCM model was implemented on a retail supermarket dataset, which was subsequently updated with additional transaction data from the same customers over eleven months, culminating in a total of eleven updates to the initial dataset. Experimental results demonstrate that the MDFCM method outperforms both CDFCM and Fathabadi's dynamic fuzzy c-means (FDFCM) algorithms, the latter being another excellent algorithm documented in the literature [16, 17], hence enhancing the efficacy of DCS. The MDFCM results are corroborated by fuzzy clustering assessment metrics, including the Xie-Beni index, weighted sum of squared errors (WSSE), root mean squared error (RMSE), Kwon index, and Tang index.

Recently, Memon [18] sought to understand the behavior of clients of a Swedish pharmacy company by evaluating segmentation strategies. Their research concentrated on customer segmentation, a conventional customer analytics method and a fundamental marketing principle, by effectively segmenting a Swedish pharmacy retail dataset

through the integration of segmentation methods and clustering models. The identified consumer segments were characterized and designated utilizing customer profiling methodologies, including neural networks, Agglomerative clustering, and Mean-Shift clustering algorithms. The experiment demonstrated that Agglomerative clustering algorithms utilizing the recency, frequency, monetary, and periodicity (LRFMP) model are the best appropriate method for customer segmentation in the dataset analyzed in the research study. The integration of LRFMP with K-means and LRFMP with Agglomerative clustering produced two client segments, classified as "low-contribution customers" and "high-contributing loyal customers." The Silhouette score, which is a validation index, was employed to assess the efficacy of the clustering models utilized in the experiment.

This study employs clustering algorithms, specifically CDFCM, to achieve DCS by substituting coded numbers in conventional RFM-based models with continuous values, hence enabling a more flexible categorization of customer actions. This work employs the philosophy of Chang et al. [19] and Wu et al. [20] to utilize continuous values for RFM models, hence facilitating the effective adoption of Recency, Frequency, and Monetary metrics in customer segmentation. Upon finalizing the RFM model, we proceed to implement the CDFCM and MDFCM clustering methodologies for comparison. Table 1 presents an overview of related research and the focus of this study.

Table 1. Overview of related research on RFM models and clustering techniques

Ref	Research Focus	Context/Domain	Model Used	Findings
[12]	This study utilizes RFM analysis to identify profitable clients within the framework of CRM	Retailing sector (Taiwan)	RFM analysis, Apriori algorithm	Most of the customers were identified as 'Lost Customers', who stay for shorter periods and spend less than other groups. The authors offered CRM strategies for these and other customer groups
[18]	This experimental study focuses on customer segmentation, a typical customer analytics technique, and a traditional concept in marketing. The thesis aimed to perform an experiment by efficiently segmenting Swedish pharmacy retail dataset by combining segmentation models and clustering algorithms	Pharmacy Retail (Sweden)	K-means, Agglomerative and Mean-Shift clustering algorithms, segmentation models RFM, LRFM and LRFMP	This experiment showed that both K-means and Agglomerative clustering algorithms with the LRFMP model are the most suitable solution for customer segmentation for the dataset used in this study

(*continued*)

Table 1. (*continued*)

Ref	Research Focus	Context/Domain	Model Used	Findings
[17]	Creating a groundbreaking DCS framework. The study created an MDFCM algorithm for operational efficiency. Client purchase behaviors are initially evaluated using a dynamic clustering-based DCS architecture and MDFCM to cluster recent data updates to identify cluster setup changes. Next phase clusters are segmented using their initial RFM pattern	Retail Supermarket (India)	Dynamic Fuzzy C-Means (DFCM), Modified Dynamic Fuzzy-C means (MDFCM)	The results demonstrated the effectiveness of the proposed MDFCM algorithm compared to the existing DFCM algorithm
[21]	This study presents the segmentation and classification technique for insurance industry via data mining approaches: K-Modes Clustering and Decision Tree Classifier	Dutch data mining company (Amsterdam)	K-means clustering algorithm combined with the Self Organising Map (SOM) technique (based on Artificial Neural Networks)	While K-Means showed promising clustering results, SOM outperformed in the sense of speed, quality of clustering ad visualization
This work	This current study uses clustering algorithms such as CDFCM and CDFCM to achieve DCS, by replacing the coded numbers in traditional RFM-based models by implementing continuous values of RFM-based models to more flexibly categorize customer behaviors	Real dataset from a Pharmacy Retail (South Africa)	cascaded RFM model, Modified Dynamic Fuzzy C-Means (MDFCM) algorithm, Crespo's Dynamic Fuzzy C-Means (CDFCM) algorithm	The experiment showed that the MDFCM performed better than the CDFCM. The cascaded RFM model shows promise in assisting with DCS task

3 Research Methodology

3.1 Data Collection and Preprocessing

This study used a quantitative transactional data set that contains customer transactions from a South African-based retail organization. It has 55150 records with 7 features as shown in Table 2.

The dataset described in Table 2 was stored in a CSV file. The CSV file was loaded and pre-processed by cleaning, aggregating based on the requirements, and converting it into a new data format. The pre-processing stage entailed the removal of null values and defining a new attribute named "Amount", which represents the product of Quantity and Unit Price attributes. This process also involved the creation of two more datasets using aggregation, which are: (i) customer aggregate data, and (ii) invoice aggregate data. The missing entries in the customerID field, duplicate entries, cancelled transactions, and

Table 2. Transactions dataset attributes for Nivea products

Features	Data Type	Description
Quantity	Numeric	The amount of each item in every transaction
StoreNo	Numeric	Store number
Stock Keeping Unit (SKU)	Numeric	SKU Number code. A unique number is assigned to each distinct product
Transaction Date	Date	Transaction date and time. Numeric
CustomerID	Numeric	Customer identification. Nominal, special integral value assigned to each customer. This is not a loyalty number
Unit Price	Float	Unit price
Supplier Cost	Float	Supplier Price

invoices that do not relate to customers are removed. The methodology steps used in the study is depicted in Fig. 1, which shows three stages i.e. data processing, data analysis and results.

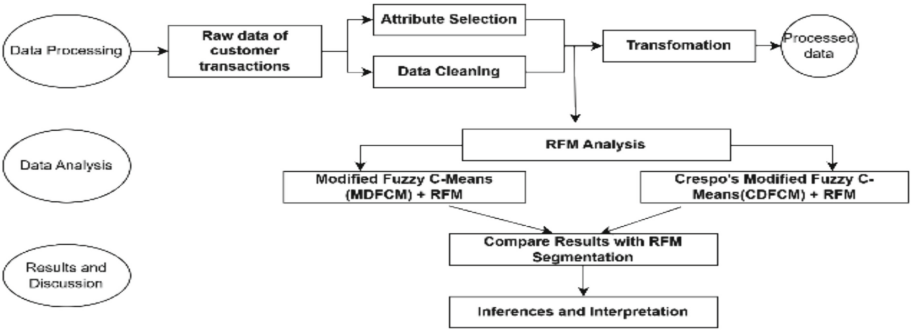


Fig. 1. A Depiction of the Research Methodology

3.2 Models Used

In this section, we explain the Customer Segmentation models used in this study, i.e. RFM Segmentation model was used as a base data model. Tavakoli et al. [22] states that a number of goals should be put into consideration when developing an effective Customer Segmentation model. The goals suggest that organizations need to have a static or semi-dynamic definition for R(Recency). This goal helps organizations to have appropriate insight into their active, lapsing, and lapsed customers which is extremely significant for a business. Moreover, they can monitor the success or failure of a marketing campaign or plan since according to the static definition of recency, the segments before a marketing

campaign or plan can be compared with the segments after that marketing campaign or plan. Also, the relationship between frequency and monetary should be considered [5].

The relationship between Frequency and Monetary is highly important in a successful customer segmentation model since they have several effects on each other, which should be handled by decision-makers. Therefore, in the case of a low-quality or low correlation model, organizations may merge or manipulate some segments to handle this relationship. Furthermore, changes in business and user behaviour should affect the customer segmentation model. Businesses, especially e-Commerce, are faced with several dynamics that can impact on customer segmentation clusters, hence the models need to align with managing the dynamics.

3.2.1 Cascaded RFM Model

This research used combinations of Recency + Frequency (RF), Recency + Monetary (RM), Frequency + Monetary (FM) in identifying relevant clusters. The aim is to provide an organisation the ability to compare over a time period and construct a customer segmentation that meets their goals. Moreover, given that Recency only reveals the time of the last purchase for each customer and does not necessarily indicate their loyalty, this type of recency is different from frequency and monetary and such isolation would be appropriate. Accordingly, this is the reason the model is called cascaded RFM. In determining the best segments for recency, information was gathered from customer purchase history with a target of four categories, which are Best Customers, Loyal Customers, Almost Lost Customers, and Lost Customers. These segments, apart from being easy and relevant, also help the retail sectors with different strategies for their customer retention and are defined as follows.

1. The ones who have purchased recently within 1–30 days, are active or best customers.
2. Those who have not purchased lately but have been involved in high purchases in the near past (30–60 days) are loyal customers.
3. The ones who have within 60–90 days purchased a product are almost Lost customers.
4. Those who have not purchased in 90 + days comprises of lost customers

According to Ullah et al. [23], these four significant segments can be used by any organization to cover the current consumer. As a result, companies can base all their recency-related strategies on these four segments. A quantile method plot, dividing the customers into four equal segments, was used to determine the thresholds for the Recency measure. Customers were segmented according to their Recency, Frequency and Monetary values. The different clustering techniques such as RFM, Fuzzy C-Means (FCM), CDFCM and MDFCM were used for building clusters and then compared.

RFM is a model in customer value analysis. The model in this study examined the customer value based on three parameters: purchase frequency, monetary, and recency. These parameters are calculated based on the attributes of the raw dataset. The RFM scores to assign customers with similar behaviour into one group or cluster was determined for the RFM segmentation. To find the best features for our clustering step, we selected Recency and Monetary as the two main features of focus. These features are simple and can be calculated fast, but they are also a meaningful feature that provides

flexibility for establishing a balance for proper customer segmentation [18]. An important point is that these features and our definition of Recency and Monetary help us have a dynamic segmentation over time since in case of changes in customer behaviours, new clusters are made and the best segments will be identified according to the new values of Recency, and Monetary.

The initial step in RFM is having a training dataset. Furthermore, to process the training data, the K-Means algorithm in the data processing stage acts as a primary step on groups with randomly selected centroids, which are used as the beginning points for each cluster or group [7]. While our core algorithms exclude K-means, it is however necessary to use this primary process to build the training data, and afterwards perform iterative calculations to optimize the positions of the centroids. It is also essential to decide the number of clusters to build by indicating the value of K. For deciding this value, the Elbow method is used.

3.2.2 K-Means and Elbow Method

The authors in [7] assert that K-means is a widely utilized clustering approach based on partitioning. It is a straightforward and rapid approach for selecting cluster heads. Zamri et al. [24] assert that the Elbow technique evaluates the percentage of variation explained relative to the number of clusters.

This strategy is predicated on the notion that many clusters should be selected so that the inclusion of an additional cluster does not significantly enhance the data modelling. The proportion of variance elucidated by the clusters is graphed in relation to the number of clusters.

The initial clusters will provide substantial information; however, at a certain juncture, the marginal gain will diminish markedly, resulting in an elbow-like angle on the graph. The appropriate number of clusters, denoted as K, is determined at this stage, thus employing the elbow criterion [24]. The objective is to commence with $K = 2$ and increment it by 1 at each subsequent step. The stabilization of the value of K signifies the greatest number of clusters achievable; hence, this study utilized $K = 4$ based on the completed experiment.

3.2.3 The Fuzzy C-Means Algorithm

In traditional clustering approaches, every input data point is assigned to a single cluster and cannot belong to multiple clusters; thus, the clusters are mutually exclusive. Consider a data point exhibiting features akin to many clusters; in traditional clustering, it is necessary to determine the most appropriate cluster for this data point and assign it exclusively to one cluster. The primary distinction between classical and fuzzy clustering is that in fuzzy clustering, a data point may belong to multiple clusters simultaneously. Like the traditional c-means model, the fuzzy c-means algorithm also requires prior knowledge of the number of clusters [25]. The algorithm is introduced as follows:

The objective function of the FCM algorithm E_z is to be minimized concerning fuzzy membership u_{ij} and cluster center X_j.

$$E_z = \sum_{i=0}^{c} \sum_{j=0}^{n} \left(u_{ij}\right)^z d^2(X_j,\ V_i) \tag{1}$$

Step 1: Initialize memberships u_{ij} of X_j belonging to cluster i such that

$$\sum_{i=1}^{c} u_{ij} = 1 \tag{2}$$

Step 2: Compute the cluster center V_i for i = 1, 2, …, c using Eq. (3)

$$\mu_{ij} = \frac{\sum_{j=1}^{n} (u_{ij})^z X_j}{\sum_{j=1}^{n} (u_{ij})^z} \tag{3}$$

Step 3: Update the fuzzy membership μ_{ij} using Eq. (4)

$$E_z = \frac{\left(\frac{1}{d^2(x_j,\ v_i,)}\right)^{\frac{1}{z-1}}}{\frac{1}{d^2(x_j,\ v_i,)^{\frac{1}{z-1}}}} \tag{4}$$

Step 4: Redo steps 2 and 3 until the value E_z is no longer decreasing. Table 3 presents the description of the parameters and notations used.

Table 3. Symbols and Notations Description

Symbol	Description
E_z	The objective function of minimizing within the sum of squared error
c	the amount of clusters
n	the number of observations
Xj	The feature vector of observations j, j = 1, …, n
Vi	Cluster centers i; where i = 1…., c
X	Feature vector of the mean of a dataset
μij	Degree of membership of observations j to cluster i, i = 1, …., c; j = 1, …, n
1/c	Index prepared as a result of checking conditions 1 and 2 for new data
z	The fuzzifier is to be fixed by the user, this process takes a precise or specific input and decides the degree to which this input belongs to each of several predefined fuzzy sets
c + 1	Creation of a new cluster
c − 1	Elimination of cluster
α	The threshold for checking cluster condition

3.2.4 The Crespo's Dynamic Fuzzy C-Means (CDFCM)

The CDFCM algorithm is introduced by authors in [26] and detailed as follows:

For every cycle, 'm' new data updates would be determined. Suppose $k = n + 1$,….., $n + m$ is the index of 'm' recent data updates.

Step 1: To determine recent data updates that have emerged, the distance between each pair of the existing cluster centers is calculated.

$$d\ (v_i, v_p) \forall i\ \neq\ pi,\ p \in \{1, .., \mathrm{c}\} \tag{5}$$

Furthermore, compute the distance between the b existing cluster centers, *i,* and new data *k.*

$$\hat{d}_{ik} = \hat{d}_{ik}(x_k, v_i),\ \text{where i belongs to the set } \{1, \ldots, c\} \\ \text{and k belongs to the set } \{\mathrm{n}+1, \ldots, \mathrm{n}+\mathrm{m}\}. \tag{6}$$

Now compute the membership of the existing cluster centers, *i,* and new data k

$$\hat{\mu}_{ik} \mathrm{i} \epsilon \{1, \ldots . \mathrm{c}\},\ \mathrm{k} \epsilon \{\mathrm{n}+1,\ \ldots \ldots,\ \mathrm{n}+\mathrm{m}\} \tag{7}$$

Examine the two subsequent conditions (1 and 2) to determine the difference in the cluster structure. The first condition determines the new data *k* with membership values near 1/c. It is impossible to cluster data satisfying this condition well into existing clusters; therefore, it necessitates the creation of new clusters, with user-defined α value.

Condition 1:

$$\hat{\mu}_{ik} - \frac{1}{c}| \leq \alpha \forall \mathrm{k} \in \{\mathrm{n}+1,\ \ldots \ldots,\ \mathrm{n}+\mathrm{m}\},\ \forall \mathrm{i} \in \{1,\ \ldots .,\ \mathrm{c}\} \tag{8}$$

The second condition determines the updated data, *k*, that is different from the existing clusters. If the second condition is satisfied, then new clusters are needed.

Condition 2:

$$\hat{d}_{ik} - \frac{1}{2}\{d(v_i,\ \mathrm{v_p})\} \forall \mathrm{k} \in \{\mathrm{n}+1,\ \ldots \ldots,\ \mathrm{n}+\mathrm{m}\},\ \forall \neq \mathrm{ip} \in \{1,\ \ldots .,\ \mathrm{c}\} \tag{9}$$

Depending on the result arising from these two conditions, 1/c index is created for each new data

$$1/\mathrm{c}(x_k) = \begin{cases} 1\ \textit{if}\ x_k\ \textit{fulfils conditions}\ 1\ \textit{and}\ 2, \\ 0 \qquad\qquad\qquad \textit{otherwise} \end{cases} \tag{10}$$

Observe that 1/c = 1, only if object *k* cannot be properly categorised by the existing clusters.

$$\text{If } \sum\nolimits_{k=n+1}^{n+m} \frac{1}{c}(x_k) = 0 \text{ proceed with } \textbf{Step 3.1},\ \text{else go to } \textbf{Step 2} \tag{11}$$

Step 2: To identify changes of cluster type, i.e., determination of a new cluster or change in cluster structure, Eq. (12) follows, with user-defined β value.

$$\frac{\sum_{k=n+1}^{n+m} \frac{1}{c}(x_k)}{m} \geq \beta\ \textit{with parameter}\ \beta,\ 0 \leq \beta \leq 1 \tag{12}$$

If the amount of recent data observations depicting the changes is above a predefined threshold β, new clusters are formed (see Step 3.2). Otherwise, the existing clusters are then migrated accordingly (see Step 3.1).

Step 3: Movement of relevant clusters and formation of new clusters

The migration of a cluster is dependent on two possibilities.

(1) Activate the FCM model after combining both new and existing observations resulting in movement of clusters v_u.
(2) Compute the movement of cluster centers denoting the recent data v_i^* and merging them with original cluster centers v_i. The function in Eq. (13) is used to compute the updated data k assigned for existing clusters i.

$$1/\mathrm{c}(x_k) = \begin{cases} 1 \; object \; k \; is \; assigned \; to \; i, \\ 0 \qquad\quad otherwise \end{cases} \tag{13}$$

Equations (14), (15) and (16) are subsequently used to identify the new data k assigned for existing clusters, *i,* by determining movement of cluster centers denoting the new data v_i^* and combining them with existing cluster centers v_i.

$$v_i^* = \frac{\sum_{k=n+1}^{n+m} 1_{ci}(X_k) * (1 - 1/c(X_k)\hat{\mu}_{ik})}{\sum_{k=n+1}^{m}(1 - 1/c(x_k))(\hat{\mu}_{ik})^z} \; 1 \leq \mathrm{i} \leq \mathrm{c} \tag{14}$$

Merging the original cluster centers with updated cluster centers v_i^*, the new cluster centers v_i are computed by:

$$\hat{v}_i = (1 - \gtrdot_i)v_i + \gtrdot_i v_i^* \tag{15}$$

Such that weight $\gtrdot_i$ represents the proportion updated data k assigned to cluster i

$$v_i^* = \frac{\sum_{k=n+1}^{n+m} 1_{ci}(X_k) * (1 - 1/c(X_k)\hat{\mu}_{ik})}{\sum_{j=1}^{n}\left(1_{ci}\left(X_j\right) * \mu_{ij}\right) + \sum_{k=n+1}^{n+m} 1_c(X_k) * (1 - 1/c(X_k)\hat{\mu}_{ik})} \tag{16}$$

To compute the amount of new clusters that should be created, Eqs. (17), loss function L(c), and (18), structure strength formula S(c), are utilised.

$$L(c) = \sum_{k=n+1}^{n+m} \sum_{i=1}^{c} \left(\hat{\mu}_{ik}\hat{d}_{ik}^2\right) \tag{17}$$

The number of clusters c is determined as follows, where γ is user-defined:

$$S(c) = \gamma \; \log\left(\frac{n}{c}\right) + (1 - \gamma) \; \log\left(\frac{L(1)}{L(c)}\right) \tag{18}$$

The adequate amount of clusters is determined by the cluster c with maximum S(c) value.

Step 4: Identify cluster path.

To determine the emergence of cluster structure or trajectory during various cycles, set a counter c_i^t for every cluster i in cycle *t*. If a cluster is created in cycle t − 1, set its counter as $c_i^t = 1$. If a cluster is moved in cycle t − 1, set its counter as $c_i^{t-1} + 1$.

Step 5: Eliminate static clusters.

The goal here is to eliminate unchanging clusters that have not received new data updates for a long period. Therefore, a cluster is discarded if it does not receive new data for *T cycles.*

The goal of any dynamic clustering algorithm is to find changes in the cluster structure by the formation, migration and elimination of clusters. In several practical applications, in particular, DCS necessitates the implementation of effective dynamic clustering models [25].

The RFM is considered to have certain limitations since a true segment means that the customers in each segment must respond differently. According to this definition, RFM, coded intuitively in terms of behaviour, does not clear up a true segment as follows: behavioural difference versus responsive difference: the former determines inferred insights while the latter directly resembles future outcomes. Behaviour is an ordinal type of data that puts forward no quantitative benchmark for true segments, also visual difference versus statistical difference. Furthermore, there is consistent need around how to understand the difference in translating research to practical outcomes.

Intuitive RFM selection hierarchy states the nominal difference instead of the real difference [11, 27], hence it is difficult to use it in solving the DCS problem. Therefore, we implemented the FCM model, by dynamically feeding new data updates to the FCM.

This MDFCM algorithm can be used to find the changes in the cluster structure effectively. The MDFCM algorithm is described as follows.

1. Initial dataset

The initial dataset X_j is clustered using the FCM algorithm. Its cluster centers and membership values are stored to cluster the subsequent new data updates.

2. New data updates

New data updates X_k are clustered by the MDFCM algorithm. The convergence criteria enhanced Fuzzy C-means clustering (EFCM) and parameter α are fixed by the user. The fixing of parameter α, which is user-defined, implies Condition 1, which identifies the new data k that has its membership values close to 1/c. The data which satisfy the condition cannot be clustered well into existing clusters; therefore, it requires the creation of new clusters.

3.3 Evaluation Metrics

This study used standard metrics to evaluate the performance of the models considered. The metrics are described as follows:

3.3.1 Silhouette Score

The Silhouette coefficient is an effective method for determining the number of fuzzy clusters by using both membership values and geometric distance [28]. The coefficient is the ratio of intra-cluster compactness to inter-cluster separation. Consequently, reducing the coefficient value yields superior clustering outcomes. The Silhouette score is computed for the obtained clusters, assessing the degree of separation or distinctiveness among them. It exists within the interval [0, 1]. A score close to +1 indicates that objects (customers) are distanced from neighbouring clusters, while a value of 1 suggests potential misassignment of objects (customers) to incorrect clusters or inaccuracies in data preparation. A value of 0 signifies improper clustering of objects. Let K be the number of clusters; Silhouette values are evaluated for $K = 2$ and $K = 5$ to determine the optimal

clustering configuration based on these values. Subsequent to the analysis, we juxtapose the recency of the sale with the sales volume.

$$s_i = \frac{b-a}{\max(b-a)} \tag{19}$$

where a = mean intra-cluster distance, and b = the mean nearest-cluster distance for each cluster.

3.3.2 Xie-Beni (XB) Index

According to Xiong et al. [29], the Xie-Beni (XB) index method was introduced in 1991. XB index is focused on separation and compactness. Separation is a measure of the distance between one cluster and another cluster, while compactness is a measure of proximity between data points in a cluster. According to this method, the optimal cluster is the one with the smallest XB value V_{xb}. The expression of Xie-Beni is as shown in Eq. 20.

$$V_{xb} = \frac{\sum_{i=1}^{c}\sum_{j=1}^{n}\mu_{ij}^{2}\left||v_i - v_j\right||^2}{nmin_{i,j}\left||v_i - v_j\right||^2} \tag{20}$$

4 Results and Discussion

The experiment was conducted using a desktop computer with an Intel(R) Celeron(R) processor of N4020 CPU @ 1.10GHz 1.10 GHz, RAM of 12.0 GB (11.8 GB usable) and 64-bit operating system, x64-based processor. The data was analysed in Jupyter notebook using Python. The resultant dataset is made up of recency of trade, frequency of trade, and monetary value. The analysis performs the clustering by using CDFCM and MDFCM clustering algorithm, and customer data is anonymized due to ethical and privacy reasons.

4.1 Cascaded Recency Frequency Monetary

In this study, the analysis of the RFM model was only conducted on each customer who has a Loyalty card.

The RFM table (see Table 4) is grouped by customer ID, a specific customer can have multiple loyalty cards depending on whether the customer also has staff card. However, purchase transactions in the database on different customer loyalty cards are grouped per customer ID as they are specific to a customer. Table 5 is the detailed description of customer purchases obtained.

Table 4. RFM table with a list of each customer's transactions based on Recency, Frequency, and Monetary

CustomerID	Recency	Frequency	Monetary
015701	27	2	652.19
141233	90	1	143.48
151355	7	1	29.57

Table 5. Description of RFM values

Features	Recency	Frequency	Monetary
Count	23498	23498	23.49800e+04
Mean	46.434888	1.214146	8.031142e+02
Std	30.253542	0.860001	1.011071e−02
Min	0.000000	1.000000	1.011071e+02
25%	20.00000	1.000000	3.390000e+01
50%	44.00000	1.000000	7.825000e+01
75%	72.00000	1.000000	1.773600e+02
Max	99.00000	100.000000	1.549880e+07

In Table 5, it is noted that the average recency of the customers is 46 days (approximately). On average, the customers purchase the product only once and spend an average of 804.37 ZAR.

It can be observed from Table 5 that the values of monetary are quite large in comparison to frequency values. They are scaled properly to be given to the MDFCM and CDFCM algorithm as input. Table 5 has four quartiles with equal ranges of the percentile, like 25%, 50%, and 75%, Percentiles are the percentages of values that fall below or at a certain number. The numbers show that 20% of the customers have been to the store recently and spent over 339 ZAR, similar patterns can be observed for 50% and 75% quartiles. The cluster centers for each criterion were determined. In Table 5 it can be observed that a recency with high score is not good, but high monetary and frequency values are good.

Table 6 shows the RFM table that has been created from the initial dataset to show the RFC score for each customer.

Now that we have the score of each customer on Table 6, we can represent our customer segmentation by combining the scores (R_Quartile, F_Quartile, M_Quartile).

Table 6. Customers RFM Segmentation Table with scores

Customer Loyalty Card	Regency	Frequency	Monetary	R_Quartile	F_Quartile	M_Quartile	RFM Score
413788	15	2	1208.71	4	4	4	444
413888	27	2	652.19	3	4	4	344
414557	90	1	143.48	1	1	1	113
416142	7	1	29.57	4	1	1	411
416215	12	1	84.34	4	4	3	314

It is important to note that: Best Recency score = 4, most recent purchase; Best Frequency score = 4, most quantity purchase; and Best Monetary score = 4, spent the most amount of money.

Identifying the number of customers in each segment is important, based on the list of segments identified, which are Active or Best Customers, Loyal Customers, Almost Lost Customers, Lost Customers. According to Table 6, an RFM score of over 412 indicates Best Customers, an F-Quartile of 4 indicates Loyal Customers, M-Quartile of 4 indicates Big Spenders. Figure 2 shows a linear distribution of customers in 4 clusters with a recency of 0 to 20 days, this reveals a high concentration on 2 clusters which are loyal and best customers.

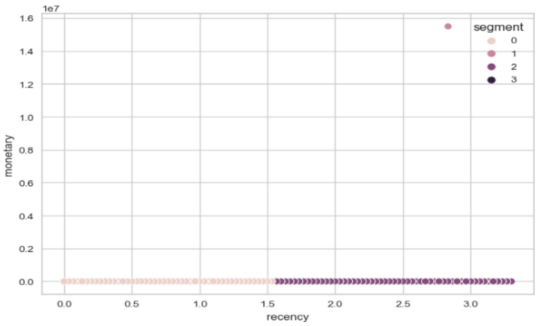


Fig. 2. Recency vs Monetary output of customer distribution existing in four segments **Legend**: 0 = Best Customers, 1 = Loyal Customers, 2 = Almost Lost Customers, 3 = Lost Customers.

4.2 Modified Dynamic Fuzzy C-Means

Figure 3 shows that four clusters are the best number of clusters for the study, hence the study continues to employ four clusters or groups for customer purchase distribution.



Fig. 3. Recency vs Monetary output of customer distribution existing in four segments **Legend**: 0 = Best Customers, 1 = Loyal Customers, 2 = Almost Lost Customers, 3 = Lost Customers

4.3 Crespo's Dynamic Fuzzy C-Means

This study uses four clusters for the CDFCM, the focus for the CDFCM is the various colours of clusters (bottom left corner) in Fig. 4 (best viewed in colour mode), the other clusters showing in Fig. 4 are created by the initial training dataset from the first and second (iterations) run of the algorithm on the dataset.

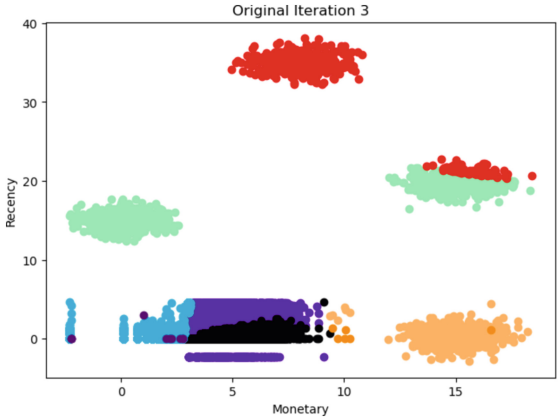


Fig. 4. CDFCM customer cluster distribution Recency vs Monetary. **Legend**: purple = Loyal Customers, black = Best Customers, Orange = Almost Lost, blue = Lost Customers. (Color figure online)

4.4 Evaluation

4.4.1 Cascaded Recency Frequency Monetary

The result of the Elbow method in Fig. 5 shows that we can have four optimal clusters created from the transactional dataset, these clusters are the point of focus form the core of the study and discussion.

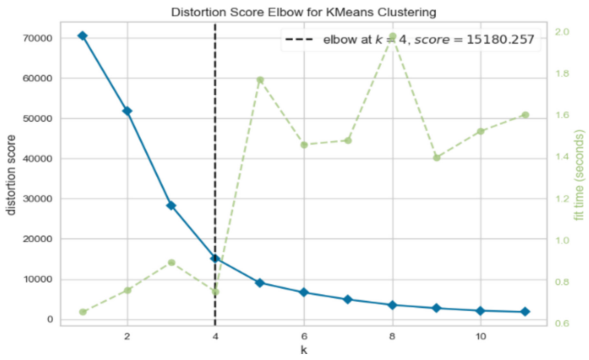


Fig. 5. Number of clusters generated based on the Elbow method, the number of clusters is 4

It was observed in Fig. 5 that clusters range from two (2) to nine (9), with their Silhouette scores, it is notable that the two clusters are the most profitable. However, we can create up to four clusters of interest based on the analysis.

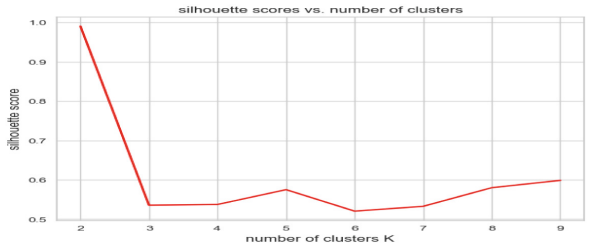


Fig. 6. Silhouette Scores vs Number of Clusters

In Fig. 6 two clusters are most optimal with a Silhouette score of 0.991, this means the customers in these clusters are properly segmented and are the most profitable customers.

The y-axis shows the silhouette values, and the height of each silhouette indicates the number of points in the corresponding cluster. The clusters look well separated. Silhouette scores for different number of clusters 2: 0.991, 3: 0.536, 4: 0.537. These clusters can be selected for further consideration and analysis for marketing purposes. The average score for all the clusters is 0.532. This means that the clusters are even condensed and separated.

4.4.2 Modified Dynamic Fuzzy C-Means

Figure 7 shows the silhouette scores for modified fuzzy c-means with number of clusters equal to 2.

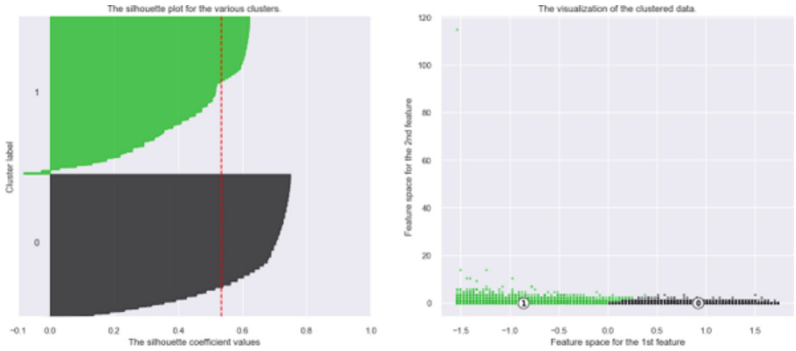


Fig. 7. Silhouette Analysis for MDFCM clustering on sample data with **clusters = 2 Legend**: black = Loyal Customers, green = Best Customers (Color figure online)

Table 7 shows the comparison of the Cascaded RFM Segmentation, CDFCM, and MDFCM clustering based on the number of clusters and Silhouette values.

Table 7. Number of Clusters and Silhouette Scores for RFM, CDFCM and MDFCM

Number of clusters	RFM	CDFCM	MDFCM
2	0.991	0.1614	0.534
3	0.536	0.1726	0.429
4	0.537	0.1830	0.511
5	0.575	0.145	0.547

According to Table 7, the study revealed that using the RFM segmentation produced two (2) most effective clusters. While CDFCM is effective with 2 to 3 clusters and MDFCM grows steadily as the number of clusters increases. Table 8 shows the results per cluster using the XB-Index metric.

Table 8. The XB-Index for different clusters using RFM, CDFCM and MDFCM

Number of clusters	RFM	CDFCM	MDFCM
2	0.17	0.622	0.69
3	0.18	0.536	1.63
4	0.19	0.537	1.50
5	0.19	0.575	1.47
6	0.19	0.520	6.02
7	0.19	0.533	5.64

In Table 8 we observe the RFM XB-Index is lower for 2 clusters, which suggests high cohesion within the customers in that cluster and thus the most optimal clustering outcome for the customers with similar features. We observe that for 5 clusters CDFCM performs better, the results also show that MDFCM performs better with the increase in the number of clusters.

5 Conclusion

Data mining and clustering techniques have great potential in promoting strategic and targeted marketing in the retail sector, to drive decisive customer engagement and competitive success. This work proposed a cascaded RFM clustering model and performed a comparative analysis with Crespo's Dynamic Fuzzy C-Means (CDFCM) and Modified Dynamic Fuzzy C-Means (MDFCM) models.

Customers are categorized into similar clusters using these models, where customers with similar demands and characteristics are put together. Businesses often aim to offer valuable goods and services to its clients after determining their requirements and values in order to boost customer satisfaction and make sure they stick with the business. As a result of the identified clusters results in this study, direct marketing to clients can be accomplished by offering customers with specialized goods and services based on their peculiar purchase characteristics and interests. However, the key objective is identifying and evaluating the behavioural traits of the organization's most important customers. The study found that MDFCM outperforms RFM and CDFCM as the value of K increases, as compared to cascaded RFM and CDFCM.

For future research, the models implemented can be expanded with more features and parameters for DCS tasks in various retail sectors. One potential approach is to create a practical application with multiple goals and assess scalability in light of the high volume of data generated in retail enterprises.

References

1. Lo, F., Campos, N.: Blending Internet-of-Things (IoT) solutions into relationship marketing strategies. Technol. Forecast. Soc. Change **137**(1), 10–18 (2018)

2. Shah, N., Engineer, S., Bhagat, N., Chauhan, H., Shah, M.: Research trends on the usage of machine learning and artificial intelligence in advertising. Augment. Hum. Res. **5**(1), 1–15 (2020)
3. Hole, Y., Pawar, M., Khedkar, E.: Omni channel retailing: an opportunity and challenges in the Indian market. J. Phys.: Conf. Ser. **132**(1), 012121 (2019)
4. Müller, J., Christandl, F.: Content is king–But who is the king of kings? The effect of content marketing, sponsored content & user-generated content on brand responses. Comput. Hum. Behav. **96**(1), 46–55 (2019)
5. Bagheri, F., Tarokh, M.: Customer behavior mining based on RFM model to improve the customer relationship management. J. Ind. Eng. Manag. Stud. **1**(1), 43–57 (2014)
6. Aggelis, V., Christodoulakis, D.: RFM analysis for decision support in e-banking area. WSEAS Trans. Comput. **4**(8), 943–950 (2005)
7. Bholowalia, P., Kumar, A.: EBK-means: a clustering technique based on elbow method and k-means in WSN. Int. J. Comput. Appl. **105** (2014)
8. Ho, T., Nguyen, S., Nguyen, H., Nguyen, N., Man, D., Le, T.: An extended RFM model for customer behaviour and demographic analysis in retail industry. Bus. Syst. Res.: Int. J. Soc. Adv. Innov. Res. Econ. **14**(1), 26–53
9. Haouas, F., Dhiaf, Z., Hammouda, A., Solaiman, B.: A new efficient fuzzy cluster validity index: application to images clustering. In: 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), vol. 1, no. 1, pp. 1–6 (2017)
10. Chen, T., et al.: Application of data mining methods in grouping agricultural product customers. Math. Probl. Eng. **1**(1), 3942374 (2022)
11. Dursun, A., Caber, M.: Using data mining techniques for profiling profitable hotel customers: an application of RFM analysis. Tour. Manag. Perspect. **18**(1), 153–160 (2016)
12. Memon, N.: Customer segmentation in Retail: An Experiment in Sweden (2023)
13. Sivaguru, M.: Dynamic customer segmentation: a case study using the modified dynamic fuzzy c-means clustering algorithm. Granular Comput. **8**(2), 345–360 (2023)
14. Abdul-Rahman, S., Arifin, N.F.K., Hanafiah, M., Mutalib, S.: Customer segmentation and profiling for life insurance using k-modes clustering and decision tree classifier. Int. J. Adv. Comput. Sci. Appl. **12**(1) (2021)
15. Tavakoli, M., Molavi, M., Masoumi, V., Mobini, M.: Customer segmentation and strategy development based on user behavior analysis, RFM model and data mining techniques: a case study. In: 2018 IEEE 15th International Conference on e-Business Engineering (ICEBE), vol. 1, no. 1, pp. 119–126 (2018)
16. Ullah, I., Raza, B., Malik, A., Imran, M., Islam, S.: A churn prediction model using random forest: analysis of machine learning techniques for churn prediction and factor identification in telecom sector **1**(1), 60134–60149 (2019)
17. Xiong, H., Li, Z.: Clustering validation measures. Data Cluster. **1**(1), 571–606 (2018)
18. Crespo, F., Weber, R.: A methodology for dynamic data mining based on fuzzy clustering. Fuzzy Sets Syst. **150**(2), 267–284 (2005)
19. Boutaba, R., et al.: A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. J. Internet Serv. Appl. **9**(1), 1–99 (2018)
20. Yang, A.: How to develop new approaches to RFM segmentation. J. Target. Meas. Anal. Mark. **13**(1), 50–60 (2004)
21. Hughes, M.: Strategic Database Marketing. Probus Publishing Company, Chicago (1994)
22. Wei, J., Lin, S., Weng, C., Wu, H.: A case study of applying LRFM model in market segmentation of a children's dental clinic. Expert Syst. Appl. **39**(5), 5529–5533 (2012)
23. Chiang, W.: Mining high value travelers using a new model designed for online CRM systems: a case study in Taiwan. Aviation **18**(3), 157–160 (2014)
24. Munusamy, S., Murugesan, P.: Modified dynamic fuzzy c-means clustering algorithm–Application in dynamic customer segmentation. Appl. Intell. **50**(1), 1922–1942 (2020)

25. Wu, H., Lin, S., Liu, C.: Analyzing patients' values by applying cluster analysis and LRFM model in a pediatric dental clinic in Taiwan. Sci. World J. **1**(1), 685495 (2014)
26. Zamri, N., Azman, W., Pairan, M., Abas, S.: An analysis of finding the best strategies of water security for water source areas using an integrated IT2FVIKOR with machine learning. Front. Environ. Sci. **10**, 9711 (2023)
27. D'urso, P., Massari, R.: Fuzzy clustering of mixed data. Inf. Sci. **505**(1), 513–534 (2019)
28. Namvar, M., Gholamian, M., KhakAbi, S.: A two phase clustering method for intelligent customer segmentation. In: 2010 International Conference on Intelligent Systems, Modelling and Simulation, vol. 1, no. 1, pp. 215–219 (2010)
29. Chang, E., Huang, S., Wu, H.: Using K-means method and spectral clustering technique in an outfitter's value analysis. Qual. Quant. **44**(1), 807–815 (2010)



Automatic Assessment of Speech Impediment for South African Early Literacy Readers

Jaco Badenhorst(✉)

Voice Computing Research Group, CSIR Next Generation Enterprises and Institutions Cluster, Pretoria, South Africa
jacbadenhorst@gmail.com

Abstract. Many children in South Africa cannot read for comprehension. This finding came from a recent international reading literacy study in which South Africa was placed last out of all 50 countries who participated [12]. During the study, it was pointed out that 75% of the learners participating came from disadvantaged backgrounds, a contributing factor to the overwhelming majority of Grade 4 learners that had already been deprived of proper school instruction. The development of automatic oral assessment applications could potentially alleviate this tragedy. A core component required by such foreseen software solutions is automatic pronunciation error detection (APED). Such APED capable of evaluating the speech produced by school learners in turn requires the development of local language child speech recognition (CSR) to succeed. CSR system development in the South African context may be attempted by refining adult pre-trained models. To this end, the paper provides a detailed breakdown of child data sets as well as reading assessment categories used to build and evaluate APED. While it is possible to select APED thresholds that do not penalise sentences read correctly, the more subtle reading disfluencies remained challenging to identify.

Keywords: Child speech recognition · Automatic pronunciation error detection · Computer assisted language learning · Oral assessment

1 Introduction

Apart from the many use cases of automatic speech recognition (ASR), enabling these systems to also recognise child speakers accurately can unlock many more potential applications. Imagine, for example, the many existing speech applications in education, entertainment, and communication services [18]. From a research perspective, applications in the educational domain did previously receive attention and in particular to increase literary development. Promoting good reading skills, automatic oral assessment strategies have been developed.

This method could be an advantage in settings where basic education fails the children of rural populations. In India, millions of children for example do not

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 74–90, 2025.
https://doi.org/10.1007/978-3-031-78255-8_5

achieve a basic English reading skill at the end of primary school [19]. The rapidly growing number of learners in many developing countries results in teacher shortages, which introduces a shortage of learning time that adversely affects systematic language learning [20]. Locally, the progress in an International Reading Literacy Study (PIRLS) round, in which South Africa participated in 2016, similarly concluded that a high number of Grade 4 children cannot read for comprehension. The study found that this was particularly true for those learners who attended schools in townships and rural areas [12]. When it comes to a shortage in teaching, one way developed countries have been attempting to address the issue was by maturing research interest in computer assisted language learning (CALL) systems. These systems have the inherent potential to offer increased flexibility to the education service. The flexibility comes from the fact that computer assisted pronunciation training (CAPT) is designed to assist with the oral practise component, fast-tracking a learner's communication skill development [27].

Assessment of the oral component can be made by measuring the oral reading fluency. For a child reading a grade level text, this can be based on three main criteria: the ability to read accurately, whether the speaking rate is natural and lastly if the text is read with proper expression [16]. To study reading ability a Words Correct Per Minute (WCPM) measure have been defined that combines two of the above assessment criteria [5]. The authors in [11] showed that the average gain in WCPM was greater for children in lower grades over the course of a year. In fact, during elementary school development, change in WCPM has widely been used as an acceptable indicator of reading skill. A good indicator of an experienced reader is if the reader can read texts in the same way that he/she normally converses. Unfortunately, traditional ways of assessing reading skills sometimes erroneously tend to focus much on word accuracy and reading rate. This teaching method may even have adverse effects, biasing the learner towards fast reading, while neglecting appropriate phrasing, expressivity and comprehension [3].

To automatically assess phrasing and expressivity for child reading, recent study [20] defined two types of fluency measures: prosodic and lexical. Prosodic fluency is measured in terms of phrasing, the ability to break up complete sentences to meaningful chunks, and word prominence. On the other hand, lexical classification incorporated false starts and hesitations, important aspects of the speech data to consider that were also included for the datasets analysed in the current work. False starts and hesitations can be assessed using refined language modelling, while implementing the prosody and phrasing tests usually requires further and even labelling of expert annotators to mark phrasal breaks and emphasised words.

Locally, a recent pilot study have been conducted that implements a newly developed mobile language learning application [15]. Importantly, this application was designed to focus both on developing the necessary language comprehension and reading skills of Grade 3 learners. The current version of the application supports isiZulu as a first language. Experimental work have been carried out to achieve automatic oral assessment of the sentences learners would read during usage. This paper presents the development of the necessary child ASR

systems and scoring metrics required. To this end the analysis below relied upon samples of recorded learner sentences, grouped into relevant reading assessment categories. Subsequent verdicts of the learner reading attempts were then automatically inferred within a CAPT framework. Due to practical considerations, speech recording in the application also required some restrictions on the text prompt length given a speech rate estimate to ensure reading was attempted appropriately at a moderate pace.

2 Background

Language learning strategies employing CALL systems is advantageous, since it can enable flexible, self-paced learning options, which the learner can access anytime and anywhere. These offerings can be cheaper than learning through human-to-human interaction alone and immersive learning is possible if real-time feedback can be provided in conjunction with personalised guidance. Since many CALL systems implements CAPT, an important consideration in the design of CALL is the ability to accurately detect pronunciation errors to provide constructive corrective feedback to guide language learners [26].

Automatic pronunciation error detection (APED) approaches usually present a predefined utterance text to the language learner, after which the learner attempts to say the text correctly. To aid in the learning process, a reference speech sample of the same utterance made by a fluent speaker of the language can be included. The APED system then evaluates the spoken audio to determine potential mispronunciations [27].

2.1 Second Language Data

The ability of CAPT systems to assess reading mistakes also depends on the availability of learner data sets. Unfortunately obtaining adequate speech samples from either young or second language (L2) learner data remains a difficulty. According to [22] even for the technologically well-equipped Dutch language non-native L2 ASR corpora is difficult to obtain. In their opinion, a main problem with ASR research of L2 learners was the insufficiency of resources which seems to currently apply to all languages other than English. Testing L2 speech fluency as a CAPT tool, the authors of [7] agreed that most research on the topic focused on English and the work's reliance on ASR based assessment techniques. The challenges to solve the task at hand included applying cost effective measures to target learner data as well as setting various thresholds on acoustic and the perceptual phonetics, not forgetting the level of education of the L2 learners. In our work also, most of these factors had to be considered during data collection and subsequent analysis in the local isiZulu language. Initially, the required annotated isiZulu learner corpora were non-existent.

Apart from the amount of L2 data, another aspect of consideration is that ASR accuracy is lower for non-native speakers, since there exists acoustic mismatch in these speakers' pronunciation compared to native speakers [17]. Similarly, in the context of this article, the same problem arises, given the additional

acoustic differences resulting from speech samples produced by children. In the L2 case, the mismatch is due to the considerably larger variability of non-native speech. Non-native speakers tend to make specific pronunciation errors related to their mother tongue. This means that L2 learners also require dedicated L2 instruction, including individual guidance and feedback. ASR-based application development for L2 learners, such as CALL approaches that support L2 learners in acquiring oral skill, is hindered by lower ASR performance of non-native speech [22].

2.2 Child Speech Recognition

Child speech recognition (CSR) is more difficult to achieve because of a diverse set of challenges. Limited availability of child speech corpora are a major development impediment. For this reason, an initial strategy to create CSR is to focus on the use of adult speech models first. Unfortunately, earlier experiments, in for example [18] did confirm large degradation of two and up to five times the word error rates. Back then, a secondary finding of the same work already showed performance degradation with decreasing age. Overall, the speech and language characteristic of children are so substantially different than adult speech, that recent modelling approaches based on deep learning techniques still test significantly less accurate than for adults [25].

Considering children's speech characteristics there are a number of important factors, which all contribute to an inherent high level of variability: Physical, articulatory and also linguistic expressions [21]. Moreover, these attributes change with the developmental stage of a child's life. Studies have shown age-dependant characteristics in formant and fundamental frequency measurement of child speech. On average the speaking rate of children is slower than adults with higher variability in speech rate and vocal effort. Recorded child speech samples in [18] also contained more disfluencies and mispronunciations, particularly for younger children, ages 8–10 years as opposed to an 11–14 year group.

The authors in [18] compared the dynamic range of formant values for various vowels spoken by child and adult speakers and showed increased variability in the formant values for child speakers. This results in greater overlap among phonetic classes used in ASR modelling approaches which increases the complexity of the classification problem. In terms of ASR, a greater variability, as seen in child data, usually also means a larger data requirement. This highlights the major challenge to CSR: a lack of child data [21].

To circumvent the data sparseness, researchers have produced numerous interventions in ongoing efforts to improve CSR. These methods include testing age dependent acoustic models to control for the child's developmental stage [18]. While training of L2 acoustic models from scratch will reduce the acoustic mismatch [14], the shortage of this data type forced the use of several other approaches to tackle the data insufficiency problem. Multi-task learning [8] is for example a method to create acoustic models for related tasks such as recognising adult and child speech with one model. Transfer learning is another approach to update deep neural acoustic models for low-resource tasks [22]. Recently,

specially designed data-efficient TDNN-F network acoustic models have been developed for child ASR [13]. Although the amount of training data required is less, other methods to incorporate adult training data into the deep acoustic modelling estimation has been shown to also be effective. Model pre-training using data from adult speech have been applied to develop deep LSTM-based hybrid systems successfully [10]. In fact, with enough speech data, it has been shown that a child speech LSTM-based hybrid acoustic model can perform just as well as an adult speech model in a large vocabulary speech recognition (LVSR) task [13].

2.3 APED Techniques

It is difficult to distinguish the difference between a right and a wrong pronunciation. There exists an entire scale of pronunciations ranging from unintelligible speech to native speech. It follows that attempting to quantify a pronunciation error is a challenging task [23]. Under the hood, most APED is based on state-of-the-art ASR techniques and therefore the steady advancement in ASR systems usually enables updated APED implementation [26].

The typical way in which APED systems can be implemented is by applying a forced alignment of the language learner's speech with the fixed phone context emanating from the presented prompt. The corresponding sequence of phone models from a standard speech recognition system can then be used to obtain likelihood or probability scores as confidence measures. Together, these measures derived from the phone model decodes provide an indication of the similarity of pronunciations between training data from which the ASR system was developed and the pronunciations in the learner's utterance [26]. Confidence metrics have also been derived that required a second recognition attempt, implementing a free phone loop, and subsequently incorporating the phone segment time alignments in goodness of pronunciation (GOP) scoring [24]. To perform the speech scoring for APED, our implementation relies on a phone dynamic programming (PDP) speech scoring algorithm which previously had been verified to achieve similar sensitivity than GOP scoring [6]. To implement a CAPT system, decision thresholds based on the scores derived from the APED system are required. These scores can be obtained from an equal error analysis (EER), with sufficient test examples [22].

3 Data

As introduced above, a necessary step in creating and evaluating a CAPT system for learner speakers would be to train and evaluate on sufficient L2 data. Furthermore, the additional inclusion of CSR in this work meant that suitable child speech data sets were needed. Apart from using the child speech data to reconfigure pre-trained ASR models to the child domain, some recordings would also be needed to support the CAPT system analysis later on. Automatic CAPT evaluation procedures require access to well annotated samples of imperfect reading

attempts in order to verify the reading assessment verdicts made. To build the required child ASR systems, the child speech data sets described in the following sections have been used together with the matching language component data sets of the locally developed NCHLT ASR speech corpora [4].

3.1 English Child Data

The Carnegie Mellon University (CMU) Kids corpus [9] is an English data set previously published by the Linguistic Data Consortium, which was designed to facilitate development of CSR early on. It comprises 9 h of sentences read aloud by 76 children of ages 6 to 11 years old. In total, 5180 of the utterances read by the child speakers were incorporated in the subsequent analysis. This set of recordings can be divided by origin based on the speakers that come from two separate populations: the majority were "good" readers to obtain good examples of reading aloud (44 speakers and 3333 utterances), and another group of less skilled readers to provide examples of reading errors and dialectic variants (32 speakers and 1847 utterances).

According to the corpus documentation, children were rarely asked to reread a sentence because their patience and attention spans were much shorter than those of adult speakers. Most speech were from the first reading attempt only which frequently resulted in utterances also containing pronunciation errors, hesitations, repetitions, etc. Some children would also make sounds, sing, or change prosody to alleviate the pressure. According to a file naming convention, the utterances were grouped into two bins. Sentences that were correctly read the same as the sentence prompt had been assigned the first bin (bin1). The second bin (bin2) meant a recorded sentence contained one or more divergences. Bin2 transcription text therefore contained additional special bracketed notations for phone labels that describe speech noises, partial, repeated or mispronounced words, and for miscellaneous acoustic events in the recording (e.g., "[noise]", "[begin_whisper]", etc.). The phone labels are strings of upper case Arpabet phonetic segment labels, separated by spaces and bounded by slashes (e.g., "/AE N/").

Evaluation Set. The top 10% of child speakers were selected to create a suitable evaluation set from the CMU Kids data. Eight good speakers were chosen for this set, which consisted out of the top four male and top four female speakers. Since the most fluent speakers in the data would be those who were able to record the highest number of "bin1" utterances, these were collated first. All these "bin1" utterances combined resulted in about 27 min of recordings. Next, "bin2" utterances had to be incorporated into the evaluation set to increase the total duration. Manual selection from this set of utterances were performed taking into account the various annotation labels on a per utterance basis. To include a recording, it should have contained minimal pronunciation errors, acceptable levels of noise, and no words that were not pronounced fully. Adding approximately 28 min of "bin2" utterances to the evaluation set increased the total duration to 55 min using 671 utterances.

3.2 IsiZulu Child Data

A similarly sized child speech corpus had to be produced in the isiZulu language. For this purpose, appropriate text prompts that would support the planned mobile language learning application were recorded. During the data collection process, the important considerations had been age and gender of the child speakers, the matching prompt difficulty and amount of data that can be collected per speaker. Mitigating the imposed restrictions, a prompt set design has been carried out by extracting sentences from story sections of isiZulu Grade 3 learner content from the VulaBula project. Systematic iteration through the various subsets of prompts on a per speaker and age group basis made it possible for all prompts to be recorded adequately. In total, the corpus consisted of approximately 6.3 h of recordings with 30 speakers who recorded more than 5 min each. In terms of the four age groups recorded, the recordings amounted to 0.25 h, 1.47 h, 1.78 h, 1.15 h and 1.65 h of data for each age group (7 to 11 year old children) respectively.

As with the English data, several recordings were made that contained imperfect reading attempts. To this end, the initial data collection guidelines stipulated that up to 2 of the specified 6 h could be made up of unsuccessful reading attempts. Afterwards, such recordings had been identified in a two-step process that included a second round of detailed annotations using the services of a transcription company. Initially, a 1.87 h (Retry) subset of 919 recordings were marked to indicate reading mistakes, leaving a 4.43 h subset of acceptable annotations. Additional recording attempts were made for these prompts, sometimes including a second or third example. To identify the different types of reading errors that occurred, a transcription protocol was developed for manual annotation of these utterances. This protocol specified the use of tags to indicate noises, fillers, partial words, and words with pronunciation errors. From the 4.43 h subset that remained (2388 recordings), later the transcriptions of a further 51 min set of 452 recordings were also verified manually, after initial automatic assessment had been carried out. Transcription updates to this (Verify) set included hesitation, pronunciation and larger transcription errors.

Evaluation Set. Initially, a proof of concept (POC) data collection phase was undertaken to compile a first hour's worth of well read isiZulu recordings. This included 502 recordings from 24 speakers. Since recordings containing reading mistakes were initially excluded, this clean, first hour set of recordings were subsequently designated as the isiZulu evaluation set in further analysis. The POC may be excluded from the 4.43 h subset above, which leaves approximately 3.4 h of data for training purposes.

4 Experimental Setup

The strategy to develop CSR started by using data-efficient TDNN-F modelling, fine-tuning adult NCHLT models [2] for an additional epoch on child data. If

successful, the Kaldi I (K1) TDNN-F models would subsequently be used to deploy the first pilot version of the proposed mobile language learning application. Improved CSR would logically benefit subsequent APED. Following this work, a decision was made to refine the speech scoring analysis by developing a new version of CSR models using next generation Kaldi (K2) techniques. This development consisted of fine-tuning a connectionist temporal classification (CTC) attention model, pre-trained on the NCHLT isiZulu ASR training component. At the time, the proposed K2 model refinement strategy had only been demonstrated successfully on broadcast domain-specific data sets [1].

4.1 CSR Model Development

Pre-trained model development preceded CSR model development training on the NCHLT corpus language components only, as this local accented data would be comparable in size for both languages. Since subsequent model refinement required compatible training transcriptions, all annotation labels were converted to appropriate phone transcriptions. In order to develop the best CSR models, the word error rate (WER) evaluation measures were converted to a phone error rate (PER) by configuring the decoding graph to support a set of equiprobable 1-grams for the same phone labels. All subsequent model adaptation was then achieved by estimating the PER of each model in order to find models generating low PERs. This rule had also been applied to determine the best K2 model (epoch) that were selected before fine-tuning with child data.

The adaptation data used for fine-tuning were as follows. Data from all remaining CMU kids English speakers not assigned to the evaluation set were included. Apart from converting the words to phone format, this meant that all bin2 type transcriptions needed were transformed by, removing all bracketed notations (e.g., "[noise]") from the text and converting Arpabet phone transcriptions appropriately. The final selection of recordings consisted out of 4277 utterances with a total duration of 7.8 h. Further processing for the isiZulu data set led to the selection of the 4.43 h of successfully made recordings as an initial adaptation training data set. Since model evaluation required appropriate test data, the 3.4 h subset of these recordings that excludes the POC set were also used for training.

4.2 APED

APED implementation consisted of the PDP scoring solution on top of CSR. Since this scoring system may be decoupled from the ASR, it readily allows comparison to be made between various ASR system outputs, fast-tracking development of different CSR models. To detect reading mistakes, identified utterances subsequently described by reading assessment classes in both CMU Kids and isiZulu learner data sets were then evaluated to test the capability of the APED configurations. Importantly, successful CSR development would mean an increase in correct classifications of utterances without reading mistakes first, compared to APED using a pre-trained NHCLT model. Subsequent PDP score

analysis took into account the expected and manually re-transcribed phone transcriptions for each "bin2" and "Retry" data set utterance in both the English and isiZulu languages during error detection. Lastly, computation of automatic reading assessment utterance detection rates required setting of dynamic programming (DP) scoring decision thresholds that may be based on experimental results included in the analysis below.

4.3 Reading Assessment Categories

Initially, the transcriptions and the accompanying meta data within the CMU Kids corpus were analysed for all speakers excluding the evaluation data. Pertaining to the findings, a set of categories were defined to describe the various aspects that could be taken into account for speech assessment. Intuitively, the markers associated with some of these categories are related to reading mistakes, while others pertain to unique acoustic properties in the recorded data.

At first, each utterance was labelled according to four main descriptive categories: "Pronunciation", "Noise", "Location", and "Age". A breakdown definition of these main utterance classes has been provided in the sections below. Finally, designing a similar category specification for the annotated isiZulu Retry data, sometimes, refined sub-classes were required to support each main category.

Utterance Class Definitions. The "Pronunciation" category describes the phonetic pronunciation errors found in the child's spoken sentence when compared to the reference sentence. Converting to phone representations, ground truth statistics were then calculated in terms of phone substitutions, insertions and deletions based on the edit distance of the phone labels contained within the strings of the two phone transcriptions for each utterance. Other than these counters describing the changes of phone labels, Table 1 provides detailed descriptions on three additional sub-classes that were required.

Table 1. Pronunciation category sub-class definitions, preparation and selection of utterances pertaining to the respective child data sets in both languages

Sub-category	CMU kids utterances	isiZulu utterances
Stutter	Bin2 utterance short phone transcriptions, with notations of two or less phones	Four or less character partial word part transcripts, contributing two or less phones
Incomplete sentence	Transcriptions including the "[cut]" notation which was not read fully	Recorded sentence transcription contained fewer words than the reference transcription
Perfect	Bin1 utterances read correctly	POC set utterances

In all data sets the transcribers included noise markers to indicate when various noisy acoustic events occurred in the audio, mostly alongside the spoken

parts. To inform this analysis, these annotations were subsequently consolidated under the Noise main category. This way, the effect that noisy utterances have on the APED analysis becomes measurable. In the CMU kids data, these tags were included for the "bin2" utterance transcriptions to indicate the start of any noisy interferences such as general noise, crosstalk, microphone noise, whispering and yawning. In addition, there were also bracket notations that indicated the start and ending of noises if its effect was localised coinciding with particular words. According to protocol, the isiZulu transcribers were also asked to include noise tag annotations. In general, any external noises that did not originate from the child reader were assigned the background noise tag. Since children do not always speak consistently, the transcribers were tasked to include whisper or shout annotation tags when speaking volume changed drastically. Apart from this, transcribers also indicated other speaker noises such as filler words, laughter or clearing of throat sounds.

Lastly, there were a bag of softer effects that could be deduced from the transcriptions. It was possible to extract hesitation indications from the transcriptions after a decision was made with the transcription company to include it as this was sometimes indicated instead of using a partial word indicator correctly. Basically, a hesitation occurred when the partial word indicator was inserted next to an original token from the reference sentence. For clarity, Table 2 presents a breakdown of the utterance category classes provided for further analysis of the designated APED evaluation data sets in both languages.

Utterance Class Breakdowns. The CMU Kids recordings mentioned in Sect. 3.1 were made at two locations, mapping to distinct data sets described in Table 2 as follows:

- SUM95: recorded in the summer of 1995, and were enrolled in either the Chatham College Summer Camp, or the Mount Lebanon Extended Day Summer Fun program in Pittsburgh
- FP: recorded in April 1996 at Fort Pitt School in Pittsburgh

Between the two sets of recordings, the children recorded at SUM95 were said to be the most skilled readers. The corpus designers required examples of reading mistakes and dialectic variants. Therefore, FP was included as a recording location because the children in that school were at risk of becoming poor readers. Similarly the manually refined isiZulu utterances data sets were as follows:

- Retry subset: The 1.87 h marked subset of recordings with identified reading mistakes, some of which were re-recorded.
- Verify subset: The additional 51 min set of recordings which were supposed to have correctly matched transcriptions but have been displaying a reduced acoustic match after initial ASR analysis.

CMU data noise level categories were defined specifying three sub-categories; low, medium and high pertaining to utterances containing meta label counts of 0–3, 4–6 and 7–9 respectively. After manually verified meta data were available, the isiZulu noise utterance sub-categories could also be defined as follows:

Table 2. Breakdown of utterance category counts collected at the SUM95 and FP locations for the CMU kids data and manually refined isiZulu Retry and Verify data sets respectively.

Main category	Sub-categories	#SUM95	#FP	#Retry	#Verify
Pronunciation	Substitutions	1648	2679	826	105
	Insertions	2914	3948	187	14
	Deletions	1437	3222	15476	125
	Stutter	290	311	56	–
	Incomplete sentence	58	76	118	4
	Perfect	1260	330	–	502 (POC)
Noise	Low	2183	855	–	–
	Medium	351	510	–	–
	High	125	305	–	–
	Hesitations	–	–	236	74
	Speech noises	–	–	80	14
	Soft noises	–	–	39	–
	Background noises	–	–	238	94
Age	Age 6	134	212	–	–
	Age 7	495	656	36	3
	Age 8	1289	503	163	155
	Age 9	742	302	243	150
	Age 10	–	–	231	66
	Age 11	–	–	245	78

- Hesitations: Short stops and pauses transcribers marked after certain word transcriptions when the children did not read words fluently.
- Speech noises: Filler words, laughter, and other noises from the speakers such as clearing the throat.
- Soft noises: Any whispers, sighs and unintelligible word markings pertained to this category.
- Background noises: All other external noise not originating from the speaker.

Finally, since this meta data was collated as part of the data collection process, of the various data sets, each utterance could be counted in terms of the age of the child speaker that produced the utterance in the recording.

5 Results

Table 3 now summarises the phone error rate (PER) assessments made during CSR model development. In both languages, initial pre-trained NCHLT based models were created first and subsequently evaluated on the selected evaluation data. Since much less isiZulu data (only 4 h) were available for training, initially the isiZulu model development focused on complete set training and evaluation on all the data, before excluding the POC data as the designated evaluation set. All error rates were obtained from 1-best decodes, which extracts the best path from the Kaldi decoding lattices. Clearly, the NCHLT pre-trained K2 conformer CTC

models initially produced the largest acoustic mismatch from the test data (compare 50.48% and 33.1% PER). The new models did not generalise as well to child domain data than the older TDNN-F acoustic modelling strategy. Also, the significantly higher PERs of the English pre-trained models compared to the isiZulu contrasted the additional CSR error obtained using a South African accented model adapted to cross-accent American English child data.

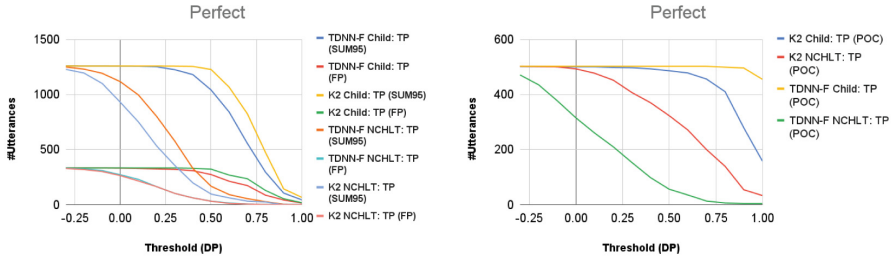
Table 3. Summary of PER values for various acoustic models developed

Language	Model	Test data	PER
English	K2 conformer CTC (NCHLT)	Evaluation set	50.48%
	TDNN-F (NCHLT)	Evaluation set	45.08%
	Adapted TDNN-F (7.8 h)	Evaluation set	26.46%
	Adapted K2 conformer CTC (7.8 h)	Evaluation set	16.1%
isiZulu	K2 conformer CTC (NCHLT)	Complete set	33.1%
	TDNN-F (NCHLT)	Complete set	20.36%
	Adapted K2 conformer CTC (3.4 h subset)	POC set	7.42%
	Adapted TDNN-F (4.43 h)	Complete set	2.55%
	Adapted TDNN-F (3.4 h subset)	POC set	2.31%
	Adapted K2 conformer CTC (4.43 h)	Complete set	0.37%

As expected, adapting the English models to the child domain obtained much lower PERs with the adapted K2 conformer CTC model significantly reducing the recognition error to a relatively low 16.1%. Interestingly, adaptation on the isiZulu child data proved even more successful with limited data, lowering the PER to 7.42%. However, this was not the best result that could be obtained. The older TDNN-F models outperformed the transformer based CTC model with a PER as low as 2.31%. Testing the K2 conformer CTC model on the complete set instead finally did result in the lowest error rate, transcribing the isiZulu data with phone labels producing only 0.37% PER.

5.1 APED

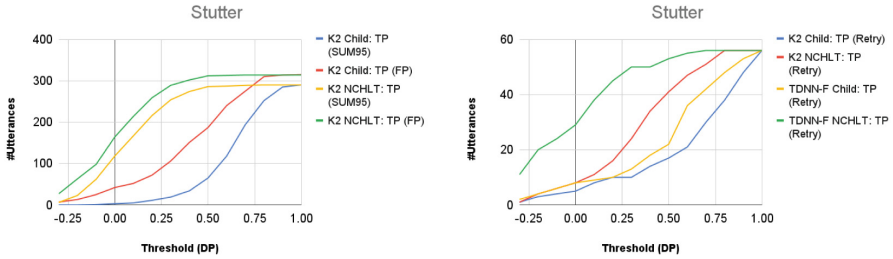
To consider the performance of the APED classification over a range of DP classification thresholds a receiver operating characteristic (ROC) curve analysis have been performed. Generally, ROC analysis compares true positive (TP) and false positive (FP) detection rates to make assessment verdicts. To this end, TP detection rates of utterances belonging to the Perfect reading assessment category were analysed first (Fig. 1). The purpose of this experiment was to determine appropriate DP thresholds for the APED to not penalise correctly read utterances. Figures 1(a) and 1(b) compare the selected TP rates for the English and isiZulu child data separately, including plots of the detection rates obtained with the various ASR modelling options. The DP score numeric values itself ranged from -1.00 to 1.00. A higher DP score (closer to 1.00) meant that better acoustic match was achieved to originally expected phone transcription.



(a) English (b) isiZulu

Fig. 1. Comparison of TP detection rates for utterances pertaining to the Perfect category at various DP scoring thresholds and various ASR models

At DP score values lower than 0.00, most of the utterances were correctly classified as Perfect category sentences. Also the child model options performed much better than the pre-trained NCHLT models, correctly classifying more utterances at higher DP thresholds. The CMU kids data plot in Fig. 1(a) showcased unique findings for the respective recording locations. Clearly, when designing a CSR system to evaluate more proficient readers (SUM95), the K2 model improved TP detection rates substantially over the TDNN-F CSR model. Unfortunately, although present, these gains were not as pronounced for the FP data detection plot. Finally, as observed in Table 3 above, the successful TDNN-F isiZulu CSR also translated to impeccable TP rates.



(a) English (b) isiZulu

Fig. 2. Comparison of TP detection rates for utterances pertaining to the Stutter category at various DP scoring thresholds and various ASR models

Subsequently, the detection of stuttering could be considered a good example study of a prominent reading impediment present in the learner data. Therefore, to acquire an initial sense of how effective the APED technique could be, similar comparison of TP error detection rates could be plotted in Fig. 2. Unfortunately, in general these plots seemed to confirm a tendency that better CSR models perform worse at detecting the stutters.

More APED thresholds were required for analysis. To ensure most correct sentences across all data sets received the TP verdict label, various CSR model specific thresholds were read off the Perfect sentence category plots (Fig. 1) from above. Consequently, in Table 4, percentages close to 100% were obtained for the Perfect reading category, selecting the 0.5 threshold for K2 conformer CTC CSR systems, 0.3 for TDNN-F (K1) English and finally a value of 0.7 for the well-performing K1 isiZulu CSR model.

Table 4. Percentage of TP APED reading assessment category utterance detection rates, setting particular DP scoring thresholds of 0.5 for the K2 conformer CTC CSR models as well as 0.3 for the TDNN-F (K1) English and 0.7 for the TDNN-F (K1) isiZulu CSR models respectively.

Sub-categories	SUM95 K1	SUM95 K2	FP K1	FP K2	Retry K1	Retry K2	Verify K1	Verify K2
Substitutions	16.38	32.71	40.13	58.64	82.81	56.54	26.67	39.05
Insertions	25.26	56.49	54.76	74.09	92.51	45.99	64.29	57.14
Deletions	30.48	37.79	67.97	77.28	97.65	87.34	16.00	28.80
Stutter	14.48	22.41	45.66	59.16	75.00	30.36	–	–
Incomplete sentence	25.86	44.83	59.21	93.42	62.71	19.49	50.00	75.00
Low Noise	6.00	10.63	23.74	30.76	–	–	–	–
Medium Noise	8.83	23.36	19.02	34.51	–	–	–	–
High Noise	16.00	30.40	32.13	44.59	–	–	–	–
Hesitations	–	–		–	57.20	27.12	85.14	4.05
Speech noises	–	–		-	91.25	67.50	78.57	7.14
Soft noises	–	–		–	89.74	66.67	–	–
Background noises	–	–		–	70.17	42.86	78.72	0.00
Perfect	97.22	97.70	96.06	96.97	–	–	100.00	97.61
Age 6	91.79	76.87	61.79	49.06	–	–	–	–
Age 7	89.09	85.25	76.83	67.99	75.00	100.00	100.00	100.00
Age 8	93.10	88.21	77.14	70.97	26.38	45.40	74.84	96.13
Age 9	95.69	92.05	81.79	73.84	25.93	48.15	85.33	96.00
Age 10	–	–	–	–	32.90	68.83	86.36	96.97
Age 11	–	–	–	–	32.65	57.96	88.46	100.00

As expected, the observed correct APED was higher for the FP data set than the SUM95 one, with the improved K2 conformer CTC model clearly outperforming the K1 TDNN-F model significantly. Unfortunately though, not all reading errors could be detected in this data. See, for example, the 22.41% APED detection rate for utterances containing stutters of the SUM95 K2 conformer CTC model. While isiZulu K2 APED fared similarly, clearly the isiZulu K1 detection method seemed to be most effective, detecting most reading mistakes from the Pronunciation assessment category classes within the Retry data set. Interestingly, though, APED detection of noises affected the APED system verdicts rather differently. With the exception of the Verify data set, better performing CSR models were more effective in detecting utterances from the noise

category. Lastly, the reading attempts classified APED acceptable on a per age group basis were included for comparison. Correctly, less FP utterances were classified as good reading. Also, the higher CSR TP percentages for age groups of 8 and 9 years may indicate more successful phone recognition for the speech of older children.

6 Conclusion

The limited availability of child speech data as well as the greater variability within that data presents unique challenges for CSR development. To this end the choice of ASR modelling and the subsequent successful implementation of a suitable data sharing strategy are of key importance. The preceding analysis proved successful CSR development is possible using both the well known CMU kids ASR corpus as well as a new locally developed isiZulu child data set based on a similar collection strategy. In particular it was possible to adapt local NCHLT-based ASR successfully to the new domains, where in the case of supporting CSR in the local isiZulu language less than 4 h data were used for the purpose of creating APED implementation in support an early literacy application. Interesting effects were observed comparing older data-efficient TDNN-F and more recent transformer-based conformer CTC modelling strategies. For starters, the powerful overtraining tendency of the latter model proved particularly beneficial in adapting NCHLT adult speech models, cross-accent to the American English child domain. However, it remains unclear why the same was not true for the isiZulu data where the TDNN-F modelling strategy outperformed. Considering the paradoxically low PER produced in Table 3 when evaluating on the training data for example, considerable conformer CTC overtraining was still achieved in isiZulu. This finding did suggest the model architecture as a possible cause of the discrepancy. Moreover, the low number of reading mistakes identified for the utterances of the Verify data set, confirmed that the data collection was in order. However, it was not yet clear why the detection of the more subtle noises in this data were not as detectable with conformer CTC modelling, but succeeded with the TDNN-F models. Subsequent data release with the South African Centre for Digital Language Resources[1] (SADiLaR) have been planned. Clearly, comparably successful APED was achieved in both languages, in the end identifying high percentages of reading mistakes made by the isiZulu readers.

References

1. Badenhorst, J., Mak, F., Govender, A.: Exploring ASR fine-tuning on limited domain-specific data for low-resource languages. Digit. Humanit. Assoc. South. Afr. Digit. Humanit. Incl. **5**(1), 1–8 (2024). https://doi.org/10.55492/dhasa.v5i1.5024

[1] https://www.sadilar.org/.

2. Badenhorst, J., de Wet, F.: Gauging the accuracy of automatic speech data harvesting in five under-resourced languages. Digit. Humanit. Assoc. South. Afr. Spec. Issue: Crossroads Digit. Humanit. **4**(2), 1–17 (2023). https://doi.org/10.55492/dhasa.v4i02.4031
3. Bailly, G., Godde, E., Piat-Marchand, A.L., Bosse, M.L.: Automatic assessment of oral readings of young pupils. Speech Commun. **138**, 67–79 (2022). https://doi.org/10.1016/j.specom.2022.01.008
4. Barnard, E., Davel, M.H., van Heerden, C., de Wet, F., Badenhorst, J.: The NCHLT speech corpus of the South African languages. In: Proceedings of the 4th Workshop on Spoken Language Technologies for Under-Resourced Languages, St. Petersburg, Russia, pp. 194–200, May 2014
5. Bolanos, D., Cole, R.A., Ward, W.H., Tindal, G.A., Schwanenflugel, P.J., Kuhn, M.R.: Automatic assessment of expressive oral reading. Speech Commun. **55**(2), 221–236 (2013). https://doi.org/10.1016/j.specom.2012.08.002
6. Davel, M.H., van Heerden, C.J., Barnard, E.: Validating smartphone-collected speech corpora. In: Third Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU), pp. 68–75 (2012)
7. Detey, S., Fontan, L., Le Coz, M., Jmel, S.: Computer-assisted assessment of phonetic fluency in a second language: a longitudinal study of Japanese learners of French. Speech Commun. **125**, 69–79 (2020). https://doi.org/10.1016/j.specom.2020.10.001
8. Duan, R., Kawahara, T., Dantsujii, M., Zhang, J.: Pronunciation error detection using DNN articulatory model based on multi-lingual and multi-task learning. In: 10th International Symposium on Chinese Spoken Language Processing (ISCSLP), pp. 1–5. IEEE (2016). https://doi.org/10.1109/ISCSLP.2016.7918389
9. Eskenazi, M., Mostow, J., Graff, D.: The CMU Kids Corpus (1997). https://doi.org/10.35111/b4v0-ff65
10. Fan, R., Afshan, A., Alwan, A.: Bi-APC: bidirectional autoregressive predictive coding for unsupervised pre-training and its application to children's ASR. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7023–7027. IEEE (2021). https://doi.org/10.1109/ICASSP39728.2021.9414970
11. Hasbrouck, J., Tindal, G.A.: Oral reading fluency norms: a valuable assessment tool for reading teachers. Read. Teach. **59**(7), 636–644 (2006). https://doi.org/10.1598/RT.59.7.3
12. Howie, S.J., Combrinck, C., Roux, K., Tshele, M., Mokoena, G., McLeod Palane, N.: PIRLS Literacy 2016 : South African Highlights Report (Grade 4). Technical report, Centre for Evaluation and Assessment, Faculty of Education, University of Pretoria, Pretoria (2016)
13. Liao, H., et al.: Large vocabulary automatic speech recognition for children. In: Interspeech, pp. 1611–1615 (2015). https://doi.org/10.21437/Interspeech.2015
14. Lin, J., Gao, Y., Zhang, W., Wei, L., Xie, Y., Zhang, J.: Improving pronunciation erroneous tendency detection with multi-model soft targets. J. Sig. Process. Syst. **92**, 793–803 (2020). https://doi.org/10.1007/s11265-019-01485-2
15. Marais, L., Wilken, I., Pretorius, L., Davel, L.C.P.: Multimodal, multilingual dynamic stories for literacy development and language learning. In: Proceedings of the 5th International Conference on Conversational User Interfaces, pp. 1–5. ACM (2023). https://doi.org/10.1145/3571884.3604303
16. National Reading Panel (US), National Institute of Child Health and Human Development (US): Teaching children to read: an evidence-based assessment of the

scientific research literature on reading and its implications for reading instruction: reports of the subgroups. University of Virginia Libraries, The City (2000)
17. Park, S., Culnan, J.: A comparison between native and non-native speech for automatic speech recognition. J. Acoust. Soc. Am. **145**(3), 1827 (2019). https://doi.org/10.1121/1.5101679
18. Potamianos, A., Narayanan, S.: Robust recognition of children's speech. IEEE Trans. Speech Audio Process. **11**(6), 603–616 (2003). https://doi.org/10.1109/TSA.2003.818026
19. Rao, P., Swarup, P., Pasad, A., Tulsiani, H., Das, G.G.: Automatic assessment of reading with speech recognition technology. In: Proceedings of the 24th International Conference on Computers in Education, pp. 1–3. Asia-Pacific Society for Computers in Education (2016)
20. Sabu, K., Rao, P.: Automatic assessment of children's oral reading using speech recognition and prosody modeling. CSI Trans. ICT **6**(2), 221–225 (2018). https://doi.org/10.1007/s40012-018-0202-3
21. Shivakumar, P.G., Georgiou, P.: Transfer learning from adult to children for speech recognition: evaluation, analysis and recommendations. Speech Commun. **63**, 101077 (2021). https://doi.org/10.1016/j.csl.2020.101077
22. Wei, X., Cucchiarini, C., van Hout, R., Strik, H.: Automatic speech recognition and pronunciation error detection of Dutch non-native speech: cumulating speech resources in a pluricentric language. Speech Commun. **144**, 1–9 (2022). https://doi.org/10.1016/j.specom.2022.08.004
23. Witt, S.M.: Automatic error detection in pronunciation training: where we are and where we need to go. In: Proceedings of International Symposium on Automatic Detection on Errors in Pronunciation Training, pp. 1–8. KTH Computer Science and Communication (2012)
24. Witt, S., Young, S.: Phone-level pronunciation scoring and assessment for interactive language learning. Speech Commun. **30**(2–3), 95–108 (2000). https://doi.org/10.1016/S0167-6393(99)00044-8
25. Yu, F., et al.: The SLT 2021 children speech recognition challenge: open datasets, rules and baselines. In: IEEE Spoken Language Technology Workshop, pp. 1117–1123. IEEE (2021). https://doi.org/10.1109/SLT48900.2021.9383608
26. Zhang, L., et al.: End-to-end automatic pronunciation error detection based on improved hybrid CTC/attention architecture. Speech Acoust. Audio Sig. Process. Appl. Sens. **20**(7), 1809 (2020). https://doi.org/10.3390/s20071809
27. Zhang, Z., Wang, Y., Yang, J.: Text-conditioned transformer for automatic pronunciation error detection. Speech Commun. **130**, 55–63 (2021). https://doi.org/10.1016/j.specom.2021.04.004



Automated Enhancement of isiZulu Data Collection for the African Health Research Institute

Jaco Badenhorst(✉) and Avashna Govender

Voice Computing Research Group, CSIR Next Generation Enterprises and Institutions Cluster, Pretoria, South Africa
jacbadenhorst@gmail.com, agovender1@csir.co.za

Abstract. IsiZulu, despite being one of the most widely spoken languages of South Africa, is classified as under-resourced. As a consequence, development in Automatic Speech Recognition (ASR) in these languages is slow. Furthermore, data available for training ASR in isiZulu is currently constrained to domains that comprise mainly of content that are government, biblical or news in nature. This poses limitation to developing ASR systems that are more fit-for-purpose. This paper investigates various methods for improving the accuracy of an isiZulu ASR system that is intended to transcribe audio in a domain that does not exist in the pre-trained model. The experimental results in this work shows promising results when applying adaptation techniques using limited domain-specific data. In addition, the paper presents a semi-automated transcription system that includes speech scoring and human-in-the-loop verification of the output transcriptions. The paper also discusses the challenges of adapting ASR models to domain-specific data and presents the contributions and novelty made to the field from the perspective of handling an under-resourced language.

Keywords: Automatic speech recognition · Fine-tuning · Automatic transcription system · Code-switched data

1 Introduction

The majority, 10 of the 11 official written languages of South Africa, are under-resourced. This poses a challenge in the speech technology landscape as offering solutions that are localised to South African languages becomes limited. isiZulu is the most spoken African language in South Africa and yet applications that include speech recognition are few and far between. Developing typical ASR models requires large amounts of transcribed speech and text data (100+ hours), and often the type of data used to train ASR models informs the accuracy that can be achieved. One of the greatest challenges that further exacerbates the problem is that the only suitable data available for training high performing ASR systems comprises mainly of biblical readings or government speeches. Such content is

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 91–106, 2025.
https://doi.org/10.1007/978-3-031-78255-8_6

not suitable for the development of real-world ASR application as the content that one typically wishes to transcribe falls outside of these domains. Therefore, achieving good accuracy on under-resourced languages in South Africa remains a challenge. The paper explores a method of adapting a pre-trained model to a domain outside of the domains that exists within the pre-trained model. The specific domain chosen is healthcare and therefore we would expect that by attempting to transcribe healthcare data using only government speeches and biblical content would perform extremely poorly. In addition, the style of healthcare data transcribed in this work is conversational in nature. So not only does the vocabulary differ but also the speech style which are two fundamental differences between the source and target data in this work. To disclose how such evaluation and adaptation data was obtained, after introducing the project background, we describe the necessary interview style data development. After this, we investigate various strategies to leverage the limited domain-specific data that we have in order to optimize the performance of ASR transcriptions. The paper discusses the development of these ASR models and the challenges of text modelling for a language like isiZulu. In future practical implementation, we foresee development of a software solution featuring a semi-automatic transcription design that involves speech scoring and human-in-the-loop verification of the transcription.

2 Project Background

The African Health Research Institute (AHRI) research context revolves around the meticulous collection of content from interviews and group discussions, a method to acquire the sufficient health care data necessary to inform scientific study. As such, the extensive repository of high-quality audio recordings derived from past interviews, presents an opportunity to customise existing ASR models to content that is conversational and therefore more fit-for purpose to real-world applications. These datasets, already transcribed, translated, and quality-checked, serve as a foundational resource for developing potentially highly accurate ASR systems aligned with AHRI's unique requirements. The overarching application of the ASR would then be to expedite the transcription, subsequent transcription verification, and translation of the conversational content and thus enable researchers at AHRI to carry out their tasks more efficiently.

Recognising the nature of AHRI's data, ASR models of the Council of Scientific and Industrial Research (CSIR) needed to be refined through a process called adaptation. This customization aimed to enhance the accuracy of transcriptions produced for AHRI's specific style of data. During the defining of this project, several critical research questions had been considered. These were as follows:

- How well does the existing (baseline) CSIR ASR models transcribe AHRI isiZulu conversational speech?
- How much improvement in terms of accuracy can be achieved by adapting the baseline ASR models using AHRI's dataset?

- What are the data requirements (i.e. how much data is needed) for adapting the CSIR isiZulu ASR baseline models to transcribe conversational speech of high accuracy?
- How effective would subsequent scoring methods be with regard to supporting human verification of the accuracy of the transcriptions?

2.1 Audio Capturing Process

In the initial discussions with AHRI staff members, the protocol for conducting interviews was thoroughly examined. AHRI clarified that, preceding the actual data collection for their studies, meticulous preparation of research questionnaires takes place. These questionnaires encompass a wide array of topics tailored to the specific focus of each study. Typically involving two individuals, an interviewer (facilitator) and a participant, each interview is conducted in a carefully structured manner to ensure the comfort and trust of the participants given the sensitive nature of the content. The interviews are deliberately set up in an informal fashion to foster a conducive atmosphere.

For audio recording purposes, AHRI utilizes a non-intrusive digital recorder setup. The recorder is strategically placed within clear view and at an optimal distance from the facilitator and participant to ensure an adequate recording volume for capturing the speech signal. The recording setup is designed to accommodate the natural flow of conversation, resulting in recordings that include multiple speaking turns and verbal speaker acknowledgements. It is noteworthy that these acknowledgements may occasionally lead to overlapping speech segments in the audio. Additionally, speakers may exhibit phenomena such as fillers, laughter, or repetitions of words during hesitations and stuttering. An important consideration that was highlighted, was the occurrence of code-switching by isiZulu speakers during AHRI interview conversations. This linguistic phenomenon would add complexity affecting the ASR model's ability to accurately recognise the audio content. The duration of AHRI's interviews is typically in the range of 30–120 min.

2.2 Design Considerations for an Annotation Interface

It is envisioned that a few processing stages would be applicable informing the application design of an automated transcription service. Prior to performing any transcriptions, reprocessing of the complete interview recording is required to provide a suitable audio segmentation for ASR consumption. Subsequently, following automatic transcription of the prepared input, a speech scoring analysis is performed on each shorter segment of transcribed audio. The speech scoring analysis in turn supplies a web interface application with the necessary information, indicating and ranking potential transcription errors still present after automatic transcription. Detailed human-in-the-loop verification of the transcription may then be performed, ending with the provision of a final speech transcript in a suitable format.

To achieve an effective speech scored solution, ASR systems have to be applied in such a way as to effectively distinguish the difference between correct and wrong pronunciations in an existing first transcript of the audio data. This process had been a challenging task in the past, where previously one use had been to measure pronunciation quality of second language speech examples in a learning environment [11]. It is expected that employing a similar strategy, locating errors in ASR transcripts, would be just as effective. A typical way to implement the required automatic pronunciation quality assessment is by applying a forced alignment of the incoming audio with fixed phone context emanating from ground truth transcriptions. The corresponding sequence of phone models from a standard speech recognition system can then be used to obtain likelihood or probability scores as confidence measures. It also follows that under the hood, such scoring solutions requires state-of-the-art ASR techniques which then usually enables improved transcription error detection [13].

Confidence metrics have also been derived that require a second recognition attempt, implementing a free phone loop, and subsequently incorporating the phone segment time alignments in goodness of pronunciation (GOP) scoring [12]. We expect our initial speech scoring implementation to be based on a phone dynamic programming (PDP) scoring algorithm which previously had been verified to achieve similar sensitivity to GOP scoring [3]. This strategy will effectively enable comparison of the initially produced ASR transcript phone contexts with the phone labels produced by the free phone loop.

3 Interview Style Data Development

To assess the accuracy of our existing ASR models on AHRI's interview recordings, a strategic approach was adopted. Initially, a series of interview simulations was conducted using recently developed short questionnaires. This facilitated the creation of approximately one hour manually annotated evaluation data for six participants, encompassing both genders. It is important to note that this data collection occurred before obtaining full ethics approval for broader data sharing. Collaboratively, the CSIR and AHRI established a standardized operating procedure (SOP) annotation protocol for the evaluation data.

3.1 Operating Procedure Annotation Protocol

The SOP's transcript format was defined, featuring a document header containing metadata such as recording date, participant and interviewer identification, and an indication of the AHRI project categorising the type of interview content recorded. Following CSIR specifications, annotations were made on a per-speaker-turn basis, with each turn annotated on a separate line. Speaker identifiers ("<i>" for interviewer or "<p>" for participant) were employed at the beginning of each line to distinguish between interviewer and participant speech. The annotation protocol also incorporated the use of tags to mark foreign

words, fillers, instances of laughter, and partial word fragments. This comprehensive annotation approach aimed to provide an understanding of the spoken content and account for various linguistic nuances and phenomena present in the AHRI interview recordings. The precision in developing this annotation protocol ensures the creation of a robust evaluation dataset for an accurate ASR evaluation.

3.2 Timestamps

In ASR development, the use of appropriately sized short audio segments is crucial for both training and evaluation phases. To address memory and processing constraints, model training typically employs short durations of a few seconds for training segments. Conversely, for assessing different complexities in evaluation data, subsets of short audio segments are selected. To enhance the evaluation dataset, a decision was made to implement timestamps after every speaker turn in the one-hour manually transcribed evaluation datasets. AHRI staff collaborated in introducing timestamps using transcription software, aligning with the requirement for precise segmentation.

3.3 Real-World Annotations

Collaborative efforts extended to the analysis of real-world interview data, whilst adhering to any ethical considerations. Identifying information was systematically removed from interview recordings to comply with ethical requirements. As part of understanding the implications sharing AHRI recordings, initially, two real-world interviews and their annotations were shared in the original AHRI database format. Analysis revealed that while verbatim AHRI transcriptions were intended, consistency in word tokens posed occasional challenges. Annotations for filler words, inaudible segments, incomplete words, or stuttering were sometimes provided in short descriptions, often enclosed in brackets. Additionally, remarks on pauses, names, background noises, and speaker noises (e.g., "clearing her throat") were present. The AHRI annotations adhered to a structure of interview questions and answers annotated on separate lines, preceded by a speaker ID. Notably, the real-world AHRI annotation texts included standard punctuation, capitalisation, and abbreviations.

3.4 Audio Dataset Construction

In preparation for ASR system development using real-world data, a comprehensive AHRI data preparation strategy was devised. The strategy encompassed four key steps for potential annotation updates:

- Step 1: Identifying information was marked with timestamps for subsequent exclusion in further processing.
- Step 2: AHRI personnel introduced timestamps at 5-minute intervals, forming an initial segmentation of audio.

- Step 3: Transcriptions were edited to ensure verbatim representation by applying the CSIR transcription protocol and removing additional markings and descriptions.
- Step 4: English and other foreign words were identified and marked within the transcriptions.

Subsequent discussions clarified that future data sharing, involving additional hours of audio and transcripts, may not require all annotation update steps. Depending on the size of the dataset, only executing the first and second steps would suffice. This outcome made an annotation update of an additional 7 real-world interviews possible, which was subsequently also consumed to test the development of the ASR model. Aligning the training transcripts and applying a further silence detection based automated segmentation of the 5-minute audios made extraction of the additional training data possible.

4 Experimental Setup

Recent international progress in ASR research has prompted a shift towards ongoing model development employing end-to-end techniques. One such model re-configured with a technique called Connectionist Temporal Classification (CTC) attention modelling was applied in answer to all the project research questions. This model incorporated a 12-layered conformer encoder, that at the time already resulted in significant recognition improvements in other work on newly developed radio news domain-specific data sets and disseminated in publication [1]. The conformer encoder essentially is a convolution-augmented transformer neural model [8]. A recipe [10] from the new next-generation Kaldi (K2) toolkit's Icefall repository was adapted to facilitate model training in support of the present ASR analysis.

4.1 Data

The different audio data utilised for ASR model development and testing is presented in Table 1, comprising 6 sets, three of which were only used for model training. In addition, Table 1 provides the total duration, total number of segments and speakers (spk) or news bulletins (bul) of each data set is comprised of.

The isiZulu NCHLT training set is the largest set consisting of more than 48 h of two-to-three-word short audio segments. These audio segments had previously been collected mostly from 202 university student participants, which read prompts in quick succession on a smartphone application. It should be noted that the characteristics or speaking style of the speech contained in these recordings are different from the long utterances that are typically recorded during interview conversations. The News training set is an existing dataset that was used in previous work [1]. This training data set contains short segments of news reading and has a duration of 8 h. This data was created using a selection of previously harvested short news reading segments (originating from 433 SABC isiZulu

Table 1. Summary of ASR audio data set duration, sub-category description and segment statistics

Data set	Dur (h)	#Description	#Segments
NCHLT train	48.5	202 spk	38 926
News train	8.25	433 bul	4 238
Real-world train	5.17	13 spk	2 815
News eval	0.51	18 bul	265
Simulated eval	1.04	8 spk	306
Real-world eval	0.78	4 spk	500

news bulletins) and were transcribed verbatim. Real-world interview adaptation data comprised 5 h of short audio segments. For the evaluation, 3 evaluation datasets were evaluated. The first, comprising news speech segments amounting to half an hour (originating from 18 SABC isiZulu news bulletins). The other two data sets comprise manually annotated and verified simulated and real-world interview data received from AHRI, amounting to a total of 306 and 500 short audio segments, respectively. The 500 segments of real-world interview data excluded any marked segments containing identifying information, which was removed before conducting any experiments.

4.2 Acoustic Modelling

In specific scenarios, development of ASR models can be performed at the phoneme level prior to the creation of word recognition systems. A phone recognition ASR model is designed to recognise individual speech sounds (phones) in the evaluated audio recordings taking into consideration the model's limited inventory of phone labels. Our isiZulu models comprised a total of 46 phone labels. The ASR models were configured to perform phone recognition using an n-gram (using the ARPA-MIT file format) language model consisting of equiprobable 1-grams. To assess recognition performance, corresponding phone error rate (PER) values were calculated. These PER values were derived from the number of string edit operations (substitution, deletion, and insertion) used to compare the reference phone label sequence with the recognised sequence for each audio segment in the evaluation set.

A few experiments were conducted to test various training options for conformer CTC ASR models. Table 4 presents the results for each of the training strategies. Apart from systems developed on the individual training data sets in Table 1 (NCHLT and News), an experiment was also performed to test model development through training data set pooling as an adaptation strategy. The last three experiments in Table 4 all represent ASR model fine-tuning, a method of adaptation where training on single data sets is completed on a one at a time basis, before commencing additional training of the same model on secondary data sets. In this way, the learning rate for each data set can be controlled for

effective data sharing. All the PERs were obtained from 1-best decodes, which extracts the best path from the decoding lattice. To create the NCHLT News model for example, we started with the NCHLT Epoch 90 model as the initial model and then continued training on the news training set data. Similarly, the NCHLT + News + Simulated model is obtained by training further on the AHRI simulated interview data, given the NCHLT+News Epoch 170 model.

4.3 Automatic Transcription Modelling

Automated transcription system development requires additional electronic text resources, especially when open-ended vocabularies need to be supported. Currently, it is still hard to obtain these resources in most South African languages. Achieving model representations that support all the different combinations of vocabulary items in various domains therefore remains a challenging aspect of ASR system development. Some languages, such as the agglutinative isiZulu language, also present unique challenges to text modelling. The words formed in agglutinative languages are made up of smaller constituents called morphemes that are combined to express compound ideas. In the isiZulu language this means that a larger than usual vocabulary needs to be supported if a transcription system's labels incorporate word level annotation.

To date we have performed a comprehensive analysis of diverse electronic texts available in the isiZulu language for ASR modelling. Primarily sourced from previous project outputs, the corpus predominantly stems from the government domain, reflecting the limited availability of isiZulu text on the internet. Initiated with TTS text corpora from the Lwazi projects, these corpora were selected for their phonetically balanced sentences, featuring minimal out-of-language tokens. This collection, which constitutes the smallest vocabulary set, prioritises sentences that represent frequently occurring words within the target language.

Additionally, a recently created set of manually transcribed news bulletins was incorporated as training data. Given that news represents narrated spoken isiZulu data, the processed transcriptions from this dataset were utilized as a supplementary text source to expand the base vocabulary with new lexical items. To further enrich the linguistic diversity and mitigate the occurrence of out-of vocabulary (OOV) lexical items, larger sets of texts from the NCHLT text project [4] and the Leipzig Corpora Collection [7] were integrated into the mix of text data.

4.4 Text Normalisation

Acknowledging that the texts in the larger sets were not as pristine as the smaller ones, a text normalisation procedure was implemented. This procedure targeted the two larger sets to eliminate empty lines, lines containing document headers, and bulleted lists. The locally developed Speect TTS software played a crucial role in this process, applying a rule-based number spell-out algorithm. This algorithm, for instance, facilitated the explicit representation of numbers, monetary amounts, and time in isiZulu [6]. Moreover, the Speect normalisation

engine [9] systematically removed undesired lines containing URLs and other graphemes, such as strings of dashes. Subsequent checks ensured compliance with the allowed graphemes in the grapheme-to-phoneme (G2P) rule set [2] for isiZulu. Any remaining incompatible graphemes were either replaced or removed semi-automatically. The G2P rule sets were then applied to extend the NCHLT pronunciation dictionaries for system development, generating pronunciations for the additional words in all text data sets.

4.5 Vocabulary Estimates

Table 2. Summary of vocabulary sizes of different isiZulu text data sets.

N	TTS	News	TTS News	NCHLT	TTS News NCHLT	TTS News NCHLT Leipzig
1	12 370	30 925	38 827	185 293	198 406	604 852
2	26 014	84 173	108 500	935 707	1 014 863	3 651 189
T	27 288	105 829	133 117	1 506 787	1 639 904	7 106 937

Table 2 summarises the vocabulary sets of the various texts used in our investigations for ASR modelling. The first two rows in the table shows the number of unique (N = 1) and bi-gram (N = 2) word tokens as well as the text data set sizes as the sum totals of all tokens in each data set. For example, the isiZulu TTS prompt sentences consisted of more than 27k words in total, containing 12 370 unique words. The manual annotations of the complete set of news data were approximately three times as large as the TTS text, followed by the NCHLT text corpus which was six times larger than the news text set. Pooling the set of normalised Leipzig texts created a text set that was three times larger than the size of all the other texts combined.

A fundamental consideration in developing an ASR model is the representativeness of the included text data for the intended test domain. Model-generated automatic transcriptions rely on the text-level representations integrated during development. To ensure low error rates, it is imperative that a subset of the development texts aligns with manual test transcriptions from the target domain. This alignment is partially quantified by the count of OOV words within the test data. Note that analysis of OOV words only does not describe the word sequence level information. The percentage of OOV items in each evaluation dataset was computed and presented in Table 3. Another crucial metric, perplexity, was employed to gauge text sufficiency by predicting test transcriptions using n-gram models. Specifically, trigram ARPA models were trained for each text dataset, aligning with the K2 recipe's capability to integrate ARPA language models into ASR models. Together, these metrics provide a comprehensive evaluation of the text modelling representation match, crucial for achieving accurate ASR transcriptions in the intended domain.

Table 3. Out-of-vocabulary (OOV) word rates and perplexity estimates of evaluation set data compared to various sets of text data.

Evaluation set	TTS	News	TTS News	NCHLT	TTS News NCHLT	TTS News NCHLT Leipzig
Percentage OOV						
News	52.18%	0%	0%	15.07%	0%	0%
Simulated	45.02%	27.92%	24.35%	13.92%	12.35%	6.66%
Real-world	43.13%	29.04%	24.87%	14.71%	14.11%	8.20%
Perplexity						
News	721 713	-	-	75 129	-	-
Simulated	300 409	87 986	66 967	52 021	38 097	26 242
Real-world	277 289	104 847	78 685	58 730	47 645	29 835

The perplexity estimates in Table 3, when considered alongside the OOV word rates in Table 3 provide insight into the representational sufficiency of the text data for AHRI ASR evaluation sets. Substantial OOV and perplexity values were prominent in the TTS texts, with nearly 50% of evaluation data words absent in this corpus. Transitioning to news transcriptions markedly improved coverage, and the amalgamation of TTS and news texts further enhanced this improvement. The OOV word percentage halved, accompanied by a more than three-fold decrease in measured perplexity, aligning these new estimates with those from the larger NCHLT text set. Although the NCHLT text's expanded vocabulary yielded lower OOV rates, the integration of TTS and news texts with NCHLT substantially reduced perplexity for AHRI data, contributing unseen sentences or word sequences despite a more modest boost in vocabulary coverage. Evaluation of the largest text set, encompassing Leipzig texts, notably decreased OOV rates, albeit with a lesser impact on perplexity estimates. This suggests that word sequences from Leipzig might lack representativeness of the AHRI target domain beyond what was captured by previously analysed texts.

4.6 Code-Switching

A consideration for advancing our ASR systems in future should involve scrutinizing and mitigating transcription errors arising from code-switched word tokens. The SOP transcript format employed by the AHRI incorporates markup identifying out-of-domain language tokens, including partial, foreign, and filler words. To assess the impact of foreign and filler words on transcription error, unique word token lists were derived from the evaluation data. By selectively excluding word token errors associated with identified foreign and filler word IDs, a refined WER estimate was generated, capturing errors attributable to remain-

ing tokens. Notably, examination of the AHRI's manual annotations revealed that foreign word annotations comprised English code-switched lexical items.

An evaluation was conducted to assess the impact of excluding English code-switched and filler lexical items using the best-matching acoustic model. Employing this model aimed to minimize transcription errors specifically for isiZulu target language words. The potential failure in correctly recognizing code switched lexical tokens could be attributed to pronunciation modelling considerations. Given the focus on developing a monolingual transcription system, all pronunciations for code-switched lexical items were generated using limited isiZulu target language phonetic labels. The analysis incorporated a rule-based G2P technique for auto pronunciation generation of out-of-language lexical items.

The effectiveness of these system design choices in accurately auto-generating pronunciations for code-switched words, aligned with isiZulu pronunciation, remained uncertain in this study. While improved recognition of these items could be anticipated with better-matched acoustics to new conversational data, potential transcription errors might persist due to inadequate pronunciation modelling.

5 Results

The initial results in Table 4 shows high PERs for the pure NCHLT model. Clearly, even the phoneme strings in the broadcast news test set segments could be recognised with lower error than the AHRI data. It was also possible to build a News only model, trained only on the broadcast news training set. Impressively, with this limited set of training data, PERs dropped to 51.85% for the simulated interview test data and below 60% for the real-world data. It may be challenging to combine ASR training data, especially if the data sets are from different domains. To further investigate, an experiment was conducted to test whether combining the 1 h of AHRI simulated interview data with the 8 h of news training data would have any effect on recognition performance. This data pooling approach, however, did not yet yield the best possible improvements in recognition even though results showed a large decline in error rate to a PER of 34.70%.

Table 4. Phone error rates (PERs) of various conformer CTC recognition systems on broadcast news, simulated and real-world interview data.

System	Epoch	News	Simulated	Real-world
NCHLT	90	65.00	82.06	76.15
News+Simulated (pool)	116	62.01	34.70	61.27
News	88	13.63	51.85	59.91
NCHLT+News	170	12.95	42.71	53.65
News+Simulated	117	19.51	24.96	50.37
NCHLT+News+Simulated	282	15.33	22.71	45.31
NCHLT+News+Real	348	24.36	35.17	42.69
NCHLT+News+Real+Simulated	461	21.52	20.50	38.32

5.1 Fine-Tuning

A key finding was found in fine-tuning the NCHLT model with the News training data over an additional 80 Epochs which resulted in a marked reduction in PERs. Notably, the PER for the AHRI simulated interview evaluation data reached a noteworthy decrease, now standing at 42.71%, representing the first instance where PERs dropped below the forties without the inclusion of this data during training. Concurrently, both the broadcast news and real-world evaluation datasets exhibited improved PERs.

Building upon this insight, two subsequent experiments were conducted to assess the efficacy of fine-tuning with the AHRI data. The results for the News + Simulated system demonstrated notably superior accuracy compared to the system trained on the pooled dataset, affirming the effectiveness of the fine-tuning modelling strategy for AHRI training data. Sequential fine-tuning of NCHLT models with both news and AHRI data provided further reduction in the PER of real-world interview audio segments to 45.31%. Similarly, the PER of simulated interview segments approximately halved, measuring at 22.71%, in contrast to the NCHLT + News system result. Lastly, adding real-world training data, culminated in the most optimal performance for the available datasets. These outcomes underscore the efficacy of sequential fine-tuning with diverse datasets, providing a robust foundation for improved performance in real-world and simulated interview scenarios within the AHRI context.

5.2 Word Recognition

Subsequently, the configured K2 models generated automatic word transcriptions using ARPA language models derived from each text selection outlined in the preceding section. Computing the Word Error Rate (WER), akin to the PER, Table 5 presents comprehensive WERs for each experiment, including results for the news test data, with additional rows incorporating findings when applying language modelling using combinations of the larger Leipzig text as well as the simulated evaluation data transcriptions. Notably, the integration of news texts into ASR models consistently resulted in markedly lower WERs in the news test data set experiments, showcasing the substantial advantages of including accurate word sequences, even in models influenced by extensive other text sources.

Despite the earlier reductions in OOV word rates and perplexity estimates, the measured decrease in WER for the AHRI evaluation data sets was initially less pronounced. Acoustic mismatch emerged as a significant factor, with the lower PERs achieved by the NCHLT + News and NCHLT + News + Simulated as well as NCHLT + News + Real + Simulated models now demonstrating substantial improvement. Moreover, also lower WERs were consistently measurable in results for these models upon the inclusion of news text data in language model estimation. However, it is noteworthy that the ASR system not incorporating AHRI audio data (NCHT + News) still produced WERs of 76.79% and 84.97%, indicating the complexity of achieving substantial improvements in WER for the AHRI evaluation data sets.

Table 5. WERs of conformer CTC recognition systems on simulated and real-world interview data using trigram ARPA language models derived from combinations of the following texts: 1) TTS, 2) News, 3) NCHLT, 4) Leipzig and 5) Simulated evaluation.

LM Text	News	Simulated	Real-world	News	Simulated	Real-world
	NCHLT			*NCHLT+News*		
1	91.02	96.07	96.11	82.91	82.48	87.69
2	72.26	96.33	96.39	17.73	82.05	88.04
1,2	72.43	96.15	95.98	17.82	80.63	87.37
3	85.75	95.45	95.17	50.69	77.39	85.48
1,2,3	74.95	95.42	95.30	17.94	77.11	85.46
1,2,3,4	77.35	95.42	95.02	19.87	76.79	84.97
1,2,3,5	74.98	92.93	95.34	17.91	59.47	85.40
1,2,3,4,5	77.35	93.67	95.02	19.87	62.35	84.86
	NCHLT+News+Simulated			*NCHLT+News+Real+Simulated*		
1	86.89	76.36	86.28	-	-	-
2	24.20	73.14	85.61	29.88	71.40	80.99
1,2	23.79	68.71	84.71	30.29	66.83	80.00
3	57.10	58.89	80.48	-	-	-
1,2,3	24.99	57.37	80.09	31.55	54.83	74.79
1,2,3,4	27.16	53.82	79.12	33.92	50.23	72.55
1,2,3,5	24.96	29.59	79.94	31.69	23.85	74.23
1,2,3,4,5	27.16	31.37	78.91	33.92	25.62	72.29

Fortunately, this picture started to change with the inclusion of only one hour of AHRI data. The trends observed for the NCHLT + News + Simulated experiment conformed more strongly to text representation improvements measured in OOV word counts. At least the WER for the simulated interview evaluation set could be reduced to 53.82% and 50.23% fine-tuning to 5 h of real-world training data. Upon exhaustively utilising the analysed data sets for ASR modelling, the question that remained then was what WER error rate would have been possible if the correct word sequences could be included in the language models. Subsequently, would this model refinement approach be successful given the acoustics of the AHRI data?

To answer these questions, a last set of experiment results were generated when including the simulated interview transcriptions with the other texts for ASR modelling. It was then possible to achieve a much lower WER of 23.85% transcribing the simulated interview data. Although this model seemed to remain stable when adding large amounts of additional text (consider the result of 25.62% obtained for inclusion of the Leipzig text in the model) the additional text in current form did not provide more benefit at this size of error. Also, the reduction to the real-world evaluation set's WER were minimal, alluding to the fact

that the word sequences of the simulated interview transcriptions might still be a poor match for the real-word set's transcriptions. It would presumably still take significantly more text data from the interview domain to build better generalised AHRI ASR models. It was also not clear if the simulated evaluation sets' WER could be further improved than the 25.62%. At the very least, the analysis confirmed the lower error rate as an achievable possibility for future AHRI ASR model development using this set of data.

5.3 Code Switched Word Errors

Table 6. Token error rates on a per simulated interview basis for NCHLT + News + Simulated Conformer CTC recognition systems for various token modelling options.

		Simulated LM	Standard LM		
Eval data	PER	WER	All	No Eng	No Fil
AssessRec_08Apr2022	17.49	21.23	46.37	35.66	33.30
AUTOT_SIDI1_07Jul2023	16.81	19.05	41.82	37.27	31.69
AUTOT_SIDI2_11Jul2023	17.07	19.76	47.69	44.16	39.89
AUTOT_SIDI3_14Jul2023	**43.55**	**63.68**	**77.09**	**49.06**	**48.12**
AUTOT_SIDI4_14Jul2023	22.42	37.49	57.25	50.24	47.52
AUTOT_SIDI5_19Jul2023	17.54	19.43	48.91	45.39	44.72
Simulated	22.71	31.37	53.82	43.49	40.60
Real-world	45.31	78.91	79.12	67.98	64.60

Evaluating the number of English code-switched words in error revealed insights into such transcription errors for the current dataset. For the simulated evaluation data, approximately 75% of English code-switched words (626 out of 836) contributed to recognition errors. Similarly, an even higher percentage of English code switched words (519 out of 583) were counted as transcription errors for the real-world evaluation data.

Table 6 presents a comparative analysis of WERs for simulated and real-world interview data given all tokens (All) against WERs generated when excluding code-switched (No Eng) and additionally omitting filler tokens (No Fil). Overall, the results indicated that approximately 10% of word errors stemmed from English code-switched tokens, with an additional 3% reduction in WER achieved by removing filler tokens. Further clarity on these WERs was obtained by reporting the same set of values per simulated interview basis.

Considering an analysis of the segments of the individual simulated interview recordings, the inclusion of corresponding PERs to assess the acoustic match revealed a significantly higher PER for one interview, highlighted in bold font. As anticipated, this discrepancy translated into higher WERs, especially even when incorporating the text from the manual transcriptions of the simulated interview into language modelling. A notable WER of 63.68% was reported. Intriguingly, the results demonstrated that excluding code-switched and filler category tokens

mitigated this elevated WER, aligning it with WERs observed for other interviews. This finding underscores the need for enhanced pronunciation modelling in future developments.

6 Discussion

In the recent concept note, it was postulated that current research in ASR technology we conduct could aid the AHRI use case. Particularly, more powerful modelling techniques to develop the necessary domain adapted models for conversational speech transcription have recently been created. The timing therefore seemed appropriate to test the potential of the new techniques to create transcription systems for the interview conversation recordings collected and transcribed by the AHRI research team. To support the experiments, a few data sets from different domains were successfully combined, facilitating appropriate data sharing. This strategy resulted in producing significantly lower word error rates for AHRI evaluation data compared to test error rates using previously existing ASR models.

Importantly, it was confirmed that the addition of broadcast news data was beneficial in model training and that adding in even more data from AHRI interview recordings themselves has a positive net effect. The latter also proved that the model fine-tuning technique applied to the new conformer CTC ASR models was effective where one can start training from as little as one hour of adaptation data from the AHRI interview domain. Since then, the inclusion of an additional 5-hour interview based ASR adaptation data set could be shown to provide additional benefit to acoustic model development although reducing error by a similar margin required more adaptation data. Future work could provide even further error reductions by localising recent weakly supervised ASR strategies [5] of the same models, improving model estimation with less accurate training transcriptions. Additional model improvements could therefore still be on the horison when this resource becomes available. Recently, in response to the AHRI need, an appropriate web-based solution to a human-in-the-loop interface that could facilitate the requested semi-automatic transcription service has been tested.

Another property of the AHRI interview recordings was that these isiZulu conversations contained a fair amount of code-switched lexical items. Closer inspection revealed that most of the code-switched words contained within the evaluation data sets were of an English origin and as such introduced noticeable transcription error for the monolingual ASR. This was the case, even if these words were included using pronunciations of the monolingual ASR base language. Around 10% of the word errors detected in the analysis of automatically transcribed AHRI data were stemming from the English words in the reference transcript. Apart from the code-switching, fillers and short audible confirmations by participants also introduced additional word errors of 3 to 4%. In future, it should therefore be possible to reduce the automated word transcription errors pertaining to the above-mentioned categories. This would potentially require

the development of suitable bi-lingual ASR models that can provide improved support to the required vocabulary refinements.

References

1. Badenhorst, J., Mak, F., Govender, A.: Exploring ASR fine-tuning on limited domain-specific data for low-resource languages. Digit. Humanit. Assoc. South. Afr. Digit. Humanit. Incl. **5**(1), 1–8 (2024). https://doi.org/10.55492/dhasa.v5i1.5024
2. Barnard, E., Davel, M.H., van Heerden, C., de Wet, F., Badenhorst, J.: The NCHLT speech corpus of the South African languages. In: Proceedings of the 4th Workshop on Spoken Language Technologies for Under-Resourced Languages, St Petersburg, Russia, pp. 194–200 (2014)
3. Davel, M.H., van Heerden, C.J., Barnard, E.: Validating smartphone-collected speech corpora. In: Third Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU), pp. 68–75 (2012)
4. Eiselen, R., Puttkammer, M.: Developing text resources for ten South African languages. In: Chair, N.C.C., et al. (eds.) Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014), Reykjavik, Iceland, pp. 3698–3703. European Language Resources Association (ELRA) (2014)
5. Gao, D., Xu, H., Raj, D., Perera, L.P.G., Povey, D., Khudanpur, S.: Learning from flawed data: weakly supervised automatic speech recognition. In: Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 1–8. IEEE (2023). https://doi.org/10.1109/ASRU57964.2023.10389684
6. Gillam, R.: A rule-based approach to number spellout. Unicode Consortium [Uni98a], page (1998)
7. Goldhahn, D., Eckart, T., Quasthoff, U.: Building large monolingual dictionaries at the leipzig corpora collection: from 100 to 200 languages. In: Chair, N.C.C., et al. (eds.) Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012), Istanbul, Turkey, pp. 31–43. European Language Resources Association (ELRA) (2012)
8. Gulati, A., et al.: Conformer: convolution-augmented transformer for speech recognition. arXiv preprint arXiv:2005.08100 (2020). https://doi.org/10.48550/arXiv.2005.08100
9. Louw, J.A., Moodley, A., Govender, A.: The speect text-to-speech entry for the blizzard challenge 2016. In: Blizzard Challenge Workshop 2016, Cupertino, United States of America (2016)
10. Povey, D.: K2 icefall conformer CTC example recipe (2024). https://github.com/k2-fsa/icefall/tree/master/egs/librispeech/ASR/conformer_ctc2. Accessed 14 Feb 2024
11. Witt, S.M.: Automatic error detection in pronunciation training: where we are and where we need to go. In: Proceedings of International Symposium on Automatic Detection on Errors in Pronunciation Training, pp. 1–8. KTH Computer Science and Communication (2012)
12. Witt, S., Young, S.: Phone-level pronunciation scoring and assessment for interactive language learning. Speech Commun. **30**(2–3), 95–108 (2000). https://doi.org/10.1016/S0167-6393(99)00044-8
13. Zhang, L., et al.: End-to-end automatic pronunciation error detection based on improved hybrid CTC/attention architecture. Speech Acoust. Audio Signal Process. Appl. Sens. **20**(7), 1809 (2020). https://doi.org/10.3390/s20071809

AI in Education: An Analysis of Large Language Models for Twi Automatic Short Answer Grading

Alex Agyemang and Tim Schlippe(✉)

IU International University of Applied Sciences, Bad Honnef, Germany
tim.schlippe@iu.org

Abstract. Automatic short answer grading can significantly enhance the speed and fairness of grading, making it particularly valuable in areas with a shortage of teachers, such as Africa [1]. However, for most African languages it is very challenging to build automatic short answer grading systems due to the limited availability of natural language processing corpora. Furthermore, only experts can deal with the complex algorithms, required for training and fine-tuning traditional automatic short answer grading systems. Given that state-of-the-art large language models have the potential to address these problems through their growing capabilities and ease of use through prompting, particularly in *zero-shot* and *few-shot* learning, we investigated their performance for grading student answers in the African language Twi. To address the absence of a Twi corpus, we translated and validated the University of North Texas benchmark corpus [2], creating the first Twi automatic short answer grading corpus. On this corpus, we evaluated the performances of the large language models GPT-4o [3], Claude 3 Sonnet [4], and LLaMA 3 [5] as well as for comparison two more traditional approaches: a fine-tuned AfroLM and a cross-lingual M-BERT approach. Among individual models, the cross-lingual M-BERT had the best performance with a mean absolute error of 0.79 points out of 5 points, followed by fine-tuned AfroLM at 0.73 points and Claude 3 Sonnet at 1.00 points. However, combining AfroLM and M-BERT outputs achieved the lowest mean absolute error of 0.64 points, which is less than the human grader variance of 0.75 points in the original corpus [6]. Combining the outputs of the large language models GPT-4o, Claude 3 Sonnet, and LLaMA 3, obtained through *few-shot* learning, yielded a mean absolute error of 1.10 points.

Keywords: AI in Education · Automatic Short Answer Grading · Natural Language Processing · Africa · Large-Language Models · LLMs

1 Introduction

United Nations have issued a global alert about a severe teacher shortage, highlighting the need for 44 million new teachers by 2030 to meet the Sustainable

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 107–123, 2025.
https://doi.org/10.1007/978-3-031-78255-8_7

Development Goals [1,7]. This shortage is most critical in sub-Saharan Africa, where 15 million teachers are needed [1]. The field of "AI in Education" explores the use and assessment of Artificial Intelligence (AI) techniques within educational and training environments [8–10]. A key focus of this research is to analyze and enhance teaching and learning processes. Advances in automatic short answer grading (ASAG) have progressed to a point where it can support educators by providing faster and fairer grading [6,11].

As the automation associated with ASAG can save money, ASAG is particularly interesting in countries with weaker economies. However, only a few ASAG systems have been developed for languages spoken in these countries. In particular, there is no ASAG corpus or systems for the African language Twi.

Recently, large language models (LLMs) have become widely accessible, and initial analyses have demonstrated their impressive *zero-shot* capabilities across various tasks in English and other languages. However, there has been no research on utilizing LLMs for ASAG in Twi. Therefore, in this study, we evaluate the *zero-shot* and *few-shot* performance of state-of-the-art LLMs GPT-4o [3], Claude 3 Sonnet [4] and LLaMA 3 [5] in automatically grading student answers in Twi. Our contributions are:

- We produce an ASAG corpus for the African language Twi, which we provide to the research community in our GitHub repository[1].
- We analyse state-of-the-art LLMs' ASAG performances for Twi.
- We compare the LLMs' *zero-shot* and *few-shot* performances in Twi ASAG.
- For comparison, we explore fine-tuning AfroLM—a pre-trained African natural language processing model (NLP)—and a cross-lingual M-BERT for ASAG using our new corpus.

In the following section, we will give a brief insight into the linguistic categorization and the peculiarities of Akuapem Twi. In Sect. 3, we will describe related work regarding ASAG, LLMs and NLP for Twi. The experimental setup which includes our analyzed pre-trained language models and LLMs as well as of our ASAG data collection will be presented in Sect. 4. In Sect. 5, we will demonstrate the results of our experiments. Finally, we will summarize our work and indicate possible future steps in Sect. 6.

2 The Language Twi

Twi is a collection of dialects within the Akan language [12]. The number of Twi speakers worldwide is estimated to be around 18 million [13]. The majority of Twi speakers are located in Ghana [14], where it is a widely spoken language among the Akan people. Additionally, there are smaller communities of Twi speakers in neighboring countries such as Cote d'Ivoire and Benin as well as in diaspora communities in countries like Suriname, Jamaica, and the United States [15].

[1] https://github.com/AgyemangOpambour/Twi-Automatic-Short-Answer-Grading/tree/main/Corpus/Twi.

The Akan dialects include Agona, Akuapem, Akwamu, Asante, Akyem, Assin, Bono, Fante, Kwahu, Wassa, Sefwi, Anyi, and Guan [16]. These dialects are classified into two categories: Fante and Twi, with Twi encompassing all non-Fante dialects [16]. Since mutual intelligibility is not universal among the dialects, Akuapem Twi serves as a pivot language for educational purposes in schools. The Akan Orthography Committee (AOC) established a unified Akan orthography in 1978, primarily based on Akuapem Twi [17]. Consequently, Twi dialects without their orthography use Akuapem Twi (marked in red) [18], while Asante and Bono have their writing systems. Fante, not a Twi dialect, has its orthography. Twi is a tonal language with high, mid, and low tones, each carrying distinct semantic meanings [19]. The Twi alphabet comprises 22 letters, including 15 consonants and 7 vowels [20]. Additionally, the letters C, J, V, and Z are used, mostly in loanwords. Twi features 10 diphthongs and many words with multiple meanings that can vary based on context [21]. For instance, "me papa" can mean "good mood" or "my dad," depending on the context.

3 Related Work

In this section, we will describe related work concerning ASAG, LLMs and NLP for Twi.

3.1 Automatic Short Answer Grading

The field of ASAG is increasingly significant as many educational institutions, both public and private, conduct online courses and examinations [6,11]. A comprehensive overview of pre-deep learning ASAG approaches is provided in [22]. Works like [6] and [23] demonstrated that BERT-based deep learning systems outperform others for English and German. [23]'s multilingual RoBERTa model [24] exhibits strong cross-lingual generalization. In [6], ASAG was extended to 26 languages using the smaller M-BERT model [25], achieving mean absolute errors between 0.41 points and 0.72 points out of 5 points, better than the 0.75 points variance between two graders, suggesting AI's viability in supporting answer scoring. Concerning state-of-the-art LLMs, GPT-3.5 and GPT-4 have been evaluated for automatic short answer grading. For instance, [26] examined Finnish student answers in *zero-shot* and *one-shot* settings, finding that GPT-4 significantly outperformed GPT-3.5 in the *one-shot* scenario. Additionally, [27] emphasized the effectiveness of fine-tuned GPT-3.5 for English ASAG. Additionally, [28] indicate that the quantized LLaMA-2 13B model, when fine-tuned with the Short Answer and Feedback dataset [29], exhibits exceptional performance in English ASAG. To the best of our knowledge, no publication describes an analysis of ASAG for African languages.

3.2 Large Language Models

The latest advancements in LLMs have showcased remarkable progress in NLP tasks, driven primarily by innovations in transformer architecture. Models like

GPT-4 [3], GPT-4o, Claude 3 [4], and LLaMA 3 [5] have set new benchmarks in terms of performance and versatility. These models leverage the self-attention mechanisms inherent to transformers, allowing them to process and understand context with unprecedented accuracy [30].

GPT-4, developed by OpenAI, exemplifies this progress with its ability to generate coherent, contextually appropriate text and perform tasks such as summarization, translation, and sentiment analysis with high accuracy [31]. The voice-to-voice capabilities of GPT-4o further extend the model's applicability to speech-related tasks, demonstrating significant advancements in multimodal processing. Claude 3, designed by Anthropic, focuses on alignment and safety, aiming to reduce harmful outputs while maintaining high performance in generative tasks. LLaMA 3, from Meta, offers strong performance in NLP benchmarks, particularly excelling in tasks requiring deep contextual understanding and nuanced language processing. Moreover, in question answering, GPT-4 and Claude 3 have demonstrated remarkable proficiency, achieving high scores on benchmarks such as SQuAD [32].

The success of these LLMs is attributed to their extensive pre-training on diverse and large-scale datasets, enabling them to generalize well across various domains and tasks. As the field continues to evolve, these models set the stage for even more sophisticated and capable NLP systems, promising further advancements in AI. Consequently, we selected GPT-4o, Claude 3 [4], and LLaMA 3 for our experiments.

To the best of our knowledge, there has been no scientific publication that described the use of LLMs for Twi NLP tasks. But other African languages have been analyzed with regards to the use of LLMs. For example, [33] analyzed the performance of four LLMs mT0, Aya, LLaMA 2, and GPT-4 on six tasks: topic classification, sentiment classification, machine translation, summarization, question answering, and named entity recognition across 60 African languages. The evaluation used the LLMs from scratch. The results indicated that all LLMs performed worse on African languages compared to high-resource languages. GPT-4 showed average performance in classification tasks but struggled with generative tasks. mT0 excelled in cross-lingual question answering, Aya was competitive and outperformed mT0 in topic classification, while LLaMA 2 performed the worst, likely due to its English-centric pre-training. The study concludes that there is a significant performance gap for African languages in current LLMs, highlighting the need for improvements, especially in generative tasks and overall support for African languages. [34] investigated the accuracy of LLMs, specifically GPT-4.0, GPT-3.5, and Bard (LaMDA), in answering basic day-to-day financial questions. The study focused on two languages: English and Yoruba, using a financial questions dataset in both languages. The results showed that GPT-4.0 outperformed GPT-3.5 and Bard (LaMDA) in all tested phases. The study suggests that while LLMs can be effective for financial queries, there is potential for improving these models to better handle low-resource languages like Yorùbá.

3.3 Natural Language Processing for Twi

[35] introduced the first transformer-based language models for Twi, specifically focusing on the Akuapem and Asante dialects. Their models, ABENA and BAKO, were fine-tuned on Akan corpora and demonstrated applicability in named entity recognition, neural machine translation, sentiment analysis, and part-of-speech tagging.

Machine translation for Twi has seen notable progress with several key studies. [36] created a transformer-based model using an English-Twi parallel corpus, highlighting the importance of native speakers in refining translations. [37] introduced the Twi-2-ENG parallel corpus from various sources, emphasizing the need for linguistic professionals to enhance corpus quality. [38] developed a Twi-French parallel corpus, showing improved translation performance using English as a pivot. Additionally, [39] addressed gender bias in machine translation, finding that participatory data design can improve translation equality between gendered terms.

Named entity recognition for Twi has been explored through the use of word embeddings and multilingual models. [40] compared word embeddings for Yorùbá and Twi using the Global Voices corpus, demonstrating the effectiveness of multilingual BERT for named entity recognition tasks and highlighting the significance of data quality.

Sentiment analysis for Twi has been advanced using ensemble learning approaches. [41] participated in SemEval 2023's AfriSenti task, focusing on sentiment analysis for multiple African languages, including Twi. They combined multilingual LLMs with language-independent features, achieving notable F1-scores for Twi.

Significant strides have been made in developing automatic speech recognition and synthesis systems for Twi. [42] developed an ASR-based mobile application, TwiGrad, to facilitate learning and maintaining the Twi language. [43] created an end-to-end text-to-speech system for Twi, demonstrating the feasibility of deploying such systems on resource-constrained devices.

Various corpora have been developed to support Twi NLP tasks. The Twi dictionary corpus by [44] is a notable example, consisting of 1,367 words derived from the TypeCraft corpus of Interlinear Glossed Texts. [45] developed a parallel Bible corpus for Twi and other languages, which included aligned sentences and 2D vector representations of Twi vocabulary. Additionally, the Twi-English Parallel Corpus by [19] consists of 25,421 sentence pairs, verified and corrected by native speakers, enhancing the quality of the dataset for machine translation. Furthermore, [38] developed another Twi-French parallel corpus with 10,708 parallel sentences, further contributing to the resources available for Twi language processing. However, to the best of our knowledge, no corpus for Twi ASAG was collected in related work.

4 Experimental Setup

In this section, we will describe our analyzed pre-trained language models and LLMs as well as of our ASAG data collection.

4.1 Overview of the Automatic Short Answer Grading Approaches

An overview of the investigated ASAG approaches for Twi is given in Fig. 1. For our three state-of-the-art LLMs GPT-4o, Claude 3 Sonnet and LLaMA 3, we compare their *zero-shot* and *few-shot* performances. In the *zero-shot* approaches, we instructed the LLMs to assign points between 0 and 5 without giving them examples of exam question, model answer, student answer and the corresponding score assigned by the graders. In contrast, in the *few-shot* approaches, we instructed the LLMs to assign points between 0 and 5 giving them 100 examples of exam question, model answer, student answer and the corresponding score from our Twi ASAG training data.

For comparison, we explore fine-tuning AfroLM—a pre-trained African NLP model—and a cross-lingual M-BERT for ASAG using our Twi ASAG training data. To solve the problems of low-resource languages in building NLP systems, some researchers propose cross-lingual NLP approaches. Thus is it possible to benefit from rich-resource languages like English [46–49]. They usually translate the text from the original low-resource language to English. This allows to do the machine learning task with well-performing models trained with a lot of English resources. Therefore, for the cross-lingual experiment we first machine-translated the Twi text to English and then used the English M-BERT for the grading, which was originally pre-trained with the BooksCorpus (800M words) [50] and English Wikipedia (2,500M words) [51] and then fine-tuned by us using the English ASAG training data.

4.2 Our Analyzed Large Language Models

To compare *zero-shot* and *few-shot* ASAG performances of the state-of-the-art LLMs, we analyzed GPT-4o, Claude 3 Sonnet and LLaMA 3.

GPT-4o. GPT-4o (GPT-4 Omni)[2], developed by OpenAI, was released in May 2024. Unlike GPT-3.5 and GPT-4, which rely on other models to process sound, GPT-4o natively supports voice-to-voice since the model was trained end-to-end across text, vision, and audio, so that all inputs and outputs are processed by the same neural network. While specific training data details for GPT-4o are not publicly disclosed, it can be assumed that it also follows the trend of massive data utilization seen in GPT-3 and GPT-4. For GPT-3, it is known that 570 GB of text data was used [52]. The multilingual and multimodal GPT-4o can process up to 128k tokens per input[3] and supports more than 50 languages[4]. Although there is no specific publication where GPT-4o has been used for ASAG, GPT-3.5 and GPT-4 have been tested to automatically grade student answers. For example, [26] tested Finnish student answers under *zero-shot* and *one-shot* settings. They report a good performance for GPT-4 in the *one-shot* setting, clearly outperforming GPT-3.5. Furthermore, [27] highlight the potential of fine-tuned GPT-3.5 for English ASAG.

[2] https://openai.com/index/hello-gpt-4o.
[3] https://platform.openai.com/docs/models/gpt-4o.
[4] https://openai.com/index/gpt-4o-and-more-tools-to-chatgpt-free.

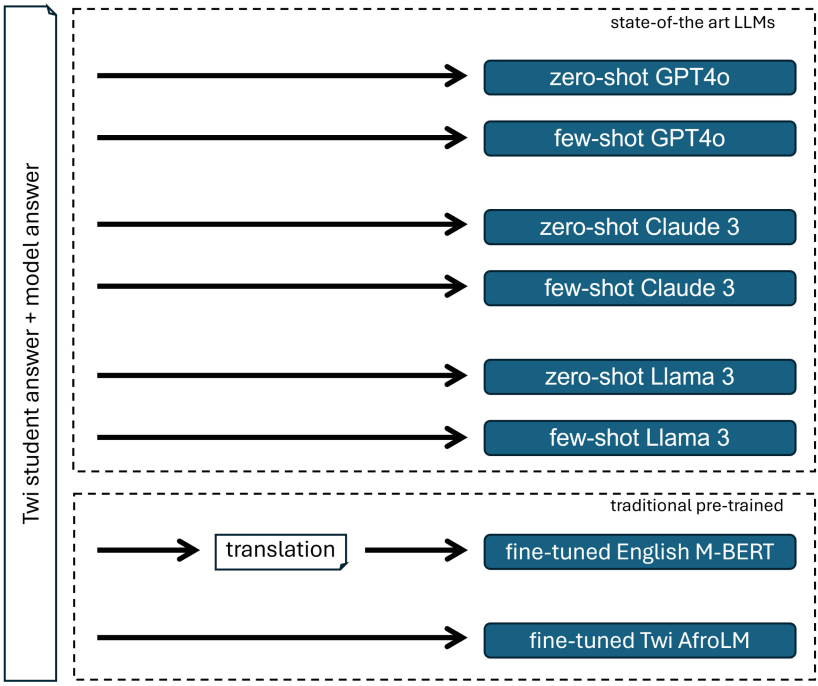


Fig. 1. Overview of the investigated models

Claude 3 Sonnet. Claude 3 Sonnet, developed by Anthropic, was released in late 2023. While specific training data details for the Claude 3 are not publicly disclosed, [4] report that "Claude 3 models are trained on a proprietary mix of publicly available information on the Internet as of August 2023, as well as non-public data from third parties, data provided by data labeling services and paid contractors, and data" that they "generate internally". Claude 3 Sonnet can process up to 200k tokens per input [4]. There are no specific publications detailing the use of Claude 3 Sonnet for ASAG.

LLaMA 3. LLaMA 3 is a large language model provided by Meta AI, launched in early 2024[5]. The model is pre-trained on over 15T tokens that were all collected from publicly available sources[6]. It can process up to 8k tokens per input. There are two LLaMA 3 versions: One model with 8B and one with 70B parameters. We used *LLaMA-3-70B-Instruct*. While there are no specific publications on the use of LLaMA 3 for ASAG, [28] report that the predecessor—quantized LLaMA-2 13B model, fine-tuned with data from the Short Answer and Feedback dataset [29]—shows exceptional English ASAG performance.

[5] https://github.com/meta-LLaMA/LLaMA3.
[6] https://ai.meta.com/blog/meta-LLaMA-3.

4.3 Our Analyzed Pre-trained Language Models

To compare the state-of-the-art LLMs with more traditional approaches of fine-tuning pre-trained language models, we analyzed Afro-LM and a cross-lingual approach using M-BERT.

Afro-LM. AfroLM, developed by [53] in 2022, is a multilingual language model specifically tailored for African languages including Twi. The model was pre-trained on a 0.73 GB dataset of 23 African languages from the news domain, which covers many topics such as health, politics, society, sport, environment [53]. Even the model was pre-trained on only 1.6 MB of Twi text, it outperforms other models like AfriBERTa-Large [53]. AfroLM is able to process 256 tokens in a single input sequence[7]. There are no specific publications detailing its use in ASAG yet. To build an ASAG system for Twi, we fine-tuned AfroLM using the validation set of our Twi ASAG corpus.

M-BERT. Multilingual BERT (Multilingual Bidirectional Encoder Representations from Transformers, M-BERT)[8] is a large language model developed by Google AI and released in 2019 [25,51]. The model was trained on Wikipedia (2,500M words) and BookCorpus [50] (800M words) across 104 languages. M-BERT can process up to 512 tokens per input, offering versatile performance across 104 languages [51]. Although primarily designed for cross-lingual understanding, M-BERT and its related BERT models have been adapted for various tasks, including ASAG, e.g. [54–56]. While most research focused only on one language, [6] investigated ASAG for 26 languages and report that their fine-tuned models are able to fairly score free-text answers in those languages. Since M-BERT was not pre-trained on Twi text, we used the model to build an English ASAG system that is an essential component of our cross-lingual ASAG system.

4.4 Data

To address the absence of a Twi corpus, we translated and validated the University of North Texas benchmark corpus [2], creating the first Twi ASAG corpus. This corpus was used for fine-tuning Afro-LM and for *few-shot* learning of the analyzed LLMs GPT-4o, Claude 3 Sonnet and LLaMA 3 as well as for testing the Twi ASAG performance. The original English corpus was used for training and fine-tuning the English BERT system which is a part of our cross-lingual Twi ASAG system.

Corpus of the University of North Texas. We used the English short answer grading dataset of the University of North Texas [2] to create a Twi ASAG corpus. The advantage of this dataset is that it contains scored student answers, while the answers of other short answer grading corpora, e.g., the SemEval-2013Task7 data

[7] https://github.com/bonaventuredossou/MLM_AL.
[8] https://huggingface.co/google-bert/bert-base-multilingual-cased.

sets [57] are only categorized into 3 classes—there is no point-based grading. [58], [11] and [6] investigated and compared state-of-the-art deep learning techniques for ASAG using the data set. The dataset contains 87 questions with corresponding model answer and on average 28.1 manually graded answers per question about the topic *Data Structures* from under-graduate studies. Each student answer received a score from 0–5 points from two independent graders. We used the average of these two scores as our prediction target and randomly selected 70% of the English ASAG data set (1.818 student answers) for training and 10% (1.82 student answers) for fine-tuning the English BERT system which is a part of our cross-lingual Twi ASAG system. The Twi translation of the remaining 20% (455 student) was used for evaluation.

Creating a Twi Automatic Short Answer Grading Corpus Based on the Corpus of the University of North Texas. In order to create a Twi ASAG corpus based on the Corpus of the University of North Texas [2] for the adaptation, evaluation and comparison of Twi ASAG, we translated all English exam questions, model answers and student answers into Twi using (1) Google's Cloud Translation API[9], which is based on [59], and (2) had the translations corrected by two native speakers who are linguistic experts from the Language Department at Aperade Senior High Technical School in Ghana.

Prompt:

> Please grade the student answer, given the question and the model answer below. You can assign points between 0 and 5, where a completely correct answer received 5 points and a completely false answer receives 0 points. Decimal places are also possible. Please provide the question, model answer, student answer, and the number of points in a table format.
>
> <test set in csv format>

Fig. 2. Prompt for *zero-shot* approaches

4.5 Prompts

Figure 2 and Fig. 3 demonstrate the prompts which we elaborated to have the LLMs do the *zero-shot* and *few-shot* grading.

All prompts were executed in the user interface, not via API, as this was cheaper and the LLMs were able to include the context with the examples from the *prompt for teaching how to grade* for grading in the *few-shot* learning as shown in Fig. 3. For prompt engineering, it was important for us to pass a precise description to the LLMs. We figured out that all LLMs can handle tables in

[9] https://cloud.google.com/translate/docs/reference/rest.

Prompt for teaching how to grade:

> Here are examples of exam questions (question), model answers (desired_answer), student answers (student_answer), and corresponding grades (score_avg) in each line in csv format:
>
> <examples from training set in csv format>
>
> From these examples, please learn how to grade, i.e. assign scores for the student answers. The reason is that in the next prompt you will be provided with exam questions (question), model answers (desired_answer), student answers (student_answer) in csv format, and you need to include the corresponding grades (score_avg) in each line.

Prompt for instructing to grade:

> Thanks for learning how to grade based on the examples provided. Now you will be provided with new exam questions (question), model answers (desired_answer), student answers (student_answer) in csv format in Twi, and you need to include the corresponding grades (score_avg) in each line. Grade all the questions provided within the range of 0 to 5. you can assign decimal values. Display your output in table format with the headings "question", "desired_answer", "student_answer", and "score_avg".
>
> <test set in csv format>

Fig. 3. Prompt for *few-shot* approaches

csv format well. Therefore, we handed over the 100 examples for the *few-shot* learning and the student answers to be graded together with the model answers and the exam questions to the LLMs in csv format. We transferred the csv text with the prompt.

5 Experiments and Results

In this section, we will present the Twi ASAG performances of the more traditional approaches of fine-tuning the pre-trained language models AfroLM and M-BERT. Then, we will explore the *zero-shot* and *few-shot* performances of the state-of-the-art LLMs GPT-4o, Claude 3 Sonnet, and LLaMA 3.

5.1 Pre-trained Language Models

To fine-tune the pre-trained Afro-LM for Twi ASAG, we trained 6 epochs—i.e. the model learned 6 times from the entire training set—with a batch size of 8—i.e. in one training iteration 8 samples were used to update the model

weights—using the AdamW optimizer with an initial learning rate of 0.00002. To fine-tune M-BERT—which was used to grade the Twi student answers after they were machine-translated to English—we trained 6 epochs with a batch size of 16 using the AdamW optimizer with an initial learning rate of 0.00002. We supplemented each model with a linear regression layer that outputs a prediction of the score given an answer. The model expects the model answer and the student answer as input.

Table 1 shows the ASAG performances of the single models, represented by the mean absolute error (MAE). AfroLM, fine-tuned with the Twi ASAG data (*AfroLM*) exhibits the best performance in grading our Twi test set with an MER of 0.73 points out of 5 points. The cross-lingual approach using M-BERT trained with English ASAG data for the machine-translated test set (*TW-EN translation + GPT-4o*$_{zero-shot}$) shows a slightly higher MER of 0.79 points. For comparison, [6] report that their best M-BERT model achieves an MER of 0.45 points for English ASAG, while other systems reach up to 0.86 depending on the language.

Table 1. ASAG performances of single models.

Model	Lang$_{Prompt}$	Lang$_{toGrade}$	MER
AfroLM	—	TW	**0.73**
TW-EN translation + M-BERT	—	EN	0.79
GPT-4o$_{zero-shot}$	TW	TW	2.36
TW-EN translation + GPT-4o$_{zero-shot}$	EN	EN	2.51
GPT-4o$_{zero-shot}$	EN	TW	2.15
GPT-4o$_{few-shot}$	EN	TW	1.53
Claude 3.5 Sonnet$_{zero-shot}$	EN	TW	1.01
Claude 3.5 Sonnet$_{few-shot}$	EN	TW	1.00
LLaMa 3$_{zero-shot}$	EN	TW	1.48
LLaMa 3$_{few-shot}$	EN	TW	1.11

5.2 Large Language Models

Next, we wanted to find out whether the LLM performance is higher when the LLMs are prompted in Twi ($Lang_{Prompt}$ = TW) or in English ($Lang_{Prompt}$ = EN) and whether the performance is higher when the text to be graded is in Twi ($Lang_{toGrade}$ = TW) or in English ($Lang_{toGrade}$ = EN). We analyzed this using GPT-4o with *zero-shot* learning (GPT-4o$_{zero-shot}$). Table 1 indicates that using an English prompt ($Lang_{Prompt}$ = EN) and the original Twi text of the test set to be graded ($Lang_{toGrade}$ = TW) (MER = 2.15 points) performs better than using a Twi prompt ($Lang_{Prompt}$ = TW) (MER = 2.36 points) or grading the test set after is was machine-translated into English ($Lang_{toGrade}$ = EN) (MER = 2.51 points).

5.3 Impact of Providing Examples to Large Language Models

Comparing the *zero-shot* performances—where we passed no ASAG examples to the LLM in the prompt—with the *few-shot* performances—where we passed 100 ASAG examples in the prompt—we see the following: Of all three LLMs, the *zero-shot* and *few-shot* performances of Claude 3 Sonnet are the best, although there is no real difference (1% relative) between *zero-shot* (MER = 1.01 points) and *few-shot* (MER = 1.00 points). In contrast, GPT-4o and LLaMa 3 benefit from the 100 ASAG examples: GPT-$4o_{few-shot}$ (MER = 1.53 points) is 40.52% relatively higher than GPT-$4o_{zero-shot}$ (MER = 2.15 points). LLaMa $3_{few-shot}$ (MER = 1.11 points) outperforms LLaMa $3_{zero-shot}$ (MER = 1.48 points) by 33.33% relative.

5.4 Combination of the Outputs

To further improve ASAG performance, we analyzed a combination of the LLM outputs using the average number of points assigned for each student answer. Our idea was to mitigate the incorrect grading of individual LLMs using this procedure. Table 2 shows that combining AfroLM and M-BERT outputs achieves the lowest MER of 0.64 points, which is 17.19% relatively less than the original corpus's human grader variance of 0.75 points [6]. Combining the outputs of the LLMs GPT-4o, Claude 3 Sonnet, and LLaMA 3, obtained through *zero-shot* learning, yielded an MER of 1.26 points. Combining the outputs of the LLMs GPT-4o, Claude 3 Sonnet, and LLaMA 3, obtained through *few-shot* learning, yielded an MER of 1.10 points, which means a relative improvement of 14.54% compared to *zero-shot*. Combining the outputs of AfroLM and M-BERT and GPT-$4o_{few-shot}$, Claude 3 Sonnet$_{few-shot}$, and LLaMA $3_{few-shot}$ could not mitigate the worse performances of the LLMs and resulted in an MER of 0.99 points as well as combining the outputs of all tested models which resulted in an MER of 1.15 points.

Table 2. ASAG performances of single models.

Model Combination	MER
Pre-trained LMs (AfroLM + M-BERT)	**0.64**
$LLMs_{zero-shot}$ (GPT-4o + Claude 3.5 Sonnet + LLaMa 3)	1.26
$LLMs_{few-shot}$ (GPT-4o + Claude 3.5 Sonnet + LLaMa 3)	1.10
Pre-trained LMs + $LLMs_{few-shot}$	0.99
Pre-trained LMs + $LLMs_{few-shot}$ + $LLMs_{zero-shot}$	1.15

6 Conclusion and Future Work

We have evaluated the performance of the state-of-the-art LLMs GPT-4o, Claude 3 Sonnet, and LLaMA 3 for ASAG in the African language Twi. Due to the

lack of a Twi corpus, we translated and validated the University of North Texas benchmark corpus to create the first Twi ASAG dataset. Our findings reveal that the traditional pre-trained models AfroLM and M-BERT, particularly when their outputs are combined, outperform individual LLMs in terms of MAE. Specifically, combining AfroLM and M-BERT achieved an MAE of 0.64 points, which is lower than the human grader variance. However, training and fine-tuning such systems can only be done by experts due to these tasks' complexity. In contrast, LLMs which are easier to instruct through prompting demonstrate potential even if their performance varies significantly depending on the language of the prompts and the text to be graded, with English prompts generally yielding better results. Additionally, providing LLMs with a few examples improves their performance, though the extent varies across LLMs.

For future work, it would be interesting to investigate whether a weighted combination of LLM outputs leads to improvements. Since LLMs are already good at other tasks without fine-tuning or *few-shot* learning compared to traditional pre-trained models, it could be investigated how well the LLMs explain the points assigned for the student answers. Preliminary work in the area of Explainable AI in ASAG already exists, e.g. [60], but not yet with LLMs and not in an African language such as Twi. Furthermore, it would be interesting to collect a real Twi ASAG corpus and analyze the Twi ASAG performance in exams in other subjects.

References

1. United Nations News: UN Issues Global Alert over Teacher Shortage. UN News (2024). https://news.un.org/en/story/2024/02/1147067
2. Mohler, M., Bunescu, R., Mihalcea, R.: Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In: The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, Oregon, USA, pp. 752–762. ACL (2011). https://aclanthology.org/P11-1076
3. OpenAI, Achiam, J., Adler, S., Agarwal, S., et al.: GPT-4 Technical Report (2024). https://arxiv.org/abs/2303.08774
4. Anthropic: The Claude 3 Model Family: Opus, Sonnet, Haiku (2024). https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf
5. Touvron, H., et al.: LLaMA: Open and Efficient Foundation Language Models (2023). https://arxiv.org/abs/2302.13971
6. Schlippe, T., Sawatzki, J.: Cross-lingual automatic short answer grading. In: The 2nd International Conference on Artificial Intelligence in Education Technology (AIET), Wuhan, China (2021)
7. United Nations: Sustainable Development Goals: 17 Goals to Transform our World (2024). https://www.un.org/sustainabledevelopment/sustainabledevelopment-goals. Accessed Aug 2024
8. Chen, L., Chen, P., Lin, Z.: Artificial intelligence in education: a review. IEEE Access **8**, 75264–75278 (2020). https://doi.org/10.1109/ACCESS.2020.2988510

9. Schlippe, T., Cheng, E.C.K., Wang, T.: Artificial Intelligence in Education Technologies: New Development and Innovative Practices. Springer, Cham (2023). https://doi.org/10.1007/978-981-99-7947-9
10. Cheng, E.C.K., Wang, T., Schlippe, T., Beligiannis, G.N.: Artificial Intelligence in Education Technologies: New Development and Innovative Practices. Springer, Cham (2023). https://doi.org/10.1007/978-981-19-8040-4
11. Sawatzki, J., Schlippe, T., Benner-Wickner, M.: Deep learning techniques for automatic short answer grading: predicting scores for English and German answers. In: Cheng, E.C.K., Koul, R.B., Wang, T., Yu, X. (eds.) Artificial Intelligence in Education: Emerging Technologies, Models and Applications. LNDECT, vol. 104, pp. 65–75. Springer, Singapore (2022). https://doi.org/10.1007/978-981-16-7527-0_5
12. Akan (Twi) at Rutgers (2022). https://www.amesall.rutgers.edu/languages/128-akan-twi. Accessed Jan 2023
13. Yakubu, M.: Check Out Other Countries That Speak Twi Apart From Ghana (2024). https://www.primenewsghana.com/entertainment/check-out-other-countries-that-speak-twi-apart-from-ghana.html. Accessed 29 July 2024
14. Akan Twi (2023). https://celt.indiana.edu/portal/Akan%20Twi/index.html. Accessed Jan 2023
15. Lingual, P.: Check Out 4 Countries Where They Speak Twi Aside Ghana That You Never Knew (2024). https://paullingual.com/check-out-4-countries-where-they-speak-twi-aside-ghana-that-you-never-knew-check-out. Accessed 29 July 2024
16. Osam, E.K.: An introduction to the verbal and multi-verbal system of akan. In: Workshop on Multi-verb Constructions, Trondheim, Norway (2003)
17. Kouadio, N.J.: A Unified Orthography for the Akan Languages of Ghana and Ivory Coast: General Unified Spelling Rules, Monograph Series/Centre for Advanced Studies of African Society, vol. 20. Centre for Advanced Studies of African Society, CASAS, Cape Town (2003)
18. Schachter, P., Fromkin, V.: A Phonology of Akan: Akuapem, Asante, Fante. Working papers in phonetics, University of California (1979)
19. Azunre, P., et al.: English-Twi Parallel Corpus for Machine Translation. arXiv abs/2103.15625 (2021)
20. The African Linguists Network Blog: Language Guide. https://alnresources.wordpress.com/african-culture-and-language. Accessed 15 May 2023
21. Alabi, J.O., Amponsah-Kaakyire, K., Adelani, D.I., España-Bonet, C.: Massive vs. curated embeddings for low-resourced languages: the case of Yorùbá and Twi. In: The 12th Conference on Language Resources and Evaluation (LREC 2020) (2020)
22. Burrows, S., Gurevych, I., Stein, B.: The eras and trends of automatic short answer grading. Int. J. Artif. Intell. Educ. **25**(1), 60–117 (2015). https://doi.org/10.1007/s40593-014-0026-8
23. Camus, L., Filighera, A.: Investigating transformers for automatic short answer grading. In: Bittencourt, I.I., Cukurova, M., Muldner, K., Luckin, R., Millán, E. (eds.) AIED 2020. LNCS (LNAI), vol. 12164, pp. 43–48. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-52240-7_8
24. Liu, Y., et al.: RoBERTa: A Robustly Optimized BERT Pretraining Approach. CoRR abs/1907.11692 (2019). http://arxiv.org/abs/1907.11692
25. Pires, T., Schlinger, E., Garrette, D.: How multilingual is multilingual BERT? In: The 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, pp. 4996–5001. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/P19-1493. https://aclanthology.org/P19-1493

26. Chang, L.H., Ginter, F.: Automatic short answer grading for finnish with ChatGPT. In: The AAAI Conference on Artificial Intelligence, vol. 38, no. 21, pp. 23173–23181 (2024). https://doi.org/10.1609/aaai.v38i21.30363. https://ojs.aaai.org/index.php/AAAI/article/view/30363
27. Latif, E., Zhai, X.: Fine-tuning ChatGPT for automatic scoring. Comput. Educ. Artif. Intell. **6**, 100210 (2024). https://doi.org/10.1016/j.caeai.2024.100210. https://www.sciencedirect.com/science/article/pii/S2666920X24000110
28. Katuka, G.A., Gain, A., Yu, Y.Y.: Investigating Automatic Scoring and Feedback using Large Language Models (2024). https://arxiv.org/abs/2405.00602
29. Filighera, A., Parihar, S., Steuer, T., Meuser, T., Ochs, S.: Your answer is incorrect... would you like to know why? Introducing a bilingual short answer feedback dataset. In: The 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, pp. 8577–8591. Association for Computational Linguistics (2022). https://doi.org/10.18653/v1/2022.acl-long.587. https://aclanthology.org/2022.acl-long.587
30. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017)
31. Brown, T., et al.: Language models are few-shot learners. In: Advances in Neural Information Processing Systems, vol. 33, pp. 1877–1901. Curran Associates, Inc. (2020)
32. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(140), 1–67 (2020). http://jmlr.org/papers/v21/20-074.html
33. Ojo, J., Ogueji, K., Stenetorp, P., Adelani, D.I.: How good are Large Language Models on African Languages? (2024). https://arxiv.org/abs/2311.07978
34. Sikiru, R.D., Adekanmbi, O., Soronnadi, A.: Comparative study of LLMs for personal financial decision in low resource language. In: 5th Workshop on African Natural Language Processing (2024). https://openreview.net/forum?id=9gDt0ZUk8H
35. Azunre, P., et al.: Contextual text embeddings for Twi. In: 2nd AfricaNLP Workshop Proceedings, AfricaNLP@EACL 2021, Virtual Event, 19 April 2021 (2021). https://arxiv.org/abs/2103.15963
36. Bannerman, S., Agyei, E., Sarpong, S., Quaye, A.B., Yussif, S.B., Agbesi, V.K.: Machine translation from English-Twi in parallel corpus: low resource Ghanaian. Language (2023). https://doi.org/10.2139/ssrn.4761197
37. Agyei, E., Zhang, X., Bannerman, S., et al.: Low resource Twi-English parallel corpus for machine translation in multiple domains (Twi-2-ENG). Discov. Comput. **27**, 17 (2024). https://doi.org/10.1007/s10791-024-09451-8
38. Gyasi, F., Schlippe, T.: Twi machine translation. Big Data Cogn. Comput. **7**(2) (2023). https://doi.org/10.3390/bdcc7020114. https://www.mdpi.com/2504-2289/7/2/114
39. Oppong, A.: Building a participatory data design approach to examine gender bias in english-twi machine translation. In: Extended Abstracts of the 2023 CHI Conference on Human Factors in Computing Systems. CHI EA 2023. Association for Computing Machinery, New York (2023). https://doi.org/10.1145/3544549.3583942
40. Alabi, J.O., Amponsah-Kaakyire, K., Adelani, D.I., España-Bonet, C.: Massive vs. curated embeddings for low-resourced languages: the case of Yorùbá and Twi. In: International Conference on Language Resources and Evaluation (2019)
41. García-Díaz, J.A., Caparros-laiz, C., Almela, Á., Alcaráz-Mármol, G., Marín-Pérez, M.J., Valencia-García, R.: UMUTeam at SemEval-2023 task 12: ensemble learning of LLMs applied to sentiment analysis for low-resource African lan-

guages. In: The 17th International Workshop on Semantic Evaluation (SemEval-2023), Toronto, Canada, pp. 285–292. Association for Computational Linguistics (2023). https://doi.org/10.18653/v1/2023.semeval-1.38. https://aclanthology.org/2023.semeval-1.38
42. Quartson, P.: TWIGRAD: An ASR-Based Application for Learning Twi, applied Project, Department of Computer Science, Ashesi University College. B.Sc, Computer Science (2021)
43. Aboagye, F., Akolly, E.: Text-to-Speech for Ghanaian Language (Akuapem Twi) on an Embedded System, capstone Project, Department of Engineering, Ashesi University College. B.Sc, Electrical/Computer Engineering (2021)
44. Beermann, D., Hellan, L., Mihaylov, P., Struck, A.: Developing a Twi (Asante) dictionary from Akan interlinear glossed texts. In: The 1st Joint Workshop on Spoken Language Technologies for Under-resourced languages (SLTU) and Collaboration and Computing for Under-Resourced Languages (CCURL), Marseille, France, pp. 294–297. ELRA (2020). https://aclanthology.org/2020.sltu-1.41
45. Adjeisah, M., Liua, G., Nortey, R.N., Song, J.: English Twi parallel-aligned bible corpus for encoder-decoder based machine translation. Acad. J. Sci. Res. **8**(12), 371–382 (2020)
46. Balahur, A., Turchi, M.: Comparative experiments using supervised learning and machine translation for multilingual sentiment analysis. Comput. Speech Lang. **28**, 56–75 (2014)
47. Lin, Z., Jin, X., Xu, X., Wang, Y., Tan, S., Cheng, X.: Make it possible: multilingual sentiment analysis without much prior knowledge. In: IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT), vol. 2, pp. 79–86 (2014). https://doi.org/10.1109/WI-IAT.2014.83
48. Vilares, D., Alonso Pardo, M., Gómez-Rodríguez, C.: Supervised sentiment analysis in multilingual environments. Inf. Process. Manag. **53** (2017). https://doi.org/10.1016/j.ipm.2017.01.004
49. Can, E.F., Ezen-Can, A., Can, F.: Multilingual sentiment analysis: an RNN-based framework for limited data. In: ACM SIGIR 2018 Workshop on Learning from Limited or Noisy Data (2018)
50. Zhu, Y., et al.: Aligning books and movies: towards story-like visual explanations by watching movies and reading books. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp. 19–27 (2015). https://doi.org/10.1109/ICCV.2015.11
51. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: The 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, Minnesota, pp. 4171–4186. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/N19-1423. https://aclanthology.org/N19-1423
52. Brown, T.B., et al.: Language models are few-shot learners. In: The 34th International Conference on Neural Information Processing Systems. NIPS 2020. Curran Associates Inc., Red Hook (2020)
53. Dossou, B.F.P., et al.: AfroLM: a self-active learning-based multilingual pretrained language model for 23 African languages. In: The Third Workshop on Simple and Efficient Natural Language Processing (SustaiNLP), Abu Dhabi, United Arab Emirates (Hybrid), pp. 52–64. Association for Computational Linguistics (2022). https://doi.org/10.18653/v1/2022.sustainlp-1.11. https://aclanthology.org/2022.sustainlp-1.11

54. Sung, C., Dhamecha, T., Saha, S., Ma, T., Reddy, V., Arora, R.: Pre-training BERT on domain resources for short answer grading. In: The 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, pp. 6071–6075. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/D19-1628. https://aclanthology.org/D19-1628
55. Ghavidel, H.A., Zouaq, A., Desmarais, M.C.: Using BERT and XLNET for the automatic short answer grading task. In: The 12th International Conference on Computer Supported Education - Volume 1: CSEDU, pp. 58–67. INSTICC, SciTePress (2020). https://doi.org/10.5220/0009422400580067
56. Zhu, X., Wu, H., Zhang, L.: Automatic short-answer grading via BERT-based deep neural networks. IEEE Trans. Learn. Technol. **15**(3), 364–375 (2022). https://doi.org/10.1109/TLT.2022.3175537
57. Dzikovska, M., et al.: SemEval-2013 task 7: the joint student response analysis and 8th recognizing textual entailment challenge. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: The Seventh International Workshop on Semantic Evaluation (SemEval 2013), Atlanta, Georgia, USA, pp. 263–274. Association for Computational Linguistics (2013). https://aclanthology.org/S13-2045
58. Gomaa, W.H., Fahmy, A.A.: Ans2vec: a scoring system for short answers. In: Hassanien, A.E., Azar, A.T., Gaber, T., Bhatnagar, R., F. Tolba, M. (eds.) AMLTA 2019. AISC, vol. 921, pp. 586–595. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-14118-9_59
59. Johnson, M., et al.: Google's multilingual neural machine translation system: enabling zero-shot translation. Trans. Assoc. Comput. Linguist. **5**, 339–351 (2017). https://doi.org/10.1162/tacl_a_00065. https://aclanthology.org/Q17-1024
60. Schlippe, T., Stierstorfer, Q., Koppel, M.T., Libbrecht, P.: Explainability in automatic short answer grading. In: Cheng, E.C.K., Wang, T., Schlippe, T., Beligiannis, G.N. (eds.) AIET 2022, pp. 69–87. Springer, Cham (2023). https://doi.org/10.1007/978-981-19-8040-4_5



Enhancing Credit Risk Assessment Through Transformer-Based Machine Learning Models

Elekanyani Siphuma(✉) and Terence van Zyl

Institute for Intelligent Systems, University of Johannesburg, Johannesburg, South Africa

elekanyanisiphuma@hotmail.co.za, tvanzyl@uj.ac.za

Abstract. This study evaluates the effectiveness of transformer-based deep learning models in improving credit risk assessment for predicting default probabilities among credit card customers. By employing the CNN-SFTransformer and GRU-Transformer models, this research aims to enhance predictive accuracy and robustness compared to traditional machine learning methods. The models were trained and tested on diverse datasets from Taiwan, Germany, and Australia, representing various credit risk scenarios. The experimental setup included rigorous hyperparameter tuning and utilized key evaluation metrics such as ROC AUC, KS statistic, and G-$\tilde{\mu}$ to assess model performance comprehensively. The CNN-SFTransformer model demonstrated superior performance, consistently surpassing baseline models like LSTM, Support Vector Machines (SVM), and Random Forest across all datasets. This performance indicates its enhanced capability in differentiating between defaulters and non-defaulters. The GRU-Transformer model also showed promising results, further validating the effectiveness of transformer architectures in this domain. Statistical significance of the results was confirmed through the McNemar test, ensuring the robustness and reliability of the proposed models. This research introduces a novel approach to credit risk management by providing scalable and adaptable models that improve the precision of default predictions, thereby aiding financial institutions in making more informed lending decisions with greater confidence.

Keywords: Credit Scoring · CNN-SFTransformer · GRU-Transformer

1 Introduction

Credit risk, the risk associated with lending to borrowers who may default on their credit obligations, is a significant challenge for lending institutions [1,2]. Basel regulations specify that customers default if they fail to make payments for 90 consecutive days [3]. Traditionally, statistical models like logistic regression and linear discriminants have been used to model this risk due to their ease of

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 124–143, 2025.
https://doi.org/10.1007/978-3-031-78255-8_8

implementation and interpretability [4]. However, these models often fall short when dealing with large, complex, and non-linear datasets [5].

The rapid increase in financial data has opened opportunities to enhance credit risk modelling using machine learning techniques. Machine learning, which builds algorithms trained on data to recognize patterns and support decision-making, can be divided into traditional machine learning and deep learning [6–8]. Traditional models, such as logistic regression, support vector machines, decision trees, and random forests, handle well-structured datasets effectively. In contrast, deep learning models, including recurrent neural networks (RNN), long short-term memory (LSTM), convolutional neural networks (CNN), and transformers, excel with complex, non-linear, and unstructured data [9].

Many researchers have applied traditional machine learning models to credit risk modelling, achieving reasonable prediction accuracy with simple datasets [10–12]. However, these models perform poorly with large, high-dimensional, sequential datasets like those from credit card and P2P lending platforms [13]. To overcome these limitations, scholars have turned to deep learning methods. For instance, Wang et al. [9] used LSTM models to analyze customer behaviour on a P2P lending platform, demonstrating better performance than traditional models. Similarly, Ala'raj et al. [14] applied LSTM to evaluate credit card spending behaviour, achieving higher prediction accuracy. In contrast, Zhang et al. [15] showed that combining textual and hard features with a transformer encoder improved default prediction.

More recently, transformer models have gained attention in the credit risk domain due to their success in natural language processing (NLP) tasks [16]. Wang and Xiao [13] introduced a feature-embedded transformer model for predicting customer defaults, showing that it outperformed LSTM, logistic regression, and gradient-boosted decision trees in terms of ROC-AUC and KS statistics. These findings indicate the potential of transformer models to address some of the limitations inherent in LSTM and traditional approaches, particularly when dealing with high-dimensional and sequential financial data.

The Taiwan credit default crisis underscored the necessity for better lending risk management strategies [17,18]. As financial data volumes and complexity have grown, researchers have turned to machine learning models to classify borrowers. Yet, traditional approaches still fall short with high-dimensional and sequential datasets [13]. While LSTM has mitigated some of these limitations, it remains susceptible to noise and requires longer training times [13]. With their ability to handle long-term dependencies and perform parallel processing, transformer models provide a promising alternative.

This research aims to model credit card customer behaviour using the CNN-SFTransformer and GRU-Transformer models, compare their effectiveness with LSTM and traditional classifiers, and demonstrate the statistical significance of their performance improvements using the McNemar test.

Given this background, the following research questions guide this study:

- To what extent do the CNN-SFTransformer and GRU-Transformer models outperform LSTM and traditional models in predicting default probability, as measured by AUC, KS statistic, and G-$\tilde{\mu}$?
- Is the performance improvement of the CNN-SFTransformer and GRU-Transformer models over traditional models statistically significant according to the McNemar test?

By addressing these questions, this study aims to contribute to more effective credit risk management strategies. The ultimate goal is to provide lending institutions with enhanced tools for categorizing customers based on their risk profiles, enabling more informed decision-making in the context of credit risk.

2 Proposed Architectures for Credit Risk Prediction

This study proposes two architectures for credit risk prediction: the CNN-SFTransformer and the GRU-Transformer. The CNN-SFTransformer, adopted from Wang et al. [19], demonstrated good performance on German and Australian datasets. However, the results presented by Wang et al. [19] cannot be fully relied upon as they did not perform statistical significance tests to validate the performance of their models. This study extends their work by incorporating the McNemar test to ensure the robustness and reliability of the CNN-SFTransformer results.

Wang et al. [19] initially applied the CNN-SFTransformer to the German and Australian datasets. In contrast, this research extends its application to the Taiwan credit card dataset, adapting and fine-tuning the model to suit the unique characteristics of this dataset. By addressing the limitations of the original study and enhancing the model with statistical significance validation, this study aims to provide a more robust and reliable credit risk prediction framework.

In addition to the CNN-SFTransformer, the newly proposed GRU-Transformer model is introduced. This model integrates Gated Recurrent Units (GRUs) with multi-head attention mechanisms to capture sequential and global dependencies in credit risk prediction.

2.1 CNN-SFTransformer Architecture

The CNN-SFTransformer integrates Convolutional Neural Network (CNN) and Semantic Feature Transformer (SFTransformer) components in a parallel structure for comprehensive feature extraction, as shown in Fig. 1. The model processes the feature data in two parallel networks:

SFTransformer: Feature data is fed into the SFTransformer to extract feature information between different feature parameters. Initially, a Gaussian distribution-weighted tokenization module converts the feature data into tokenized semantic features, which are then fed into the SFTransformer to construct semantic queries (Q) and keys (K). The SFTransformer's output is

normalized, capturing global dependencies and interactions among features. The multi-head attention mechanism is defined as:

$$\text{Multi-Head}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_h)W^O \tag{1}$$

where Q represents the queries derived from the tokenized semantic features, K denotes the keys also derived from the tokenized semantic features, V signifies the values extracted from the tokenized semantic features, and W^O is the learnable weight matrix. The attention for each head (head_i) is computed as:

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \tag{2}$$

where W_i^Q, W_i^K, and W_i^V are the learnable weight matrices specific to each attention head head_i. The attention function is:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{3}$$

where d_k denotes the dimensionality of the keys K.

CNN: Feature data is processed by a two-layer 1D CNN. The first convolutional layer contains 16 kernels with size three and stride 1usesocal correlations. The operation of the convolutional layer is expressed as:

$$\text{Conv}(x) = (x * w) + b \tag{4}$$

where x is the input, w is the filter, and b is the bias. The output undergoes ReLU activation:

$$\text{ReLU}(x) = \max(0, x) \tag{5}$$

which is then followed by another convolutional layer and batch normalization. Dropout layers prevent overfitting. The flattened output from the 1D CNN is combined with the SFTransformer output in the Features Fusion layer. The final prediction uses a dense layer with a sigmoid activation function, averaging losses from the CNN and SFTransformer components to guide training.

2.2 GRU-Transformer Architecture

The GRU-Transformer combines Gated Recurrent Units (GRU) with Multi-Head Attention blocks to capture sequential and global dependencies, as shown in Fig. 2. The architecture includes:

Input Layer: Ensures data is correctly structured for subsequent layers.

Bidirectional GRU Layer: Contains 64 GRU units capturing information from past and future states. The GRU operations include:

- **Update Gate:** $z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$
- **Candidate Hidden State:** $\tilde{h}_t = \tanh(W_h \cdot (r_t \odot h_{t-1}, x_t))$
- **Hidden State:** $h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$

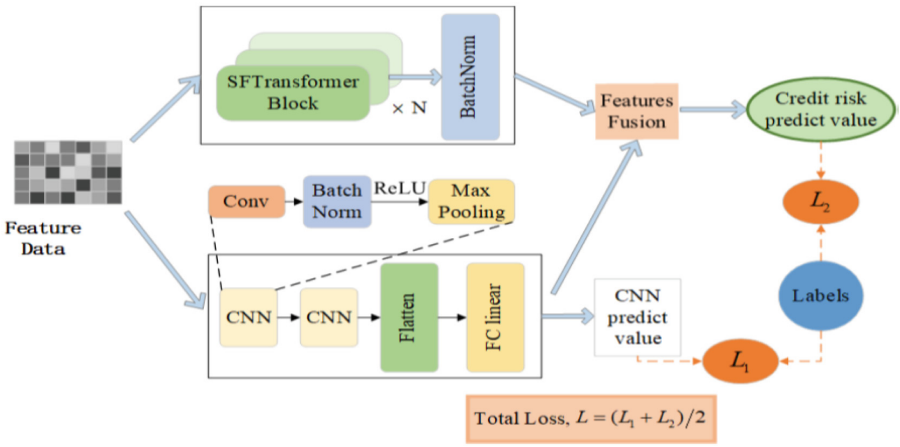


Fig. 1. CNN-SFTransformer [19]

where z_t is the update gate vector at time step t, σ is the sigmoid activation function, W_z and W_h are the weight matrices for the update gate and hidden state respectively, h_{t-1} is the hidden state vector from the previous time step $t-1$, x_t is the input vector at the current time step t, tanh is the hyperbolic tangent activation function, and $r_t \odot h_{t-1}$ is the element-wise multiplication of the reset gate and the previous hidden state.

Transformer Blocks: Include multi-head attention, dropout, add layers (skip connections), and layer normalization.

Flatten Layer: Converts the 3D tensor into a 2D tensor, preparing it for the final dense layer.

Dense Layer: With a sigmoid activation function, outputs the probability of default.

The GRU-Transformer effectively captures sequential patterns and long-term dependencies, making it suitable for credit risk prediction.

2.3 Comparative Analysis and Justification

The CNN-SFTransformer is effective for data with strong local patterns, while the GRU-Transformer is tailored for time-series data with significant temporal patterns. Both architectures were empirically evaluated against baseline models (LSTM, Logistic Regression, Random Forest, SVM), ensuring robust and reliable credit risk predictions.

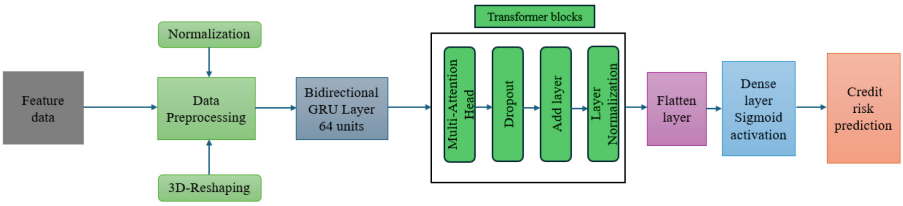


Fig. 2. The GRU-Transformer Architecture

3 Methods

This study employs a confirmatory experimental design methodology, focusing on the hypothesis-driven evaluation of transformer-based models to enhance credit risk prediction. Confirmatory experimental design emphasizes testing hypotheses through structured experiments [20], ensuring that the proposed models are rigorously evaluated for their effectiveness in improving credit risk assessment. The methodology encompasses dataset descriptions, experimental setup, hyperparameter selection, and evaluation metrics. Additionally, the methodology includes a detailed description of the comparison models used for benchmarking.

3.1 Dataset Description

The study utilized three datasets from the UCI Machine Learning Repository: the Taiwan Credit Card dataset, the German Credit dataset, and the Australian Credit Approval dataset. The Taiwan Credit Card dataset contains 30000, the German Credit dataset includes 1000, and the Australian Credit Approval dataset has 690 records. Table 1 shows detailed summaries of these datasets.

Table 1. Credit Card Dataset Description

Dataset	Default	Non-Default	Features
Taiwan Credit Card	6000	24000	24
German Credit	300	700	21
Australian Credit	307	383	15

3.2 Experimental Setup

The study uses Python 3.10 in Jupyter Notebook with the TensorFlow and Scikit-learn libraries to conduct the experiments. Datasets were split into 80% training and 20% testing data to ensure robust model evaluation. Data

preprocessing involved transforming temporal features into three-dimensional sequences for the Taiwan dataset using a sliding window of size three with a step size of one, which was crucial for capturing temporal dependencies effectively. The study uses a fixed sliding window of size for the German and Australian datasets. The Standard-Scaler was used for normalization, ensuring that all features contributed equally to the model's learning process. The proposed models, CNN-SFTransformer and GRU-Transformer, were trained with dropout layers, layer normalization, skip connection and the Adam optimizer with a learning rate 0.001. The study uses binary cross-entropy as the loss function.

3.3 Hyperparameter Selection

Due to the computational expense of hyperparameter tuning techniques, the study employed a less computationally expensive approach. This approach involved systematically adjusting three key hyperparameters-number of transformer blocks, attention heads, and batch size-to enhance the performance of both the CNN-SFTransformer and GRU-Transformer models. The ranges for these hyperparameters are summarized in Table 2. To systematically explore the impact of each hyperparameter, one hyperparameter was varied at a time while keeping others constant, with the validation set used to evaluate the effect on model performance. These hyperparameters were selected by values commonly used in the literature.

Table 2. Hyperparameters Ranges

Hyperparameter	Range
Transformer blocks	1–3
Attention heads	6, 8, 10, 12
Batch size	32, 64, 128

3.4 Evaluation Metrics

The study employs ROC-AUC, KS Statistic, and G-$\tilde{\mu}$ to comprehensively assess the models' performance, predictive power and robustness. These metrics were chosen because they are widely used in credit risk modeling due to their ability to handle imbalanced datasets and assess the predictive power of models in distinguishing defaulters from non-defaulters.

ROC-AUC: This metric measures the area under the receiver operating characteristic curve, which plots the true positive rate (TPR) against the false positive rate (FPR) across different threshold settings. A higher ROC-AUC indicates better performance in distinguishing between positive and negative classes.

KS Statistic: This statistic measures the maximum difference between the cumulative distribution functions of the true and false positive rates. A higher KS value reflects the greater discriminatory power of the model, showing its ability to separate positive and negative cases effectively.

G-$\tilde{\mu}$: This metric assesses the model's accuracy in classifying both majority and minority classes by balancing the true positive and negative rates. A higher G-$\tilde{\mu}$ indicates the model performs well in detecting positive cases and avoiding false positives.

McNemar test: To assess the statistical significance of the results, the McNemar test was utilized to compare the CNN-SFTransformer model against each baseline algorithm. This test examines discordant pairs of predictions to evaluate performance differences. The McNemar test statistic follows a chi-square distribution with 1 degree of freedom. A threshold value of 3.841 was used to reject the null hypothesis, as it corresponds to the 0.05 significance level in the chi-square distribution. A McNemar test statistic exceeding 3.841 indicates a significant difference in performance between the models, confirming meaningful discrepancies in classification results. The higher the McNemar test statistic value, the greater the difference in performance between the models.

3.5 Comparison Models

The machine-learning models used for comparison in this study include Long Short-Term Memory (LSTM) networks, Support Vector Machines (SVM), Random Forests (RF), and Logistic Regression (LR). LSTM networks are recurrent neural networks (RNN) designed for processing sequential data, making them particularly effective for tasks involving temporal dependencies and historical patterns, such as credit risk modelling [21,22]. SVMs are powerful supervised learning models that find the optimal hyperplane to separate data points of different classes. They enhance generalization to unseen data and make them suitable for binary classification tasks like predicting credit defaults [23,24]. Random Forest is an ensemble learning method that constructs multiple decision trees and aggregates their predictions. It provides robust performance by addressing overfitting and high variance associated with individual decision trees [25,26]. Logistic regression is a statistical method for binary classification that models the probability of a given input belonging to a specific class, offering transparent probabilistic estimates of default risk and insights into the significance of different features, making it effective for many fields of data analysis, including credit risk modelling [27,28].

4 Results and Discussion

The results analysis and discussion cover the performance evaluation of the Transformer-based models (CNN-SFTransformer and GRU-Transformer) on the Taiwan, Australian, and German credit datasets. These models are compared against baseline models like LSTM, Logistic Regression, Random Forest, and SVM, using performance metrics including ROC AUC, KS Statistic, and G-$\tilde{\mu}$. The statistical significance of the observed results is validated through the McNemar test.

The performance of both the CNN-SFTransformer and GRU-Transformer models is influenced by several hyperparameters. Detailed experimentation was conducted for both models, focusing on three critical hyperparameters: the number of transformer blocks, attention heads, and batch size. The study uses a systematic approach to identify the optimal configuration for each dataset (Taiwan, German, and Australian). One hyperparameter was varied at a time while keeping the others fixed, allowing for a systematic evaluation of their impact on model performance.

4.1 CNN-SFTransformer Hyperparameter Selection

Transformer Blocks: The number of transformer blocks varied from 1 to 3 as shown in Table 4 in the appendix. For the Taiwan dataset, the best performance was observed with three transformer blocks, achieving an AUC of **77.46%**, a KS of **42.45%**, and a G-$\tilde{\mu}$ of **59.28%**. One transformer block yielded the highest values for the German and Australian datasets, indicating fewer blocks might be more effective for these datasets.

Attention Heads: The number of attention heads varied from 6 to 12, as shown in Table 5 in the appendix. Performance improved consistently by increasing attention heads up until 10, which provided the best results across all datasets. However, increasing the attention heads to 12 led to a slight decline in performance. This decrease may be attributed to overfitting or the increased complexity of the model, which can lead to diminished returns in performance when the attention mechanisms become too dense. The optimal number of attention heads balances capturing sufficient context and maintaining model generalization.

Batch Size: Batch size was varied from 32 to 128 as shown in Table 6 in the appendix. The results show that the best performance for the model for the Taiwan dataset was achieved with a batch size of 64. For the German and Australian datasets, a batch size of 32 yielded the highest values. Increasing the batch size beyond 32 for the German and Australian datasets and beyond 64 for the Taiwan dataset did not improve further performance.

The study then used the optimal hyperparameters to compare the performance with baseline algorithms. Overall, the results indicate that the CNN-SFTransformer model, with the best hyperparameters, demonstrates superior performance across different datasets.

4.2 GRU-Transformer Hyperparameter Selection

Transformer Blocks: The number of transformer blocks varied from 1 to 3, as shown in Table 7 in the appendix. Similar to the observations with the CNN-SFTransformer model, the optimal number of transformer blocks varied across datasets. Three blocks provided the best performance for the Taiwan dataset, capturing more complex patterns effectively. However, one transformer block yielded the best results across all metrics for the German and Australian datasets, indicating that simpler configurations were more effective.

Attention Heads: The number of attention heads was varied from 6 to 12 as shown in Table 8 in the appendix. The evaluation showed that the best performance was observed with ten attention heads for all datasets.

Batch Size: Batch size was varied from 32 to 128 as shown in Table 9 in the appendix. The batch size of 64 was optimal for the Taiwan dataset, whereas a batch size of 32 was ideal for the German and Australian datasets.

Both the CNN-SFTransformer and GRU-Transformer models were retrained using the identified optimal hyperparameters. Although the CNN-SFTransformer generally outperformed the GRU-Transformer, the final evaluations of both models using the test datasets are detailed in the following subsection. The results provide a comprehensive comparison of their performance with the optimized settings.

4.3 Model Analysis

Table 3 presents a detailed analysis of the experimental results obtained from Taiwan, Australian, and German credit datasets. The models' performance is evaluated based on ROC AUC, KS Statistic, and G-$\tilde{\mu}$ metrics, comprehensively comparing their predictive capabilities.

Taiwan Dataset: The CNN-SFTransformer achieved the highest ROC-AUC (77.69%), KS (42.60%), and G-$\tilde{\mu}$ (60.09%) as shown in Table 3. These results indicate balanced performance and strong discriminatory power. The GRU-Transformer also demonstrated robust performance, though slightly lower than the CNN-SFTransformer. The LSTM model follows closely, with a ROC-AUC of 77.64%, KS of 42.25% and G-$\tilde{\mu}$ of 58.88%. SVM and Logistic Regression (LR) demonstrated limited effectiveness, with LR being the lowest performer.

Australian Dataset: Similar to the observations from the Taiwan dataset, the CNN-SFTransformer outperformed other models with the highest AUC (94.34%), KS (78.92%), and G-$\tilde{\mu}$ (87.77%) as shown in Table 3, demonstrating superior classification and balance. The GRU-Transformer followed closely with an AUC of 93.65%, KS of 78.87%, and G-$\tilde{\mu}$ of 86.50%. The SVM also performed strongly with an AUC of 93.27%, KS of 78.69%, and G-$\tilde{\mu}$ of 86.64%. LSTM, RF, and LR also yielded good results but outperformed the transformer-based models.

Table 3. Model Performance on Different Datasets

Classifier	Taiwan Dataset		
	AUC (%)	KS (%)	G-$\tilde{\mu}$ (%)
GRU-Transformer	77.23	42.59	59.61
CNN-SFT	**77.69**	**42.60**	**60.09**
LSTM	77.64	42.25	58.88
SVM	71.86	34.83	56.57
LR	70.79	36.39	48.48
RF	74.54	38.14	58.10
Classifier	Australian Dataset		
	AUC (%)	KS (%)	G-$\tilde{\mu}$ (%)
GRU-Transformer	93.65	78.87	86.87
CNN-SFT	**94.34**	**78.92**	**87.77**
LSTM	92.57	76.58	86.69
SVM	93.27	78.69	86.64
LR	92.22	75.11	86.02
RF	92.81	75.97	86.24
Classifier	German Dataset		
	AUC (%)	KS (%)	G-$\tilde{\mu}$ (%)
GRU-Transformer	80.63	55.13	69.73
CNN-SFT	**81.25**	**56.24**	**69.98**
LSTM	80.05	54.84	68.37
SVM	78.85	50.94	66.49
LR	79.92	50.51	69.61
RF	79.00	49.39	65.67

German Dataset: The CNN-SFTransformer continued to show its robust performance, achieving an AUC of 81.25%, KS of 56.24%, and G-$\tilde{\mu}$ of 69.98% (Table 3), highlighting its superior classification capability and well-balanced performance. It outperformed the GRU-Transformer and LSTM models, with the LSTM achieving an AUC of 80.05%, KS of 54.84%, and G-$\tilde{\mu}$ of 68.37%. The SVM and Logistic Regression (LR) models demonstrated moderate performance, while Random Forest (RF) was the lowest performer.

The CNN-SFTransformer consistently demonstrated superior performance across all three datasets, outperforming state-of-the-art models like LSTM. This performance underscores its ability to effectively utilize attention mechanisms, capturing complex patterns and enhancing model robustness.

4.4 Statistical Significance Analysis

To validate whether the performance improvements of the CNN-SFTransformer model over baseline algorithms are statistically significant, the McNemar test was conducted five times for each baseline algorithm across the Taiwan, German, and Australian datasets. A p-value of 0.05 and a critical value of 3.841 were used to assess statistical significance. This analysis ensures that the observed performance improvement of the CNN-SFTransformer was not due to random chance.

Taiwan Dataset: Table 10 in the appendix shows the McNemar test results for the Taiwan dataset. The results indicate that the CNN-SFTransformer model consistently outperforms baseline algorithms, with significant differences in most runs. Logistic Regression (LR) showed significant differences in all five runs, with χ^2 values ranging from 30.91 to 44.34 and p-values < 0.001. SVM also demonstrated significant differences in four out of five runs. LSTM showed significant differences in all five runs, while Random Forest (RF) exhibited fewer significant differences (three out of five), indicating closer performance to CNN-SFTransformer. GRU-Transformer showed significant differences in two out of five runs.

German dataset: Table 11 in the appendix shows that the CNN-SFTransformer model consistently demonstrated robust performance, maintaining statistical significance in several runs against baseline algorithms. The LSTM model showed statistically significant differences in four out of five runs, indicating that CNN-SFTransformer consistently outperforms LSTM. The GRU-Transformer exhibited significant differences in three out of five runs. Random Forest (RF) showed significant differences in all five runs, highlighting that CNN-SFTransformer outperforms RF. Logistic Regression (LR) displayed significant differences in three runs. Support Vector Machine (SVM) showed significant differences in only two runs, suggesting a closer alignment in performance with CNN-SFTransformer in this dataset.

Australian Dataset: Table 12 in the appendix shows similar observations to the Taiwan and German datasets. The CNN-SFTransformer model demonstrated robust performance with statistical significance against baseline algorithms in multiple runs. LSTM showed significant differences in all five runs, with extremely high χ^2 values ranging from 97.39 to 122.87 and p-values < 0.001. RF also showed significant differences in all runs, with χ^2 values ranging from 42.75 to 43.91 and p-values < 0.001. GRU-Transformer showed significant differences in three out of five runs. SVM and LR exhibited significant differences in three runs, further validating the robustness of the CNN-SFTransformer model.

Overall, the McNemar test results confirm the superior performance of the CNN-SFTransformer model across all datasets, with statistically significant improvements over baseline algorithms.

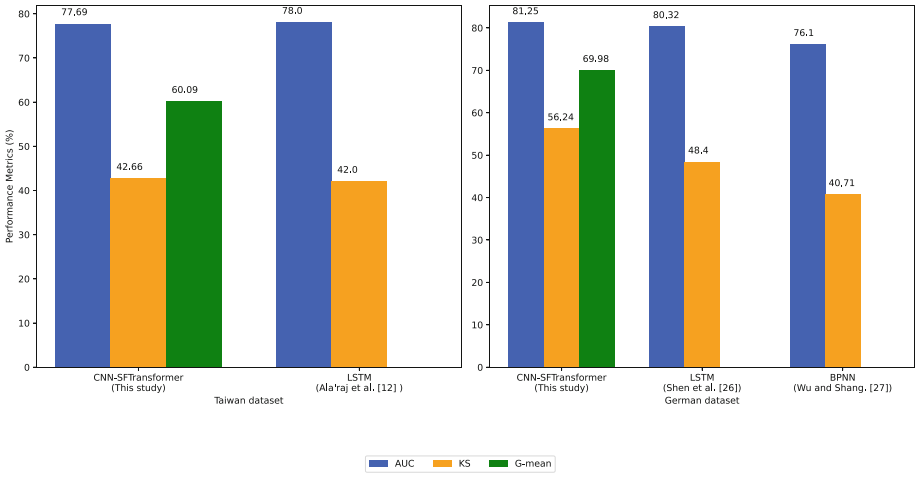


Fig. 3. Comparative performance of CNN-SFTransformer and state-of-the-art models

4.5 Discussion

The results of this study underscore the CNN-SFTransformer model's effectiveness in predicting credit card default probabilities. Statistical significance testing confirmed the robust performance of the CNN-SFTransformer model, affirming that the observed results are not due to random chance. Figure 3 illustrates the CNN-SFTransformer's performance compared to state-of-the-art approaches reported in the literature for the Taiwan and German credit card datasets.

For the Taiwan dataset, the CNN-SFTransformer achieved an AUC of 77.69%, a KS statistic of 42.66%, and a G-$\tilde{\mu}$ of 60.09%. While Ala'raj et al. [14] reported a slightly higher AUC of 78.00% with their LSTM model, the CNN-SFTransformer exhibited a superior KS statistic, indicating improved discriminatory power. For the German dataset, the CNN-SFTransformer significantly outperformed existing models, achieving an AUC of 81.25%, a KS of 56.34%, and a G-$\tilde{\mu}$ of 69.98%. As shown in Fig. 3, these results surpass those reported by Shen et al. [29] for their LSTM model (AUC of 80.32% and KS of 48.40%) and Wu and Shang. [30] with their BPNN model (AUC of 76.10% and KS of 40.71%).

The CNN-SFTransformer's architecture, which integrates Convolutional Neural Networks (CNN) with a Semantic Feature Transformer (SFTransformer), allows for effective local and global feature extraction. This comprehensive approach, capturing both temporal and non-temporal features, enhances the model's ability to understand customer behaviour and improve predictive accuracy.

Implementing the CNN-SFTransformer offers several advantages for lending institutions, including enhanced risk management capabilities and more informed lending decisions, which could reduce default rates and improve credit

portfolio quality. The model's interpretability, supported by attention mechanisms, provides valuable insights into credit risk factors, facilitating refined credit scoring policies and targeted customer management strategies. Furthermore, its scalability ensures effective application to large datasets.

Despite these strengths, the study acknowledges limitations such as high computational costs and the requirement for large datasets, which may pose challenges for smaller institutions. Future research should explore additional features and alternative model architectures and test the model's applicability across different financial contexts to further validate its robustness and generalizability.

5 Conclusion

This study explored deep learning techniques, focusing on transformer-based models to enhance credit risk management by predicting default probabilities among credit card customers. The CNN-SFTransformer and GRU-Transformer models were developed and validated, with their performance compared to baseline models including LSTM, RF, GB, SVM, and LR. The CNN-SFTransformer demonstrated superior discriminatory power on the Taiwan dataset, achieving an AUC of 77.69%, KS of 42.66%, and G-$\tilde{\mu}$ of 60.09%. Similar superior performance of the CNN-SFTransformer was observed on the German and Australian datasets, confirming its generalizability. The GRU-Transformer also improved over traditional models, indicating its effectiveness in credit risk management. The statistical significance of the CNN-SFTransformer's performance improvements was validated using the McNemar test, ensuring these improvements were not due to random chance.

The findings of this study suggest that transformer-based models, especially the CNN-SFTransformer, can significantly enhance credit risk prediction. Financial institutions are encouraged to adopt these advanced models to improve risk management and inform lending decisions. Future research should explore these models across various loan types and economic conditions and optimize hyperparameters for better performance. The scalability and interpretability of these models offer valuable insights into credit risk factors, which can aid in developing targeted credit scoring policies.

Overall, the CNN-SFTransformer and GRU-Transformer models represent significant advancements in credit risk prediction, offering potential benefits for risk management and portfolio quality.

A Appendix

Table 4. CNN-SFTransformer: Transformer Blocks Hyperparameter Selection

Blocks	Taiwan			German			Australian		
	AUC	KS	G-$\tilde{\mu}$	AUC	KS	G-$\tilde{\mu}$	AUC	KS	G-$\tilde{\mu}$
1	77.25	42.38	59.17	**81.08**	**55.65**	**69.42**	**93.30**	**77.54**	**87.04**
2	77.33	42.39	59.22	80.54	55.41	69.39	93.15	77.51	86.98
3	**77.46**	**42.45**	**59.28**	80.38	55.27	69.31	92.99	77.49	86.97

Table 5. CNN-SFTransformer: Multi-Attention Head Hyperparameter Selection

Heads	Taiwan			German			Australian		
	AUC	KS	G-$\tilde{\mu}$	AUC	KS	G-$\tilde{\mu}$	AUC	KS	G-$\tilde{\mu}$
6	77.46	42.45	59.28	81.08	55.65	69.42	93.30	77.54	87.04
8	77.49	42.48	59.33	81.15	55.74	69.58	93.44	77.60	87.20
10	**77.53**	**42.51**	**59.41**	**81.19**	**55.87**	**69.67**	**93.64**	**77.82**	**87.29**
12	77.51	42.50	59.41	81.17	55.88	69.65	93.61	77.75	87.23

Table 6. CNN-SFTransformer: Batch Size Hyperparameter Selection

Batch	Taiwan			German			Australian		
	AUC	KS	G-$\tilde{\mu}$	AUC	KS	G-$\tilde{\mu}$	AUC	KS	G-$\tilde{\mu}$
32	77.53	42.51	59.41	**81.19**	**55.87**	**69.67**	**93.64**	**77.82**	**87.29**
64	**77.56**	**42.55**	**59.47**	81.11	55.82	69.62	93.61	77.79	87.24
128	77.52	42.53	59.44	81.08	55.78	69.59	93.59	77.74	87.20

Table 7. GRU-Transformer: Transformer Blocks Hyperparameter Selection

Transformer block	Taiwan			German			Australian		
	AUC	KS	G-$\tilde{\mu}$	AUC	KS	G-$\tilde{\mu}$	AUC	KS	G-$\tilde{\mu}$
1	76.74	41.66	58.87	**79.43**	**54.67**	**69.32**	**93.27**	**77.42**	**85.04**
2	76.77	41.69	58.89	79.40	54.60	69.11	93.24	77.39	84.99
3	**76.79**	**41.74**	**58.91**	79.37	54.57	69.10	93.24	78.35	84.96

Table 8. GRU-Transformer: Multi-Attention Head Hyperparameter Selection

Heads	Taiwan			German			Australian		
	AUC	KS	G-$\tilde{\mu}$	AUC	KS	G-$\tilde{\mu}$	AUC	KS	G-$\tilde{\mu}$
6	76.79	41.74	58.91	79.43	54.67	69.32	93.27	77.42	85.04
8	76.82	41.77	58.94	79.51	54.74	69.56	93.31	77.68	85.08
10	**76.86**	**41.84**	**58.98**	**79.67**	**54.87**	**69.70**	**93.36**	**77.74**	**85.12**
12	76.85	41.81	58.96	79.63	54.84	69.68	93.34	77.71	85.10

Table 9. GRU-Transformer: Batch Size Hyperparameter Selection

Batch	Taiwan			German			Australian		
	AUC	KS	G-$\tilde{\mu}$	AUC	KS	G-$\tilde{\mu}$	AUC	KS	G-$\tilde{\mu}$
32	76.86	41.84	58.98	**79.67**	**54.87**	**69.70**	**93.36**	**77.54**	**85.12**
64	**76.91**	**41.88**	**59.12**	79.65	54.84	69.56	93.34	77.51	85.11
128	76.87	41.86	59.11	79.61	54.82	69.51	93.30	77.48	84.99

Table 10. Taiwan Dataset: McNemar's test results for multiple classifiers over five runs

Classifier	χ^2	p-value
LSTM		
Run 1	13.26	2.71×10^{-4}
Run 2	6.09	1.36×10^{-2}
Run 3	32.49	1.20×10^{-8}
Run 4	4.89	2.70×10^{-2}
Run 5	23.65	1.16×10^{-6}
GRU-Transformer		
Run 1	9.27	2.56×10^{-3}
Run 2	8.74	3.12×10^{-3}
Run 3	0.47	4.92×10^{-1}
Run 4	8.81	3.00×10^{-3}
Run 5	0.87	3.98×10^{-1}
RF		
Run 1	6.57	2.65×10^{-2}
Run 2	2.97	8.50×10^{-2}
Run 3	4.42	3.55×10^{-2}
Run 4	5.08	2.42×10^{-2}
Run 5	2.97	8.50×10^{-2}
SVM		
Run 1	22.96	1.65×10^{-6}
Run 2	33.36	7.67×10^{-9}
Run 3	19.70	9.08×10^{-6}
Run 4	19.35	1.09×10^{-5}
Run 5	2.82	9.31×10^{-2}
LR		
Run 1	32.97	9.40×10^{-10}
Run 2	41.37	1.00×10^{-10}
Run 3	44.34	1.03×10^{-10}
Run 4	36.97	1.20×10^{-10}
Run 5	30.91	1.00×10^{-8}

Table 11. German Dataset: McNemar's test results for multiple classifiers over five runs.

Classifier	χ^2	p-value
LSTM		
Run 1	4.01	4.55×10^{-2}
Run 2	6.02	1.42×10^{-2}
Run 3	0.47	4.94×10^{-1}
Run 4	4.87	2.72×10^{-2}
Run 5	7.56	5.98×10^{-3}
GRU-Transformer		
Run 1	4.86	6.25×10^{-2}
Run 2	4.98	2.56×10^{-2}
Run 3	1.09	2.95×10^{-1}
Run 4	6.63	3.47×10^{-2}
Run 5	2.17	1.41×10^{-1}
RF		
Run 1	5.02	2.88×10^{-3}
Run 2	6.78	9.22×10^{-3}
Run 3	5.85	3.22×10^{-3}
Run 4	3.91	4.81×10^{-2}
Run 5	3.91	4.81×10^{-2}
SVM		
Run 1	5.45	3.25×10^{-2}
Run 2	7.11	7.65×10^{-2}
Run 3	0.01	9.10×10^{-1}
Run 4	3.24	7.18×10^{-2}
Run 5	3.26	7.25×10^{-2}
LR		
Run 1	4.99	2.88×10^{-2}
Run 2	8.27	4.04×10^{-3}
Run 3	7.85	3.98×10^{-3}
Run 4	2.55	1.10×10^{-1}
Run 5	2.58	1.11×10^{-1}

Table 12. Australian Dataset: McNemar's test results for multiple classifiers over five runs

Classifier	χ^2	p-value
LSTM		
Run 1	122.87	1.70×10^{-9}
Run 2	98.42	1.70×10^{-9}
Run 3	105.19	1.70×10^{-9}
Run 4	115.78	1.72×10^{-9}
Run 5	97.39	1.01×10^{-9}
GRU-Transformer		
Run 1	1.64	2.00×10^{-1}
Run 2	6.89	8.66×10^{-3}
Run 3	3.75	5.28×10^{-2}
Run 4	5.14	2.33×10^{-2}
Run 5	9.14	2.50×10^{-3}
RF		
Run 1	43.91	3.45×10^{-11}
Run 2	42.75	6.22×10^{-11}
Run 3	42.78	6.22×10^{-11}
Run 4	43.91	3.45×10^{-11}
Run 5	42.98	3.45×10^{-11}
SVM		
Run 1	1.59	2.08×10^{-1}
Run 2	4.24	3.58×10^{-2}
Run 3	4.07	3.88×10^{-2}
Run 4	1.95	1.62×10^{-1}
Run 5	4.04	4.12×10^{-2}
LR		
Run 1	2.82	9.33×10^{-2}
Run 2	4.34	3.73×10^{-2}
Run 3	4.88	4.72×10^{-2}
Run 4	3.75	5.28×10^{-2}
Run 5	4.14	3.88×10^{-2}

References

1. Bluhm, C., Overbeck, L., Wagner, C.: Introduction to Credit Risk Modeling. Chapman and Hall/CRC (2016)
2. Boahene, S.H., Dasah, J., Agyei, S.K.: Credit risk and profitability of selected banks in Ghana. Res. J. Finan. Account. **3**(7), 6–14 (2012)

3. Ben Lahouel, B., Taleb, L., Ben Zaied, Y., Managi, S.: Financial stability, liquidity risk and income diversification: evidence from European banks using the CAMELS-DEA approach. Ann. Oper. Res. **334**(1), 391–422 (2024)
4. Desai, V.S., Crook, J.N., Overstreet, G.A., Jr.: A comparison of neural networks and linear scoring models in the credit union environment. Eur. J. Oper. Res. **95**(1), 24–37 (1996)
5. Machado, M.R., Karray, S.: Assessing credit risk of commercial customers using hybrid machine learning algorithms. Expert Syst. Appl. **200**, 116889 (2022)
6. Bhatore, S., Mohan, L., Reddy, Y.R.: Machine learning techniques for credit risk evaluation: a systematic literature review. J. Bank. Financ. Technol. **4**(1), 111–138 (2020). https://doi.org/10.1007/s42786-020-00020-3
7. Mathonsi, T., van Zyl, T.L.: Multivariate anomaly detection based on prediction intervals constructed using deep learning. Neural Comput. Appl. 1–15 (2022)
8. Laher, S., Paskaramoorthy, A., Van Zyl, T.L.: Deep learning for financial time series forecast fusion and optimal portfolio rebalancing. In: 2021 IEEE 24th International Conference on Information Fusion (FUSION), pp. 1–8. IEEE (2021)
9. Wang, C., Han, D., Liu, Q., Luo, S.: A deep learning approach for credit scoring of peer-to-peer lending using attention mechanism LSTM. IEEE Access **7**, 2161–2168 (2018)
10. Pandey, T.N., Jagadev, A.K., Mohapatra, S.K., Dehuri, S.: Credit risk analysis using machine learning classifiers. In: 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), pp. 1850–1854. IEEE (2017)
11. Bequé, A., Lessmann, S.: Extreme learning machines for credit scoring: an empirical evaluation. Expert Syst. Appl. **86**, 42–53 (2017)
12. Shi, S., Tse, R., Luo, W., D'Addona, S., Pau, G.: Machine learning-driven credit risk: a systemic review. Neural Comput. Appl. **34**(17), 14327–14339 (2022)
13. Wang, C., Xiao, Z.: A deep learning approach for credit scoring using feature embedded transformer. Appl. Sci. **12**(21), 10995 (2022)
14. Ala'raj, M., Abbod, M.F., Majdalawieh, M., Jum'a, L.: A deep learning model for behavioural credit scoring in banks. Neural Comput. Appl. **34**(8), 5839–5866 (2022)
15. Zhang, W., Wang, C., Zhang, Y., Wang, J.: Credit risk evaluation model with textual features from loan descriptions for P2P lending. Electron. Commer. Res. Appl. **42**, 100989 (2020)
16. Patwardhan, N., Marrone, S., Sansone, C.: Transformers in the real world: a survey on NLP applications. Information **14**(4), 242 (2023)
17. Chou, M.: Cash and credit card crisis in Taiwan. Business Weekly, pp. 24–27 (2006)
18. Chang, C.-H.: Information asymmetry and card debt crisis in Taiwan. Bull. Appl. Econ. **9**(2), 123–145 (2022)
19. Wang, M., Zhou, L., Meng, Q., Kong, Y., Sun, J.: Credit risk prediction network based on semantic feature transformer and CNN. In: 2023 IEEE 6th International Conference on Electronic Information and Communication Technology (ICEICT), pp. 723–728. IEEE (2023)
20. Nilsen, E.B., Bowler, D.E., Linnell, J.D.C.: Exploratory and confirmatory research in the open science era. J. Appl. Ecol. **57**(4), 842–847 (2020)
21. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
22. Lipton, Z.C., Berkowitz, J., Elkan, C.: A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019 (2015)

23. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**, 273–297 (1995)
24. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)
25. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
26. Maimon, O.Z., Rokach, L.: Data Mining with Decision Trees: Theory and Applications, vol. 81. World Scientific (2014)
27. Hosmer Jr, D.W., Lemeshow, S., Sturdivant, R.X.: Applied Logistic Regression. Wiley (2013)
28. Menard, S.: Applied Logistic Regression Analysis, vol. 106. Sage (2002)
29. Shen, F., Zhao, X., Kou, G., Alsaadi, F.E.: A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique. Appli. Soft Comput. **98**, 106852 (2021)
30. Xinyu, W., Shang, J., et al.: Analysis of credit customer delinquency based on BP neural network model. Financ. Eng. Risk Manag. **7**(3), 181–186 (2024)



Predicting and Discovering Weather Patterns in South Africa Using Spatial-Temporal Graph Neural Networks

Adeeb Gaibie[1], Hamza Amir[1], Irene Nandutu[1,3], and Deshendran Moodley[1,2(✉)]

[1] Artificial Intelligence Research Unit and Neuroscience Institute, University of Cape Town, Cape Town, South Africa
{GBXADE002,AMRHAM001}@myuct.ac.za,
{irene.nandutu,deshen.moodley}@uct.ac.za
[2] Centre for Artificial Intelligence Research, Cape Town, South Africa
[3] Busitema University, P.O. Box 236, Tororo, Uganda

Abstract. This study explores Spatial-Temporal Graph Neural Networks (STGNNs) for predicting and discovering weather patterns in South Africa. We compared the predictive performance of three recent STGNNs, i.e. the Adaptive Graph Convolutional Recurrent Network (AGCRN), Conditional Local Convolution Recurrent Network (CLCRN) and Graph WaveNet (GWN) to a Temporal Convolution Network (TCN) to predict temperature, humidity, pressure and wind speed at 45 weather stations in South Africa. We also analysed the quality and usability of the spatial-temporal dependency graph learnt by the different STGNNs. Using the TCN as a baseline we analyse the increase in performance by adding spatial information and identify the stations most likely to be influenced by changes in neighbouring stations. While AGCRN achieved the best overall predictive performance, the dependency graph learned by CLCRN captured the most plausible spatial-temporal dependencies. The findings highlight the strengths and limitations of AGCRN and CLCRN for weather prediction and for automatic discovery of prominent weather patterns in South Africa.

Keywords: Spatial Temporal Graph Neural Networks · Weather prediction · Spatial Temporal Data Mining

1 Introduction

A weather system can be characterised as a highly dynamic and chaotic system that exhibits complex and often latent spatial and temporal dependencies or flows [4]. While traditional deep neural network (DNN) approaches, like the Temporal Convolutional Network (TCN), are able to learn temporal dependencies, they are not designed to deal with spatial dependencies. Spatial-Temporal Graph Neural Networks (STGNNs) [7], an emerging set of DNNs, can deal with both

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 144–160, 2025.
https://doi.org/10.1007/978-3-031-78255-8_9

temporal and spatial dependencies and generally outperform traditional DNNs on flow prediction applications in dynamic environments.

Formulating the graph structure is an essential aspect of STGNNs. For weather prediction, each weather station is represented as a node in a graph that captures the local weather dynamics in the area around the station, whilst the system level or global dynamics are captured by weighted edges that reflect the strength of the dependencies between stations [4,10]. Thus, a station (node) will have strong connections (links) to stations that are affected by changes in weather conditions at its location and weaker links to stations that it does not affect. In this way, a node captures local dynamics (single location), while the overall graph structure captures the global (inter-location) dynamics of the system.

While STGNNs have been widely applied for predicting traffic flow at different points in a city traffic network [18], they are less explored for weather modeling and prediction. In a recent study [4] the authors explored two STGNN techniques, GraphWaveNet (GWN), one of the most established STGNNs, with the Low Rank Weighted Graph Neural Network (WGN), and compared these with a Long Short-Term Memory and a TCN for temperature prediction at 21 weather stations in South Africa. Their results showed that GWN outperformed the other techniques including WGN. Newer and more advanced STGNN techniques have appeared since GraphWaveNet. The Conditional Local Convolution Recurrent Network (CLCRN) [10] is a specialised STGNN designed for global scale meteorological prediction and was shown to outperform other STGNNs for coarse grain weather prediction on the global WeatherBench dataset, while the Adaptive Graph Convolutional Recurrent Network (AGCRN) [1] was designed to capture fine-grained node-specific spatial and temporal dependencies and outperforms other STGNNs for city traffic prediction.

In this study we evaluate and compare AGCRN and CLCRN for weather prediction in South Africa. We use a dataset with six weather attributes observed at 45 weather stations and totalling 113 929 observations from the Western Cape, Northern Cape, and Eastern Cape regions of South Africa. We compare the STGNNs for predicting temperature, pressure, humidity and wind speed at these stations over five different time horizons. We performed a systematic node level and diffusion path analysis on the dependency graph and highlight the potential of STGNNs for automatic discovery of weather patterns.

This paper is organised as follows: Sect. 2 describes related work on STGNNs and DNNs for weather prediction; Sect. 3 describes the design of the experiments; Sect. 4 describes the results of the experiments including a comparison of prediction performance, an analysis of the performance at different stations, the dependency analysis and the path analysis; Sect. 5 summarises the key findings; and finally Sect. 6 discusses the limitations of the study and directions for future work.

2 Background and Related Work

2.1 Problem Formulation

We define a graph as:

$$G = (N, E) \tag{1}$$

where N is the set of nodes representing the different weather stations, and E is the set of edges representing the dependencies between stations.

The adjacency matrix $A \in \mathbb{R}^{n \times n}$, where n is the number of weather stations, represents the structure and strength of the dependencies between the nodes (stations) [4]. The observations at a weather station i, is a time series denoted as:

$$X_{it} = \{X_{i1}, X_{i2}, \ldots, X_{iT}\} \tag{2}$$

where an observation X_{it} consists of the features observed at the weather station, i.e. wind speed, wind direction, pressure, humidity, rainfall and temperature at each time step, t over a total of T time steps [4].

Given a sequence of historical weather observations for all features, along with the adjacency matrix A, the goal is to predict a series of future values $y_1, y_2, \ldots, y_H$ for a given weather feature at each station. The prediction horizon or number of steps to predict is denoted by H [4,10].

2.2 Spatial-Temporal Graph Neural Networks for Weather Prediction

Spatial-Temporal Graph Neural Networks (STGNNs) have been explored widely for flow prediction in dynamic environments, including traffic prediction [7], stock market prediction [13] and electricity forecasting [15]. Their applicability for weather predictions is starting to be explored. Ma *et al.,* [11] shows how the Hierarchical Spatio-Temporal Graph Neural Network (HiSTGNN) outperforms on real-time meteorological datasets. The study recommends using large-scale data with more variables and stations to enhance forecasting accuracy.

Sáenz et al. [5] apply a Graph Convolutional Network (GCN) combined with a Long Short-Term Memory (LSTM) network for predicting temperatures in Spain, achieving high prediction accuracy and robust performance.

The CLCRN implemented by Lin et al. [10] explores four weather variables and compares their performance against seven baseline models. The results indicate high performance, although there are limitations related to over-smoothness in learned local patterns and instability in the training process.

Similarly, Davidson [4] implements Graph Wave Networks (GWNs) and Weighted Graph Networks (WGNs), which outperform the baseline models, TCN and LSTM. Further, GWN exhibits superior stability and robustness in temperature prediction compared to WGN.

2.3 TCN Architecture

TCNs are deep learning models typically composed of several blocks of dilated causal convolutions, where each convolution layer only considers past sequence values and has no access to future values. The dilation factor regulates the size of each layer's receptive field, allowing TCNs to capture patterns at different time scales [12]. TCN was shown to outperform other temporal DNNs [4] for weather prediction and was therefore selected as our first baseline model.

2.4 GWN Architecture

GraphWaveNet (GWN) is the second baseline model. The architecture combines temporal convolution (Gated TCN) and GCN modules. The GCN module includes a self-adaptive adjacency matrix that can be learned using gradient descent and does not need prior information. The Gated TCN module uses one-dimensional convolutions to capture long-range temporal sequences and avoid the vanishing gradient problem. The GCN module extracts node-level dependencies by aggregating neighbourhood features. GWN has shown to produce strong prediction performance across several domains, including traffic flow prediction, stock market prediction and weather prediction [4,13,18].

2.5 AGCRN Architecture

The Adaptive Graph Convolutional Recurrent Network (AGCRN) approach [1] combines the capabilities of Graph Convolutional Networks (GCNs) with the sequential features of Gated Recurrent Units (GRU), and similar to GWN is able to dynamically learn spatial and temporal dependencies. The spatial patterns are captured using the Node Adaptive Parameter Learning (NAPL) module and Data Adaptive Graph Generation (DAGG). The NAPL processes temporal patterns by learning unique patterns and assigning parameters to each node for a specific period. In addition, DAGG computes spatial patterns by learning hidden spatial relationships among nodes and refining diverse and complex spatial patterns.

2.6 CLCRN Architecture

The Conditional Local Convolution Recurrent Network (CLCRN) [10] was designed specifically for meteorological forecasting. The architecture comprises a Conditional Local Convolution (CLC) layer and a Recurrent Neural Network (RNN) layer. The CLC layer captures spatial dependencies by performing convolution operations on specific portions of the input data. The RNN is designed to capture temporal dependencies. It constructs a hidden state using the feature map provided by the CLC layer as input at each step, which is fed back into the RNN layer at the next step. To ensure computational efficiency, a sliding window approach is adopted, enabling convolutions only on specific segments of the input data. CLCRN was shown to outperform state-of-the-art

techniques for weather prediction, specifically the Multi-Scale Spatial Temporal Graph Convolutional Network (MSTGCN), Diffusion Convolutional Recurrent Neural Network (DGCRNN), and the Spatio-Temporal Graph Convolutional Networks (STGCN).

3 Experimental Design

3.1 Data

We use a dataset of weather observations from 70 automatic weather stations across South Africa obtained from the South African Weather Service (SAWS). The dataset contains hourly data for six weather attributes, temperature, wind speed, wind direction, pressure, humidity, and rainfall, over 13 years (2010 to 2022) covering three provinces, the Western Cape, Northern Cape, and Eastern Cape. However, our study focuses on predicting only four of the attributes: temperature, wind speed, humidity, and pressure.

3.2 Pre-processing and Partitioning

We followed the pre-processing approach outlined by Hewage et al. [6] and Davidson et al. [4]. A weather station is only included in the dataset if less than 5% of its data is missing. Of the original 70, 25 stations did not meet this criteria which results in a final dataset of 45 stations, with 15 from each province. The inverse distance weighting (IDW) method was used for imputing missing data at the these stations [16]. Next, the data for the weather features is normalised to the range $[0, 1]$, and then a sliding window is applied to create input/output samples of the time series for each weather feature. Given a time series X, where x_0 represents the current time step, and with window length w and prediction horizon H, the input-output pair of the sliding window technique is represented as $[x_0 - w, \ldots, x_0]$ and $[x_1, \ldots, x_1 + H]$, respectively.

We make use of the holdout validation technique as our primary validation method, consistent with the approach adopted by several STGNN studies [2,9, 19]. We divided our 13 years of data into a training set of 11 years (2010 to 2020), 1 year (2021) for validation and 1 year (2022) for testing which gives a 11:1:1 split in terms of years or 84%:8%:8%. Our total dataset contained 113 929 observations, with a training set of 79750 samples, a validation set of 17089 samples, and a test set of 17089 samples for each of the 45 stations.

3.3 Model Implementation and Evaluation

Four different techniques were implemented, a TCN and three STGNN techniques, i.e. GWN, AGCRN and CLCRN. Four independent experiments were conducted to predict temperature, pressure, humidity and wind speed at 3, 6, 9, 12 and 24 h horizons. In each experiment, the STGNNs are trained to predict across all 45 stations concurrently, while a dedicated TCN model is created

for each weather station. Hyperparameter optimisation (HPO) is performed for each technique using a random search approach. We generated random combinations of hyperparameters and used each configuration to train the model for the first two years of the training set, but only for a 24-h prediction horizon. The validation loss was calculated for each iteration, and the combination that yielded the lowest loss was chosen as the optimal hyperparameters. A breakdown of the hyperparameters explored and the optimal values found and eventually used in the experiments is shown in Table 1.

Table 1. Parameter Search Space and Optimal Parameters Found for Each Model

Parameter	Search Space	TCN	GWN	CLCRN	AGCRN
# of Epochs	[20, 30, 40]	40	40	40	40
Batch Size	[32, 64, 125, 250]	[32, 64, 125]	64	32	64
Learning Rate	(0.0001, 0.05)	(0.0003, 0.003)	0.01	0.05	0.01
Input Window Size (hours)	[12, 24, 48, 72, 168]	[48, 72, 168]	[24, 48]	12	12
# of Layers	[1, 2, 3, 4]	1	[1, 2]	2	2
Dropout Rate	[0.1, 0.2, 0.3, 0.4]	[0.1, 0.2]	[0.1, ..., 0.5]	-	-
# of hidden units	[30, 40, 50, 60]	[30, 40, 50, 60]	-	16	-
L1, L2 Regularisers	[0.0001, 0.001, 0.01]	[0.001, 0.01]	-	-	-
Layer Normalisation	[True, False]	-	True	True	-
Adjacency Matrix Init.	[Random, Distance]	-	Random	Distance	Random
# of ST layers	[1, 2, 3, 4]	-	1	1	1
# of RNN units	[16, 32, 64]	-	-	32	64
# of Filters	[32, 64, 128]	-	[32, 64, 128]	32	-

We used root mean squared error (RMSE), and mean absolute error (MAE) as commonly employed in the evaluation of STGNNs for prediction tasks [4,17]. The RMSE, represents the square root of the average squared differences between predicted and actual values, offering a measure of the typical deviation from the observed values. In comparison, MAE calculates the average of the absolute differences between predicted and actual values, providing a more interpretable metric of the average prediction error.

3.4 Experiment Infrastructure

The TCN was implemented in Keras, while GWN, AGCRN and CLCRN was implemented in PyTorch. All experiments were conducted using the facilities provided by the University of Cape Town's High-Performance Computing Cluster on a A100 GPU or on a Dell XPS 15 laptop with an Intel Core i7-12700H @ 4.70 GHz and an NVIDIA RTX 3050 GPU (4GB GDDR6).

4 Results and Analysis

This section is divided into four main parts. First, we compare the performance of each algorithm for predicting temperature, humidity, pressure, and wind speed.

Next, we analyse the models' performance at the station level, examining how it varied across different stations. Following this, we explore the spatial dependencies highlighted by the adjacency matrices. Lastly, we identify the most prominent dependency paths.

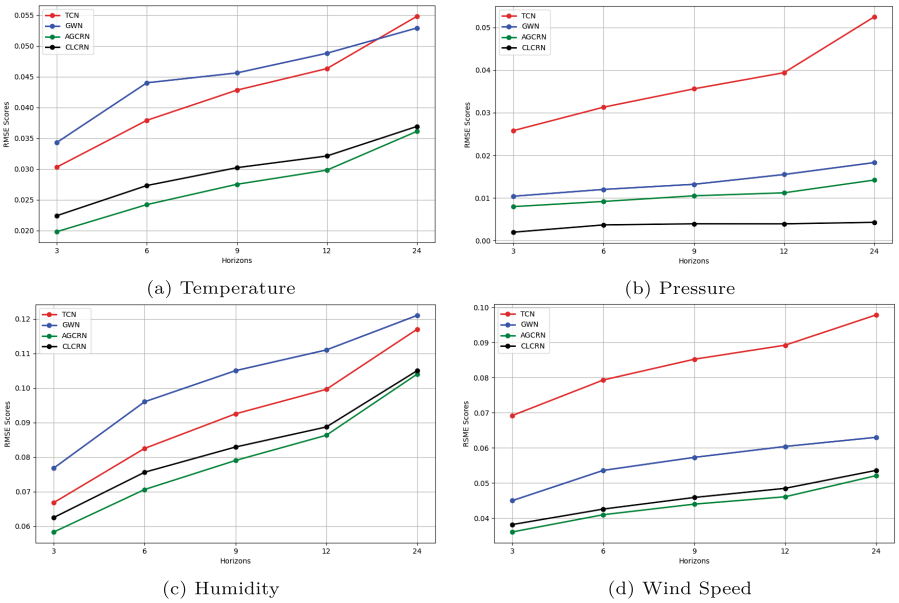
4.1 Predictive Performance

Table 2 shows the average performance of the AGCRN, CLCRN, GWN, and TCN across all 45 weather stations and over different prediction horizons. The MAE and RMSE scores shown are the mean values over three independent runs.

Table 2. Average prediction performance of each algorithm on different weather attributes and horizons. For each variable the best values (lowest) are shown in bold to highlight the best performing algorithm.

Attribute	Metrics	Models	Prediction Horizons					Average
			3	6	9	12	24	
Temperature	RMSE	TCN	0.03031	0.03793	0.04286	0.04639	0.05486	0.04247
		GWN	0.03430	0.04408	0.04566	0.04889	0.05295	0.04518
		AGCRN	**0.01982**	**0.02423**	**0.02757**	**0.02988**	**0.03610**	**0.02752**
		CLCRN	0.02249	0.02737	0.03024	0.03216	0.03692	0.02983
	MAE	TCN	0.02102	0.02678	0.03066	0.03333	0.04024	0.03041
		GWN	0.02435	0.03156	0.03321	0.03647	0.04019	0.03316
		AGCRN	**0.01345**	**0.01680**	**0.01927**	**0.02113**	**0.02618**	**0.01937**
		CLCRN	0.01534	0.01912	0.02155	0.02296	0.02673	0.02114
Pressure	RMSE	TCN	0.02580	0.03136	0.03567	0.03942	0.05259	0.03697
		GWN	0.01043	0.01201	0.01320	0.01554	0.01832	0.01390
		AGCRN	0.00796	0.00918	0.01055	0.01128	0.01420	0.01063
		CLCRN	**0.00196**	**0.00367**	**0.00393**	**0.00392**	**0.00428**	**0.00355**
	MAE	TCN	0.00903	0.01264	0.01592	0.01869	0.02891	0.01704
		GWN	0.00541	0.00619	0.00678	0.00891	0.01053	0.00756
		AGCRN	0.00217	0.00274	0.00363	0.00413	0.00644	0.00382
		CLCRN	**0.000732**	**0.00168**	**0.00191**	**0.00187**	**0.00209**	**0.00166**
Humidity	RMSE	TCN	0.06687	0.08255	0.09250	0.09961	0.11746	0.09180
		GWN	0.07682	0.09607	0.10507	0.11192	0.12156	0.10229
		AGCRN	**0.05832**	**0.07065**	**0.07902**	**0.08630**	**0.10419**	**0.07970**
		CLCRN	0.06255	0.07586	0.08293	0.08872	0.10467	0.08295
	MAE	TCN	0.04493	0.05705	0.06486	0.07052	0.08448	0.06437
		GWN	0.05422	0.06816	0.07514	0.08079	0.08990	0.07364
		AGCRN	**0.03892**	**0.04807**	**0.05448**	**0.05981**	**0.07285**	**0.05483**
		CLCRN	0.04152	0.05177	0.05763	0.06189	0.07364	0.05729
Wind Speed	RMSE	TCN	0.06922	0.07937	0.08529	0.08926	0.09785	0.08420
		GWN	0.04506	0.05368	0.05732	0.06045	0.06300	0.05590
		AGCRN	**0.03617**	**0.04105**	**0.04402**	**0.04613**	**0.05217**	**0.04391**
		CLCRN	0.03826	0.04263	0.04591	0.04850	0.05367	0.04579
	MAE	TCN	0.05155	0.05952	0.06443	0.06779	0.07506	0.06367
		GWN	0.03374	0.04051	0.04374	0.04650	0.04957	0.04281
		AGCRN	**0.02633**	**0.03022**	**0.03279**	**0.03446**	**0.03937**	**0.03263**
		CLCRN	0.02811	0.03140	0.03405	0.03603	0.04048	0.03401

From the table, it is evident that pressure exhibits the highest predictive accuracy, followed by temperature and wind speed. In contrast, humidity consistently demonstrates the poorest predictive performance. These findings hold



(a) Temperature (b) Pressure

(c) Humidity (d) Wind Speed

Fig. 1. RMSE values for each model across different forecasting horizons and weather attributes.

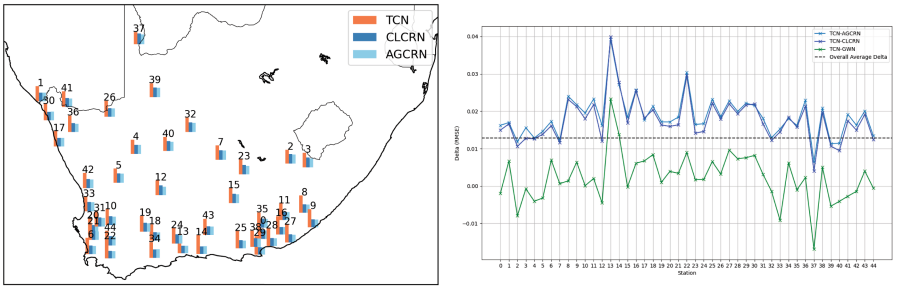
consistently across all models, likely due to the inherent differences between these attributes. In terms of predictive performance, AGCRN outperforms the other techniques for temperature, wind speed, and humidity, while CLCRN yields the best performance for pressure. For our baseline models, on average GWN outperformed TCN for the pressure and wind speed variables. However, TCN outperformed GWN on average for the humidity and temperature variables.

We visualise the RMSE values from Table 2 of the different models over different prediction horizons in Fig. 1. As anticipated, all models display decreased accuracy with increasing horizon length. Notably, TCN degrades more rapidly with longer horizons compared to the STGNN models. Although AGCRN outperformed CLCRN for temperature, wind speed, and humidity, Fig. 1 shows that the AGCRN and CLCRN lines lie very close together and are nearly parallel, indicating minimal differences in prediction accuracy across all horizons for these two models. Both models clearly outperform the baseline TCN and GWN models, consistently achieving substantially lower RMSE scores for most weather attributes (Temperature, Humidity and Wind Speed) than the TCN and GWN. The outlier is Pressure, with CLCRN showing better performance across all horizons than AGCRN. AGCRN still outperforms GWN, but not by a large margin.

4.2 Station Level Analysis

For our analysis at the station level we hone in on temperature prediction, the attribute which is predominantly explored in primary studies [4,17] that our work builds upon. We examine and compare the 24 h temperature prediction models for the TCN and the STGNNs at each station to better understand the extent to which the incorporation of spatial information impacts on predictive performance.

In Fig. 2(a), the RMSE values of the TCN, CLCRN and AGCRN for 24 h temperature prediction are overlaid on a geographic map. Figure 2(b) plots the difference between the RMSE scores for each STGNN to the TCN at each station ($RMSE_{TCN} - RMSE_{STGNN}$). A value of zero indicates the same RMSE value as the TCN, while values above zero indicate an improvement in prediction performance over the TCN. We also plotted the average improvement achieved across all stations to better identify stations where there was a substantial improvement in performance. From this, we can see which stations had the most significant improvements over the purely temporal TCN and where they are situated on the map. We observe that our STGNN models generally outperform the purely temporal TCN model, with the most significant improvements at stations 13 and 22, while the improvement over the TCN is the least pronounced at station 37. In general, we notice that the biggest improvements over the TCN occur mostly at stations along the coast, whereas the least improvements are observed at more inland stations. This suggests that the additional spatial information utilized by the STGNN architecture provides more significant benefits for coastal stations.



(a) RMSE plotted on a geographic map

(b) Difference in RMSE of TCN over STGNNs

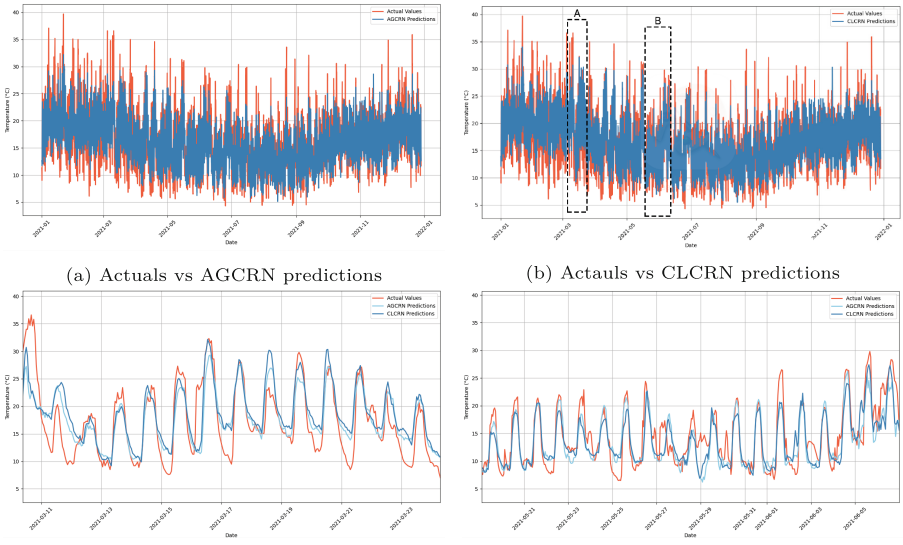
Fig. 2. Comparisons of RMSE scores achieved by each model for temperature predictions on the 24-h horizon at each station.

While the GWN improvement over TCN is less pronounced compared to CLCRN and AGCRN, with some stations where the TCN performed better, it is interesting to note that the relative performance between stations follows a similar pattern to the other STGNNs. For stations where CLCRN and AGCRN show

the greatest improvement over TCN, GWN also demonstrates the most substantial relative improvement. Conversely, for stations where CLCRN and AGCRN show less improvement, GWN shows a similar relative performance. This consistent trend in relative performance to the TCN across different stations is interesting and indicates which stations are most impacted by changes at neighbouring stations.

Given that AGCRN and CLCRN performed similarly, achieving nearly identical RMSE values for temperature, we decided to investigate these two models' predictive results at the station level. Figure 3 plots the actual values in the test set against the values predicted by each model for station 13.

Although this visual is specific to station 13, similar patterns were observed for other stations. From the observations, CLCRN tends to predict higher or more extreme peaks more accurately, while AGCRN generally provides more accurate predictions for average peaks. This indicates that CLCRN is better at detecting extreme temperatures, whereas AGCRN performs better on average temperature predictions. The two bottom plots in Fig. 3 show zoomed-in sections: Section A from March 10 to March 24, 2021, and Section B from May 19 to June 7, 2021. Section A includes average and extreme values, where CLCRN more accurately predicts the highest peaks, while AGCRN is closer to the actual values for lower average peaks. Section B contains fewer extreme values, but the same pattern is observed, with CLCRN excelling in predicting extreme values and AGCRN performing better for average values.



(a) Actuals vs AGCRN predictions

(b) Actauls vs CLCRN predictions

(c) Zoomed into section A

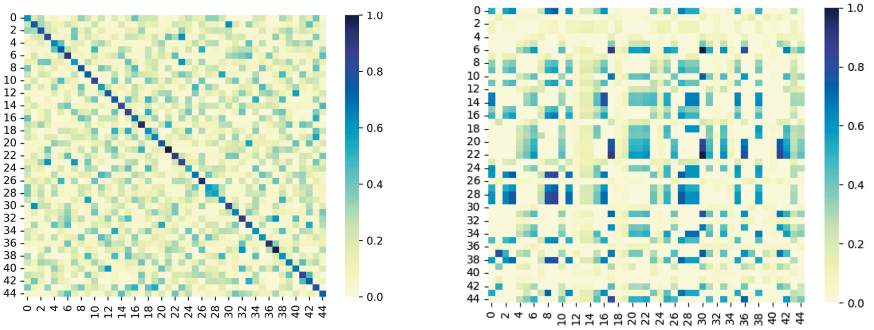
(d) Zoomed into section B

Fig. 3. The actuals and predicted values for AGCRN and CLCRN models at station 13.

4.3 Dependency Analysis

In this section, we analyse the adjacency matrices generated by each model for predicting temperature over a 24-h horizon. It is important to note that AGCRN's matrix does not rely on any pre-existing knowledge for initialization, as it starts with random initialization. Consequently, any observed patterns in the final matrix of this model are solely the result of the model's self-learning processes. In contrast, CLCRN's matrix is initialised by leveraging the K-nearest neighbours algorithm, based on the induced spherical distance calculated from the spatial locations of the weather stations.

First, Fig. 4 represents heat maps depicting the matrices produced by each model, which illustrates the significant differences in how each model utilises and learns from the matrices.



(a) AGCRN (b) CLCRN

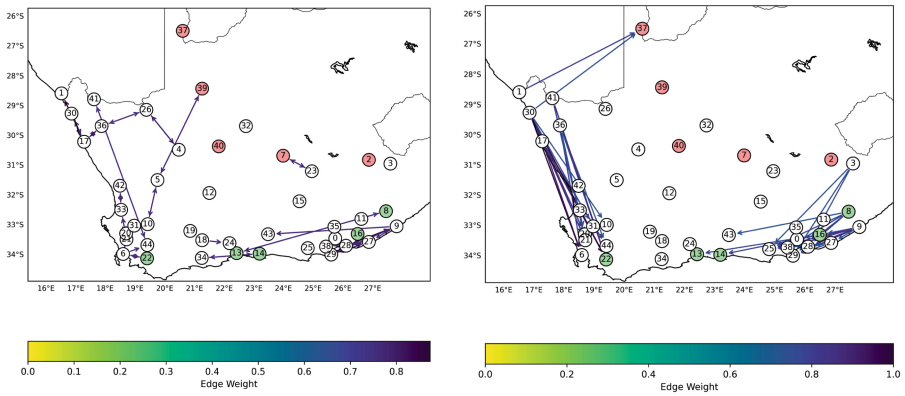
Fig. 4. Heatmaps of the adjacency matrices for 24-h temperature predictions.

The heatmap for the AGCRN model shows a clear emphasis on the diagonal, indicating that AGCRN prioritises each station's own past information when predicting their future values. Despite this self-emphasis, there are still notable values off the diagonal, distributed relatively evenly across the matrix. This suggests that AGCRN does not assign overriding importance to any single station, as no specific row or column in the heatmap stands out significantly darker than the others. In contrast, the heatmap for CLCRN displays a distinct pattern with a higher prevalence of influential stations and/or stations that are more dependent on others, evident from columns that are noticeably more dominant. Additionally, there is a marked reduction in self-referential dependencies for individual stations compared to AGCRN. This indicates that CLCRN's modelling approach places greater emphasis on inter-station relationships.

Next, in Fig. 5, we plotted the strongest edges from each matrix onto a map to understand the spatial relationships between the most prominent dependencies. An edge from one station to another indicates that the station receiving the arrow is dependent on the station from which the arrow originates. We excluded

the diagonal (self-referencing) edges and calculated the 98th percentile of the strongest edges to identify the top 40 edges for each model. The 98th percentile for AGCRN was 0.7592, while for CLCRN, it was 0.7039. The comparative analysis of CLCRN and AGCRN reveals significant differences in their dependency structures.

For AGCRN, the strongest dependencies predominantly point to nearby stations, highlighting that a station primarily influences its immediate neighbours. This is particularly noteworthy since station location data was not provided to the model, suggesting that AGCRN learned about its nearest nodes independently. This observation aligns with AGCRN's architecture, where the NAPL module provides unique parameter spaces for each node, allowing it to capture local patterns more effectively rather than network-wide effects. The presence of bidirectional edges is also evident, indicating that two stations exert mutual influence on each other.



(a) AGCRN (b) CLCRN

Fig. 5. Strongest dependencies maps drawn from the adjacency matrices for AGCRN and CLCRN models for 24-h temperature predictions.

Additionally, strong dependencies are minimal inland, with most concentrated along the coast. In contrast, CLCRN shows a more distinct pattern in its strongest dependencies. Similar to AGCRN, CLCRN exhibits fewer strong dependencies inland. However, CLCRN's strong dependencies along the coasts are more pronounced and follow a consistent pattern, extending southward along both the eastern and western coasts.

We then investigate whether stations with strong dependencies exhibit a direct correlation with improved prediction accuracy. To do this, we compare our model to the purely temporal TCN, which has no dependencies. From Fig. 2(b), we identify the top 5 stations with the greatest improvements over TCN (stations 13, 22, 14, 16, and 8) and the 5 stations with the least improvements (stations 37, 39, 40, 2, and 7). Accordingly, in Fig. 5, we have shaded the top 5

stations green and the bottom 5 stations red. As previously noted, STGNN models perform particularly well compared to the TCN model along coastal areas. Interestingly, the stations with the greatest improvements (green) are located along the coast, where dependencies are more prominent. In contrast, more inland areas, where strong dependencies are less prevalent, show less improvement over TCN, with more red stations. For both CLCRN and AGCRN, four out of the five top-improved stations have incoming edges. In contrast, four out of five of the least improved stations for CLCRN have no incoming edges. This suggests that stations receiving strong influential dependencies demonstrate the best improvement and utilization of the STGNN architecture compared to the purely temporal TCN. However, it is worth noting that station 37, which was the station with the least improvement over TCN, did not align with this general pattern as it has two strong incoming dependencies. Therefore, we should be aware that not all dependencies can be said to be true dependencies, and some occurrences may deviate from the pattern.

4.4 Path Analysis

Figure 6 shows the paths found by CLCRN. We lowered the threshold for edge weight to 0.25, and only show paths or chains of at least 3 nodes, i.e. a source node must influence at least one node via another node to be included. This analysis yields some interesting insights into the dynamics of the system.

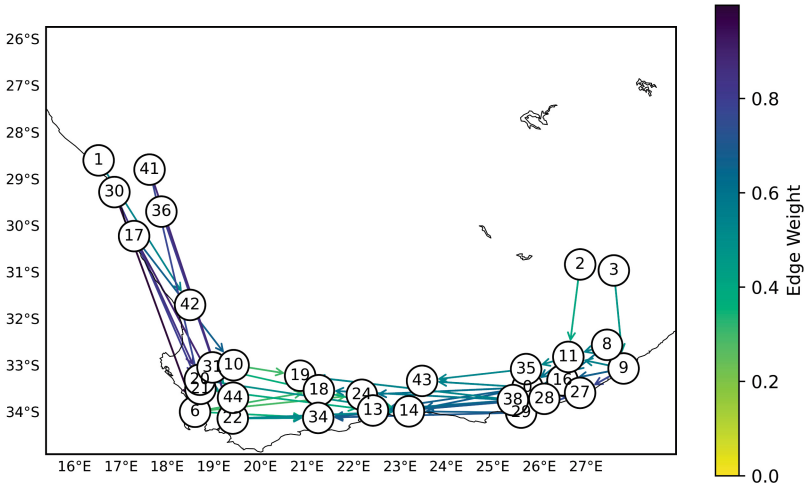


Fig. 6. Visualising Spatial-Temporal Dependency Paths in CLCRN Model for 24-h Temperature Prediction.

Easternmost Stations' Influence. In terms of the most prominent paths, Fig. 6 shows that there is a strong path of influence originating from a cluster of four stations 2, 3, 8 and 9 in the east.

These stations influence stations to the south and west along a path to a cluster of five stations (35, 0, 38, 28 and 29) on the south east coast. These stations in turn influence their western neighbours stations 13, 14, and 24, which in turn influence stations 18, 19, and 34. Stations 18, 19 and 34 appear to fall at the centre of the dependency network. This sequence illustrates a clear path of spatial-temporal dependencies, from the eastern stations towards and along the southern coast to the centre of the Western Cape.

Northernmost Stations' Influence. A similar prominent path along the western coast emerges originating from this cluster of northern coastal stations (1, 30, 41, 17 and 36). They exert a strong influence in a southern path along the coast through station 42 to a cluster of seven stations in the south west. The path of dependencies traced from stations 1, 30, 41, 36, and 17, extending down to 31, 20, 21, 22, 6, 10, 44, and finally to 19, 18, and 34, reveals a significant influence pathway. This dependency chain indicates that the climatic conditions at the areas where these five northernmost coastal stations are located plays a crucial role in influencing climatic conditions along the coast.

Convergence of Paths. A noteworthy observation is the convergence of these two critical paths at stations 18, 19, and 34, where the influence from the eastern and north west stations converge. This convergence underscores the central role these stations play in the network, acting as key nodes where the influence from both the easternmost and northernmost stations interconnects. A more detailed analysis of the results suggests that this pattern may be linked to the geographical terrain of the study area. Further research corroborates this hypothesis, pointing to the Great Escarpment as a potential contributing factor [3]. This aligns with a previous study [8] that highlighted distinct weather patterns in the coastal regions and Cape Fold mountains compared to the central regions of South Africa. Further substantiating this hypothesis, our analysis of Fig. 4 and comprehensive examination of the per station dependencies reveal a distinct characteristic among Stations 4, 12, and 40, which are positioned centrally within the spatial map of our data. These stations display minimal influence on other stations and limited dependence on their neighbouring counterparts. Such observations could be attributed to the relative scarcity of nearby weather stations or specific geographical factors unique to these regions.

5 Conclusions

In this study we evaluated two recently developed Spatial-Temporal Graph Neural Network (STGNN) techniques, i.e. the Adaptive Graph Convolutional Recurrent Network (AGCRN) and the Convolutional Long Short-Term Memory Graph

Neural Network (CLCRN) for weather prediction and knowledge discovery in South Africa. This work builds on and expands on the study done by Davidson and Moodley [4]. Their study showed that GraphWaveNet achieved the best prediction performance for temperature prediction on 21 weather stations in South Africa.

Our study used a much larger dataset (45 stations) and also explored STGNNs for predicting three other weather attributes, i.e. pressure, humidity and wind speed, in addition to temperature. We found that the newer STGNNs, AGCRN and CLCRN, outperformed GraphWaveNet, with AGCRN achieving the best overall performance. Moreover, we also explored the efficacy of STGNNs for scientific knowledge discovery. We used the TCN performance as a baseline to identify the effect of adding spatial information, which allowed us to further analyse the dependency graph learnt by the STGNNs. Through visual analysis of the dependency graph learnt by CLCRN and systematic path analysis we were able to discover complex spatial temporal dependency chains which drive the dynamics of the weather system in South Africa.

It must be noted that the approach does not require extensive prior domain knowledge. The adjacency matrix for AGCRN was initalised randomly, while the adjacency matrix for CLCRN was initialised using the induced spherical distance calculated from the spatial locations and leveraging the K-nearest neighbours algorithm (stations). The STGNNs adjust the adjacency matrix during training, so the dependency graphs are automatically learnt from the observational data and the techniques do not require prior domain knowledge.

We believe that this result will have significant impact on researchers in climate science. Not only do STGNNs achieve superior predictive performance, but they can also be used to discover complex weather patterns based purely on observational data. While the study focused on weather data, the approach has broader applicability and can potentially be applied for automatic discovery and explanation of spatial-temporal patterns in other complex and dynamic systems.

6 Limitations and Future Work

This research uncovers multiple avenues for further exploration. Firstly to explore how the prominent dependencies and paths shown in Figs. 5 and 6 align with known theories about the South African weather system. Since mountains are known to have a substantial effect on weather patterns, one immediate avenue for further visual analysis is to overlay these results on a topological map to understand the influence of mountains on the paths.

Another avenue for future exploration could focus on implementing the approach on a global-scale dataset, like WeatherBench [14], to extract dependencies and paths. This would allow for a more comprehensive evaluation of the approach to discover global weather patterns.

Lastly, given that weather predictions often require longer prediction horizons, future work could investigate the applicability of the AGCRN and CLCRN for long term weather prediction beyond 24 h.

Acknowledgments. This work is based on the research supported in part by the National Research Foundation of South Africa (Grant Numbers: 151217). In addition, we acknowledge the South African Weather Service who provided us with the dataset used in this research. Finally, we are grateful for the feedback from the anonymous reviewers that helped improve our manuscript.

References

1. Bai, L., Yao, L., Li, C., Wang, X., Wang, C.: Adaptive graph convolutional recurrent network for traffic forecasting. Adv. Neural. Inf. Process. Syst. **33**(7), 17804–17815 (2020)
2. Cao, D., et al.: Spectral temporal graph neural network for multivariate time-series forecasting. In: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (eds.) Neural Information Processing Systems. Annual Conference. 31st 2017. (10 VOLS) Advances In Neural Information Processing Systems 30, vol. 33, pp. 17766–17778. Curran Associates, Inc., Long Beach, California, USA (2020)
3. Clark, V.R., Barker, N.P., Mucina, L.: The great escarpment of southern Africa: a new frontier for biodiversity exploration. Biodivers. Conserv. **20**, 2543–2561 (2011)
4. Davidson, M., Moodley, D.: ST-GNNs for weather prediction in South Africa. In: Pillay, A., Jembere, E., Gerber, A. (eds.) SACAIR 2022. CCIS, vol. 1734, pp. 93–107. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-22321-1_7
5. García-Duarte Sáenz, L., Cifuentes Quintero, J., Marulanda, G.: Short-term spatio-temporal forecasting of air temperatures using deep graph convolutional neural networks. Stoch. Env. Res. Risk Assess. **37**, 1649–1667 (2021)
6. Hewage, P., Trovati, M., Pereira, E., Behera, A.: Deep learning-based effective fine-grained weather forecasting model. Pattern Anal. Appl. **24**(1), 343–366 (2021)
7. Jin, G., et al.: Spatio-temporal graph neural networks for predictive learning in urban computing: a survey. IEEE Trans. Knowl. Data Eng. **36**, 5388–5408 (2023)
8. Kruger, A.C., Shongwe, S.: Temperature trends in South Africa: 1960–2003. Int. J. Climatol. **24**(15), 1929–1945 (2004)
9. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: data-driven traffic forecasting. https://arxiv.org/abs/1609.03499. Accessed 10 Sept 2023
10. Lin, H., Gao, Z., Xu, Y., Wu, L., Li, L., Li, S.Z.: Conditional local convolution for spatio-temporal meteorological forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 7, pp. 7470–7478 (2022)
11. Ma, M., et al.: Histgnn: hierarchical spatio-temporal graph neural network for weather forecasting. Inf. Sci. **648**, 119580 (2023)
12. van den Oord, A., et al.: Wavenet: a generative model for raw audio. https://arxiv.org/abs/1609.03499. Accessed 21 Mar 2024
13. Pillay, K., Moodley, D.: Exploring graph neural networks for stock market prediction on the JSE. In: Jembere, E., Gerber, A.J., Viriri, S., Pillay, A. (eds.) SACAIR 2021. CCIS, vol. 1551, pp. 95–110. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-95070-5_7
14. Rasp, S., Dueben, P.D., Scher, S., Weyn, J.A., Mouatadid, S., Thuerey, N.: Weatherbench: a benchmark data set for data-driven weather forecasting. J. Adv. Model. Earth Syst. **12**(11), e2020MS002203 (2020)
15. Simeunović, J., Schubnel, B., Alet, P.J., Carrillo, R.E.: Spatio-temporal graph neural networks for multi-site PV power forecasting. IEEE Transactions on Sustainable Energy **13**(2), 1210–1220 (2021)

16. Sree Dhevi, A.: Imputing missing values using inverse distance weighted interpolation for time series data. In: 2014 Sixth International Conference on Advanced Computing (ICoAC), Chennai, India, pp. 255–259 (2014)
17. Wilson, T., Tan, P.N., Luo, L.: A low rank weighted graph convolutional approach to weather prediction. In: 2018 IEEE International Conference on Data Mining (ICDM), Singapore, pp. 627–636. IEEE (2018). https://doi.org/10.1109/ICDM.2018.00078
18. Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C.: Graph wavenet for deep spatial-temporal graph modeling. https://arxiv.org/abs/1906.00121. Accessed 09 Jan 2023
19. Zhao, L., et al.: T-GCN: a temporal graph convolutional network for traffic prediction. IEEE Trans. Intell. Transp. Syst. **21**(9), 3848–3858 (2020)

Benchmarking Political Bias Classification with In-Context Learning: Insights from GPT-3.5, GPT-4o, LLaMA-3, and Gemma-2

Eduan Kotzé(✉) and Burgert A. Senekal

Department of Computer Science and Informatics, University of the Free State, Bloemfontein, South Africa
kotzeje@ufs.ac.za

Abstract. ChatGPT, which provides access to the large language models GPT-3.5 and GPT-4, received wide attention worldwide after its release in 2022, with many noting the great potential this conversational agent holds, the many dangers and limitations of this technology. This study investigates the capabilities of large language models, specifically GPT-3.5, GPT-4o, GPT-4o mini, LLaMA-3 and Gemma-2 for classifying political bias (left, centre, and right) using a specialised dataset. The models were evaluated for their performance in text classification, focusing on accuracy, precision, recall, and F1 scores. The results indicate that GPT-4o outperforms GPT-4o mini, GPT-3.5 Turbo, LLaMA-3, and Gemma-2 for in-context learning. GPT-4o also surpassed the TF-IDF word bi-gram SVM baseline for this text classification task; however, none of the other models achieved better results, as they frequently misclassified biased sentences as unbiased. Despite its strengths, GPT-4o still exhibits some limitations. Overall, the findings suggest that GPT-4o is a promising tool for in-context learning in detecting political bias, as it generally assigns the correct classifications.

Keywords: GPT-3.5 Turbo · GPT-4o · LLaMA-3 · Gemma-2 · political bias · large language models · in-context learning · text classification

1 Introduction

ChatGPT, a conversational agent that provides access to the GPT-3.5 and GPT-4 large language models (LLMs), has received a lot of attention in the media and academia since its release in November 2022, with some highlighting dangers such as plagiarism, while others emphasize the potential of this conversational agent. However, it is important to distinguish between potential and actual ability, which is why ChatGPT has already been tested in a variety of studies with different tasks. One promising application of ChatGPT is in text classification, where ChatGPT' ability to perform multiple text classification tasks can make it a versatile tool that can address a variety of Natural Language Processing (NLP) challenges.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 161–175, 2025.
https://doi.org/10.1007/978-3-031-78255-8_10

The widespread use of ChatGPT, or the GPT-3.5 language model underlying ChatGPT, requires an evaluation of its capacity and limitations. Therefore, ChatGPT has already been tested for text classification tasks such as sentiment analysis, emotion recognition, genre recognition and stance recognition [1, 2]. The current study joins this research by investigating the ability of the GPT-3.5, GPT-4o, GPT-4o-mini, LLaMA-3 and Gemma-2 language models to perform political bias classification. For this, all five language models are compared to the Spinde et al. [3] dataset of political bias.

The article is structured as follows. First, an overview of previous research related to political bias classification using machine learning, as well as previous research regarding large language models' ability to perform text classification tasks are provided. The methods followed in the current study are discussed next, after which the results are presented and discussed. The article draws to a close with concluding remarks and suggestions for further research.

2 Background

Political bias refers to a text's political leaning or ideology, usually classified as left, centre or right, which can influence the reader's political opinion and, ultimately, their voting behaviour [4]. Left-wing ideals include liberal ideas such as individual liberty, equality, fraternity, human rights, progress, reform and internationalism, while right-wing ideals include conservative ideas such as tradition, authority, hierarchy, order, duty and nationalism [5].

Because political bias is such an important issue regarding trust in the media and within today's information sphere, political bias classification using machine learning has already been undertaken in various studies, to the extent that it has become a well-established field in NLP. Previous studies have investigated various classification methods for political bias classification, including BERT, support vector machines (SVM), RoBERTa, various deep learning models, and Transformers [4, 6–12]. Raza, Reji and Ding [13] also developed Dbias, an open source Python package that detects biased words in news articles and suggests bias-free or less biased alternatives.

OpenAI, a company founded by Elon Musk, Sam Altman, and others in 2015, released ChatGPT in November 2022 [14]. As a conversational agent, ChatGPT was developed from the Generative Pre-trained Transformer-3 (GPT-3) model developed by Brown et al. [15], which is an improvement on the GPT model proposed by Radford et al. [16]. The model by Radford et al. [16] is based on the Transformer model developed by Vaswani et al. [17]. ChatGPT was further refined with reinforcement learning from human feedback, as described by Ouyang et al. [18], and the model itself has been improved to the GPT-3.5 model. GPT-1 was trained with 117 million parameters, GPT-2 with 1.5 billion parameters, and GPT-3 with 175 billion parameters. However, GPT-3.5 is a smaller version of GPT-3, and was trained with 6.7 billion parameters [14]. GPT-4, released in 2023, builds upon the foundation of its predecessors with even more advanced capabilities, although it is not employed in all responses generated by ChatGPT [19].

GPT-3.5's general knowledge has already been investigated in various contexts [20, 21]. Several studies have also investigated ChatGPT's ability to perform text classifications, often by using the GPT-3.5 model. Belal, She and Wong [2] investigated the

use of ChatGPT for labelling data in sentiment analysis tasks. They demonstrated the superior performance of ChatGPT compared to lexicon-based algorithms, as well as its potential for labelling in various sentiment analysis tasks. Li et al. [22] used ChatGPT to detect harmful content on social media and compared its performance with human tagging. They found that ChatGPT achieved around 80% accuracy and exhibited consistent classifications for non-harmful comments. Kuzman et al. [23], in turn, compared the performance of ChatGPT and a fine-tuned multilingual XLM-RoBERTa model in automatic genre recognition, and found that ChatGPT outperformed the fine-tuned model on unseen English and Slovenian test datasets.

In contrast with the promising findings mentioned above, several studies point out ChatGPT's limitations. Reiss [24] for example, tested the consistency of ChatGPT's ability for text labelling and classification, and found that even slight wording changes of inputs to ChatGPT can lead to different outputs, necessitating caution when using ChatGPT for these tasks. In a study titled, *ChatGPT: Jack of all trades, master of none*, Kocoń et al. [1] examined the ability of ChatGPT in relation to 25 different NLP tasks, most of them subjective, for example sentiment analysis, emotion recognition, offensiveness recognition and attitude recognition. They showed that ChatGPT in the form of the GPT-3.5 model performed 25% worse on average for text classification purposes compared to the best text classification methods. Although the GPT-4 model outperformed the GPT-3.5 model, it still performed worse than the best text classification models. However, they noted that the best text classification methods can only perform a single task, while ChatGPT can perform multiple tasks.

While our primary focus is the GPT-3.5 and GPT-4 language models by OpenAI, we also wanted to compare our findings with language models released by other companies such as Meta and Google. These companies are currently at the forefront of LLM development, and although OpenAI's ChatGPT has achieved widespread adoption, we did not want to limit our findings to a model released by a single company. In April 2024, Meta introduced the LLaMA-3 model, offering 8 billion and 70 billion parameter configurations [25]. These models were pretrained on 15 trillion data tokens. We evaluated the 4-bit quantization LLaMA-3-8B-Instruct model. Soon after, Google introduced the Gemma-2 model in June 2024, offering 9 billion and 27 billion parameter configurations [26]. These models were trained on a dataset of text data that includes a wide variety of sources. The 27B model was trained with 13 trillion tokens and the 9B model was trained with 8 trillion tokens. We evaluated the 4-bit quantization Gemma2-9B-Instruct model. OpenAI released GPT-4o-mini ("o" for "omni") in July 2024. According to the OpenAI website, the GPT-4o-mini is their most advanced model in the small models category, and the model is cheaper and more capable than GPT3.5 Turbo, but just as fast [27]. It is also multimodal, meaning it accepts text or image inputs and outputs text.

Although large language models have already been investigated for a variety of text classification tasks, such as sentiment, attitude and emotion recognition, they have not yet been tested for political bias classification according to our knowledge. The next section discusses how the language models were tested in the current study.

3 Methodology

3.1 Dataset

The dataset by Spinde et al. [3] was used in the current study as a benchmark to test the various LLM's ability to handle political bias classification. Spinde et al.'s [3] dataset consists of 1,700 sentences manually classified according to political bias. The authors used articles from three left (HuffPost, MSNBC and AlterNet), three right (The Federalist, Fox News and Breitbart) and two centre media houses (USA Today and Reuters), which were classified by 784 human annotators. From this, sentences on which most annotators agreed were extracted, and Spinde et al. [3] provided the sentence together with the label assigned to the sentence by most annotators. Spinde et al. [3] also assigned a topic to sentences, with a total of 14 topics, which were not used in the current study. Table 1 shows the distribution of labels in this dataset.

Table 1. Number of labels by Spinde et al. [3]

Label	Count
Centre	315
Left	694
Right	691
Total	**1700**

3.2 Zero-Shot, One-Shot and Few-Shot Text Classification

Zero-shot, one-shot, and few-shot in-context are strategies used in deep neural language models to generalise from limited examples. Zero-shot learning refers to the model's ability to make predictions about classes or tasks it has never seen during training, relying on the model's understanding of language and context. This approach leverages knowledge transfer from related tasks or domains to infer new concepts. One-shot learning involves training a model to recognise or categorise new classes based on only a single example. This technique often utilises advanced methods such as metric learning or Siamese networks, which learn to measure similarity between examples effectively. Few-shot learning, on the other hand, extends this concept to scenarios where only a small number of examples (more than one but still limited) are available.

To conduct one-shot and few-shot learning, we needed example sentences that reflect stances associated with a *left*, *right* and *centre* political leaning. Against the backdrop of Heywood's [5] distinction between left and right, ChatGPT was asked (via the web-based Graphical User Interface or GUI) to provide five examples of *left*, five examples of *right* and five examples of *centre* statements, which the authors evaluated for accuracy.

ChatGPT's suggestions for *left-wing statements* were the following:

1. We should implement universal healthcare to ensure that everyone, regardless of income, has access to quality medical care.
2. Increasing the minimum wage is essential to reduce income inequality and ensure that all workers can support themselves and their families.
3. Climate change is an urgent crisis that requires immediate action, including transitioning to renewable energy sources and reducing carbon emissions.
4. We need comprehensive immigration reform that provides a pathway to citizenship for undocumented immigrants and respects their human rights.
5. Public education should be fully funded and accessible to all, with measures in place to address disparities in resources and opportunities between different communities.

ChatGPT's suggestions for *right-wing statements* were the following:

1. Lowering taxes on businesses and individuals will stimulate economic growth and create more jobs.
2. The Second Amendment rights should be protected, and law-abiding citizens should have the freedom to own and carry firearms.
3. Strict immigration policies and stronger border security are necessary to protect national sovereignty and ensure public safety.
4. Government regulations on businesses should be minimised to encourage innovation, entrepreneurship, and economic freedom.
5. Traditional family values and cultural heritage should be preserved and promoted as the foundation of a stable and prosperous society.

Finally, ChatGPT's suggestions for *centre statements* were the following:

1. While it is important to support free markets and encourage entrepreneurship, we must also ensure that there are safety nets in place to protect the most vulnerable members of our society.
2. We should aim to provide universal healthcare coverage while also allowing for private options to ensure that everyone has access to the care they need without compromising on choice or quality.
3. Public education should be adequately funded to provide equal opportunities for all students, but we should also encourage innovation and improvement through charter schools and other alternative educational models.
4. We need to address climate change through sustainable practices and investments in renewable energy, but we should also consider the economic impact on industries and workers, ensuring a just transition to a greener economy.
5. It is crucial to uphold and respect individual rights and freedoms, including those related to marriage and gender identity, while also maintaining a respectful dialogue and considering the diverse beliefs and values of all community members.

We followed the commonly adopted N-way K-shot setting, where N is the number of classes and K is the number of examples per class. An example of a 3-way 5-shot scenario is given below (Table 2). We used the first sentence of the above-provided examples for the 3-way 1-shot text classification and no examples for 3-way zero-shot text classification.

Table 2. Example of a *3*-way *5*-shot text classification. We omit instances for class 2–3 for simplicity. The given query belongs to Class 3: RIGHT

Support Set	
Class 1: *CENTER*	**Instance 1**: *While it is important to support free markets and encourage entrepreneurship, we must also ensure that there are safety nets in place to protect the most vulnerable members of our society* **Instance 2**: *We should aim to provide universal healthcare coverage while also allowing for private options to ensure that everyone has access to the care they need without compromising on choice or quality* **Instance 3**: *Public education should be adequately funded to provide equal opportunities for all students, but we should also encourage innovation and improvement through charter schools and other alternative educational models* **Instance 4**: *We need to address climate change through sustainable practices and investments in renewable energy, but we should also consider the economic impact on industries and workers, ensuring a just transition to a greener economy* **Instance 5**: *It is crucial to uphold and respect individual rights and freedoms, including those related to marriage and gender identity, while also maintaining a respectful dialogue and considering the diverse beliefs and values of all community members*
Class 2: *LEFT*	…
Class 3: *RIGHT*	…
Query Instance	
"Lowering taxes on businesses and individuals will stimulate economic growth and create more jobs."	

3.3 Performance Metrics

For a more detailed calculation of the effectiveness of the relevant language models for text classification, we used *accuracy*, *precision*, *recall* and *F1-scores*. Accuracy measures the effectiveness of a model by calculating the ratio between the number of correct predictions and the total number of predictions made [28]. Precision refers to the ratio of the correctly identified positive classifications to the total number of positive classifications [28]. In other words, precision considers false positive classifications, which refers to classifications that indicate a particular label when, in fact, the text should not receive the label. Recall refers to the ratio of the actual positive classifications that are correctly identified versus the positive classifications that are not identified [28]. In other words, recall considers false negatives, which is a term that refers to situations where a label was not assigned when it should have been indicated. The F1 score is a metric that measures the balance between precision and recall in machine learning.

Given that we were utilising a multi-class classifier (i.e., three labels), we must consider performance metrics that are derived from the generalisation of the binary metrics through the application of weighted, macro, or micro-averaging. In micro-averaging,

all the predictions and true labels for every class are considered when computing the metrics globally. This indicates that each individual data point, regardless of the class to which it belongs, has the same weight. Macro-averaging involves computing the metrics for each class and averaging them over all classes. All classes receive the same weight, irrespective of the quantity of samples in each class. Lastly, weighted averaging involves computing the metrics for each class and averaging them across all classes, weighted by the total number of samples in each class. As a result, the metrics calculated on larger classes are assigned greater weight, making them more significant in the evaluation.

We chose *macro-averaging* because it treats all classes equally, allowing us to evaluate the model's performance on each class separately without being impacted by imbalances in the classes. In this investigation, the measurements in the macro-averaging were represented by an *m* index. The next section provides the results of the current study.

4 Experiments

4.1 Prompt Engineering

The purpose of a prompt is to bridge the gap between the pre-train and downstream tasks performed on a pre-trained large language model. The prompt template that we used for the zero, one-shot and few-shot classification is listed below.

```
prompt = f"""
    Please label the query statement as 'right', 'left', or 'center' based on typical political
    ideology.  Only return 'right', 'left' or 'center'.

    Class 1: center
            Instance 1: While it is important to support free markets and encourage en-
            trepreneurship, we must also ensure that there are safety nets in place to
            protect the most vulnerable members of our society.
            Instance 2: …
    Class 2: left
            Instance 1: We should implement universal healthcare to ensure that every-
            one, regardless of income, has access to quality medical care.
            Instance 2: …
    Class 3: right
            Instance 1: Lowering taxes on businesses and individuals will stimulate eco-
            nomic growth and create more jobs.
            Instance 2: …

    Query: {text}
    Label:
    """
```

4.2 In-Context Learning

For the current study, queries were directed (classified by us as in-context zero-shot, one-shot, and few-shot) to OpenAI's API, with the task of assigning a label (left, right, or centre) to the sentence (using the prompt). We used GPT-3.5 Turbo, GPT-4o, and GPT-4o-mini for the OpenAI models. Thereafter, queries were directed to the LLaMA-3-8B and Gemma-2-9B models for in-context zero-shot, one-shot and few-shot text classifications. Since both these instruction-tuned models are fine-tuned with instruction datasets, we re-used the same prompt. All experiments on the two models were performed using the open-source libraries provided by Huggingface [29]. Table 3 provides information about the model version, data freshness and the cost of each model.

Table 3. Summary of classification models.

Model	Version	Data Freshness	Usage: Cost
GPT-3.5 Turbo	gpt-3.5-turbo-0125	Up to Sep 2021	$0.74
GPT-4o	gpt-4o-2024-05-13	Up to Oct 2023	$7.39
GPT-4o-mini	gpt-4o-mini-2024-07-18	Up to Oct 2023	$0.22
LLaMA-3	llama-3-8b-instruct	March 2023 for the 8B model	N/A
Gemma-2	gemma-2-9b-instruct	*Not specified	N/A

When comparing the costs and data freshness of different GPT models, it is crucial to weigh the trade-offs between performance and expense. GPT-4, for instance, is significantly more expensive than earlier versions. In our experiment, GPT-4o costs $7.39 per million tokens, whereas GPT-4o-mini costs only $0.22 per million tokens, making GPT-4o approximately 33 times more expensive. This cost disparity is primarily driven by the increased computational requirements of GPT-4o, which necessitates more inference computing resources. Thus, while the model's advanced capabilities contribute to its value, the financial investment must be carefully considered, leading us to test both models to evaluate their relative worth.

4.3 Feature Engineering and Machine Learning

In addition to the large language models, we also used a Support Vector Machine (SVM) classifier as a baseline model. We used Python's scikit-learn library by Pedregosa et al. [30] for the training and testing of the SVM model. For feature engineering, we used the term frequency-inverse document frequency (TF-IDF) feature extraction technique to create the vector representation. The TF-IDF features were based on a contiguous sequence of character or word n-grams. The character n-grams features ranged from 1-g to 4-g and the word n-gram features from unigrams (1-g) to bigrams (2-g).

The dataset was split into an 80% training dataset and a 20% testing dataset before model training was performed. Model training and hyperparameter optimisation were performed using 10-fold cross-validation in an exhaustive grid search over two possible hyperparameter configurations (TF-IDF word features and TF-IDF character features).

The grid search concentrated on optimising the values for the *C* and *tol* hyperparameters. *C* is a regularisation parameter, where the strength of the regularisation is inversely proportional to *C*. *tol* is the tolerance stopping criterion and is essential during optimisation to balance the trade-off between time and accuracy. Table 4 shows the hyperparameter combinations for model selection.

Table 4. Hyperparameters for SVM baseline model

Model	Support Vector Machine		
	Parameter	Values	Optimal
SVM with TF-IDF word n-grams (1–2)	*C*	$1e^{\{-3, -2, -1, 0, 1, 2\}}$	10
	tol	$1e\{-3, -2, -1, 0, 1\}$	0.001
SVM with TF-IDF char n-grams (1–4)	*C*	$1e\{-3, -2, -1, 0, 1, 2\}$	1
	tol	$1e\{-3, -2, -1, 0, 1\}$	0.1

All experiments were performed on a desktop computer with a NVidia 24GB RTX-4090 GPU.

5 Results

Table 5 shows the evaluation results of classification models, compared to those found in the dataset from Spinde et al. [3], in more detail.

Table 5. Test results of the classification models.

Model	Test Results				
		Acc	P_m	R_m	$F1_m$
Baseline (SVM)	TF-IDF	0.632	0.645	0.603	0.613
Model					
GPT-3.5-turbo	zero-shot	0.342	0.331	0.334	0.312
GPT-3.5-turbo	one-shot	0.327	0.324	0.331	0.310
GPT-3.5-turbo	few-shot	0.350	0.332	0.338	0.321
Model					
GPT-4o-mini	zero-shot	0.480	0.530	0.506	0.454
GPT-4o-mini	one-shot	0.443	0.515	0.494	0.429
GPT-4o-mini	few-shot	0.462	0.530	0.510	0.460

(*continued*)

Table 5. (*continued*)

Model	Test Results				
		Acc	P_m	R_m	$F1_m$
Model					
GPT-4o	zero-shot	**0.637**	**0.632**	**0.629**	**0.616**
GPT-4o	one-shot	0.587	0.610	0.600	0.576
GPT-4o	few-shot	0.593	0.590	0.594	0.576
Model					
LLaMA-3-8b	zero-shot	0.549	0.525	0.529	0.523
LLaMA-3-8b	one-shot	0.480	0.489	0.504	0.467
LLaMA-3-8b	few-shot	0.492	0.469	0.440	0.427
Model					
Gemma-2-7b	zero-shot	0.612	0.580	0.570	0.571
Gemma-2-7b	one-shot	0.569	0.568	0.573	0.555
Gemma-2-7b	few-shot	0.573	0.546	0.550	0.546

The best results are printed in **bold**.

This table shows that both LLaMA-3 and Gemma-2 performance degraded with examples and the zero-shot performed with $F1_m$ scores of 0.523 and 0.571. GPT-3.5 Turbo performed the worst of all the OpenAI models with an $F1_m$ score of 0.321, while GPT-4o-mini fared better with an $F1_m$ score of 0.460. Both these models' performance increased with 3-way 5-shot text classification prompts. GPT-4o performed the best with an $F1_m$ score of 0.616, surpassing the baseline SVM model with zero-shot classification. Its performance degraded with one-shot and few-shot classification ($F1_m = 0.576$ for both). This decline in performance could be attributed to the model overfitting to the limited examples provided, or it may stem from the examples being unrepresentative of the broader dataset used to train GPT-4o.

Figure 2 shows the confusion matrix of the best performing GPT-4o and GPT-3.5 Turbo models. GPT-4o was reliable in classifying both *left*- and *right*-wing labels but struggled in classifying the *centre* label. It misclassified (n = 175) *left*-wing labels as *centre* and (n = 148) *right*-wing labels as *centre*. From the GPT 3–5 Turbo confusion matrix, it is clear that the model classified fewer sentences as *centre* or *right* as the original dataset, and had a much larger predicted number of labels (n = 896) as *left*-wing.

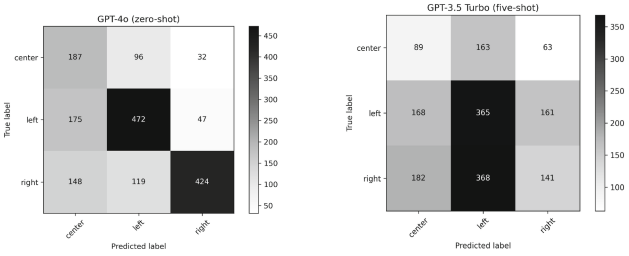


Fig. 2. Confusion Matrix of GPT-4o and GPT-3.5 Turbo

Figure 3 shows the confusion matrix of the best-performing LLaMA-3 and Gemma-2 models. The models showed substantial performance differences to the best-performing GPT-4o model (when comparing the $F1_m$ -scores). Gemma-2 misclassified more *centre* (n = 141) and *right* (n = 191) sentences as *left*, and LLaMA-3 showed similar results with more *right* (n = 147) sentences as *left* and more *centre* (n = 113) as *right*. Also, LLaMA-3 misclassified (n = 205) *left* sentences as *right*.

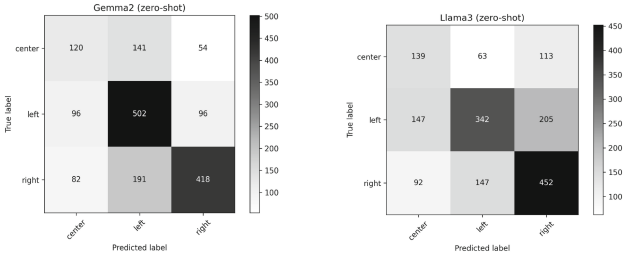


Fig. 3. Confusion Matrix of Gemma-2 and LLaMA-3

6 Discussion

The above experiments suggest that GPT-4o generally outperforms GPT-3.5 Turbo for political bias classification, with better accuracy, precision, recall, and F1 scores. Nevertheless, both models committed errors, mainly by classifying biased statements as *neutral*. GPT-3.5 takes a more cautious approach and is more likely to classify texts as *neutral.* It also does better at correctly classifying text with a moderate or balanced viewpoint.

Although the $F1_m$ score of 0.616 for GPT-4o is low, Wessel et al. [4] emphasised that text classification related to political bias is difficult and subjective, and in their study, models such as GPT-2, ELEKTRA and RoBERTa also achieved low F1 scores of between 0.6 and 0.7. Fagni and Cresci [10] achieved F1 scores of 0.43 and 0.77 with different methods, and the authors showed that other unsupervised techniques achieved $F1 \leq 0.35$ and $F1 \leq 0.71$. Raza, Reji and Ding [13] also achieved F1 scores of between 0.61 and 0.75 with various models. Other studies have achieved better results with text

classification for political bias, for example, Agrawal et al. [31], who achieved an F1 score of 0.76, and Ullah et al. [9], who achieved F1 scores between 0.87 and 0.93. Nevertheless, the current study's F1 score of 0.6 compares with that of Wessel et al. [4], Raza, Reji and Ding [13] and Fagni and Cresci [10], although GPT-4's classifications performed somewhat worse (but better than the classifier in Aksenov et al. [11]).

Based on the results, it can be suggested that while GPT-4o achieved an $F1_m$ score of 0.616 for the classification task, this performance is comparable to the baseline SVM model's $F1_m$ score of 0.613. However, the significant costs of running GPT-4o—such as time, financial expense, and environmental impact—should be carefully considered. In many cases, the traditional SVM model may offer a more efficient and responsible alternative for this task. Therefore, while GPT-4o shows promise, particularly in the classification of less polarised texts, there is still room for improvement in its performance, and its use must be weighed against the practicality and resource requirements of other methods.

An important limitation of the current study is that the comparison of performance across different large language models (LLMs) is potentially confounded by the substantial size difference between GPT-4o and the other models (GPT-3.5, GPT-4o mini, LLaMA-3, Gemma-2). This disparity in scale may provide GPT-4o with an inherent advantage, making it difficult to isolate the impact of model architecture or training data on the observed performance differences.

7 Conclusion

ChatGPT has much potential but cannot be used for tasks without thorough testing. The current study joined previous research investigating the language models underlying ChatGPT, namely GPT-3.5 Turbo, GPT-4o, and GPT-4o-mini for text classification purposes by testing the language models' ability to perform political bias classifications.

It was found that although GPT-4o outperformed both GPT-3.5 Turbo and GPT-4o-mini, all three models had difficulty with this task. In other words, if ChatGPT is used for this task, it is preferable to use the GPT-4o model rather than the GPT-3.5 Turbo or GPT-4o-mini models, although the shortcomings of the GPT-4o model must still be considered.

Misclassification of political sentiment by LLMs, such as GPT-4o and GPT-3.5 Turbo, can have detrimental effects on individuals and society. As indicated by the findings that both models often classify biased statements as *neutral*, this misclassification could amplify existing biases and deepen political divides by distorting users' perceptions of different viewpoints. The relatively low $F1_m$ scores (i.e. GTP4o with 0.616), highlight the challenges inherent in accurately classifying political bias, which could erode trust in media and institutions. This underscores the importance of regulations promoting transparency, accuracy standards, and public education to mitigate these risks and ensure responsible use of LLMs in political contexts, especially given the significant resources required to operate more advanced models like GPT-4o.

In addition to political bias classification and text classification tasks that have already been investigated in previous research, such as attitude recognition, emotion recognition and sentiment classification, there are still text classification tasks that have not been

extensively explored using language models. For instance, while studies on hate speech have been conducted in various contexts, including South Africa [32, 33], these investigations did not utilise LLMs. The current study also did not examine the ability of these language models to perform text classifications with texts in other languages. Therefore, future studies that can be undertaken to assess the capabilities of language models in these areas.

Future work will include fine-tuning language models such as Gemma-2, LLaMA-3, and GPT variants on the same dataset used in this study to assess whether task-specific adaptations can improve classification performance. Comparing fine-tuned models with the baseline zero-shot results will provide deeper insights into how tuning impacts accuracy and bias detection for political sentiment classification. In addition, future work could explore the implications of model misclassifications by adopting a social stance. This approach would deepen our understanding of how such models might influence public perception and usage, particularly in sensitive domains such as political sentiment classification.

In addition, future work could investigate the impact of slight prompt variations on model performance to determine the sensitivity of each language model to changes in input phrasing during classification tasks. This work could examine fine-tuned and baseline models. The common use of ChatGPT makes it critical to evaluate its capability and limitations, as many people use this technology and if the technology is not reliable, this issue will have to be accounted for when the technology is deployed in various practical contexts.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Kocoń, J., et al.: ChatGPT: Jack of all trades, master of none. Inf. Fusion. **99**, 1–37 (2023). https://doi.org/10.1016/j.inffus.2023.101861
2. Belal, M., She, J., Wong, S.: Leveraging ChatGPT As Text Annotation Tool For Sentiment Analysis. CoRR. abs/2306.1 (2023)
3. Spinde, T., Rudnitckaia, L., Sinha, K., Hamborg, F., Gipp, B., Donnay, K.: MBIC – A Media Bias Annotation Dataset Including Annotator Characteristics. arXiv (2021). https://arxiv.org/abs/2105.11910
4. Wessel, M., Horych, T., Ruas, T., Aizawa, A., Gipp, B., Spinde, T.: Introducing MBIB - the first media bias identification benchmark task and dataset collection. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 2765–2774. ACM, New York (2023). https://doi.org/10.1145/3539618.3591882
5. Heywood, A.: Key Concepts in Politics and International Relations. Springer, London (2015). https://doi.org/10.1007/978-1-137-49477-1
6. Ness, E., Fatima, A., Oghaz, M.M.: Data driven model to investigate political bias in mainstream media. IEEE Access **11**, 41880–41893 (2023). https://doi.org/10.1109/ACCESS.2023.3270630
7. Stefanov, P., Darwish, K., Atanasov, A., Nakov, P.: Predicting the topical stance and political leaning of media using tweets. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 527–537. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.acl-main.50

8. Olteanu, A., Cernian, A., Gâgă, S.-A.: Leveraging machine learning and semi-structured information to identify political views from social media posts. Appl. Sci. **12**, 12962 (2022). https://doi.org/10.3390/app122412962
9. Ullah, H., et al.: Comparative study for machine learning classifier recommendation to predict political affiliation based on online reviews. CAAI Trans. Intell. Technol. **6**, 251–264 (2021). https://doi.org/10.1049/cit2.12046
10. Fagni, T., Cresci, S.: Fine-grained prediction of political leaning on social media with unsupervised deep learning. J. Artif. Intell. Res.Artif. Intell. Res. **73**, 633–672 (2022). https://doi.org/10.1613/jair.1.13112
11. Aksenov, D., Bourgonje, P., Zaczynska, K., Ostendorff, M., Moreno-Schneider, J., Rehm, G.: Fine-grained classification of political bias in German news: a data set and initial experiments. In: Proceedings of the 5th Workshop on Online Abuse and Harms (WOAH 2021), Stroudsburg, PA, USA, pp. 121–131. Association for Computational Linguistics (2021). https://doi.org/10.18653/v1/2021.woah-1.13
12. Lee, S., Kim, J., Kim, D., Kim, K.J., Park, E.: Computational approaches to developing the implicit media bias dataset: assessing political orientations of nonpolitical news articles. Appl. Math. Comput.Comput. **458**, 128219 (2023). https://doi.org/10.1016/j.amc.2023.128219
13. Raza, S., Reji, D.J., Ding, C.: Dbias: detecting biases and ensuring fairness in news articles. Int. J. Data Sci. Anal. **17**, 39–59 (2024). https://doi.org/10.1007/s41060-022-00359-4
14. Ray, P.P.: ChatGPT: a comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. Internet Things Cyber-Phys. Syst. **3**, 121–154 (2023). https://doi.org/10.1016/j.iotcps.2023.04.003
15. Brown, T.B., et al.: Language models are few-shot learners. Adv. Neural. Inf. Process. Syst. **33**, 1877–1901 (2020)
16. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving Language Understanding by Generative Pre-Training (2018)
17. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems (NIPS), Long Beach, CA, USA, pp. 5999–6009 (2017)
18. Ouyang, L., et al.: Training language models to follow instructions with human feedback (2022). https://arxiv.org/abs/2203.02155
19. Achiam, J., et al.: GPT-4 Technical Report (2023)
20. Sallam, M., et al.: ChatGPT output regarding compulsory vaccination and COVID-19 vaccine conspiracy: a descriptive study at the outset of a paradigm shift in online search for information. Cureus (2023). https://doi.org/10.7759/cureus.35029
21. Senekal, B., Brokensha, S.: Is ChatGPT a friend or foe in the war on misinformation? Commun. J. Commun. Stud. Africa **42**, 3–16 (2023). https://doi.org/10.36615/jcsa.v42i2.2437
22. Li, L., Fan, L., Atreja, S., Hemphill, L.: "HOT" ChatGPT: the promise of ChatGPT in detecting and discriminating hateful, offensive, and toxic comments on social media. ACM Trans. Web **18**, 1–36 (2024). https://doi.org/10.1145/3643829
23. Kuzman, T., Mozetič, I., Ljubešić, N.: ChatGPT: Beginning of an End of Manual Linguistic Data Annotation? Use Case of Automatic Genre Identification. arXiv (2023). https://arxiv.org/abs/2303.03953
24. Reiss, M.V.: Testing the Reliability of ChatGPT for Text Annotation and Classification: A Cautionary Remark. arXiv (2023). https://arxiv.org/abs/2304.11085
25. Llama Team: The Llama 3 Herd of Models. arXiv, pp. 1–92 (2024). https://doi.org/10.48550/arXiv.2407.21783
26. Gemma Team: Gemma 2: Improving Open Language Models at a Practical Size. arXiv, pp. 1–21 (2024). https://arxiv.org/abs/2408.00118
27. OpenAI: Models. https://platform.openai.com/docs/models. Accessed 14 Aug 2023

28. Lipton, Z.C., Elkan, C., Naryanaswamy, B.: Optimal thresholding of classifiers to maximize F1 measure. In: Calders, T., Esposito, F., Hüllermeier, E., Meo, R. (ed.) Machine Learning and Knowledge Discovery in Databases, pp. 225–239. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-44851-9_15
29. Wolf, T., et al.: Transformers: state-of-the-art natural language processing. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Stroudsburg, PA, USA, pp. 38–45. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.emnlp-demos.6
30. Pedregosa, F., et al.: Scikit-learn: machine learning in python. J. Mach. Learn. Res. **12**, 2825–2830 (2011). https://doi.org/10.1007/s13398-014-0173-7.2
31. Agrawal, S., Gupta, K., Gautam, D., Mamidi, R.: Towards detecting political bias in hindi news articles. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, Stroudsburg, PA, USA, pp. 239–244. Association for Computational Linguistics (2022). https://doi.org/10.18653/v1/2022.acl-srw.17
32. Oriola, O., Kotzé, E.: Exploring neural embeddings and transformers for isolation of offensive and hate speech in South African social media space. In: Gervasi, O., Murgante, B., Hendrix, E.M.T., Taniar, D., Apduhan, B.O. (eds.) Computational Science and Its Applications – ICCSA 2022, pp. 649–661. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-10522-7_44
33. Oriola, O., Kotzé, E.: Improving the detection of multilingual South African abusive language via skip-gram using joint multilevel domain adaptation. ACM Trans. Asian Low-Resour. Lang. Inf. Process. **23**, 1–28 (2024). https://doi.org/10.1145/3638759



Uncovering the ANPR Performance Gap: A Commercial Systems Analysis

Frank Zandamela(✉), Rofhiwa Seletani, Patrick Malatjie, Dumisani Kunene, Muhammad Ahmed Nana, and Teboho Sekopa

Council for Scientific and Industrial Research, Defence and Security, Pretoria 0001, South Africa
{fzandamela,rseletani,pmalatjie,dkunene,mnana,tsekopa}@csir.co.za
https://www.csir.co.za/defence-and-security

Abstract. Automatic number plate recognition (ANPR) systems play a vital role in global security and crime prevention. However, despite their importance, there is a notable lack of research assessing commercial off-the-shelf ANPR systems. Given that ANPR systems are not universally applicable due to regional variations in number plates, performance varies per region. A recent two-year study of a commercial ANPR system revealed a concerning average accuracy of 70%. This falls significantly short of the global average of 90%, compromising the reliability of the ANPR results. This study aims to address this knowledge gap by investigating factors that influence the performance of ANPR systems and comparing the effectiveness of commercial off-the-shelf solutions on South African vehicle plates. Four cloud-based ANPR solutions and an ANPR camera were assessed with a custom dataset. The custom dataset contains vehicle plates from different provinces and a government department. The study revealed that the camera viewing angle significantly affected the accuracy of the deployed ANPR system. Cloud-based ANPR APIs outperformed the ANPR camera system by 6.67%. In addition, most ANPR systems have difficulty detecting and correctly recognising number plate characters when there are multiple vehicles in the image. These findings underscore the need for region-specific optimisation and more comprehensive evaluation frameworks to enhance the reliability and accuracy of commercial ANPR systems in diverse real-world scenarios.

Keywords: Automatic number plate recognition · Automatic number plate recognition · Commercial systems · Performance evaluation. · Optical Character Recognition

1 Introduction

Automatic Number Plate Recognition (ANPR), also referred to as Automatic number Plate Recognition (ALPR), is a technology that leverages computer vision and image processing techniques to visually read and capture the number

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 176–191, 2025.
https://doi.org/10.1007/978-3-031-78255-8_11

plate information of a vehicle. Given the unique nature of each number plate to its respective vehicle, ANPR systems are widely used in applications such as law enforcement, parking management, and access control to restricted areas such as army bases [1]. ANPR solutions are available in various packages including desktop applications, mobile applications, web applications, cloud application interfaces (APIs), and network video recorders (NVRs).

The variations of number plates across different regions pose a significant challenge to developing robust ANPR systems that work globally [2]. Common variations affecting ANPR performance include size, colour, language, font, and standards [1,2]. Due to these differences and other factors such as environmental conditions, ANPR technologies are not one-size-fits-all solutions and often require optimisation for each region. A research study that evaluated the performance of commercial off-the-shelf ANPR systems on China number plates found a significant discrepancy between the observed accuracy and the accuracy reported in systems advertised [3].

The Council for Scientific and Industrial Research (CSIR) in South Africa recently deployed a commercial ANPR camera system. The deployed ANPR system is supposed to read and capture the vehicle number plates of civilians and the South African government. The ANPR camera system obtained an average accuracy of 70% between 2020 and 2022. Generally, robust ANPR systems achieve accuracies above 90% [1], making the information collected by the deployed ANPR camera system less reliable. The subpar accuracy performance prompted the CSIR to conduct the current study which seeks to investigate the factors that affected the performance of the deployed ANPR camera system. A comparison was also made between the performance of the deployed ANPR camera system and four other commercial off-the-shelf ANPR systems. A custom dataset was used to test the ANPR systems on vehicles with South African civilian and government plates.

The contributions of this work can be summarised as follows:

- **Investigating factors affecting ANPR performance**. We present quantitative and qualitative results that highlight factors affecting the performance of the deployed ANPR commercial system. The insights drawn from the investigation could serve as a general guideline for selecting ANPR commercial systems.
- **Evaluating the performance of commercial ANPR systems on a South African dataset**. A review of the existing literature reveals that there is only one study that evaluates the performance of commercial ANPR systems [3]. The study was conducted more than five years ago. ANPR technology has advanced a lot since then. In addition, few to none of the existing studies evaluate the performance of commercial ANPR systems on South African number plates. Insights drawn from the study will indicate the robustness of the evaluated ANPR systems on South African number plates.

2 Related Work

This section aims to provide an overview of ANPR methods, commercial off-the-shelf ANPR systems, and the performance evaluation of ANPR systems. It does not provide a detailed review of existing ANPR methods. Readers interested in a detailed review are referred to [1,2,4–6].

Automatic Number Plate Recognition. Automatic number plate recognition is typically divided into three key stages, as shown in Fig. 1 [1,3,4]. In the first stage, a vehicle's number plate is located. The second stage separates the number plate characters using image segmentation techniques. Finally, the number plate characters are identified using optical character recognition techniques. The literature discusses two main themes: traditional ANPR algorithms and deep learning ANPR algorithms. Traditional-based methods typically use image thresholding techniques such as Otsu's thresholding method and edge detection methods such as Sobel and Gabor filters. Image thresholding methods separate pixels into the background and foreground based on pixel intensity values. Morphological operations are typically applied after image thresholding techniques to reduce noise. Edge detection filters detect edges based on a sharp change in pixel intensity values. Other techniques such as Histogram of Oriented (HOG) and Scale Invariant Feature Transform (SIFT) are sometimes used instead to detect number plate edge information. After highlighting the edges, line detection methods such as Hough transform and connected component analysis (CCA) [7] are used to find the bounding box of a number plate.

While traditional methods have been successful, researchers have expressed concerns about their adaptability to various number plates and environmental conditions. This is due to their reliance on hand-engineered features that are often based on expert knowledge and limited to specific number plate characteristics. Additionally, researchers have raised concerns about the slow processing speed of traditional methods, as they are often computationally expensive [5].

To address the shortcomings of traditional methods, researchers have shifted their focus towards developing deep learning-based algorithms. Deep learning methods typically follow a multi-stage approach (similar to traditional methods) or a single-stage approach. Multi-stage approaches involve using dedicated deep learning networks for at least one of the three stages in ANPR. Various ANPR methods often use artificial neural networks (ANN) and convolutional neural networks [8–11] to extract number plate features in the first stage. Other multi-stage methods use deep learning object detectors like You Only Look Once (YOLO) [12–14]. After extracting the number plate, the number plate characters are often identified using ANNs, CNNs, and recurrent neural networks (RNNs) [6]. Single-stage deep learning ANPR approaches use a single network that can find the location of the vehicle number plate and recognise the characters in one forward pass through the network.

Commercial Off-the-Shelf ANPR Systems. Commercial off-the-shelf ANPR systems typically consist of an image camera device or a special ANPR

camera device that automatically captures number plate images of vehicles, software that processes and analyses the captured images of number plates and a database that stores and compares the number plate information against known records, see Fig. 1. The ANPR software often runs on a video management system (VMS) or a dedicated user application. The robust performance of ANPR systems can be affected by numerous factors such as environmental factors, variation of number plates, image factors, and a combination of different factors [1]. As such, the selection of an ANPR system must consider requirements such as camera resolution, the inclusion of infrared illuminators, depth of field, shutter speed, image processing and OCR software capabilities, and lens design. In addition to performance requirements and functional requirements, commercial ANPR systems must meet non-functional requirements such as cost of acquisition and operation, physical dimensions, power consumptions, integration and compatibility, data storage and management, training and support, and compliance with regulations. Due to these varying factors, ANPR technology is not a one-size-fits-all solution and often needs optimization from region to region [2]. However, by carefully considering these factors, individuals and organizations can select an ANPR system that perfectly suits their needs and delivers maximum value for their specific application.

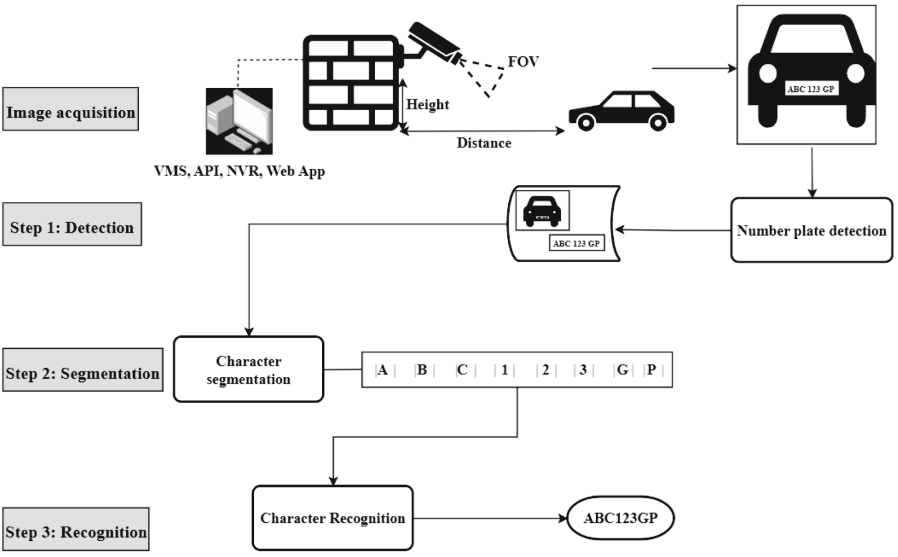


Fig. 1. Summary of the key steps involved in number plate recognition

Performance Evaluation of ANPR Systems. Due to the rapid development of ANPR technology, it is often necessary to evaluate their performance. Such methods serve as a guideline for selecting ANPR methods for specific applications, gauge technology advancement, and provide insights for future research.

Most existing studies evaluate the performance of each stage in a typical ANPR algorithm (detection, segmentation, and recognition) and the overall performance. In an early study, Ahmad *et al.* [4] evaluated the performance of six traditional ANPR algorithms which are made of three plate localisation methods (E1, E2, and E3), one character segmentation method, and two optical character recognition methods (O1, and O2). Each plate localisation method was paired with one of two optical recognition modules. The first plate localisation method used a Sobel edge detector method. The second plate localisation method was based on adaptive morphology. Finally, the third method localises plates using edge statistics and morphology. The first OCR method was based on template matching. The second OCR method was based on mapping the image into a character space and then using a probabilistic neural network to predict the plate characters. The same method for character segmentation is used in all tested ANPR methods. The authors found that the method that used the Sobel filter for edge detection provided the best results. In another study [15], the performance of various edge detectors for ANPR was compared. Both studies presented by Ahmad *et al.* [4] and Lubna *et al.* [15] focused on traditional methods and none of the evaluated methods were a commercial system. Later, Youting *et al.* [3] evaluated the performance of five commercial systems on a dataset consisting of number plates from all 31 provinces in mainland China. The authors observed a significant gap between the experimental results and the system accuracy rates advertised. This highlights the importance of studies that evaluate the performance of commercial ANPR systems.

Summary. This section provided an overview of ANPR methods in the literature, commercial ANPR systems, and the performance evaluation of ANPR systems. Based on the literature, research is currently focused on developing more robust ANPR methods based on deep learning due to the ability of convolutional neural networks to extract visual features automatically.

While ANPR performance evaluation studies exist, only a few studies focus on commercial systems [3]. ANPR technology has advanced a lot since the study by Youting *et al.* [3]. Additionally, there is a need for a research study that focuses on evaluating the performance of commercial systems on a South African dataset.

3 Experimental Setup

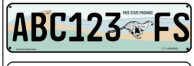
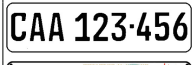
3.1 Dataset

Image Acquisition. The study's number plate dataset was collected from two ANPR camera systems in Pretoria, Gauteng Province, South Africa. One system was set up at the CSIR (Location 1), while the other was placed at the entrance of a government facility, also in Pretoria (Location 2). The ANPR camera at the government facility was installed 1.8 m above the ground and at least 2 m from stationary vehicles. The CSIR camera was installed 0.9 m above the ground and at least 3 m from stationary vehicles.

The images captured by the ANPR camera at the government facility will be used to study the factors influencing the system's performance. The dataset from this facility will be combined with the dataset obtained from the CSIR system to evaluate the performance of commercially available ANPR systems. These two datasets will also be used to compare how well the ANPR systems can read the number plates of both government and civilian vehicles.

The ANPR camera systems captured most images during the day under varying environmental conditions, such as sunny, cloudy, and rainy weather. Government vehicle number plates differ based on the public department they are affiliated with, while civilian number plates vary depending on the province to which the vehicle belongs. The final combined dataset comprises 1,222 images. Table 1 summarizes the distribution of images in the dataset and includes example images representing each province in South Africa. The dataset primarily contains government and civilian number plates from the Gauteng province, where the ANPR camera systems were installed. Efforts are underway to gather a more diverse dataset representing all nine provinces. The dataset used in the study can be obtained upon request.

Table 1. Distribution of number plate images in the dataset across South African provinces with corresponding example images

Province	Number	Plate example
Mpumalanga	5	
Kwa-Zulu Natal	0	
Government	608	
Gauteng	544	
Limpopo	21	
Freestate	0	
Western Cape	0	
Eastern Cape	0	
Northern Cape	0	
North West	0	
Blank Plates	44	

Data Pre-processing. The collected images underwent an initial inspection to identify and remove duplicates. This step was taken to prevent repetition in the datasets and to ensure that each image contributed uniquely. A perceptual hashing algorithm was employed to detect duplicates, which were then manually reviewed and removed.

Following the initial inspection, the next task involved image annotation. This process included manually transcribing the visible characters on the number plates, noting the direction of the vehicle, and indicating whether the plate was obscured or absent from the scene. The annotation process was performed using a custom ANPR verification software application. The number plates were annotated based on what a human could visually discern from the captured image. If some characters on the plate were obscured, only the visible characters were recorded as the ground truth. Similarly, if the number plate was entirely invisible, a "No plate" ground truth label was assigned.

3.2 Automatic Number Plate Recognition Systems

Deployed ANPR Camera System. The ANPR camera installation guidelines specify that the maximum pitch angle and the angle between the vertical and the vehicle's direction of travel must be less than 30°. To account for the pitch angle, the installation instructions define a specific height and minimum distance at which the camera should be installed. The cameras used to collect the datasets were installed using the simplest setup. Information on the installation of the ANPR camera systems and the manufacturer-recommended installation is provided in Table 2.

The pitch angles were within the specified range even though the civilian camera was installed at a lower height than recommended. However, the angle between the vertical and the vehicle's direction of travel in the images was not within the specified range. The government camera was slightly outside the range, and the civilian camera was almost double the recommended angle. Due to the incorrect installation, we anticipate lower accuracy; nevertheless, this setup provided a challenging testing environment for the cameras.

Table 2. Table showing the recommended installation location of the ANPR camera and the location of the installed cameras.

Camera	Height	Min Distance	Pitch	Horizontal
Camera specification	1.5 m–2 m	2.5 m–3 m	<30°	<30°
System A	Not Specified	Not Specified	<45°	<45°
System B	Not Specified	Not Specified	<40°	<40°
System C	Not Specified	Not Specified	Not Specified	Not Specified
System D	Not Specified	Not Specified	<20°	<30°
System E Location 1	0.9 m	3 m	≈16.7°	≈65°
System E Location 2	1.8 m	3 m	≈30.96°	≈32°

Selected Commercial ANPR Systems. In the study, four commercial ANPR systems were selected for evaluation. Commercial systems were mainly ANPR systems that provided cloud-based application programming interfaces (APIs). Commercial ANPR systems with API interfaces are easy to set up and require minimal software implementation, making the evaluation process simple and cost-effective. The information on the commercial ANPR APIs selected is summarised in Table 3. Table 3 summarises the subscription plans published by API developers, the corresponding costs, the number of requests or calls to the ANPR API per month, and the interface type used. The names of the plans, such as "Small," "Medium," and "Starter," correspond to the exact plan names published by the API providers.

These APIs were identified using search terms such as "automatic number plate recognition," "automatic number plate recognition," "commercial systems," and "cloud-based API." Ultimately, the four ANPR APIs were chosen based on their support for South African number plates, the minimum number of API calls (requests) allowed per month, the availability of a Python API interface, and a subscription cost of less than $100 per month. For privacy reasons, generic names (System A-D) were used instead of the actual names of the commercial ANPR systems. The deployed ANPR camera system will be called System E for easy reference. Readers interested in the actual names of the ANPR systems can contact the first author of this study.

Table 3. Summary of Subscription Plans, Costs, Monthly API Request Limits, and Interface Types for ANPR API Providers.

System A			
Plans	Small $50 monthly	Medium $150 monthly	Large $250 monthly
API calls	50 000 monthly	250 000 monthly	500 000 monthly
API interface	Shell, Ruby, Python, JavaScript, C#, PHP, Java, and C++		
System B			
Plans	Starter $40 monthly	Plus $125 monthly	Advanced $510 monthly
API calls	2,500 monthly	25,000 monthly	125,000 monthly
API interface	JAVA, Python, C#, JavaScript		
System C			
Plans	Starter €95 monthly	Growth €220 monthly	Professional €400 monthly
API calls	25,000 monthly	250,000 monthly	500,000 monthly
API interface	cURL, JavaScript, and Python		
System D			
Plans	7 day trial	$80 monthly	$816 yearly
API calls	50 daily	50 per-minute	50 per-minute
API interface	JAVA, Python, C#, Ruby, PHP, Windev, and JavaScript		

3.3 Evaluation Scope

The evaluations conducted in this study focused solely on assessing the effectiveness of the API algorithms in accurately recognising South African vehicle number plates. No efforts were made to delve into the specific implementations of the API algorithms, as this data is proprietary or comprises trade secrets of the API providers. APIs were uniformly evaluated using datasets consisting of images captured under the same camera settings, environmental conditions, camera placements, and vehicle number plate attributes.

3.4 Evaluation Approach

Evaluation Procedure. The selected commercial ANPR systems will be evaluated using the three-step approach summarised in Fig. 3. Step 1 involves data preprocessing which was covered in Sect. 3.1. In the second step, the images are converted to a binary format and then sent to the API using a post request for number plate detection and recognition. The ANPR APIs return responses in JavaScript object notation (JSON) format. Number plate data is retrieved from the JSON output and saved in a comma-separated values file, see Fig. 2. In the last step, the returned API results are compared against the ground-truth number plate characters. A prediction is deemed correct if the predicted number plate characters match the ground-truth characters (including blanks or vehicles with no number plates). The results will be analysed using the evaluation metrics discussed below.

```python
import requests
import json
import base64

# Convert the image to base64 string
image_path = 'image000016650.jpg'
with open(image_path, "rb") as image_file:
    base64_image = base64.b64encode(image_file.read()).decode('utf-8')

# Create the JSON string with the base64 image
data = {
    "image": base64_image
}
json_string = json.dumps(data)

# Basic authentication
usrPass = b'your_username:your_password'
b64Val = base64.b64encode(usrPass).decode('utf-8')

# Send the POST request
resp = requests.post(
    'https://gatiosoft.ro/platebber.aspx',
    headers={
        "Authorization": "Basic %s" % b64Val,
        'Content-Type': 'application/json',
        'Accept': 'application/json'
    },
    json=json.loads(json_string)  # Send the JSON object
)

# Get and print the response
resp_data = resp.json()
print(resp_data)
```


```
[
  {
    "id": "license_plate",
    "description": "License plate",
    "timestamp": "2020-04-15 06:53:36",
    "bounding_box_confidence": 0.9206130504608154,
    "x1": 161,
    "y1": 237,
    "x2": 1112,
    "y2": 237,
    "x3": 1112,
    "y3": 727,
    "x4": 161,
    "y4": 727,
    "plate_text": "BLR88M",
    "base64PlatePhoto": ""
  }
]
```


Fig. 2. API response JSON example

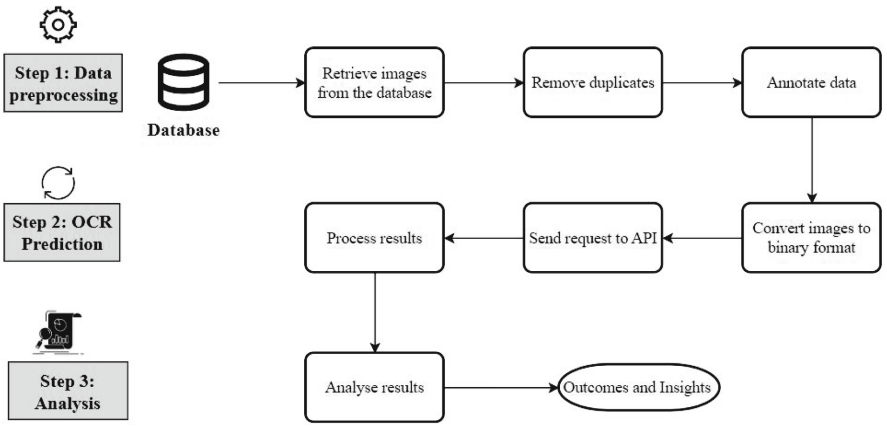


Fig. 3. Image-to-API process for ANPR performance evaluation

Recognition Performance Metrics. The character recognition performance of the ANPR systems will be measured using the accuracy rate metric in Eq. 1. The Accuracy Rate is the percentage ratio of all correct predictions (including true positives and negatives) to the total number (N) of annotated number plates in the dataset, including valid and blank number plates. In Eq. 1, P_i refers to the predicted plate characters for the i-th sample, and gt_i refers to the ground truth plate characters for the i-th sample.

$$\%AR = \frac{1}{N}\sum_{i=1}^{N} I(P_i = gt_i) * 100\% \tag{1}$$

In addition to the ANPR prediction performance, the OCR performance of the APIs will be measured by calculating the number of characters that were predicted correctly from the ground-truth values. The OCR performance consists of character-to-character matching, number plate character position error rate, and analysis of the per-digit or character performance by the APIs. OCR performance will provide a detailed analysis to cater for edge cases where number plates have unusual fonts, damaged plates, or images taken from an extreme angle.

Detection Performance Metrics. The detection performance of each ANPR API will be measured by accuracy, precision, recall, and F1-Score. The metrics will be calculated based on the number of True Positives (TP), False Positives (FP), True Negatives (NP), and True Negatives (TN) as shown in Eqs. 2 to 5 (Table 4).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

Table 4. ANPR Detection Performance Metrics.

Prediction Outcome	Confusion Matrix Reference	Outcome Description
Correct number Plate	True Positive (TP)	The ANPR API or camera correctly predicted the number in the image
Missed number Plate	False Negative (FN)	The camera API or ALPR failed to detect the licence plates in the annotated image
Blank number Plate	True Negative (TN)	The ALPR API or camera returned no number when there is no number plate in the image
False number Plate	False Positive (FP)	The ALPR API or camera returned number plates when no number plates were in the image

4 Experimental Results and Discussion

4.1 Investigating Factors Affecting ANPR

Three user-controllable factors can affect the accuracy of ANPR systems. These are the viewing angle, camera exposure, and the size of the number plate in the captured image. A rule of thumb for ANPR is that if a human can read the number plate, the ANPR system will extract it. The cameras were installed close enough to the vehicle entrances that the licence plates were large enough to be easily read. The camera manufacturer-recommended exposure settings when using the camera for licence recognition. The recommended setting was to use the fixed iris mode with 1/1000 exposure time. These settings produced well-exposed images for number plate recognition. The settings were kept throughout the data capturing. Natural light, infrared illumination, and the recommended exposure of the cameras provided well-exposed images of the licence plates. The only factor out of specification that would reduce the accuracy, from what was advertised, of the ANPR algorithms was the viewing angle.

The recommended installation of the ANPR camera defined the maximum pitch angle and the angle between the vertical and the vehicle moving direction must be less than 30°. To accommodate the pitch angle, the installation instructions define an installation height and minimum distance to which the camera

should be installed. The recommended installation of the cameras, for each of the tested systems, is shown in Table 2. System A advertises that it works up to 70° but its recommendation is much lower. Camera installation was constrained by the available infrastructure, which led to the non-optimal placement of the cameras. The pitch angles were within specification even though the location 1 camera was installed lower than recommended. The angle between the vertical and the vehicle moving direction in the images was out of the specification as shown in the table. The location 2 camera was slightly out, and the location 1 camera was almost double what it should be. On this basis, it was expected that a lower accuracy than advertised would be achieved. This expectation was not noticed, as location 1, which had the worst installation, performed better than location 2. This discrepancy indicates that the viewing angle may not be the cause of the lower accuracy than expected at location 2.

4.2 Comparing the Performance of ANPR Commercial Systems

This section compares the performance of cloud-based ANPR systems from ANPR cameras. A comparison of the performance of the ANPR systems will be conducted based on their ability to recognise number plate characters (OCR recognition performance) and detect number plates (number plate detection performance), as explained in Sect. 3.4. OCR recognition performance will demonstrate a system's ability to correctly identify number plate characters. The ability to localise and detect a number plate on a vehicle will be revealed through number plate detection.

OCR Recognition Performance. Figure 4 shows the number plate recognition accuracy and the OCR performance of the cloud-based ANPR systems (System A to D) and the ANPR cameras (System E) with images from (Location 1, L1) and (Location 2, L2). In this section, the systems are referred to by their alphabet and the locations are specified by their number. Accordingly, A1_L1 refers to System A applied to the images from Location 1.

Evaluating OCR performance through character-to-character matching can introduce biases, as the direction and position of matching influence outcomes. For instance, matching from left to right may yield different results than matching from right to left. Furthermore, positional matching penalises models harshly for missing characters. An evaluation metric combining accuracy scores with recall rates (character retrieval rates) was adopted to alleviate these biases. Specifically, the character retrieval rate (occurrence) was added to the two accuracy scores. The three values were multiplied by a third to obtain the average ANPR accuracy, as shown in Table 5.

Figure 4 presents the number plate recognition performance of the evaluated systems at Location 1. System A obtained the highest accuracy, narrowly surpassing System B. System A achieved an impressive 99.52% average accuracy at Location 1, closely trailed by System B's 99.47%. System C secured third place with a notable 90.13% average accuracy, followed by System D at 72.47%. The deployed ANPR camera system trailed behind, achieving an accuracy rate of 53.44%.

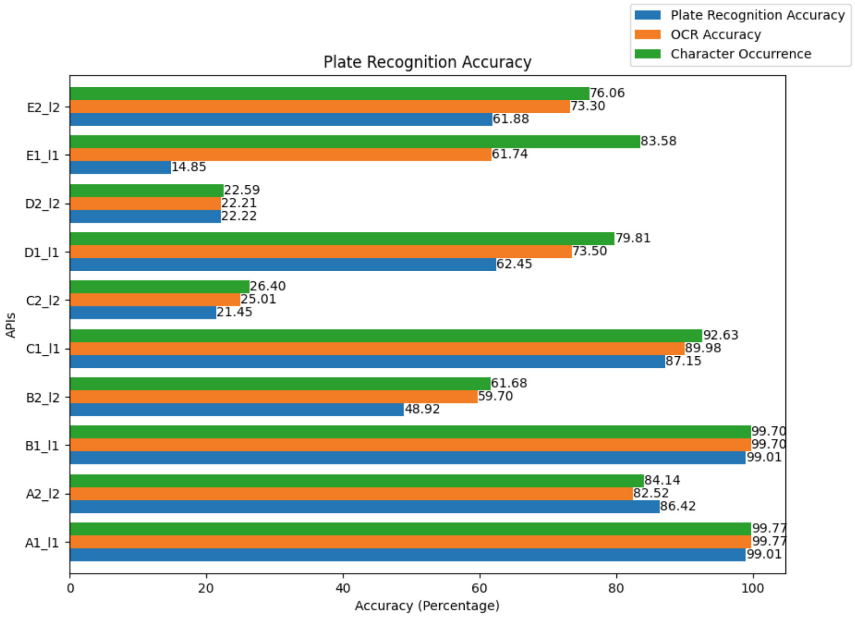


Fig. 4. Number plate recognition accuracy and OCR performance

Table 5. OCR recognition performance per system

System	Plate Recognition	OCR Accuracy	Character Occurrence	Average Accuracy
A1_l1	99.01	99.77	99.77	99.52
A2_l2	86.42	82.52	84.14	84.36
B1_l1	99.01	99.70	99.70	99.47
B2_l2	48.92	59.70	61.68	56.77
C1_l1	87.15	89.98	92.63	89.92
C2_l2	21.45	25.01	26.40	24.29
D1_l1	62.45	73.50	79.81	71.92
D2_l2	22.22	22.21	22.59	22.34
E1_l1	14.85	61.74	83.58	53.39
E2_l2	61.88	73.30	76.06	70.41

Location 2 presented a challenging environment for ANPR systems, with System A emerging as the top performer, achieving an accuracy rate of 84.36%. The camera's accuracy rate followed at 70.41%, while System B and System D trailed behind, recording 56.77% and 22.34%, respectively.

The lower performance can be attributed to the unique number plate format prevalent among vehicles captured at this site. Systems C and D struggled significantly, resulting in notably poor performances. In contrast, System A demonstrated greater robustness in adapting to unfamiliar number plate formats, outperforming its counterparts.

The cloud-based ANPR APIs demonstrated varying levels of performance across locations. At Location 1, they achieved an average accuracy of 90.20%, whereas at Location 2, their performance dropped significantly to 46.94%. In contrast, the camera (System E) recorded average accuracy rates of 53.39% at Location 1 and 70.41% at Location 2.

Averaging these values reveals that the cloud-based APIs attained an overall average performance of 68.57%, outperforming the camera's average of 61.9%. Notably, System A and System B APIs achieved impressive accuracy levels. However, the poorer performance of System C and System D substantially degraded the overall performance of the cloud-based APIs.

The results so far, represent the ANPR recognition accuracy, which measures the character-to-character matching capabilities of the systems. However, another crucial aspect is number plate detection performance, which refers to the system's ability to accurately detect the presence and visibility of a vehicle's number plate.

Vehicle Number Plate Presence Detection Performance. The number plate detection performance of the systems was measured by calculating the F1-Scores of each system and computing the average F1-Score across the two datasets (L1 and L2). The F1-Score, a decimal value between 0 and 1, measures the harmonic mean of precision and recall to provide a single metric that balances the two. A high F1-Score signals a model that is effective in correctly identifying positive instances (high recall) while also minimising false positives (high precision).

From Table 6, System A achieved the highest average F1-Score of 0.99. The ANPR camera (System E) followed closely with a respectable F1-Score of 0.96. System B had the third-best average F1-Score of 0.91. System D had the lowest score of 0.65. System C also struggled to retrieve positive samples, scoring just 0.72.

Table 6. Vehicle number plate presence detection performance

API	FP	TP	TN	FN	Accuracy	Precision	Recall	F1-Score
C_L1	3	466	5	33	0,93	0,99	0,93	0,96
C_L2	0	228	7	517	0,31	1	0,31	0,47
E_L1	0	495	8	3	0,99	1	0,99	0,99
E_L2	5	643	2	102	0,86	0,99	0,86	0,92
B_L1	0	497	8	1	1,00	1	1	1
B_L2	1	513	2	238	0,68	1	0,68	0,81
D_L1	3	413	5	86	0,82	0,99	0,83	0,9
D_L2	0	187	7	557	0,26	1	0,25	0,4
A_L1	3	498	5	1	0,99	0,99	1	0,99
A_L2	4	734	3	11	0,98	0,99	0,99	0,99

4.3 Summary of the Findings

Camera viewing angles significantly impacted accuracy. However, in this study, the camera with the larger viewing angle yielded surprisingly more accurate results. This outcome contradicts initial expectations, warranting further investigation to determine the underlying cause.

With regards to the cloud-based ANPR systems, only System A was capable of detecting multiple number plates in a single image and consistently outperformed its counterparts on datasets from both Location 1 and Location 2. System B performs as impressively as System A on the Location 1 dataset but struggles on the Location 2 dataset due to its inability to detect multiple number plates, as the government dataset consists of many images with multiple plates. In contrast, System C underperformed significantly, frequently failing to detect plates in most images. System D exhibited unique behaviour, often returning duplicate number plates, necessitating the partitioning of the plate text for accurate analysis.

It is unclear what number plate detection and recognition techniques are employed by the camera's built-in algorithms. Considering the limited processing resources available in embedded devices like cameras, these devices are likely employing traditional computer vision techniques or shallow/simple deep learning techniques. Cloud-based ANPR algorithms (API) could be performing better than the camera built-in algorithms due to field re-enforcement learning, this has not been confirmed with API providers.

5 Conclusions and Future Work

The study evaluated the performance of an ANPR camera and four commercial off-the-shelf ANPR APIs. The main goal was to determine whether commercial off-the-shelf cloud-based APIs can outperform the specific ANPR algorithm implemented on the cameras used to gather the datasets.

The study found that ANPR prediction performance varied significantly based on the API used for number plate recognition and the installation location of ANPR cameras used for data collection. The prediction accuracies for the commercial off-the-shelf cloud-based APIs were generally better than those of the ANPR cameras.

The study generated a comprehensive dataset that can serve as a valuable resource for researchers seeking to evaluate the detection and recognition performance of ANPR algorithms. Additionally, it provided critical insights into the effectiveness of cloud-based commercial ANPR APIs specifically within the South African context.

Disclosure of Interests. The authors declare that no funding was received from any of the companies that developed the commercial systems evaluated in this study.

References

1. Tang, J., Wan, L., Schooling, J., Zhao, P., Chen, J., Wei, S.: Automatic number plate recognition (ANPR) in smart cities: a systematic review on technological advancements and application cases. Cities **129**(103833), 1–18 (2022)
2. Shashirangana, J., Padmasiri, H., Meedeniya, D., Perera, C.: Automated number plate recognition: a survey on methods and techniques. IEEE Access **9**, 11203–11225 (2021)
3. Youting, Z., Zhi, Y., Xiying, L.: Evaluation methodology for number plate recognition systems and experimental results. IET Intell. Transp. Syst. **12**(5), 375–385 (2018)
4. Ahmad, I.-S, Boufama, B., Habashi, P., Anderson, W., Elamsy, T.: Automatic number plate recognition: a comparative study. In: 2015 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), pp. 635–640. IEEE, Abu Dhabi (2015). https://doi.org/10.1109/ISSPIT.2015.7394415
5. Lubna, N.M., Syed, A.-A.: Automatic number plate recognition: a detailed survey of relevant algorithms. Sensors **21**(3028), 1–35 (2021)
6. Parvin, S., Rozario, L.-J., Islam, M.-E.: Vehicle number plate detection and recognition techniques: a review. Adv. Sci. Technol. Eng. Syst. J. **6**(2), 423–438 (2021)
7. Pustokhina, I.-V., et al.: Automatic vehicle number plate recognition using optimal K-means with convolutional neural network for intelligent transportation systems. IEEE Access **8**, 92907–92917 (2020)
8. Gonçalves, G.-R., Diniz, M.-A., Laroca, R.-M., Menotti, D., Schwartz, W.-R.: Real-time automatic number plate recognition through deep multi-task networks. In: 2018 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 110–117. IEEE, Parana (2018). https://doi.org/10.1109/SIBGRAPI.2018.00021
9. Xu, Z., et al.: Towards end-to-end license plate detection and recognition: a large dataset and baseline. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11217, pp. 261–277. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01261-8_16
10. Li, H., Wang, P., Shen, C.: Toward end-to-end car number plate detection and recognition with deep neural networks. IEEE Trans. Intell. Transp. Syst. **20**(3), 1126–1136 (2018)
11. Omar, N., Sengur, A., Al-Ali, S.-G.-S.: Cascaded deep learning-based efficient approach for number plate detection and recognition. Expert Syst. Appl. **149**, 1–10 (2020)
12. Zhang, L., et al.: A robust attentional framework for number plate recognition in the wild. IEEE Trans. Intell. Transp. Syst. **22**(11), 6967–6976 (2020)
13. Hendry, C.R.: Automatic number Plate Recognition via sliding-window darknet-YOLO deep learning. Image Vision Comput. **87**, 47–56 (2019)
14. Yang, D., Yang, L.: A deep learning-based framework for vehicle number plate detection. Int. J. Adv. Comput. Sci. Appl. **15**(1), 47–56 (2024)
15. Lubna, Khan, M.-F, Mufti, N.: Comparison of various edge detection filters for ANPR. In: 2016 Sixth International Conference on Innovative Computing Technology (INTECH), pp. 306–309, IEEE, Dublin (2016). https://doi.org/10.1109/INTECH.2016.7845061



Neural Network-Based Vehicular Channel Estimation Performance: Effect of Noise in the Training Set

Simbarashe Aldrin Ngorima[1,2,3(✉)], Albert S. J. Helberg[1], and Marelie H. Davel[1,2,3]

[1] Faculty of Engineering, North-West University, Potchefstroom, South Africa
aldringorima@gmail.com, albert.helberg@nwu.ac.za
[2] Centre for Artificial Intelligence Research, Cape Town, South Africa
[3] National Institute for Theoretical and Computational Sciences, Stellenbosch, South Africa

Abstract. Vehicular communication systems face significant challenges due to high mobility and rapidly changing environments, which affect the channel over which the signals travel. To address these challenges, neural network (NN)-based channel estimation methods have been suggested. These methods are primarily trained on high signal-to-noise ratio (SNR) with the assumption that training a NN in less noisy conditions can result in good generalisation. This study examines the effectiveness of training NN-based channel estimators on mixed SNR datasets compared to training solely on high SNR datasets, as seen in several related works. Estimators evaluated in this work include an architecture that uses convolutional layers and self-attention mechanisms; a method that employs temporal convolutional networks and data pilot-aided estimation; two methods that combine classical methods with multilayer perceptrons; and the current state-of-the-art model that combines Long-Short-Term Memory networks with data pilot-aided and temporal averaging methods as post processing. Our results indicate that using only high SNR data for training is not always optimal, and the SNR range in the training dataset should be treated as a hyperparameter that can be adjusted for better performance. This is illustrated by the better performance of some models in low SNR conditions when trained on the mixed SNR dataset, as opposed to when trained exclusively on high SNR data.

Keywords: Channel estimation · deep learning · neural networks · CNN-Transformer · IEEE 802.11p · vehicular channels

1 Introduction

The integration of network infrastructure with artificial intelligence (AI) is becoming more prevalent due to the increasing demands of data-intensive applications and the need for more efficient and reliable communication systems.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 192–206, 2025.
https://doi.org/10.1007/978-3-031-78255-8_12

Incorporating AI into these systems promises to improve network management, resource allocation, and overall system performance.

This convergence is notably evident in vehicular communication, which is an important component of intelligent transportation systems. Vehicular communication systems face unique problems due to their high mobility, rapidly varying environments, and significant latency requirements. One of the challenges in this domain is obtaining accurate channel estimates, which is critical to ensuring reliable communication between vehicles and infrastructure. Channel estimation involves determining the characteristics of a communication channel and monitoring its changes to adjust to varying conditions [9].

The dynamic nature of vehicular environments, characterised by rapidly changing channel conditions, poses a challenge to traditional channel estimation methods. These methods include methods such as least squares (LS) and minimum mean square error (MMSE), which rely on mathematically estimating the channel based on the received version of known data sent as a preamble to the signal. Although effective in more stationary environments, these techniques often struggle to keep up with the rapid changes inherent in vehicular channels [4]. This limitation highlights the need for more advanced adaptive approaches to channel estimation in these high-mobility scenarios.

In recent years, researchers have increasingly focused on using data symbols directly to estimate the channel in a method referred to as data pilot-aided (DPA) channel estimation. DPA estimation addresses the challenges posed by dynamic environments, especially in vehicular communication systems, where pilot signals are evidently insufficient. Unlike traditional methods that rely heavily on a fixed number of pilot signals for channel estimation, DPA estimation uses demapped data symbols as additional pilots. This approach allows for more flexible and adaptive tracking of channel variations without the need to allocate more resources for pilot signals, thereby improving efficiency and accuracy in rapidly changing channels. However, the performance of the DPA method can be significantly affected by errors introduced during the demapping process, which may limit its overall effectiveness [12]. To improve efficiency, DPA estimation is typically used with an error compensation scheme to reduce channel distortion and prevent errors from propagating to the next symbols in a frame. Two common error compensation approaches for channel estimation are spectral temporal averaging (STA) [2] which utilises the correlations of data symbols in time and frequency domains to estimate the channel and time-domain reliable test frequency domain interpolation (TRFI) [8], as described in more detail in Sect. 2.2. However, these methods still struggle in highly dynamic scenarios [3].

Deep learning (DL) has recently gained attention for its ability to improve the accuracy of channel estimates. A method utilising autoencoders and training on a high signal-to-noise ratio (SNR) dataset (SNR=40 dB) was proposed to correct DPA estimate errors in the frequency domain [6]. However, it did not account for temporal changes, limiting its performance. Furthermore, incorporating feedforward networks into classical methods such as STA and TRFI methods, as proposed in [4], has shown promise in improving estimation accuracy.

These methods were also trained on high SNR datasets (SNR = 30 dB). These methods offer valuable baselines. Recently, a long-short-term memory (LSTM)-based architecture has been proposed to capture the temporal and frequency characteristics of vehicular channels [3]. This method was trained on a dataset generated at the SNR level of 40 dB and demonstrated significant improvements over previous estimators. However, this method is computationally expensive.

This study introduces the use of a mixed SNR dataset, where the training dataset includes data generated at different SNR levels. We hypothesise that training on a mixed SNR dataset helps an neural network (NN) generalise across a wider range of real-world conditions by exposing it to both noise-dominated and signal-dominated scenarios. This approach contrasts with the methodology adopted in several studies [3–5,12], which propose that training the NN-based estimator using a dataset generated only at a high SNR level enhances its ability to generalise in any environment: at high SNR the impact of the channel is greater than the impact of noise, therefore it is expected that the NN develops better channel knowledge.

We evaluate the use of the mixed SNR dataset using a CNN-Transformer-based estimator [11] which is a hybrid model that combines a one-dimensional convolutional neural network (CNN) with a transformer architecture, a TCN-DPA estimator [10] that uses a temporal convolutional network (TCN) to correct the DPA propagation error, two estimators based on concatenating a multilayer perceptron (MLP) with a classical method, in this case STA-MLP [4] and TRFI-MLP [5], and the LSTM-DPA-TA method [3] which combines Long Short-Term Memory (LSTM) networks with Data Pilot-Aided (DPA) estimation and Temporal Averaging (TA).

2 Background

We describe the system model, well-known channel estimation schemes in vehicular communications, and some of the NN-based channel estimation techniques currently used. We also provide a brief description of the IEEE 802.11p physical layer structure that is used in all simulations, along with a description of its specifications.

2.1 System Model

In this paper, we adopt the IEEE 802.11p physical layer structure, which is specifically designed for vehicular communication systems. This section follows a detailed discussion of the IEEE 802.11p specifications as outlined in Ngorima et al. [10]. The IEEE 802.11p standard uses Orthogonal Frequency Division Multiplexing (OFDM) to transfer data by dividing the available bandwidth into several subcarrier frequencies for simultaneous transmission of multiple signals. 52 of the 64 available subcarriers are used for data transmission and pilot symbols. The remaining subcarriers are for guard bands and direct current (DC)

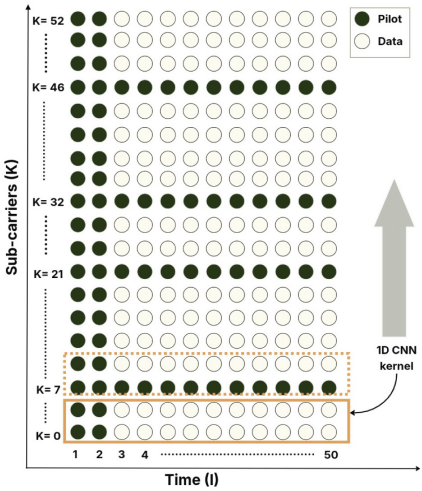


Fig. 1. The IEEE 802.11p frame structure, illustrating the allocation of subcarriers for pilot and data transmission.

offset. To illustrate this structure, Fig. 1 shows the IEEE 802.11p OFDM frame, indicating pilot and data subcarriers.

In this work, we consider a frame structure that consists of two preamble symbols for signal detection and timing synchronisation, followed by a signal field that contains transmission parameters. The received signal on subcarrier $[k]$ at time i can be represented as

$$Y_i[k] = H_i[k]X_i[k] + N_i[k], \tag{1}$$

where $H_i[k]$, $X_i[k]$ and $N_i[k]$ denote the respective channel response, transmitted signal, and additive Gaussian noise (AWGN), respectively. Note that in vehicular channels H_i varies in both the time and frequency domains.

2.2 Well-Known Vehicular Channel Estimation Schemes

This section describes some of the main vehicular channel estimation schemes that have been widely used and studied in the literature. These methods were selected based on their effectiveness in dynamic environments and their foundational role in the development of more advanced NN-based estimators.

DPA Estimation: The first step of the DPA process, as described by Pan et al. [12], is to calculate the initial estimate of the frame. This is done by applying the least squares (LS) estimation to the preamble as shown below:

$$\hat{h}_0[k] = \frac{y_1^{(p)}[k] + y_2^{(p)}[k]}{2p[k]}, \tag{2}$$

where $y_1^{(p)}[k]$ and $y_2^{(p)}[k]$ are the known orthogonal preambles received on the $[k]$ subcarrier and $p[k]$ is the OFDM data symbol transmitted over the k-th subcarrier. The next OFDM symbols in the frame are equalised using the initial estimate as the starting point as follows:

$$y_i^{\text{eq}}[k] = \frac{y_i[k]}{\hat{h}_{i-1}^{\text{DPA}}[k]}, \hat{h}_0^{\text{DPA}}[k] = \hat{h}_0[k]. \tag{3}$$

The equalised symbol is demapped and mapped to its closest constellation point as:

$$d_i[k] = \Re\left(\frac{y_i[k]}{\hat{h}_{i-1}^{\text{DPA}}[k]}\right), \hat{h}_0^{\text{DPA}}[k] = \hat{h}_{\text{LS}}[k], \tag{4}$$

Finally, the DPA channel is updated:

$$\tilde{h}_{\text{DPA}_i}[k] = \frac{y_i[k]}{d_i[k]} \tag{5}$$

STA Estimation Scheme: STA uses both spectral (frequency) and temporal correlations within data symbols to estimate the channel [4]. STA can also be viewed as a recursive filtering process that combines multiple channel estimates into a more accurate estimate. The first step is to calculate a channel estimate in the frequency domain $\hat{h}i^{\text{FD}}[k]$ as follows:

$$\hat{h}i^{\text{FD}}[k] = \sum \lambda = -\beta^{\beta}\omega_{\lambda}\hat{h}_i^{\text{DPA}}[k+\lambda],\ \omega_{\lambda} = \frac{1}{2\beta+1}, \tag{6}$$

where $2\beta + 1$ represents the number of subcarriers that are considered. Subsequently, a temporal average is computed to arrive at the final STA channel estimate, $\hat{h}i^{\text{STA}}[k]$:

$$\hat{h}i^{\text{STA}}[k] = \left(1 - \frac{1}{\alpha}\right)\hat{h}i - 1^{\text{STA}}[k] + \frac{1}{\alpha}\hat{h}_i^{\text{FD}}[k], \tag{7}$$

where α is a smoothing parameter that controls the weight given to current and previous estimates.

TRFI Estimation Scheme: TRFI leverages the high correlation within OFDM symbols to improve the accuracy of the channel estimation [5]. The process starts by creating multiple channel estimates for each symbol using the DPA method. The received symbol is equalised using the estimates of the current and previous symbols ($\hat{h}_i^{\text{DPA}}[k]$) and ($\hat{h}_{i-1}^{\text{DPA}}[k]$) respectively.

$$\begin{aligned} y_{i-1}^{\text{eq}'}[k] &= \frac{y_{i-1}[k]}{\hat{h}_i^{\text{DPA}}[k]}, \\ y_{i-1}^{\text{eq}''}[k] &= \frac{y_{i-1}[k]}{\hat{h}_{i-1}^{\text{DPA}}[k]}, \end{aligned} \tag{8}$$

Subsequently, the equalised symbols are remapped to their corresponding constellation points, $d'_{i-1}[k]$ and $d''_{i-1}[k]$. TRFI then categorises subcarriers into reliable and unreliable sets based on a reliability test between the remapped symbols. Subcarriers with consistent remapped symbols, that is, $d'_{i-1}[k] = d''_{i-1}[k]$, are considered reliable, while those with discrepancies are classified as unreliable. TRFI then interpolates channel estimates for unreliable subcarriers by using the reliable subcarriers as anchor points. This interpolation process effectively fills in the gaps in channel knowledge, resulting in a more accurate overall channel estimate. A detailed discussion on the interpolation procedure is given in [4].

2.3 MLP-Based Estimators

To enhance the performance of conventional STA and TRFI estimators, Gizzini et al. [4] introduced MLP-based channel estimation methods. In these methods, MLP layers are added to the STA and TRFI processes, allowing the model to learn complex patterns in vehicular channels and thereby improve the accuracy of the estimation. The methods STA-MLP and TRFI-MLP initially conduct DPA estimation, proceed with STA or TRFI estimation and subsequently feed the results into the MLP layers.

2.4 TCN-DPA Estimator

As proposed in [10], the TCN-DPA estimator integrates a Temporal Convolutional Network (TCN) with DPA estimation to enhance channel estimation in dynamic environments. The TCN processes the received OFDM symbols in the frequency domain, treating subcarriers as time steps. The extracted features are then passed to the DPA process, where the previous TCN output, $\hat{h}_{i-1}^{\mathrm{TCN}}[k]$, is used to equalise the current received symbol. After equalisation, the symbol is demapped and remapped to the nearest constellation point to obtain the final channel estimate, $\hat{h}_{i}^{\mathrm{TCN}}[k]$. This estimate is updated iteratively using the DPA process. This approach leverages the ability of the TCN to model long-range dependencies across subcarriers, making it effective for complex channel conditions.

2.5 LSTM-DPA-TA Estimator

The LSTM-DPA-TA estimator [3] integrates an LSTM network with DPA and Temporal Averaging (TA) techniques to enhance channel estimation in vehicular environments. In this approach, the LSTM processes the received OFDM symbols to extract key features, which are then passed into the DPA module. The resulting channel estimates are further refined using the TA technique. In the LSTM-DPA-TA method, OFDM symbols are treated as time steps, with subcarriers considered as features within each time-step sequence. This design enables the model to effectively capture frequency dependencies and improve the accuracy of channel estimation under dynamic conditions.

2.6 CNN-Transformer

The CNN-Transformer is a recently proposed hybrid architecture for vehicular channel estimation in [11]. This approach combines the strengths of CNNs and Transformer networks to analyse vehicular channel data comprehensively. The channel data can be represented in both the time and frequency domains. The frequency domain represents the subcarriers of the OFDM symbols, while the time domain represents the sequence of OFDM symbols transmitted over time. Each OFDM symbol is transmitted across multiple subcarriers. The frequency domain captures the spatial characteristics of the channel across OFDM subcarriers, while the time domain reflects the temporal dynamics as OFDM symbols are transmitted sequentially. In this CNN-Transformer architecture, subcarriers are treated as the 'time steps'. The architecture leverages CNNs to extract local features from subcarriers and uses Transformer layers to capture global dependencies across the frequency domain. The CNN component applies 1D convolutions to capture local patterns, while the Transformer component uses self-attention mechanisms to analyse patterns between subcarriers globally.

3 Methodology

This section outlines the methodological framework used in this study. We start with a detailed description of the datasets used for training and evaluation, specifically focussing on two approaches: the mixed SNR dataset and the high SNR dataset. Following this, we discuss the data preprocessing steps applied to transform complex received symbols into a format suitable for input into NNs.

3.1 Data Preparation

We simulate vehicle-to-vehicle communication following the 'Vehicle-To-Vehicle Expressway Same Direction with Wall' (VTV-SDWW) channel model [7]. Our simulation specifically modelled vehicles moving at 100 km/h, with a Doppler shift of 550 Hz. We used 16QAM (16-Quadrature Amplitude Modulation) modulation for data transmission in this urban environment scenario. This modulation scheme encodes the data by varying the amplitude and phase of the carrier signal.

Mixed SNR Dataset. The mixed SNR dataset represents a wide range of noise introduced on the VTV-SDWW channel. For this dataset, we simulate 18,000 time-specific frames with SNR values ranging from 0 to 40 dB in increments of 5 dB. Each SNR level includes 2,000 frames, with 50 OFDM symbols spread across 52 active subcarriers (48 data and 4 pilot subcarriers). During training, each model is exposed to the entire range of SNR levels. The validation set is created by reserving 25% of this training data to monitor the performance of the model during training. An independent test set consisting of 2,000 frames is generated separately from the training sets. This test set includes frames at

SNR levels of 0, 5, 10, 15, 20, 25, 30, 35, and 40 dB, ensuring that the evaluation correctly reflects the performance across different SNR conditions. The same test set is used to test all the models regardless of how they are trained.

High SNR Dataset. The high SNR dataset focusses on a fixed SNR level, representing less noisy or nearly ideal channel conditions, as opposed to the mixed SNR dataset. This dataset is used to train models with the assumption that training NNs in an environment with a clearly defined channel can lead to better generalisation even in noisy environments [3–5,12]. In this work, the high SNR dataset consists of 18,000 time-specific frames, all generated at a 40 dB SNR level, with the same frame structure as the mixed SNR data. As before, 25% of the training data is set aside for validation.

Data Preprocessing. To preprocess the data for NN models, we transform the complex 16QAM symbols received across subcarriers and time slots into a real-valued format. Each received symbol consists of both real and imaginary components. The following steps are performed to prepare the data:

1. **Separate Real and Imaginary Components:** Decompose each complex 16QAM symbol into its real and imaginary parts. Handle the data as two separate real-valued matrices instead of a single complex-valued matrix. For a given OFDM symbol, where $y[k] = r[k] + j \cdot i[k]$ represents the received complex value at subcarrier k, extract $r[k]$ as the real part and $i[k]$ as the imaginary part.
2. **Interleave Components:** After separation, interleave the real and imaginary components. This means arranging the real and imaginary parts in an alternating sequence across the time slots. By interleaving, a unified structure that preserves the relationship between the real and imaginary parts within the input data is created.
3. **Form the Input Matrix:** Structure the interleaved data into a matrix with dimensions 52×100, where 52 represents the number of subcarriers and 100 represents the sequence of interleaved real and imaginary components across the time slots. This transformation enables the NN to process the complex-valued input as a series of real-valued inputs.
4. **Prepare Input Data for NN models:** The resulting matrix is now a real-valued representation of the original complex data, ready for input into NN models.

The transformed input matrix described above is visually represented in Fig. 2. Each block in the figure represents the interleaved real and imaginary components of the received 16QAM symbols for a particular subcarrier across the OFDM symbols. The kernel size of 3 shown in the figure indicates the width of the sliding window used by the CNN during the convolution process across the subcarriers.

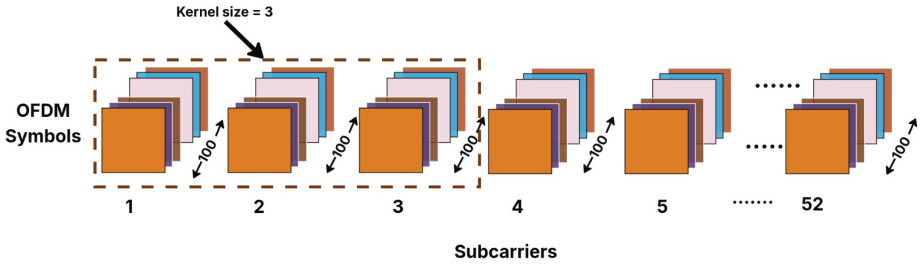


Fig. 2. Transformed IEEE 802.11p frame structure used as input to deep learning models, composed of 100 interleaved complex symbols and 52 subcarriers.

3.2 Hyperparameter Optimisation

We reimplement most of the NN-based estimators discussed in the previous sections to ensure compatibility with our experimental setup and to be able to optimise each individually for the specific datasets used. Specifically, the CNN-Transformer, TCN-DPA, STA-MLP, and TRFI-MLP models were reimplemented from scratch. The LSTM-DPA-TA model is implemented using existing code available online. To verify the accuracy of our reimplementations, we compare the performance of our implementations against the benchmark results reported in the original studies, ensuring that they perform as expected.

The final architecture of each model is determined through a hyperparameter tuning process, to identify the optimal configurations for each model. We optimise the hyperparameters of all models in both datasets using Optuna, a Bayesian optimisation framework that efficiently searches the hyperparameter space using a tree-structured Parzen estimator (TPE) [1]. This method systematically explores different hyperparameter configurations while conserving computational resources by early termination of underperforming trials.

For each model, we conduct multiple trials with different random seeds to ensure that results are not biased by a particular initialisation. Specifically, we conduct 50 trials per model, each with 3 seeds, allowing us to assess the robustness of the selected hyperparameters. The search space for each model was designed to cover a wide range of potential configurations. The key hyperparameters include the learning rate, the number of layers, the layer size (kernel size, number of attention heads, or layer width, depending on model), the learning rate scheduler parameters, and hyperparameters controlling regularisation through dropout. During the optimisation process, we monitor the convergence of validation loss. Early stopping is employed to prevent overfitting, stopping training when the validation loss ceases to improve for a predefined number of epochs.

Table 1 provides a summary of the best hyperparameters obtained for each model across the datasets. In addition to the parameters in the table, batch normalisation is applied to enhance model generalisation. The Adam optimiser and a batch size of 128 was used for TCN-DPA, LSTM-DPA-TA, STA-MLP and TRFI-MLP. For the CNN-Transformer model, we use the AdamW optimiser and manually set the batch size to 16.

Table 1. Optimised hyperparameters for CNN-Transformer, TCN-DPA, STA-MLP, TRFI-MLP, and LSTM-DPA-TA on the mixed SNR and high SNR datasets.

Hyperparameter	Search Space	Mixed SNR	40 dB
CNN-Transformer			
Learning rate	[1e–5 to 1e–2]	0.001	0.001
Transformer layers	[1 to 4]	2	4
Number of attention heads	[1 to 4]	2	4
Hidden dimension	[64 to 128]	128	128
Dropout rate	[0.001 to 0.3]	0.1	0.25
Number of epochs	[50 to 200]	110	200
Number of CNN layers	[1 to 5]	2	4
CNN kernel size	[2 to 5]	3	3
TCN-DPA			
Learning rate	[1e–5 to 1e–2]	0.0006	0.003
Number of Layers	[1 to 5]	4	4
Kernel Size	[2 to 5]	2	2
Dropout	[10^{-5} to 0.5]	0.17	0.01
StepLR Step Size	[10 to 50]	21	17
StepLR Gamma	[0.5 to 1]	0.9	0.8
Epochs	[0 to 200]	156	100
STA-MLP			
Learning rate	[1e–5 to 1e–2]	0.001	0.001
Number of layers	[1 to 5]	2	3
Size of hidden layer 0	[5 to 30]	29	15
Size of hidden layer 1	[5 to 30]	27	15
Size of hidden layer 2	[5 to 30]	N/A	15
Total training epochs	[50 to 500]	133	300
TRFI-MLP			
Learning rate	[1e–5 to 1e–2]	0.0004	0.001
Number of layers	[1 to 5]	3	3
Size of hidden layer 0	[5 to 30]	23	15
Size of hidden layer 1	[5 to 30]	29	15
Size of hidden layer 2	[5 to 30]	21	15
Total training epochs	[50 to 500]	130	160
LSTM-DPA-TA			
Learning rate	[1e–5 to 1e–1]	0.004	0.01
LSTM size	[64 to 128]	128	128
StepLR step size	[1 to 50]	35	10
StepLR step gamma	[0.1 to 1]	0.7	0.8
Training epochs	[50 to 500]	160	500

4 Results

This section evaluates the performance of channel estimators using BER as the primary metric. We compare the effectiveness of two training approaches: one using a mixed SNR dataset and the other using a high SNR dataset. The evaluated architectures are CNN-Transformer, TCN-DPA, STA-MLP, TRFI-MLP, and LSTM-DPA-TA.

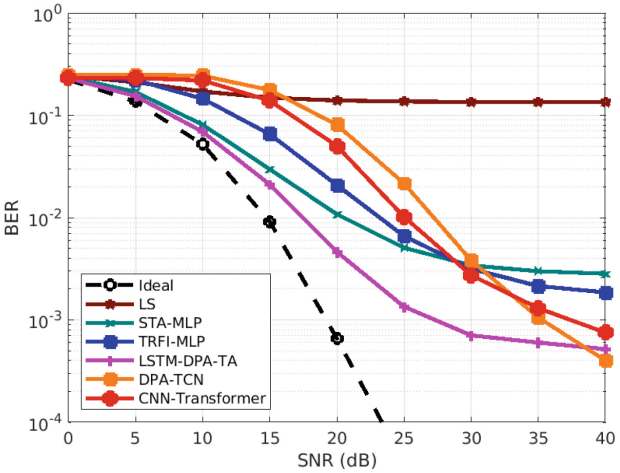
4.1 BER

BER is a crucial performance measure in digital communication. It compares transmitted and received bit sequences to assess system reliability. In channel estimation, a lower BER means more accurate signal reconstruction and fewer errors in decoded data. The following analysis details the BER performance of each model at different levels of SNR, providing a direct comparison between the two training approaches.

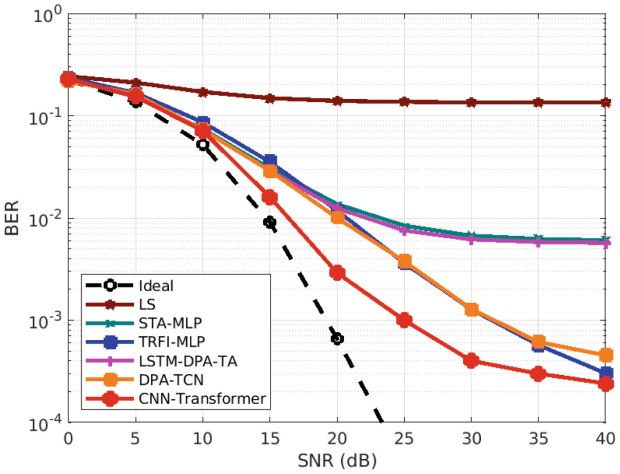
Figure 3 presents the BER performance of the channel estimators trained using a high SNR dataset (40 dB) and a mixed SNR dataset, respectively. The ideal curve, represented by the dotted black line in these figures, represents the best possible performance in channel estimation, where the received signal is assumed to be perfectly equalised without any channel estimation errors. This curve serves as a theoretical lower bound for the BER, showing the performance in a scenario where only additive white Gaussian noise (AWGN) is present and the channel effects are perfectly compensated for. The closer the BER of a model is to the ideal curve, the better its performance in accurately estimating the channel and mitigating the effects of noise. LS serves as the lower performance baseline.

In Fig. 3a, we observe several key trends across models trained on an high SNR dataset: In the low SNR range (0–10 dB), the STA-MLP and the LSTM-DPA-TA models perform relatively well, maintaining a lower BER compared to other models. As we move into the mid SNR range (15–25 dB), the CNN-Transformer and DPA-TCN models show an improvement, particularly from 15 dB onwards, with performance becoming more pronounced from 20 dB. This improvement at high SNR highlights that these models have effectively adapted to high SNR conditions. The TRFI-MLP model also improves in this range, although it lags behind the CNN-Transformer and DPA-TCN models, indicating some limitations in its ability to adapt to low SNR conditions. In the high SNR range (30–40 dB), the CNN-Transformer and DPA-TCN models continue to show further significant reductions in BER. The LSTM-DPA-TA model continues to outperform other models across all SNR levels, showing its effectiveness in less noisy conditions. In contrast, STA-MLP shows limited improvement beyond 20 dB, indicating that it struggles to capitalise on the reduced noise at higher SNR levels.

In the mixed SNR training scenario depicted in Fig. 3b, we observe that CNN-Transformer and DPA-TCN models exhibit excellent performance across the entire SNR range with CNN-Transformer outperforming all models. The



(a) High SNR training dataset



(b) Mixed SNR training dataset

Fig. 3. Comparison of BER performance for various channel estimators trained on high SNR and mixed SNR datasets.

LSTM-DPA-TA and STA-MLP models perform well at lower SNR levels (0 to 15 dB) but show reduced performance as the SNR increases, indicating diminished capabilities in high SNR conditions when trained on mixed SNR data. We can also observe the improved performance of the TRFI-MLP estimator across the entire SNR range tested compared to its performance when trained on the high SNR dataset. The unsatisfactory performance by LSTM-DPA-TA

and STA-MLP indicates that while these models can handle varying noise conditions, they may not leverage mixed SNR training as effectively as TRFI-MLP, CNN-Transformer and DPA-TCN.

Models trained on mixed SNR data exhibit lower BER even in low-SNR scenarios, compared to those trained on high SNR data. Mixed SNR training appears to be beneficial for models such as TRFI-MLP, TCN-DPA, and CNN-Transformer, which show a significant improvement in low SNR conditions than when trained on high SNR. In contrast, the LSTM-DPA-TA and STA-MLP models demonstrate better performance when trained on a high SNR dataset, suggesting that these models make use of the better channel statistics available at high SNR. This is analysed in more detail below.

4.2 Difference in BER Between Models Trained on Mixed SNR and High SNR Datasets

Figure 4 illustrates the difference in BER between models trained on high SNR datasets and those trained on a mixed SNR dataset. The graph clearly shows how the training dataset influences the performance of various models across different SNR ranges. In the low SNR range (0–10 dB), the CNN-Transformer and TCN-DPA models exhibit the highest positive delta, peaking around 10 dB. This indicates a substantial improvement in performance when these models are trained on a mixed SNR dataset compared to a high SNR dataset. A positive delta suggests that mixed SNR training better equips these models to generalise in low SNR conditions. TRFI-MLP also shows a noticeable positive delta, although less pronounced, indicating that mixed SNR training offers some advantages in low SNR environments. In contrast, the LSTM-DPA-TA and STA-MLP models display a slightly negative delta across this range, implying that training on high SNR datasets offers a marginal performance advantage in these specific models.

As the SNR increases (15–25 dB), the positive delta values for the CNN-Transformer and TCN-DPA models start to decrease but remain positive, particularly around 15 dB. This indicates that while the benefits of mixed SNR training diminish slightly, these models still gain performance advantages in this SNR range. TRFI-MLP continues to maintain positive delta values, demonstrating that mixed SNR training consistently improves its performance at varying SNR levels. Meanwhile, the LSTM-DPA-TA model continues to show a negative delta, particularly at the higher end of the mid-SNR range, around 20–25 dB. Similarly, the STA-MLP model exhibits a slight negative delta in this range.

In the high SNR range (30–40 dB), the delta values for most models, including the CNN-Transformer, TCN-DPA, and TRFI-MLP, approach zero. This suggests that as the SNR increases and the impact of noise decreases, the performance difference between models trained on high SNR and mixed SNR datasets becomes negligible.

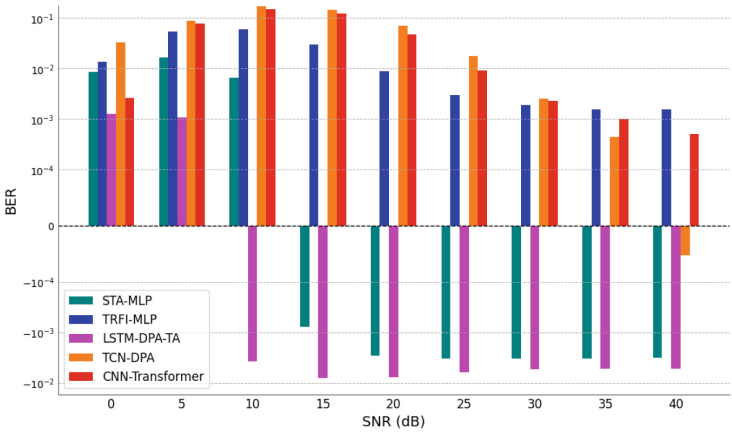


Fig. 4. Difference in BER between models trained on mixed SNR and high SNR datasets.

5 Conclusion

This study investigated the effectiveness of training NN-based channel estimators using mixed SNR datasets compared to high SNR datasets. Our results demonstrate that training with mixed SNR data significantly improves the generalisation of various estimators, especially in low SNR conditions. In particular, models such as the CNN-Transformer, DPA-TCN and TRFI-MLP exhibited substantial improvements in BER across the entire SNR range when trained on mixed SNR data compared to when trained on high SNR. Among the models tested, the CNN-Transformer, when trained on mixed SNR data, outperformed other estimators, including the current state-of-the-art LSTM-DPA-TA. However, it is important to note that some models, such as LSTM-DPA-TA and STA-MLP, showed reduced performance when trained on mixed SNR data. Mixed SNR training improves performance and generalisation across SNR levels for some models. The exact reason why only certain models benefit remains unclear and requires further investigation.

These results indicate the importance of considering the SNR range as an important hyperparameter during training, rather than following the current practice of using only high SNR training data. The channel model used in this study is an example where the impact of mobility on the channel is more significant compared to other channels with lower mobility. As a result, the tested models should be able to generalise well to those channels. This will be verified in future research that will investigate the impact of mixed SNR training on other vehicular channel models.

Acknowledgments. The authors are grateful to NITheCS and the Telkom CoE at NWU for their support of this research.

References

1. Akiba, T., Sano, S., Yanase, T., Ohta, T., Koyama, M.: Optuna: a next-generation hyperparameter optimization framework. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2623–2631 (2019)
2. Fernandez, J.A., Borries, K., Cheng, L., Vijaya Kumar, B.V.K., Stancil, D.D., Bai, F.: Performance of the 802.11p physical layer in vehicle-to-vehicle environments. IEEE Trans. Veh. Technol. **61**(1), 3–14 (2012). https://doi.org/10.1109/TVT.2011.2164428
3. Gizzini, A.K., Chafii, M., Ehsanfar, S., Shubair, R.M.: Temporal averaging LSTM-based channel estimation scheme for IEEE 802.11p standard. In: 2021 IEEE Global Communications Conference (GLOBECOM), pp. 01–07. IEEE (2021)
4. Gizzini, A.K., Chafii, M., Nimr, A., Fettweis, G.: Deep learning based channel estimation schemes for IEEE 802.11p standard. IEEE Access **8**, 113751–113765 (2020)
5. Gizzini, A.K., Chafii, M., Nimr, A., Fettweis, G.: Joint TRFI and deep learning for vehicular channel estimation. In: 2020 IEEE Globecom Workshops (GC Workkshops), pp. 1–6. IEEE (2020)
6. Han, S., Oh, Y., Song, C.: A deep learning based channel estimation scheme for IEEE 802.11p systems. In: ICC 2019-2019 IEEE International Conference on Communications (ICC), pp. 1–6. IEEE (2019)
7. Jiang, D., Delgrossi, L.: IEEE 802.11p: towards an international standard for wireless access in vehicular environments. In: VTC Spring 2008 - IEEE Vehicular Technology Conference, pp. 2036–2040 (2008). https://doi.org/10.1109/VETECS.2008.458
8. Kim, Y.K., Oh, J.M., Shin, Y.H., Mun, C.: Time and frequency domain channel estimation scheme for IEEE 802.11p. In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 1085–1090. IEEE (2014)
9. Li, Y.: Pilot-symbol-aided channel estimation for OFDM in wireless systems. IEEE Trans. Veh. Technol. **49**(4), 1207–1215 (2000). https://doi.org/10.1109/25.875230
10. Ngorima, S.A., Helberg, A., Davel, M.H.: A data pilot-aided temporal convolutional network for channel estimation in IEEE 802.11p vehicle-to-vehicle communications. In: Southern Africa Telecommunication Networks and Applications Conference (SATNAC) (2024)
11. Ngorima, S.A., Helberg, A., Davel, M.H.: Hybrid CNN-transformer model for channel estimation in vehicular communications. In: Deep Learning Indaba Conference (2024)
12. Pan, J., Shan, H., Li, R., Wu, Y., Wu, W., Quek, T.Q.: Channel estimation based on deep learning in vehicle-to-everything environments. IEEE Commun. Lett. **25**(6), 1891–1895 (2021)

Deep Learning-Based Network Intrusion Detection Systems: A Systematic Literature Review

Leonard L. Mutembei[1,2(✉)], Makhamisa C. Senekane[1,3], and Terence van Zyl[1,2]

[1] Academy of Computer Science and Software Engineering, University of Johannesburg, Johannesburg, South Africa
{lmutembei,smakhamisa,tvanzyl}@uj.ac.za
[2] Institute for Intelligent Systems, University of Johannesburg, Johannesburg, South Africa
[3] National Institute for Theoretical and Computational Systems, Stellenbosch, South Africa

Abstract. Machine Learning algorithms have been used to develop models in different fields like banking, healthcare, transportation, cybersecurity, and others. Many studies have shown the use of Machine Learning in network security by detecting network intrusions. However, the increase in user devices increases the data size and hence increases the complexity of the network. Cyber attackers continue to create advanced cyber attacks, and identifying them becomes extremely challenging. On the other hand, traditional Machine Learning models cannot efficiently handle large amounts of data and complexity. Therefore, this study examines how Deep Learning methods can be implemented for the Network Intrusion Detection Systems. The Network Intrusion Detection System (NIDS) helps to secure businesses within companies' networks from bad actors. As Deep Learning advances, network security experts must incorporate the techniques within the NIDS to minimize the effects of cyber attacks. For the investigation of the Deep Learning techniques in implementing NIDS, a study used Preferred Reporting Items for Systematic reviews and Meta-Analysis (PRISMA) framework to conduct a systematic literature review and summarizes 111 studies published between 2021 and November 2023. Articles were analyzed by categorizing them into Deep Learning algorithms, architectures and datasets. The performance comparisons have been evaluated based on available articles' results on various datasets. Methods have been compared based on precision, recall and F1 score metrics. The results revealed that the most commonly used datasets are CICIDS2017, CSE-CIC-IDS2018, CICDDoS2019, NSL-KDD and UNSW-NB15. Furthermore, it was observed that hybrid implementation approaches have been shown to produce accurate and robust models compared with traditional methods. Finally, the results further demonstrated the need to explore using Machine Learning in NIDS.

Keywords: Artificial Intelligence · Deep Learning · Machine Learning · Network Intrusion Detection System · PRISMA

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 207–234, 2025.
https://doi.org/10.1007/978-3-031-78255-8_13

1 Introduction

Connected devices and users are exponentially increasing because of the advancement of technology. However, criminals use the opportunities by creating and conducting cyber attacks [100]. These cyber attacks have also attacked most people and companies and caused financial losses [92,100]. The number of Internet users expected by 2023 was 5.3 billion, and the number of connected devices was 3.6 billion [23]. With the increase of users and devices on the Internet, our computer networks become more vulnerable to cyber attacks [9]. Despite the increased research on advanced technologies, cybersecurity threats still need to be resolved.

The prevailing cyberspace leads to the protection of connected devices within networks being important. Cybersecurity is critical to secure companies' businesses. Cyber attacks are rapidly growing with the pace of devices connected. Businesses are losing revenues because of attacks that prevent the economy's growth. Therefore, network infrastructure needs to be more resilient to cyber attacks. Many studies have shown that cybersecurity is critical in this Internet era. Different tools have been implemented to secure the company's networks, including Intrusion Detection Systems (IDS). IDS technologies play an essential role in network security by analyzing all incoming and outgoing network traffic. IDS detects unauthorised and malicious activities [52,129]. Despite the development of IDS technologies, some limitations need to be addressed by using advanced techniques.

IDS can be categorized into different areas, such as host-based IDS (HIDS) and network-based IDS (NIDS) [11]. All categories of IDS have challenges, but to secure the network, the NIDS are needed. Network security experts must monitor networks every time to ensure that security is available throughout. Despite the new approaches used in NIDS, challenges are still available. The first challenge is that the amount of data keeps growing as the number of devices and users exponentially increases. The volume of data by 2025 is predicted to be 175 Zettabytes (ZB) from 33 ZB in 2018 [85]. Thus, advanced deep learning-based NIDS models are needed because attacks are becoming complex as the technologies advance. An IDS is vital in providing network security by detecting unauthorised and malicious activities [44]. These systems can be hardware or software designed to analyse network traffic patterns and identify malicious activity [44]. However, the detection rates of existing IDS could be more reasonable and overwhelm the enormous effort associated with test data [8]. Machine Learning methods and optimization algorithms are often used in IDS to address these challenges.

Machine Learning (ML) and Deep Learning (DL) [66], among others, are the approaches that are used to combat cyber attacks. These techniques not only help in cybersecurity but are also used in other sectors like healthcare, finance, telecommunication, and manufacturing, a few to mention, to facilitate and improve workflow. Supervised ML approaches have been widely studied and have proved valuable in intrusion detection [38]. These approaches involve training a model on labelled data to classify network traffic as normal or abnormal. Different ML algorithms, such as decision trees, Support Vector Machines (SVMs), and neural networks, have been used for this purpose [7]. It has been found that using ML approaches enhances the detection accuracy of IDS and reduces the computational cost [7].

The importance of IDS as an essential element in cybersecurity frameworks has been consistently highlighted over the past few decades. This emphasis is evident in scholarly works such as [27,60]. Throughout history, the prevailing approach of these systems has primarily relied on traditional ML methodologies [28]. Nevertheless, due to the swift progress in technology and the widespread emergence of complex cyber threats, there has been a notable shift in focus toward integrating cutting-edge methodologies, particularly advanced ML.

The emergence of DL techniques in developing NIDS has been extensively explored in recent academic research. [95,120] have made significant scholarly contributions. These studies shed light on the growing prominence of DL techniques in constructing resilient and efficient NIDS, particularly when compared to traditional ML approaches. However, a significant deviation becomes apparent when examining the use of DL in the NIDS field: a noticeable avoidance of time series data by formatting datasets with a time sliding window. As mentioned earlier, the oversight can potentially obscure important temporal observations, thus diminishing the thorough examination of cyber incidents.

Many studies have shown the use of traditional ML in implementing different IDS, and performance kept increasing as the ML kept advancing. Different DL approaches have been used with traditional ML, which performed better than IDS, which used only classical ML. However, a few systematic literature reviews (SLRs) can be found within the domain of ML and IDS. Also, SLRs lack the information needed by researchers to understand the current situation and implement new models. For example, existing SLRs are not the latest and are focused on multiple categories of IDS. Thus, this work is motivated by previous studies to perform SLR on current DL models used for implementing NIDS. Thus, this study aims to review the DL techniques used in NIDS systematically. As such, the study will answer the following research questions (RQ):

- RQ1: What is the current state of the deep learning algorithms used to implement NIDS?
- RQ2: What architectures in RQ1 were used to implement NIDS?
- RQ3: What DL approaches are used to implement NIDS with time series formatting on datasets?

The study's main contributions are: We used the PRISMA framework to select all studies based on ML, DL and NIDS. This work analyzed studies within three years, from 2021 to November 2023. Lastly, this work shows the five most used network datasets for implementing DL-based NIDS.

The rest of the paper is organized as follows: Sect. 2 describes the methodology used, Sect. 3 depicts the results and discussion, and Sect. 4 describes the conclusion.

2 Methodology

The study implements a systematic literature review by applying state-of-the-art methods [48,81]. PRISMA [81] was used to formulate the work's structure, as shown in Fig. 1. The study used different methods to ensure the work can

be replicated. The literature search used inclusion criteria related to intrusion detection systems, and all articles written in English were assessed. Different keywords were used to identify literature from public scientific databases Scopus, Web of Science and IEEE Explore. The study used the publication date from 2021 to 2023 as the inclusion criteria for the relevant material on DL techniques. The study only used peer-reviewed articles from the Association for Computing and Machinery (ACM), the Institute of Electrical and Electronics Engineers (IEEE), Elsevier, and Springer as publishers.

Four steps were used to conduct the study. Step one was formulating the search keywords and queries in the identified databases. Step two was to find relevant articles according to formulated criteria. Step three was to read selected articles regarding their methodologies and results found. Finally, step four was to review and present the results obtained from the selected articles.

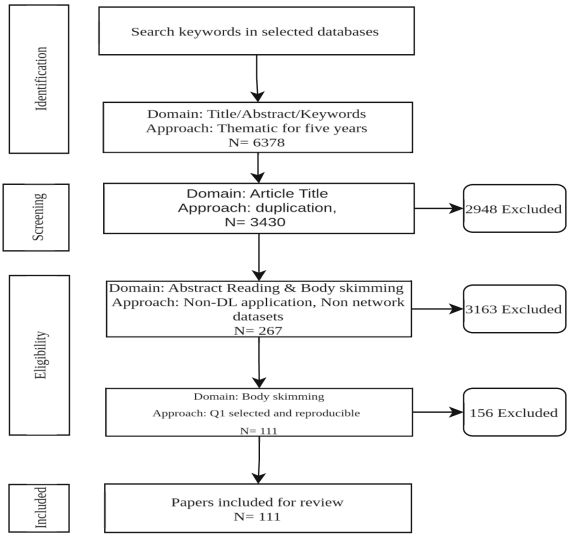


Fig. 1. Flow chart methodology.

2.1 Search Strategies

Different keywords were combined to search for the articles that were needed. The study used logical operators of 'OR' and 'AND' to add keywords. Different keywords used by the study are: 'machine learning', 'deep learning', 'deep neural network', 'artificial neural network', 'intrusion detection system', 'network intrusion detection system', 'anomaly detection', 'outlier detection', 'network anomaly detection', 'convolutional neural network', 'CNN', 'long short-term memory', 'LSTM', 'bidirectional long short-term memory', 'BiLSTM', 'gated recurrent unit', 'GRU', 'bidirectional gated recurrent unit', 'BiGRU', 'deep reinforcement learning', 'generative model', 'autoencoder', 'generative adversarial network', 'GAN' and 'transformer'.

2.2 Study Selection Strategies

Different selection strategies were used to make the selected articles relevant. The following are the ways used for inclusion and exclusion criteria:

- All surveyed papers were excluded.
- All articles related to HIDS were excluded.
- All studies that used only Internet of Things (IoT) datasets were excluded from the study.
- Studies that used only synthetic data were excluded because it becomes difficult to validate their results compared to publicly known datasets.
- All articles related to NIDS were included.
- All articles with at least one network dataset were included.
- Hybrid models of DL and traditional ML were included.
- All articles that fulfil the criteria of reproducibility were included.

2.3 Quality Assessment

All articles selected were assessed for the quality of the final review. The journal impact factor was considered, and all articles with quartile one (Q1) were selected. Furthermore, the study used all articles indexed within the Web of Science research platform.

2.4 Machine Learning Reproducible

Reproducibility is essential in ML models because it provides a way to validate results within training conditions. [34] and [89] stipulated that for the model to be reproducible, there should be clear information about methodologies, hyper-parameters and datasets. The present study excluded studies that were not transparent regarding methodology, parameters and datasets. To compare the DL model performance well, each model must use the same network topology, number of hidden layer units, learning parameters, and other features [22].

3 Results Analysis

We selected 111 articles for further analysis after using different criteria, as shown in Fig. 1. All studies were selected from 2021 to November 2023.

3.1 Deep Learning Algorithms Used in NIDS

The selected articles applied several DL algorithms with a few classical ML to implement NIDS models. Figure 2 answers RQ1 by summarizing the algorithms utilized within different studies. Algorithms used are: Auto Encoder (AE), Bidirectional Long Short-Term Memory (BiLSTM), Convolutional Neural Networks (CNN), Deep Reinforcement Learning (DeepRL), Deep Neural Networks (DNN),

Feedforward Neural Networks (FNN), Generative Adversarial Networks (GAN), Graph Neural Networks (GNN), Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Multi-Layer Perceptron (MLP), Recurrent Neural Networks (RNN), and Transformer. The CNN method led with the highest number of studies, followed by AE, while the least methods are RNN and the FFN.

3.2 Implemented Architectures

Different architectures have been used to implement different IDS. Many studies have shown that implementing hybrid models results in better performance. Combining more than one DL algorithm or using DL and classical ML algorithms has been shown as the solution to enhance detection rate and robustness. Table 1 answers RQ2 by illustrating the summary of most architectures used within different models.

Research on the application of DL in NIDS has evolved. Several implementation architectures have been proposed (see Table 2: summary of the implemented architectures). The analysis of the model architectures demonstrates that hybrid approaches provide promising results if more experiments are taken in this direction. Hybrid models are known for their accuracy compared to individual classifiers when implemented well.

3.3 Experimental Environments

Model implementation can be done in different environments. Studies show that 11.7% used the Google Colab platform, 0.9% used the Kaggle platform, 0.9% used both their local machine and Google Colab, and 71.2% used their local machines. However, 15.3% of studies did not provide any information about the working environments, which may lead to difficulty in reproducing the model results. Figure 3 summarises usable working environments.

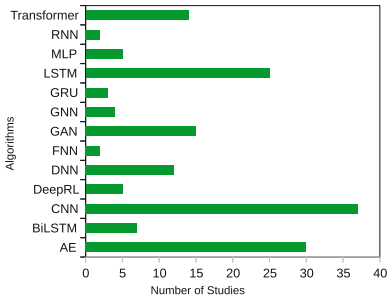


Fig. 2. Deep Learning Algorithms used.

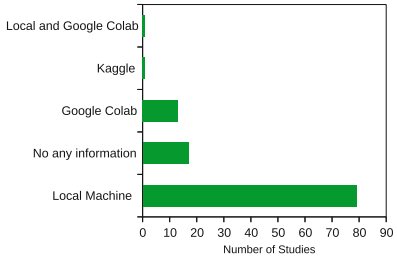


Fig. 3. Working Environment.

Table 1. Implemented Architectures

Algorithm	Model Architecture	References
AE	The use of AE, stacked AE, stacked contractive AE, denoising AE with other DL algorithms and classical ML	[2,4,13,15,24,40,50,53,57,59,61,62,69,71,73–76,86,88,104,105,110,112,115,119,121,122,131,132]
BiLSTM	The use of BiLSTM, word embedding with BiLTSM, alone or hybrid to other algorithms	[14,19,22,43,45,109,116]
CNN	The use of CNN, word embedding with CNN, or with other algorithms	[1–3,5,6,12,14,19,20,22,32,37,39,42,47,56,58,61,67,80,82,83,90,91,94,97,99,106,113,114,117,122,124,125,127,133,134]
DeepRL	The use of adversarial RL and variant of DeepRL	[18,30,63,72,78]
DNN	The use of DNN, DNN with classical ML, optimization techniques, and other DL algorithms	[2,17,36,46,49,53,70,74,87,93,102,103]
FNN	The use of FNN with conditional GAN and tree deep method	[25,107]
GAN	The use of GAN, BiGAN or conditional GAN with other DL algorithms	[12,29,33,49,50,57,61,68,79,93,96,107,117,119,123]
GNN	The utilization of GNN like GCN and NEspsGCN	[10,21,26,135]
GRU	The use of GRU and combined with LSTM and AE	[75,77,130]
LSTM	The use of LSTM, hybridization with attention mechanisms and other DL algorithms	[2,5,15,16,31,35,37,40,47,49,51,52,56,58,64,68,76,77,82,88,90,97,98,126,128]
MLP	The use of MLP with attention mechanisms and other DL algorithms	[46,59,82,84,131]
RNN	The use of RNN and attention mechanism and hybridization with DNN and MLP algorithms	[46,54]
Transformer	The use attention approaches with LSTM, CNN, MLP, RNN, transformer or vision transformer with other algorithms	[1,41,51,54–56,84,109,111,113,116,118,123,125]

3.4 Feature Selection Techniques

NIDS models depend much on network traffic patterns to detect cyber attacks. Feature engineering approaches have been implemented in a few studies, while many used the advantage of DL algorithms, which select and extract features automatically. However, most studies combine feature selection mechanisms to identify relevant features. Some studies managed to provide the number of features that have been selected compared to the original features. However, other studies did not provide the number of features used during the experiment, which made it difficult to reproduce the study.

Most DLs used to perform feature extraction are CNN, LSTM, AE, Stacked Sparse Auto-Encoder (SCAE), DNN, and attention mechanisms. This shows how DL can be utilized in NIDS implementation without the feature selection tech-

niques needed for traditional ML. Datasets can be formatted to allow required features to be extracted and used within the detection models. [86] showed how time-based features could improve the detection rate of NIDS compared to the most available flow-based statistical feature models. The study managed to detect zero-day attacks. Furthermore, due to sophisticated attacking techniques, zero-day attacks are the most dangerous intrusions for the networks. However, most available NIDS can not handle zero-day attacks [86].

3.5 Datasets

According to RQ3, many reviewed articles have shown different datasets used during NIDS implementations. Datasets can be categorized into private and public categories. For security reasons, most private datasets are not shared within the public repositories. For the model to be robust, researchers must use more than one dataset to test and validate their models. Some studies used only one dataset to validate their models. However, more than one dataset is needed to generalize the model.

Furthermore, different datasets within the same domain are not equal due to variations in the working environment. Thus, experiments can attain a high detection rate in one dataset, while the performance becomes poor in other datasets. Different advantages like generalization, minimizing overfitting and robustness can be attained when more than one dataset is used [40].

There is a problem with insufficient datasets to provide source files, such as packet capture files (.pcap). Also, normal traffic is much higher than malicious traffic, affecting models' prediction and forecast results. Most studies have shown how the datasets were preprocessed before being applied to the models. Datasets can be transformed by using sliding windows and time to perform cyber attack forecasting [86]. It is worth noting that network attacks are advancing with time. Therefore, some of the datasets need to be updated. The analysis of datasets shows their prominence across years (see Table 2- summary of the most usable datasets within different studies). Thus, all articles used NSL-KDD, CICIDS2017, UNSW-NB15, CSE-CICIDS2018 and CICDDoS2019 datasets were analyzed.

3.6 Datasets Network Attacks

The five most usable NIDS datasets within three consecutive years, from 2021 to November 2023, as depicted in Table 2, have used different network attacks. Table 3 depicts categories of attacks analyzed within different studies. It shows the need for common standards of network attack types researchers use during NIDS Machine Learning implementation. There is much similarity between Denial of Service (DoS) attacks within the four datasets of NSL-KDD, UNSW-NB15, CICIDS2017, and CSE-CICIDS2018, while CICDDoS2019 is based on categories of Distributed Denial of Service (DDoS) attacks. Researchers have shown the importance of DoS and DDoS attacks and how bad they are within the networks. They must be detected as early as possible using different techniques with DL [65,101].

Table 2. Datasets most used

Dataset	Number of Studies	Year Used
NSL-KDD	48	2021, 2022, 2023
CICIDS2017	46	2021, 2022, 2023
UNSW-NB15	41	2021, 2022, 2023
KDDcup99	18	2021, 2022, 2023
CSE-CICIDS2018	12	2021, 2022, 2023
CICDDoS2019	9	2021, 2022, 2023
BoT-IoT	9	2021, 2022
CICIDS2018	5	2022, 2023
CIDDS-001	5	2021, 2022, 2023
Synthetic dataset	5	2021

Table 3. Network Attacks

Datasets	Attacks
NSL-KDD	DoS, Probe, Remote to Local (R2L), User to Root (U2R)
UNSW-NB15	Analysis, Backdoor, DoS, Exploits, Fuzzers, Generic, Reconnaissance, Shellcode, Worms
CICIDS2017	Botnet, Bruteforce, Bruteforce-FTP, Bruteforce-SSH, DDoS, DoS, DoS-goldeneye, DoS-hulk, DoS-slowhttptest, DoS-slowloris, FTP-patator, Goldeneye, Heartbleed, hulk, Infiltration, Patator, Portscan, Slowhttp, Slowhttptest, Slowloris, SSH-patator, SSH-patator, SSH-patator, Webattack, Webattack-bruteforce, Webattack-SQL-injection, Webattack-XSS
CSE-CICIDS2018	Botnet, Bruteforce, Bruteforce-FTP, Bruteforce-SSH, Bruteforce-Web, Bruteforce-XSS, DDoS, DDoS-hoic, DDoS-loic-http, DDoS-loic-udp, DDoS-udp, DoS, DoS-goldeneye, DoS-hulk, DoS-slowhttptest, DoS-slowloris, FTP-patator, Infiltraton, SQL-injection, SSH-patator, Webattack
CICDDoS2019	Bot-Ares, DDoS, DDoS-goldeneye, DNS, LDAP, MSSQL, NetBIOS, NTP, PortMap, Portscan, SNMP, SSDP, Syn, TFTP, UDP, UDP-Lag, WebDDoS, XSS

Most studies showed how NIDS datasets are imbalanced and suggested different approaches to solve the problem. Oversampling and undersampling approaches are used in different studies to solve imbalanced datasets. Minority classes led to poor performance during multi-class classification. For example, NSL-KDD has Root to Local (R2L) and User to Root (U2R), which are minimal compared to other attack classes. More studies are needed to ensure minority classes are detected with better performance than majority classes. Malicious samples are a minority in network traffic datasets, whereby some studies have used oversampling techniques and managed to improve the performance of some experiments. The observation shows that balanced datasets can increase or decrease attack detection rates.

In a real network environment, normal and abnormal samples within the datasets can not be balanced. However, using GANs with different approaches has been demonstrated to solve imbalance problems and may improve the NIDS model's performance [96]. Data generated within the known dataset may reduce the detection rate, too, as shown in a study by [119]. However, other studies showed the success of data balancing in the detection rate using SMOTE and Tomek-Link [5]. The study obtained the precision, recall and F1 score of 99%.

Furthermore, [123] used conditional GAN to solve the imbalanced data, and the results in one dataset seem small compared to other existing studies that do not use imbalanced approaches. Therefore, there is a need for further investigation on how to use imbalanced datasets while still improving detection rates and reducing false alarm rates. Also, some studies did not specify whether the performance measurement results were based on the test or training data. Studies need to specify the performance measurement using test data or training data. However, few studies provided the data for the reported results; good examples are [32,99].

3.7 NIDS Classification Techniques

Detecting cyber attacks within networks can be difficult and time-consuming. Thus, classification approaches can be binary classification or multi-class classification. The mechanism will be binary classification when identifying two groups within network traffic, which are normal or malicious. For multi-class classification, the output will be more than two categories of network traffic, normal and other types of attacks. For the study to be more understandable, the researcher must explain which classification approach has been used. However, other studies still need clear information about the experiment category for binary or multi-class classification. Lack of clear information about the study may make replicating the experiment difficult. (See Figs. 4, 5, 6, 7, 8, and 9 – summary of comparison results based on precision, recall and F1 score for binary and multi-class classification based on specific datasets).

3.8 Evaluation Metrics

Model evaluation is the most critical area in reporting the model's performance. However, reviewed articles showed the need for standardized metrics for the NIDS. Most evaluation metrics have been used interchangeably, leading to confusion during analysis for reproducible papers. Some studies reported the detection rate and recall as the same metric, while others showed different metrics. For example, sensitivity, also known as recall or detection rate, specificity, F1 score, and Area Under the ROC curve (AUC-ROC). Positive predictive value (PPV), also known as precision; negative predictive value (NPV), accuracy, balanced accuracy, the area under the precision-recall curve (PR-AUC), also known as average precision and Matthews Correlation Coefficient (MCC) that can be used to evaluate NIDS's model performance [108]. Most evaluation metrics used within reviewed studies are accuracy, precision, recall (or sensitivity, detection rate), and F1 score. However, few studies used False Alarm Rate (FAR), AUC, specificity, and MCC.

4 Observation

A hybrid of DL and classical ML approaches for NIDS implementation can improve the detection rate performance compared to a single algorithm. Results reporting within the NIDS models revealed the need for researchers to follow a standardized way to present results. Most multi-class classification studies take time to understand their output performance due to a lack of clear ways of reporting. Studies with confusion matrices made it easy for researchers to calculate missing information like precision, recall and F1 score.

The best evaluation metrics for NIDS are precision, recall, F1 score and FAR. However, very few studies' experiments reported on FAR, making comparing studies with this metric difficult. The analysis shows that 85.8% of studies have F1 scores, 82.2% have precision, 87.3% have recall, and 26.9% of studies reported FAR values. Therefore, F1 score, precision and recall have been used to find the best implementation architectures of the NIDS.

4.1 Dataset Formatting

Datasets need to be preprocessed before being used within the model. However, some studies did not show how their datasets were preprocessed [50,52,57,67,78,79,96,130]. Datasets can be transformed by using sliding windows and time to perform cyber attack forecasting. Few studies have shown time-series formatting, while most other studies used datasets without further formatting. [86] used the CICIDS2019 dataset and showed how time-window features perform better than flow-based features. They used AE as a binary classifier of DDoS attacks and attained more than 99% of F1 score on most attacks. [68] used ten (10) time slots on each time window by implementing a bidirectional GAN with the LSTM to detect anomalies within CICIDS2017 and obtained precision of 93.87%, recall of 94.02% and F1 of 93.94%. [54] proposed a Bidirectional Independent Recurrent Neural Network (BiIndRNN) with Global Attention (GA) on UNSW-NB15 with the time steps from 1 to 11. The best results obtained are a precision of 99.71%, recall of 98.69% and F1 score of 99.15% with a FAR of 0.36%.

[83] implemented the hybrid of Random Forest (RF), CNN and Dempster-Shafer Theory (DST) with detection time in multi-class classification on CICIDS2017. The model obtained a precision of 99.4%, recall of 99.96% and F1 score of 99.68%. [135] implemented a Graph Convolutional Network (GCN) to solve the class imbalance in binary classification on the UNSW-NB15 dataset. They transform network traffic by time intervals into graphs. However, the precision of 60%, recall of 93% and F1 score of 73% are lower than other studies. More experiments can be done with other methods to find the correlation of time in cyber forecasting. [26] implemented GCN by correlating time within traffic patterns on three different datasets. However, the best result was on the UNSW-NB15 dataset with a precision of 98.66%, recall of 98.65% and F1 score of 98.65% compared to other methods. [57] developed DAE-GAN for binary classification by training the model with only normal data samples and a time window size of 1000. Furthermore, they obtain a precision of 98.6%, a recall of 81.9% and an

F1 score of 89.5% on NSL-KDD while a precision of 98.54%, a recall of 69.38% and an F1 score of 81.43% on UNSW-NB15.

4.2 Best Architectures in Binary Classification

The ten best architectures from NSL-KDD, UNSW-NB15 and CICIDS2017 datasets are chosen, but due to a small number of studies in CSE-CICIDS2018 and CICDDoS2019 datasets, all architecture results are presented. The NSL-KDD dataset attained performance from 96.08% to 99.81% of F1 score, 97.62% to 99.94% of precision, and 96.6% to 99.94% of recall. In the UNSW-NB15 dataset, the F1 score was from 98% to 99.66%, precision from 97.93% to 100%, and recall of 97.64% to 99.71%. In the CICIDS2017, the best architectures have a 99.35% to 99.996% of F1 score, 98.28% to 99.996% precision, and 99.15% to 99.996% recall. In the CSE-CICIDS2018, F1 score of 92.7% to 99.29%, precision of 93.3% to 99.44%, and recall of 92.3% to 99.15%. While in the CICDDoS2019 dataset, F1 score of 95.94% to 99.84%, a precision of 94.08% to 99.8%, and a recall of 97.43% to 100%. Hybridization of two or more DL methods with attention mechanisms performed better throughout the datasets used, as depicted in Figs. 4, 5 and 6.

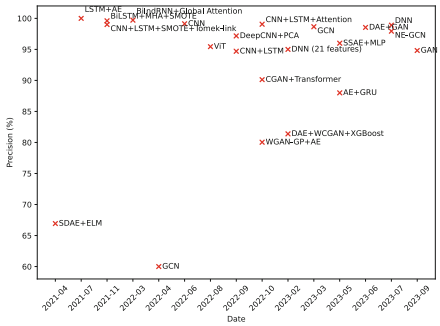
4.3 Best Architectures in Multi-class Classification

The ten best architectures from NSL-KDD, UNSW-NB15 and CICIDS2017 datasets are chosen, and due to a small number of studies in CSE-CICIDS2018 and CICDDoS2019 datasets, all architecture results are presented. The NSL-KDD dataset attained 91.46% to 99.93% of the F1 score, 92.13% to 99.93% of precision, and 94% to 99.93% of recall. In the UNSW-NB15 dataset, the F1 score was from 84% to 98.95%, precision from 82.4% to 100%, and recall of 82.6% to 98.99%. In the CICIDS2017, the best architectures have a 99.46% to 100% F1 score, a 99.49% to 100% precision score, and a 99.62% to 100% recall score. In the CSE-CICIDS2018, F1 score of 90% to 99.99%, precision of 93% to 99.99%, and recall of 90% to 99.99%. In the CICDDoS2019 dataset, the F1 score was 98.48% to 99.29%, the precision from 98.82% to 99.6%, and the recall from 98.7% to 99%. Hybridization of two or more DL methods, ensemble transfer learning and multi-modal deep transfer performed better on the specific datasets used, as depicted in Figs. 7, 8 and 9.

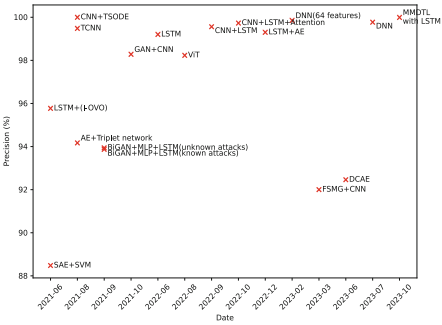
Transfer learning techniques have performed better than other methods analyzed in this study. [31] used multi-modal deep transfer learning with LSTM on CICIDS2017 and managed to obtain precision, recall and F1 score of 99.97% in binary classification and 99.97% in multi-class classification. Furthermore, [80] used transfer learning based on optimized CNN with inceptionV3, MobileNetV3Small, and EfficientNetV2B0 in the ensemble model. They achieved the performance of 99.99% precision, recall and F1 scores on CICIDS2017 and CSE-CICIDS2018 datasets during multi-class classification. However, the performance is the best, but there is a need for more experiments using different datasets on both binary and multi-class classification.



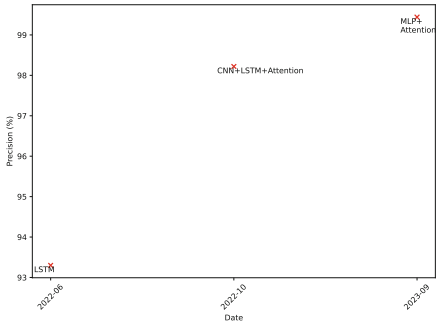
(a) NSL-KDD



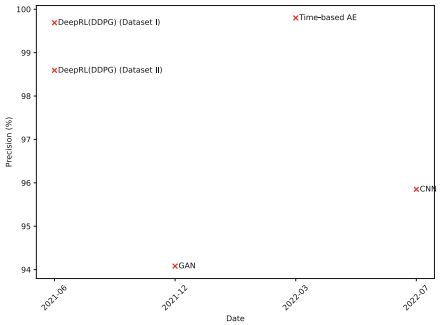
(b) UNSW-NB15



(c) CICIDS2017

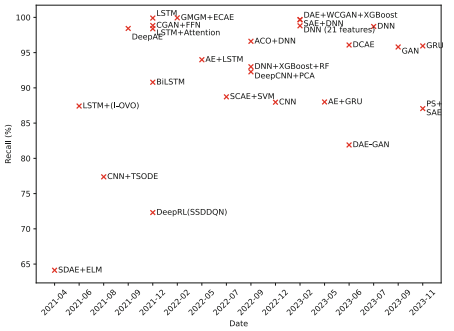


(d) CSE-CICIDS2018

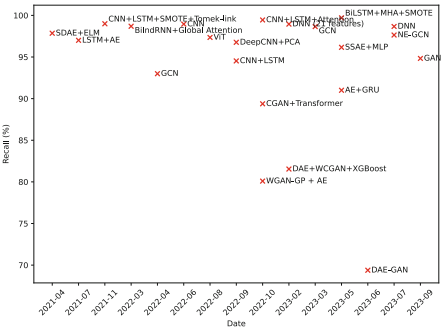


(e) CICDDoS2019

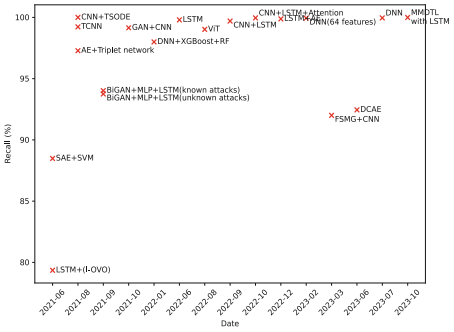
Fig. 4. Binary classification results (Precision)



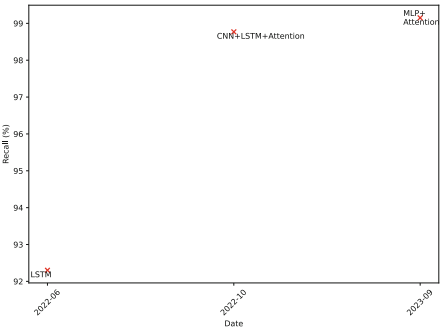
(a) NSL-KDD



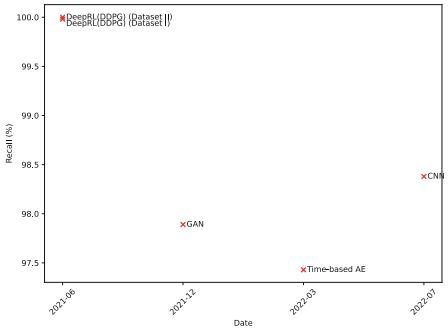
(b) UNSW-NB15



(c) CICIDS2017

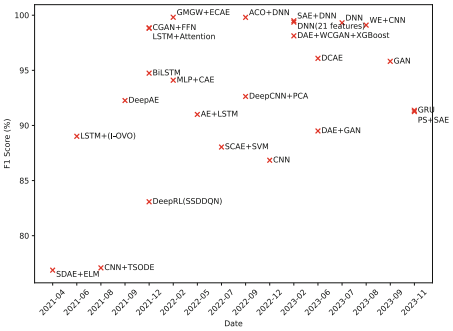


(d) CSE-CICIDS2018

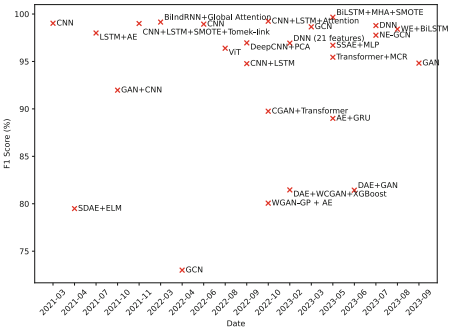


(e) CICDDoS2019

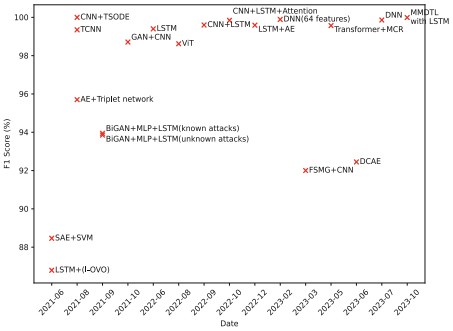
Fig. 5. Binary classification results (Recall)



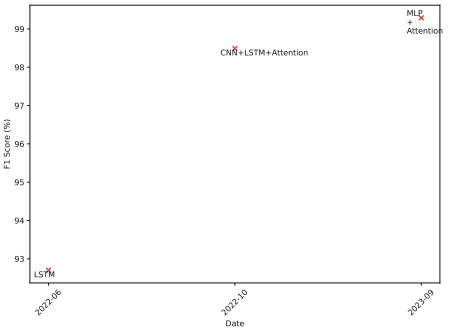
(a) NSL-KDD



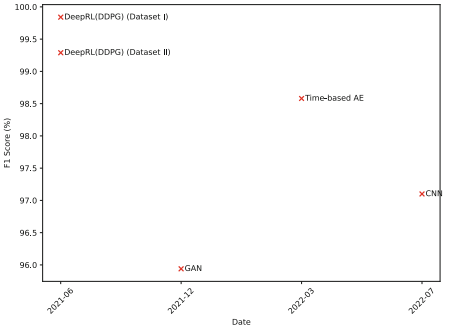
(b) UNSW-NB15



(c) CICIDS2017

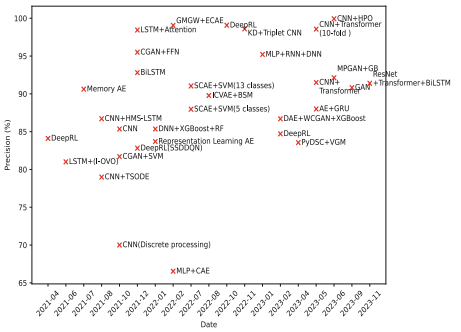


(d) CSE-CICIDS2018

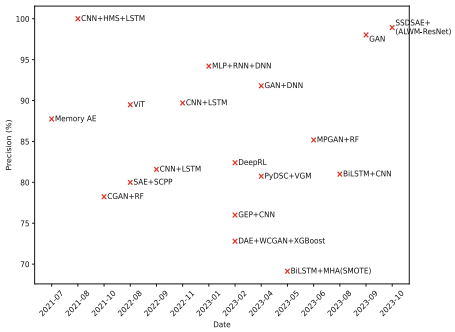


(e) CICDDoS2019

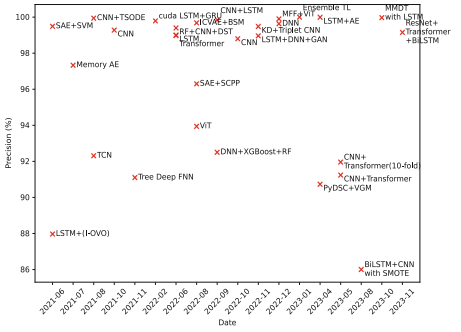
Fig. 6. Binary classification results (F1 score)



(a) NSL-KDD



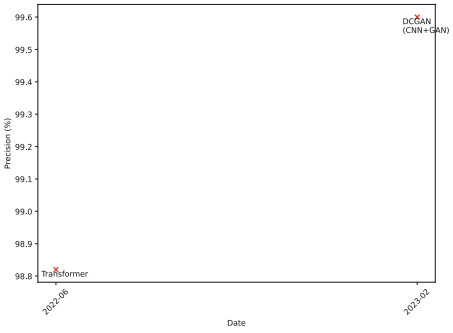
(b) UNSW-NB15



(c) CICIDS2017

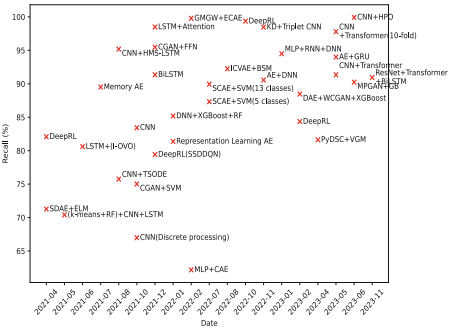


(d) CSE-CICIDS2018

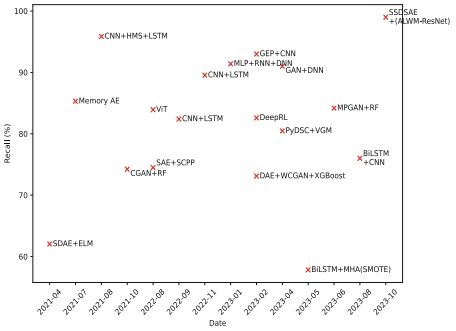


(e) CICDDoS2019

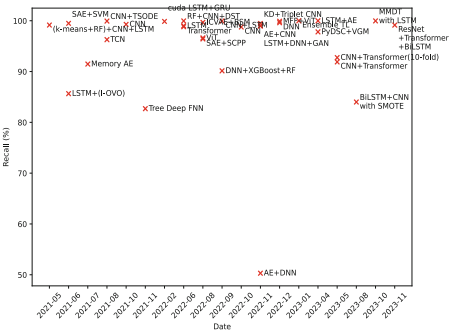
Fig. 7. Multi-class classification results (Precision)



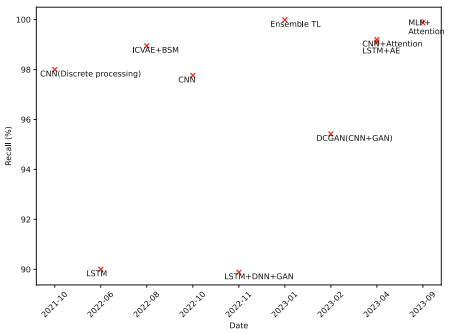
(a) NSL-KDD



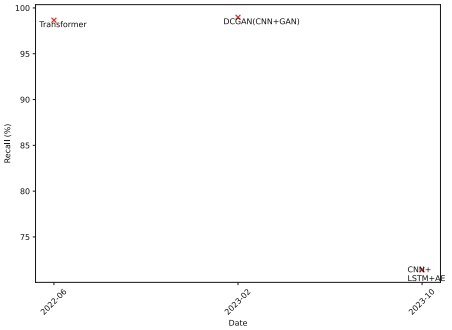
(b) UNSW-NB15



(c) CICIDS2017

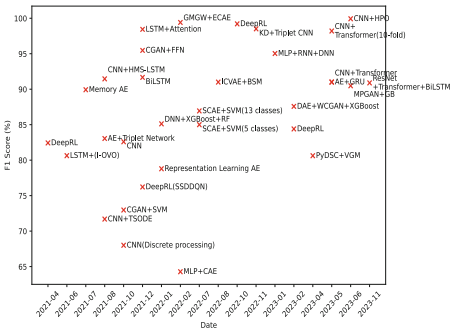


(d) CSE-CICIDS2018

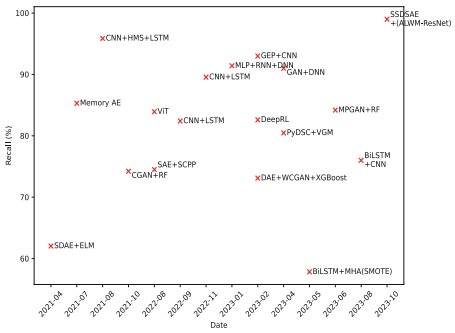


(e) CICDDoS2019

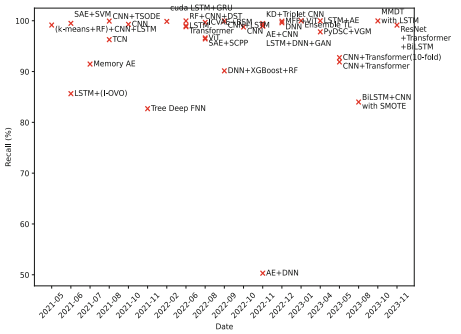
Fig. 8. Multi-class classification results (Recall)



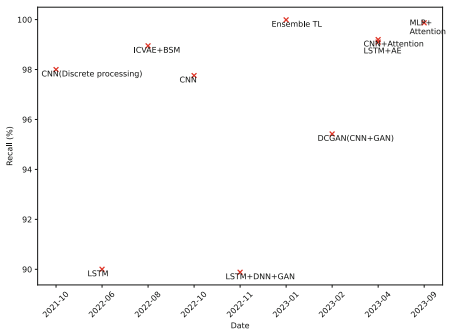
(a) NSL-KDD



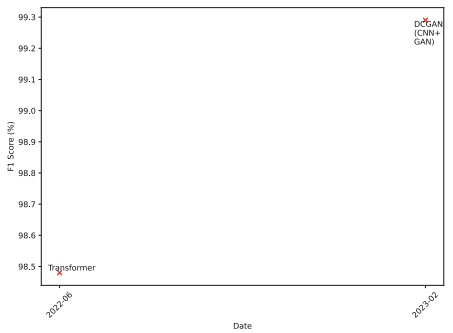
(b) UNSW-NB15



(c) CICIDS2017



(d) CSE-CICIDS2018



(e) CICDDoS2019

Fig. 9. Multi-class classification results (F1 score)

5 Limitations and Advantages of DL for NIDS

The study found some limitations within DL-based models for NIDS, which can be addressed in future studies. Developed models are complex compared to the state-of-the-art, which leads to more training time [43,59]. Due to hardware

limitations, some authors used only some of the datasets with large sizes like CICIDS2027 and CSE-CICIDS2018 [98]. Most developed models were tested over the offline environments [31]. Furthermore, imbalanced datasets led to difficult detection of other attacks [110,127].

DL mechanisms provide many advantages within the NIDS. DL performs automatic feature extraction compared to traditional ML, which speeds up the training time and increases the detection rate [15]. The developed DL-based models show the robustness of the training data compared to traditional ML [20]. Furthermore, developed models can detect attacks early [49,83]. Also, DL models provide a reasonable detection rate of attack classes with majority samples [63].

6 Conclusion

This systematic review provided a detailed analysis of 111 articles between 2021 and November 2023. The use of DL algorithms in NIDS implementation has been increasing with the performance. However, network attacks are becoming complex as technology advances, and most network intrusion datasets remain the same. Deep learning approaches have shown to be advanced with time. However, more research in NIDS implementation using DL methods still needs to be done. The study found that CNN, LSTM, and Attention mechanisms may bring better performance when used in a hybrid approach. The transfer and ensemble transfer learning approaches used within analyzed articles showed better results and needed more experiments to implement NIDS. Furthermore, The performance of DL models in NIDS depends on datasets, working environments, and hyper-parameters few to mention which gives more space for further research. In the future, we will work on using Large Language Models (LLMs) to implement NIDS.

References

1. Abdel-Basset, M., Hawash, H., Chakrabortty, R.K., Ryan, M.J.: Semi-supervised spatiotemporal deep learning for intrusions detection in IoT networks. IEEE Internet Things J. **8**(15), 12251–12265 (2021). https://doi.org/10.1109/JIOT.2021.3060878
2. Ahmim, A., Maazouzi, F., Ahmim, M., Namane, S., Dhaou, I.B.: Distributed denial of service attack detection for the internet of things using hybrid deep learning model. IEEE Access **11**, 119862–119875 (2023). https://doi.org/10.1109/ACCESS.2023.3327620
3. Akgun, D., Hizal, S., Cavusoglu, U.: A new ddos attacks intrusion detection model based on deep learning for cybersecurity. Comput. Secur. **118**, 102748 (2022). https://doi.org/10.1016/j.cose.2022.102748
4. Aktar, S., Nur, A.Y.: Towards ddos attack detection using deep learning approach. Comput. Secur. **129**, 103251 (2023). https://doi.org/10.1016/j.cose.2023.103251
5. Al, S., Dener, M.: Stl-hdl: a new hybrid network intrusion detection system for imbalanced dataset on big data environment. Comput. Secur. **110**, 102435 (2021). https://doi.org/10.1016/j.cose.2021.102435

6. Aldarwbi, M.Y., Lashkari, A.H., Ghorbani, A.A.: The sound of intrusion: a novel network intrusion detection system. Comput. Electr. Eng. **104**, 108455 (2022). https://doi.org/10.1016/j.compeleceng.2022.108455
7. Alhayali, R.A.I., Aljanabi, M., Ali, A.H., Mohammed, M.A., Sutikno, T.: Optimized machine learning algorithm for intrusion detection. Indon. J. Electr. Eng. Comput. Sci. **24**(1), 590–599 (2021)
8. Aljanabi, M., Ismail, M.A., Mezhuyev, V.: Improved tlbo-jaya algorithm for subset feature selection and parameter optimisation in intrusion detection system. Complexity **2020**(1), 5287684 (2020)
9. Alshamrani, A., Myneni, S., Chowdhary, A., Huang, D.: A survey on advanced persistent threats: techniques, solutions, challenges, and research opportunities. IEEE Commun. Surv. Tutor. **21**(2), 1851–1877 (2019). https://doi.org/10.1109/COMST.2019.2891891
10. Altaf, T., Wang, X., Ni, W., Liu, R.P., Braun, R.: Ne-gconv: a lightweight node edge graph convolutional network for intrusion detection. Comput. Secur. **130**, 103285 (2023). https://doi.org/10.1016/j.cose.2023.103285
11. Amanoul, S.V., Abdulazeez, A.M., Zeebare, D.Q., Ahmed, F.Y.: Intrusion detection systems based on machine learning algorithms. In: 2021 IEEE International Conference on Automatic Control & Intelligent Systems (I2CACIS), pp. 282–287. IEEE (2021)
12. Andresini, G., Appice, A., De Rose, L., Malerba, D.: Gan augmentation to deal with imbalance in imaging-based intrusion detection. Futur. Gener. Comput. Syst. **123**, 108–127 (2021). https://doi.org/10.1016/j.future.2021.04.017
13. Andresini, G., Appice, A., Malerba, D.: Autoencoder-based deep metric learning for network intrusion detection. Inf. Sci. **569**, 706–727 (2021). https://doi.org/10.1016/j.ins.2021.05.016
14. Anitha, T., Aanjankumar, S., Poonkuntran, S., Nayyar, A.: A novel methodology for malicious traffic detection in smart devices using BI-LSTM-CNN-dependent deep learning methodology. Neural Comput. Appl. **35**(27), 20319–20338 (2023). https://doi.org/10.1007/s00521-023-08818-0
15. Ashraf, J., Bakhshi, A.D., Moustafa, N., Khurshid, H., Javed, A., Beheshti, A.: Novel deep learning-enabled LSTM autoencoder architecture for discovering anomalous events from intelligent transportation systems. IEEE Trans. Intell. Transp. Syst. **22**(7), 4507–4518 (2021). https://doi.org/10.1109/TITS.2020.3017882
16. Aydın, H., Orman, Z., Aydın, M.A.: A long short-term memory (lstm)-based distributed denial of service (ddos) detection and defense system design in public cloud network environment. Comput. Secur. **118**, 102725 (2022). https://doi.org/10.1016/j.cose.2022.102725
17. Balamurugan, E., Mehbodniya, A., Kariri, E., Yadav, K., Kumar, A., Haq, M.A.: Network optimization using defender system in cloud computing security based intrusion detection system withgame theory deep neural network (idsgt-dnn). Pattern Recogn. Lett. **156**, 142–151 (2022). https://doi.org/10.1016/j.patrec.2022.02.013
18. Benaddi, H., Ibrahimi, K., Benslimane, A., Jouhari, M., Qadir, J.: Robust enhancement of intrusion detection systems using deep reinforcement learning and stochastic game. IEEE Trans. Veh. Technol. **71**(10), 11089–11102 (2022). https://doi.org/10.1109/TVT.2022.3186834
19. Bowen, B., Chennamaneni, A., Goulart, A., Lin, D.: Blocnet: a hybrid, dataset-independent intrusion detection system using deep learning. Int. J. Inf. Secur. **22**(4), 893–917 (2023). https://doi.org/10.1007/s10207-023-00663-5

20. Cai, S., Han, D., Li, D.: A feedback semi-supervised learning with meta-gradient for intrusion detection. IEEE Syst. J. **17**(1), 1158–1169 (2023). https://doi.org/10.1109/JSYST.2022.3197447
21. Caville, E., Lo, W.W., Layeghy, S., Portmann, M.: Anomal-e: a self-supervised network intrusion detection system based on graph neural networks. Knowl.-Based Syst. **258**, 110030 (2022). https://doi.org/10.1016/j.knosys.2022.110030
22. Chen, D., Zhang, F., Zhang, X.: Heterogeneous IoT intrusion detection based on fusion word embedding deep transfer learning. IEEE Trans. Ind. Inf. **19**(8), 9183–9193 (2023). https://doi.org/10.1109/TII.2022.3227640
23. Cisco: Cisco Annual Internet Report (2018–2023) (2020). https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf. Accessed 23 Aug 2023
24. Dao, T.N., Lee, H.: Stacked autoencoder-based probabilistic feature extraction for on-device network intrusion detection. IEEE Internet Things J. **9**(16), 14438–14451 (2022). https://doi.org/10.1109/JIOT.2021.3078292
25. Data, M., Aritsugi, M.: T-dfnn: an incremental learning algorithm for intrusion detection systems. IEEE Access **9**, 154156–154171 (2021). https://doi.org/10.1109/ACCESS.2021.3127985
26. Deng, X., Zhu, J., Pei, X., Zhang, L., Ling, Z., Xue, K.: Flow topology-based graph convolutional network for intrusion detection in label-limited iot networks. IEEE Trans. Netw. Serv. Manag. **20**(1), 684–696 (2023). https://doi.org/10.1109/TNSM.2022.3213807
27. Denning, D.: An intrusion-detection model. IEEE Trans. Softw. Eng. **SE-13**(2), 222–232 (1987). https://doi.org/10.1109/TSE.1987.232894
28. Dina, A.S., Manivannan, D.: Intrusion detection based on machine learning techniques in computer networks. Internet Things **16**, 100462 (2021)
29. Dlamini, G., Fahim, M.: DGM: a data generative model to improve minority class presence in anomaly detection domain. Neural Comput. Appl. **33**, 13635–13646 (2021). https://doi.org/10.1007/s00521-021-05993-w
30. Dong, S., Xia, Y., Peng, T.: Network abnormal traffic detection model based on semi-supervised deep reinforcement learning. IEEE Trans. Netw. Serv. Manag. **18**(4), 4197–4212 (2021). https://doi.org/10.1109/TNSM.2021.3120804
31. Elubeyd, H., Yiltas-Kaplan, D., Bahtıyar, Ş: A multi-modal deep transfer learning framework for attack detection in software-defined networks. IEEE Access (2023). https://doi.org/10.1109/ACCESS.2023.3324878
32. Fatani, A., Elaziz, M.A., Dahou, A., Al-Qaness, M.A.A., Lu, S.: Iot intrusion detection system using deep learning and enhanced transient search optimization. IEEE Access **9**, 123448–123464 (2021). https://doi.org/10.1109/ACCESS.2021.3109081
33. Fu, J., Wang, L., Ke, J., Yang, K., Yu, R.: Ganad: a gan-based method for network anomaly detection. World Wide Web **26**(5), 2727–2748 (2023). https://doi.org/10.1007/s11280-023-01160-4
34. Gundersen, O.E., Kjensmo, S.: State of the art: reproducibility in artificial intelligence. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
35. Gupta, N., Jindal, V., Bedi, P.: Lio-ids: handling class imbalance using LSTM and improved one-vs-one technique in intrusion detection system. Comput. Netw. **192**, 108076 (2021). https://doi.org/10.1016/j.comnet.2021.108076
36. Gupta, N., Jindal, V., Bedi, P.: Cse-ids: using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detec-

tion systems. Comput. Secur. **112**, 102499 (2022). https://doi.org/10.1016/j.cose.2021.102499
37. Halbouni, A., Gunawan, T.S., Habaebi, M.H., Halbouni, M., Kartiwi, M., Ahmad, R.: Cnn-lstm: hybrid deep neural network for network intrusion detection system. IEEE Access **10**, 99837–99849 (2022). https://doi.org/10.1109/ACCESS.2022.3206425
38. Hamid, Y., Sugumaran, M., Journaux, L.: Machine learning techniques for intrusion detection: a comparative analysis. In: Proceedings of the International Conference on Informatics and Analytics, pp. 1–6 (2016)
39. He, J., Wang, X., Song, Y., Xiang, Q.: A multiscale intrusion detection system based on pyramid depthwise separable convolution neural network. Neurocomputing **530**, 48–59 (2023). https://doi.org/10.1016/j.neucom.2023.01.072
40. Hnamte, V., Nhung-Nguyen, H., Hussain, J., Hwa-Kim, Y.: A novel two-stage deep learning model for network intrusion detection: LSTM-AE. IEEE Access **11**, 37131–37148 (2023). https://doi.org/10.1109/ACCESS.2023.3266979
41. Ho, C.M.K., Yow, K.C., Zhu, Z., Aravamuthan, S.: Network intrusion detection via flow-to-image conversion and vision transformer classification. IEEE Access **10**, 97780–97793 (2022). https://doi.org/10.1109/ACCESS.2022.3200034
42. Hussen, N., Elghamrawy, S.M., Salem, M., El-Desouky, A.I.: A fully streaming big data framework for cyber security based on optimized deep learning algorithm. IEEE Access **11**, 65675–65688 (2023). https://doi.org/10.1109/ACCESS.2023.3281893
43. Imrana, Y., Xiang, Y., Ali, L., Abdul-Rauf, Z.: A bidirectional lstm deep learning approach for intrusion detection. Expert Syst. Appl. **185**, 115524 (2021). https://doi.org/10.1016/j.eswa.2021.115524
44. Jain, J.K., Waoo, A.A.: An artificial neural network technique for prediction of cyber-attack using intrusion detection system. J. Artif. Intell. Mach. Learn. Neural Netw. (JAIMLNN) **3**(02), 33–42 (2023). ISSN: 2799-1172
45. Javeed, D., Saeed, M.S., Ahmad, I., Kumar, P., Jolfaei, A., Tahir, M.: An intelligent intrusion detection system for smart consumer electronics network. IEEE Trans. Consum. Electron. **69**(4), 906–913 (2023). https://doi.org/10.1109/TCE.2023.3277856
46. Kandhro, I.A., et al.: Detection of real-time malicious intrusions and attacks in IoT empowered cybersecurity infrastructures. IEEE Access **11**, 9136–9148 (2023). https://doi.org/10.1109/ACCESS.2023.3238664
47. Kanna, P.R., Santhi, P.: Unified deep learning approach for efficient intrusion detection system using integrated spatial-temporal features. Knowl.-Based Syst. **226**, 107132 (2021). https://doi.org/10.1016/j.knosys.2021.107132
48. Keele, S., et al.: Guidelines for performing systematic literature reviews in software engineering. Technical report. ver. 2.3 ebse technical report. ebse (2007)
49. Kim, T., Pak, W.: Early detection of network intrusions using a gan-based one-class classifier. IEEE Access **10**, 119357–119367 (2022). https://doi.org/10.1109/ACCESS.2022.3221400
50. Kumar, V., Sinha, D.: Synthetic attack data generation model applying generative adversarial network for intrusion detection. Comput. Secur. **125**, 103054 (2023). https://doi.org/10.1016/j.cose.2022.103054
51. Laghrissi, F., Douzi, S., Douzi, K., Hssina, B.: Ids-attention: an efficient algorithm for intrusion detection systems using attention mechanism. J. Big Data **8**(1), 149 (2021). https://doi.org/10.1186/s40537-021-00544-5

52. Laghrissi, F., Douzi, S., Douzi, K., Hssina, B.: Intrusion detection systems using long short-term memory (lstm). J. Big Data **8**(1), 65 (2021). https://doi.org/10.1186/s40537-021-00448-4
53. Lee, J.S., Chen, Y.C., Chew, C.J., Chen, C.L., Huynh, T.N., Kuo, C.W.: Conn-ids: intrusion detection system based on collaborative neural networks and agile training. Comput. Secur. **122**, 102908 (2022). https://doi.org/10.1016/j.cose.2022.102908
54. Li, H., Ge, H., Yang, H., Yan, J., Sang, Y.: An abnormal traffic detection model combined biindrnn with global attention. IEEE Access **10**, 30899–30912 (2022). https://doi.org/10.1109/ACCESS.2022.3159550
55. Li, M., Han, D., Li, D., Liu, H., Chang, C.C.: Mfvt: an anomaly traffic detection method merging feature fusion network and vision transformer architecture. EURASIP J. Wirel. Commun. Netw. **2022**(1), 39 (2022). https://doi.org/10.1186/s13638-022-02103-9
56. Li, Y., Qin, T., Huang, Y., Lan, J., Liang, Z., Geng, T.: Hdfef: a hierarchical and dynamic feature extraction framework for intrusion detection systems. Comput. Secur. **121**, 102842 (2022). https://doi.org/10.1016/j.cose.2022.102842
57. Li, Z., Chen, S., Dai, H., Xu, D., Chu, C.K., Xiao, B.: Abnormal traffic detection: traffic feature extraction and dae-gan with efficient data augmentation. IEEE Trans. Reliab. **72**(2), 498–510 (2023). https://doi.org/10.1109/TR.2022.3204349
58. Liu, C., Gu, Z., Wang, J.: A hybrid intrusion detection system based on scalable k-means+ random forest and deep learning. IEEE Access **9**, 75729–75740 (2021). https://doi.org/10.1109/ACCESS.2021.3082147
59. Liu, Q., Wang, D., Jia, Y., Luo, S., Wang, C.: A multi-task based deep learning approach for intrusion detection. Knowl.-Based Syst. **238**, 107852 (2022). https://doi.org/10.1016/j.knosys.2021.107852
60. Ludwig, S.A.: Intrusion detection of multiple attack classes using a deep neural net ensemble. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–7 (2017). https://doi.org/10.1109/SSCI.2017.8280825
61. Lunardi, W.T., Lopez, M.A., Giacalone, J.P.: Arcade: adversarially regularized convolutional autoencoder for network anomaly detection. IEEE Trans. Netw. Serv. Manag. **20**(2), 1305–1318 (2023). https://doi.org/10.1109/TNSM.2022.3229706
62. Ma, W., Ma, L., Li, K., Guo, J.: Few-shot iot attack detection based on ssdsae and adaptive loss weighted meta residual network. Inf. Fusion **98**, 101853 (2023). https://doi.org/10.1016/j.inffus.2023.101853
63. Ma, X., Shi, W.: Aesmote: adversarial reinforcement learning with smote for anomaly detection. IEEE Trans. Netw. Sci. Eng. **8**(2), 943–956 (2021). https://doi.org/10.1109/TNSE.2020.3004312
64. Mahdavisharif, M., Jamali, S., Fotohi, R.: Big data-aware intrusion detection system in communication networks: a deep learning approach. J. Grid Comput. **19**(4), 46 (2021). https://doi.org/10.1007/s10723-021-09581-z
65. Mahjabin, T., Xiao, Y., Sun, G., Jiang, W.: A survey of distributed denial-of-service attack, prevention, and mitigation techniques. Int. J. Distrib. Sens. Netw. **13**(12), 1550147717741463 (2017)
66. Mathonsi, T., van Zyl, T.L.: Multivariate anomaly detection based on prediction intervals constructed using deep learning. Neural Comput. Appl., 1–15 (2022)
67. Mendonça, R.V., et al.: Intrusion detection system based on fast hierarchical deep convolutional neural network. IEEE Access **9**, 61024–61034 (2021). https://doi.org/10.1109/ACCESS.2021.3074664

68. Meng, X., Wang, S., Liang, Z., Yao, D., Zhou, J., Zhang, Y.: Semi-supervised anomaly detection in dynamic communication networks. Inf. Sci. **571**, 527–542 (2021). https://doi.org/10.1016/j.ins.2021.04.056
69. Mighan, S.N., Kahani, M.: A novel scalable intrusion detection system based on deep learning. Int. J. Inf. Secur. **20**(3), 387–403 (2021). https://doi.org/10.1007/s10207-020-00508-5
70. Milosevic, M.S., Ciric, V.M.: Extreme minority class detection in imbalanced data for network intrusion. Comput. Secur. **123**, 102940 (2022). https://doi.org/10.1016/j.cose.2022.102940
71. Min, B., Yoo, J., Kim, S., Shin, D., Shin, D.: Network anomaly detection using memory-augmented deep autoencoder. IEEE Access **9**, 104695–104706 (2021). https://doi.org/10.1109/ACCESS.2021.3100087
72. Mohamed, S., Ejbali, R.: Deep sarsa-based reinforcement learning approach for anomaly network intrusion detection system. Int. J. Inf. Secur. **22**(1), 235–247 (2023). https://doi.org/10.1007/s10207-022-00634-2
73. Moizuddin, M., Jose, M.V.: A bio-inspired hybrid deep learning model for network intrusion detection. Knowl.-Based Syst. **238**, 107894 (2022). https://doi.org/10.1016/j.knosys.2021.107894
74. Muhammad, G., Hossain, M.S., Garg, S.: Stacked autoencoder-based intrusion detection system to combat financial fraudulent. IEEE Internet Things J. **10**(3), 2071–2078 (2023). https://doi.org/10.1109/JIOT.2020.3041184
75. Mushtaq, E., Zameer, A., Nasir, R.: Knacks of a hybrid anomaly detection model using deep auto-encoder driven gated recurrent unit. Comput. Netw. **226**, 109681 (2023). https://doi.org/10.1016/j.comnet.2023.109681
76. Mushtaq, E., Zameer, A., Umer, M., Abbasi, A.A.: A two-stage intrusion detection system with auto-encoder and lstms. Appl. Soft Comput. **121**, 108768 (2022). https://doi.org/10.1016/j.asoc.2022.108768
77. Muthanna, M.S.A., Alkanhel, R., Muthanna, A., Rafiq, A., Abdullah, W.A.M.: Towards sdn-enabled, intelligent intrusion detection system for internet of things (IoT). IEEE Access **10**, 22756–22768 (2022). https://doi.org/10.1109/ACCESS.2022.3153716
78. Nie, L., et al.: Intrusion detection in green internet of things: a deep deterministic policy gradient-based algorithm. IEEE Trans. Green Commun. Network. **5**(2), 778–788 (2021). https://doi.org/10.1109/TGCN.2021.3073714
79. Novaes, M.P., Carvalho, L.F., Lloret, J., Proença, M.L., Jr.: Adversarial deep learning approach detection and defense against ddos attacks in sdn environments. Futur. Gener. Comput. Syst. **125**, 156–167 (2021). https://doi.org/10.1016/j.future.2021.06.047
80. Okey, O.D., Melgarejo, D.C., Saadi, M., Rosa, R.L., Kleinschmidt, J.H., Rodríguez, D.Z.: Transfer learning approach to ids on cloud IoT devices using optimized CNN. IEEE Access **11**, 1023–1038 (2023). https://doi.org/10.1109/ACCESS.2022.3233775
81. Page, M.J., et al.: The prisma 2020 statement: an updated guideline for reporting systematic reviews. BMJ **372** (2021)
82. Psathas, A.P., Iliadis, L., Papaleonidas, A., Bountas, D.: Corem2 project: a beginning to end approach for cyber intrusion detection. Neural Comput. Appl. **34**(22), 19565–19584 (2022). https://doi.org/10.1007/s00521-022-07084-w
83. Qiu, W., Ma, Y., Chen, X., Yu, H., Chen, L.: Hybrid intrusion detection system based on dempster-shafer evidence theory. Comput. Secur. **117**, 102709 (2022). https://doi.org/10.1016/j.cose.2022.102709

84. Rendón-Segador, F.J., Álvarez-García, J.A., Varela-Vaca, A.J.: Paying attention to cyber-attacks: a multi-layer perceptron with self-attention mechanism. Comput. Secur. **132**, 103318 (2023). https://doi.org/10.1016/j.cose.2023.103318
85. Rydning, D.R.J.G.J., Reinsel, J., Gantz, J.: The digitization of the world from edge to core. Framingham: Int. Data Corporat. **16**, 1–28 (2018)
86. Salahuddin, M.A., Pourahmadi, V., Alameddine, H.A., Bari, M.F., Boutaba, R.: Chronos: DDOS attack detection using time-based autoencoder. IEEE Trans. Netw. Serv. Manag. **19**(1), 627–641 (2022). https://doi.org/10.1109/TNSM.2021.3088326
87. Samriya, J.K., Tiwari, R., Cheng, X., Singh, R.K., Shankar, A., Kumar, M.: Network intrusion detection using aco-dnn model with dvfs based energy optimization in cloud framework. Sustain. Comput. Inf. Syst. **35**, 100746 (2022). https://doi.org/10.1016/j.suscom.2022.100746
88. Sayed, M.S.E., Le-Khac, N.A., Azer, M.A., Jurcut, A.D.: A flow-based anomaly detection approach with feature selection method against ddos attacks in sdns. IEEE Trans. Cogn. Commun. Network. **8**(4), 1862–1880 (2022). https://doi.org/10.1109/TCCN.2022.3186331
89. Semmelrock, H., Kopeinik, S., Theiler, D., Ross-Hellauer, T., Kowald, D.: Reproducibility in machine learning-driven research. arXiv preprint arXiv:2307.10320 (2023)
90. Shahzad, F., Mannan, A., Javed, A.R., Almadhor, A.S., Baker, T., Al-Jumeily OBE, D.: Cloud-based multiclass anomaly detection and categorization using ensemble learning. J. Cloud Comput. **11**(1), 74 (2022). https://doi.org/10.1186/s13677-022-00329-y
91. Shams, E.A., Rizaner, A., Ulusoy, A.H.: A novel context-aware feature extraction method for convolutional neural network-based intrusion detection systems. Neural Comput. Appl. **33**(20), 13647–13665 (2021). https://doi.org/10.1007/s00521-021-05994-9
92. Sharif, M.H.U., Mohammed, M.A.: A literature review of financial losses statistics for cyber security and future trend. World J. Adv. Res. Rev. **15**(1), 138–156 (2022)
93. Sharma, B., Sharma, L., Lal, C., Roy, S.: Anomaly based network intrusion detection for IoT attacks using deep learning technique. Comput. Electr. Eng. **107**, 108626 (2023). https://doi.org/10.1016/j.compeleceng.2023.108626
94. Siddiqi, M.A., Pak, W.: Tier-based optimization for synthesized network intrusion detection system. IEEE Access **10**, 108530–108544 (2022). https://doi.org/10.1109/ACCESS.2022.3213937
95. Sivamohan, S., Sridhar, S.: An optimized model for network intrusion detection systems in industry 4.0 using xai based bi-lstm framework. Neural Comput. Appl. **35**(15), 11459–11475 (2023)
96. Soleymanzadeh, R., Kashef, R.: Efficient intrusion detection using multi-player generative adversarial networks (gans): an ensemble-based deep learning architecture. Neural Comput. Appl. **35**(17), 12545–12563 (2023). https://doi.org/10.1007/s00521-023-08398-z
97. Soltani, M., Ousat, B., Siavoshani, M.J., Jahangir, A.H.: An adaptable deep learning-based intrusion detection system to zero-day attacks. J. Inf. Secur. Appl. **76**, 103516 (2023). https://doi.org/10.1016/j.jisa.2023.103516
98. Soltani, M., Siavoshani, M.J., Jahangir, A.H.: A content-based deep intrusion detection system. Int. J. Inf. Secur. **21**(3), 547–562 (2022). https://doi.org/10.1007/s10207-021-00567-2

99. Song, D., et al.: Intrusion detection model using gene expression programming to optimize parameters of convolutional neural network for energy internet. Appl. Soft Comput. **134**, 109960 (2023). https://doi.org/10.1016/j.asoc.2022.109960
100. Soni, T., Kaur, R., Gupta, D., Sharma, A., Gupta, G.: The cybersecurity ecosystem: challenges, risk and emerging technologies. In: 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI), pp. 699–703. IEEE (2023)
101. Subasi, O., Manzano, J., Barker, K.: Denial of service attack detection via differential analysis of generalized entropy progressions. In: 2023 IEEE International Conference on Cyber Security and Resilience (CSR), pp. 219–226. IEEE (2023)
102. Thakkar, A., Lohiya, R.: Attack classification of imbalanced intrusion data for IoT network using ensemble-learning-based deep neural network. IEEE Internet Things J. **10**(13), 11888–11895 (2023). https://doi.org/10.1109/JIOT.2023.3244810
103. Thakkar, A., Lohiya, R.: Fusion of statistical importance for feature selection in deep neural network-based intrusion detection system. Inf. Fusion **90**, 353–363 (2023). https://doi.org/10.1016/j.inffus.2022.09.026
104. Torabi, H., Mirtaheri, S.L., Greco, S.: Practical autoencoder based anomaly detection by using vector reconstruction error. Cybersecurity **6**(1), 1 (2023). https://doi.org/10.1186/s42400-022-00134-9
105. Tu, S., Waqas, M., Badshah, A., Yin, M., Abbas, G.: Network intrusion detection system (nids) based on pseudo-siamese stacked autoencoders in fog computing. IEEE Trans. Serv. Comput. **16**(6), 4317–4327 (2023). https://doi.org/10.1109/TSC.2023.3319953
106. Tufan, E., Tezcan, C., Acartürk, C.: Anomaly-based intrusion detection by machine learning: a case study on probing attacks to an institutional network. IEEE Access **9**, 50078–50092 (2021). https://doi.org/10.1109/ACCESS.2021.3068961
107. Ullah, I., Mahmoud, Q.H.: A framework for anomaly detection in IoT networks using conditional generative adversarial networks. IEEE Access **9**, 165907–165931 (2021). https://doi.org/10.1109/ACCESS.2021.3132127
108. Varoquaux, G., Colliot, O.: Evaluating machine learning models and their diagnostic value. In: Machine Learning for Brain Disorders, pp. 601–630 (2023)
109. Wang, S., Xu, W., Liu, Y.: Res-tranbilstm: an intelligent approach for intrusion detection in the internet of things. Comput. Netw. **235**, 109982 (2023). https://doi.org/10.1016/j.comnet.2023.109982
110. Wang, W., Jian, S., Tan, Y., Wu, Q., Huang, C.: Representation learning-based network intrusion detection system by capturing explicit and implicit feature interactions. Comput. Secur. **112**, 102537 (2022). https://doi.org/10.1016/j.cose.2021.102537
111. Wang, W., Jian, S., Tan, Y., Wu, Q., Huang, C.: Robust unsupervised network intrusion detection with self-supervised masked context reconstruction. Comput. Secur. **128**, 103131 (2023). https://doi.org/10.1016/j.cose.2023.103131
112. Wang, W., Du, X., Shan, D., Qin, R., Wang, N.: Cloud intrusion detection method based on stacked contractive auto-encoder and support vector machine. IEEE Trans. Cloud Comput. **10**(3), 1634–1646 (2022). https://doi.org/10.1109/TCC.2020.3001017
113. Wang, Z., Ghaleb, F.A.: An attention-based convolutional neural network for intrusion detection model. IEEE Access **11**, 43116–43127 (2023). https://doi.org/10.1109/ACCESS.2023.3271408

114. Wang, Z., Li, Z., He, D., Chan, S.: A lightweight approach for network intrusion detection in industrial cyber-physical systems based on knowledge distillation and deep metric learning. Expert Syst. Appl. **206**, 117671 (2022). https://doi.org/10.1016/j.eswa.2022.117671
115. Wang, Z., Liu, Y., He, D., Chan, S.: Intrusion detection methods based on integrated deep learning model. Comput. Secur. **103**, 102177 (2021). https://doi.org/10.1016/j.cose.2021.102177
116. Wei, W., Chen, Y., Lin, Q., Ji, J., Wong, K.C., Li, J.: Multi-objective evolving long-short term memory networks with attention for network intrusion detection. Appl. Soft Comput. **139**, 110216 (2023). https://doi.org/10.1016/j.asoc.2023.110216
117. Wu, Y., Nie, L., Wang, S., Ning, Z., Li, S.: Intelligent intrusion detection for internet of things security: a deep convolutional generative adversarial network-enabled approach. IEEE Internet Things J. **10**(4), 3094–3106 (2023). https://doi.org/10.1109/JIOT.2021.3112159
118. Wu, Z., Zhang, H., Wang, P., Sun, Z.: Rtids: a robust transformer-based approach for intrusion detection system. IEEE Access **10**, 64375–64387 (2022). https://doi.org/10.1109/ACCESS.2022.3182333
119. Xie, J., Li, S., Zhang, Y., Sun, P., Xu, H.: Analysis and detection against network attacks in the overlapping phenomenon of behavior attribute. Comput. Secur. **121**, 102867 (2022). https://doi.org/10.1016/j.cose.2022.102867
120. Xin, Y., et al.: Machine learning and deep learning methods for cybersecurity. IEEE Access **6**, 35365–35381 (2018)
121. Xu, W., Jang-Jaccard, J., Singh, A., Wei, Y., Sabrina, F.: Improving performance of autoencoder-based network anomaly detection on nsl-kdd dataset. IEEE Access **9**, 140136–140146 (2021). https://doi.org/10.1109/ACCESS.2021.3116612
122. Yang, J., Li, H., Shao, S., Zou, F., Wu, Y.: Fs-ids: a framework for intrusion detection based on few-shot learning. Comput. Secur. **122**, 102899 (2022). https://doi.org/10.1016/j.cose.2022.102899
123. Yang, Y., Yao, C., Yang, J., Yin, K.: A network security situation element extraction method based on conditional generative adversarial network and transformer. IEEE Access **10**, 107416–107430 (2022). https://doi.org/10.1109/ACCESS.2022.3212751
124. Yao, C., Yang, Y., Yin, K., Yang, J.: Traffic anomaly detection in wireless sensor networks based on principal component analysis and deep convolution neural network. IEEE Access **10**, 103136–103149 (2022). https://doi.org/10.1109/ACCESS.2022.3210189
125. Yao, R., Wang, N., Chen, P., Ma, D., Sheng, X.: A cnn-transformer hybrid approach for an intrusion detection system in advanced metering infrastructure. Multimedia Tools Appl. **82**(13), 19463–19486 (2023). https://doi.org/10.1007/s11042-022-14121-2
126. Ye, T., Li, G., Ahmad, I., Zhang, C., Lin, X., Li, J.: Flag: few-shot latent dirichlet generative learning for semantic-aware traffic detection. IEEE Trans. Netw. Serv. Manag. **19**(1), 73–88 (2022). https://doi.org/10.1109/TNSM.2021.3131266
127. Yoo, J., Min, B., Kim, S., Shin, D., Shin, D.: Study on network intrusion detection method using discrete pre-processing method and convolution neural network. IEEE Access **9**, 142348–142361 (2021). https://doi.org/10.1109/ACCESS.2021.3120839
128. Zeeshan, M., et al.: Protocol-based deep intrusion detection for dos and ddos attacks using unsw-nb15 and bot-iot data-sets. IEEE Access **10**, 2269–2283 (2022). https://doi.org/10.1109/ACCESS.2021.3137201

129. Zhang, K., Zhao, F., Luo, S., Xin, Y., Zhu, H.: An intrusion action-based ids alert correlation analysis and prediction framework. IEEE Access **7**, 150540–150551 (2019)
130. Zhang, L., Liu, K., Xie, X., Bai, W., Wu, B., Dong, P.: A data-driven network intrusion detection system using feature selection and deep learning. J. Inf. Secur. Appl. **78**, 103606 (2023). https://doi.org/10.1016/j.jisa.2023.103606
131. Zhang, T., Chen, W., Liu, Y., Wu, L.: An intrusion detection method based on stacked sparse autoencoder and improved gaussian mixture model. Comput. Secur. **128**, 103144 (2023). https://doi.org/10.1016/j.cose.2023.103144
132. Zhang, Y., Liu, Q.: On IoT intrusion detection based on data augmentation for enhancing learning on unbalanced samples. Futur. Gener. Comput. Syst. **133**, 213–227 (2022). https://doi.org/10.1016/j.future.2022.03.007
133. Zhang, Z., Zhang, Y., Guo, D., Song, M.: A scalable network intrusion detection system towards detecting, discovering, and learning unknown attacks. Int. J. Mach. Learn. Cybern. **12**(6), 1649–1665 (2021). https://doi.org/10.1007/s13042-020-01264-7
134. Zhao, R., et al.: A novel intrusion detection method based on lightweight neural network for internet of things. IEEE Internet Things J. **9**(12), 9960–9972 (2022). https://doi.org/10.1109/JIOT.2021.3119055
135. Zola, F., Segurola-Gil, L., Bruse, J.L., Galar, M., Orduna-Urrutia, R.: Network traffic analysis through node behaviour classification: a graph-based approach with temporal dissection and data-level preprocessing. Comput. Secur. **115**, 102632 (2022). https://doi.org/10.1016/j.cose.2022.102632

Impact of Batch Normalization on Convolutional Network Representations

Hermanus L. Potgieter[1,2,4(✉)], Coenraad Mouton[1,2], and Marelie H. Davel[1,2,3]

[1] Faculty of Engineering, North-West University, Potchefstroom, South Africa
potgieterharmen@gmail.com
[2] Centre for Artificial Intelligence Research, Cape Town, South Africa
[3] National Institute for Theoretical and Computational Sciences, Stellenbosch, South Africa
[4] South African National Space Agency, Pretoria, South Africa
http://engineering.nwu.ac.za/must

Abstract. Batch normalization (BatchNorm) is a popular layer normalization technique used when training deep neural networks. It has been shown to enhance the training speed and accuracy of deep learning models. However, the mechanics by which BatchNorm achieves these benefits is an active area of research, and different perspectives have been proposed. In this paper, we investigate the effect of BatchNorm on the resulting hidden representations, that is, the vectors of activation values formed as samples are processed at each hidden layer. Specifically, we consider the sparsity of these representations, as well as their implicit clustering – the creation of groups of representations that are similar to some extent. We contrast image classification models trained with and without batch normalization and highlight consistent differences observed. These findings highlight that BatchNorm's effect on representational sparsity is *not* a significant factor affecting generalization, while the representations of models trained with BatchNorm tend to show more advantageous clustering characteristics.

Keywords: Batch normalization · Hidden Representations · Clustering · Convolutional Neural Networks

1 Introduction

Deep learning has become a particularly important set of machine learning techniques and is widely applied to solve real-world tasks. At the same time, many open questions remain with regard to the ability of these deep neural networks (DNNs) to generalize so well, that is, their ability to perform well on unseen data. Although there is not yet a theoretical framework to assist us in reasoning about these models [35], the generalization ability of DNNs has been studied from many

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 235–252, 2025.
https://doi.org/10.1007/978-3-031-78255-8_14

perspectives, such as the geometry of the loss landscape [12], statistical measures of stability and robustness [6], size of margins (distance to the decision boundary between classes) [26], and information-theoretic techniques [18], among others.

A promising research direction is to study the characteristics of the internal data representations formed by DNNs, where each representation is the vector of activation values from a specific layer for a given sample. Aspects of these representations that have been studied include the size of margins in the representation space [8,17,28]; the 'quality' of representations, evaluated using the consistency of class-specific representations and their robustness when combined [28]; and representation sparsity, that is, the number of non-zero elements in a data representation [13]. In this work, we also study the characteristics of the internal representations of DNNs, but focus on the effect that a very specific technique – Batch Normalization (BatchNorm) – has on internal representation quality.

BatchNorm [15] is a popular technique used to normalize hidden activations when training DNNs. Networks trained with BatchNorm show desirable properties such as faster convergence and better generalization ability [22,31]. Despite the success and widespread adoption of BatchNorm, the exact mechanisms by which BatchNorm achieves its performance remain unclear. In this work, we study the effect of BatchNorm on the internal representations of convolutional neural networks (CNNs). More precisely, we analyze the resulting sparsity and clustering characteristics of CNN activations for models trained with and without BatchNorm. We selected these two elements, specifically:

- Representational sparsity is generally considered a desirable property for DNNs, and is considered to be one of the reasons for the performance of rectified linear unit (RELU) activation functions [13]. Although the sparsity of neural networks parameters (parametric sparsity) is well studied, the sparsity of the internal representations (representational sparsity) is less explored [14, p. 251]. The effect of BatchNorm on representational sparsity has been studied to a limited extent for Multilayer Perceptrons (MLPs) [30] but no such analysis has yet been performed for CNNs.
- The clustering characteristics of representations have shown to be a promising avenue to better understand generalization in DNNs [8,23,28]. As such, comparing the clusters formed by models trained with and without BatchNorm could provide insights into how BatchNorm increases generalization performance.

In summary, the objective of this paper is to compare models trained with and without BatchNorm and to analyze the effect of BatchNorm on the hidden representations.

2 Background

This section introduces the BatchNorm technique as well as current views on the rationale for its performance. We also review current perspectives on the effect of representation sparsity and clustered representations on generalization performance.

2.1 Batch Normalization

Batch normalization [15] is a popular layer normalization technique which speeds up training and improves the performance of deep neural networks. The goal of BatchNorm is to normalize each batch of activation values for some layer to a learned mean and standard deviation during both training and inference. More precisely, for the activation values $\mathbf{x}^l$ at layer l for a sample $\mathbf{x}$, the BatchNorm operation is defined as:

$$\mathrm{BN}(\mathbf{x}^l) = \boldsymbol{\gamma} \odot \frac{\mathbf{x}^l - \hat{\boldsymbol{\mu}}^l_{\mathcal{B}}}{\hat{\boldsymbol{\sigma}}^l_{\mathcal{B}}} + \boldsymbol{\beta}. \tag{1}$$

where $\hat{\boldsymbol{\mu}}^l_{\mathcal{B}}$ and $\hat{\boldsymbol{\sigma}}^l_{\mathcal{B}}$ is the *per feature* mean and standard deviation vectors calculated over a batch of samples $\mathbf{X}_{\mathcal{B}}$. The vectors $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are learnable parameters of the same dimensionality as $\mathbf{x}^l$. In the case of CNNs, each channel is considered a 'feature', and the mean and standard deviation are calculated independently per channel. More precisely, the activation values for a convolution layer l for a batch of samples can be denoted as a 4D tensor such that $\mathbf{X}^l_{\mathcal{B}} \in \mathbb{R}^{N \times C \times H \times W}$. Here N, C, H, and W correspond to the number of samples in the batch, the number of channels, and the height and width of the representation, respectively. The per-channel mean and standard deviation are calculated in the same way but for a channel instead of a node.

Despite the widespread popularity of BatchNorm, there is not yet a consensus on the exact mechanism by which BatchNorm speeds up training and improves generalization. Originally the success of BatchNorm was motivated by its reduction of 'internal covariance shift' [15], meaning that BatchNorm reduces distributional differences between the inputs to different layers in the network. However, Santurkar et al. [31] show that even when the internal distributions are artificially shifted (by injecting noise after BatchNorm layers) there is no noticeable degradation in performance. Instead, they argue from an optimization perspective, and conclude that BatchNorm drastically smooths the loss landscape. From a similar perspective, Bjorck et al. [5] show that without BatchNorm, a small subset of the activations in deeper layers grow uncontrollably large during training. They further show that BatchNorm prevents this phenomenon and allows for the use of larger learning rates that result in speedier convergence. Similarly, Li and Arora [22] show that an exponential learning rate schedule is made possible by BatchNorm.

Although the optimization perspective is the dominant view, several other authors have also investigated the benefits of BatchNorm from alternative perspectives. These include the effect of BatchNorm on adversarial robustness [4], its effect on the density of linear regions in the input space [2], and the orthogonality of internal representations [9].

2.2 Sparsity

Representational sparsity refers to the extent to which a representation consists of values that are zero or close to zero [14, p. 251]. Glorot et al. [13] approach

sparsity from a biological perspective, arguing the use of rectified neurons over sigmoidal neurons is inspired by neuroscience. In addition to the use of rectified neurons, they propose the use of L_1 regularization on the activation values to promote sparsity as a mechanism for better performance. Sparsity has been attributed with benefits such as information disentangling, efficient variable-size representation, and representations being more linearly separable [13]. Representational sparsity can thus be seen as desirable in CNNs [3].

Pretorius et al. [30] investigated the effect of activation functions on Multi-Layer Perceptrons (MLPs) by analysing network characteristics such as network sparsity. They found that networks trained with ReLU activation are more sparse on average than networks trained on sigmoidal activations. They also investigated the effect of BatchNorm on representation sparsity and found that the models trained with BatchNorm tend to have *less* sparse representations. It was also found that the less sparse BatchNorm models tend to have better generalization performance, and it was suggested that sparsity is not necessarily an indication of good generalization.

2.3 Clustered Representations

Clustering is an unsupervised learning method that aims to form groups of similar elements based on different metrics. Methods can range from forming clusters by minimizing or maximizing a quality measure [24], using hierarchical methods that build a dendrogram to iteratively merge or split clusters based on a distance metric [29], or using linear algebra directly (eigenvalues and eigenvectors) to cluster [32].

Class-Based Clustering. A neural network can be conceptualized as a clustering algorithm, as it inherently groups data with similar features together within its hidden layers [25], meaning similar samples are modelled in a similar way. Since the network is trained to predict the correct class of a given sample, it tends to cluster samples from the same class together. That said, the initial layers often capture features that are shared across different classes, leading to inter-class clustering. However, as observed by Caron et al. [7], deeper layers typically contain more class-specific information, resulting in distinct clusters that align closely with the true class labels. Class-based cluster analysis therefore assigns class labels as cluster identifiers and evaluates cluster characteristics.

Class-Agnostic Clustering. An alternative viewpoint is not to consider the class label information at all but rather to consider the purity of clusters formed at different layers of a network. In this case, the number of clusters that are relevant is not known ahead of time.

Clustering Techniques. One of the most popular clustering methods is k-means clustering [24]. The method partitions the samples into k clusters. The distance

to each cluster centre is calculated, and the sample is placed in the cluster with the nearest mean. After each iteration, the cluster centres are moved to the centre of the points assigned to the cluster. The process is iterated for a number of steps. Different variations of the method exist such as k-means++ [1], Fuzzy C-means clustering [11], and k-medians clustering [16], all of which use a similar principle as k-means clustering. The k-means++ algorithm improves upon k-means with a more intelligent selection of the initial centroids. Fuzzy C-means allow data points to belong to multiple clusters, assigned via the similarity to the centroid. It focuses more on the membership relationship and uncertainty of the data points. K-medians minimise the sum of absolute differences between the centroids and data points, which is a more robust approach. The number of clusters must be specified in advance for the k-means method, which often necessitates a hyperparameter search to determine the optimal number of clusters.

Evaluating Cluster Quality. The Davies-Bouldin index (DBI) [10] is a popular metric to evaluate cluster quality. It evaluates the average similarity between each cluster and its most similar cluster, offering an indication of how well-separated and cohesive the clusters are. It is computed by comparing the average distance between clusters and the size of the clusters. This method can be used to compare the relative appropriateness of different data partitions. Importantly, the index is not sensitive to the number of clusters or the specific partitioning of the data. The DBI is defined as:

$$DBI = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} \left\{ \frac{d(x_i) + d(x_j)}{d(c_i, c_j)} \right\} \tag{2}$$

where k denotes the number of clusters, i, j are cluster labels, $d(x_i)$ and $d(x_j)$ are the mean intra-cluster distance for the clusters i and j, respectively, and $d(c_i, c_j)$ is the distance between the cluster centroids. The DBI can be used to measure the purity of the of representational clusters based on the class that the representation belongs to.

Clustered Representations and Generalization. There are limited studies that explore the link between the characteristics of representation clusters and generalization directly. The two main papers that make a direct link between these concepts are those by Natekar and Sharma [28] and Liao et al. [23].

Natekar and Sharma proposed three measures of quality of representations to predict generalization in a post-hoc diagnostic setting [28]. The first metric they measure is consistency, which is particularly relevant, as it is defined as an indication of representation cluster purity. To measure consistency, the DBI is calculated on a dimensionally reduced version of the activation values of a layer. This produces a measure of how consistent the representations are within a cluster, as well as how different representations are across clusters. Dimensionality reduction is performed with principal component analysis (PCA) and max pooling. The main observation from their study was that simple measures based on

the quality of internal representations are somewhat predictive of the generalization performance and they, therefore, conjecture that clustered representations of objects in internal layers aid in generalization.

Liao et al. [23] use the process of clustering representations as a regulariser, and demonstrate some performance gains in the process. That is, they adjust the loss function during optinization to encourage cluster-forming. They looked at three different clustering methods namely: sample clustering, spatial (channel) clustering and co-clustering of both samples and channels. The regularisation term that they add to the standard loss consists of the distance of the element to the cluster mean. The cluster centres are initialised and updated with k-means, and then resampled and updated through the $k\text{-}means++$ procedure. Their results across clustering techniques were fairly similar, with sample and spatial clustering outperforming co-clustering with a small margin.

Clustering has been studied more extensively in the unsupervised learning setting. Caron et al. [7] use the clustering of internal representations to train CNNs. The training process consists of an alternating process of clustering internal representations (using standard k-means) and updating the parameters until convergence. Once the clustering is complete, the cluster tags are used as pseudo-labels for the corresponding hidden representation vectors. The pseudo-labels are used to train a classifier network on top of the CNN to predict the pseudo-labels for each hidden representation vector. This idea was further developed by Zhan et al. [34] who introduced a batch-aware version that addresses instability caused by samples changing their batch membership; and by Li et al. [21] who forced the training process to create representations that are close to the cluster centroids.

3 Experimental Setup

In this section, we describe the approach used to investigate the effect of BatchNorm on CNN representations. We select a set of CNN architectures and datasets and develop comparative models. We then develop metrics of sparsity and techniques to evaluate the cluster characteristics of representations.

3.1 Models

We consider two different datasets and network architectures for experimentation. For datasets, we select MNIST [20] and CIFAR10 [19]. MNIST, while simple, is useful for probing generalization from a clustering perspective as it consists of well-separated classes. On the other hand, CIFAR10 consists of more natural images and a greater amount of overlap among classes. This allows us to verify whether results on MNIST also hold on CIFAR10. We select architectures that match the datasets: a simple CNN with 4 convolutional layers trained on the MNIST dataset, and a standard, deeper architecture with 13 convolutional layers trained on the CIFAR10 dataset. For each of these architectures, we train a model with and without BatchNorm using 4 different random initialization seeds each.

MNIST. For the MNIST dataset, we make use of an architecture similar to the 'standard architecture' described by Nakkiran et al. [27]. This 5-layer CNN consists of 4 convolutional layers with $[16, 32, 64, 128]$ channels, respectively. There are also max pooling layers after the 2nd and 3rd convolutional layers with a stride of 2. We alter the architecutre by adding a fully connected hidden layer of 100 nodes preceding the output layer of 10 nodes, as we are interested in the representations formed by both the convolutional and fully connected layers. This is in contrast to the architecture of Nakkiran et al. [27] which only has a single fully connected output layer of 10 nodes.

Hyperparameters are chosen based on a hyperparameter sweep that tracks validation accuracy at the point of interpolation. We apply a ReLU activation after each hidden layer, cross-entropy as loss function, and SGD as optimizer for both the BatchNorm and non-BatchNorm models. We use a batch size of 64, a learning rate of 0.01, a learning rate schedule that multiplies the learning rate with 0.99 every 10 epochs, and train for 1 000 epochs. We also include a Nesterov momentum of 0.99. We terminate the optimization once the network interpolates the training data (reaches 100% training accuracy). The accuracy achieved during training is displayed in Table 1. Both sets of models interpolated on the training data with the BatchNorm models performing a bit better on the validation and test sets than the non-BatchNorm. The variation between seeds is minimal for both the BatchNorm and non-BatchNorm models.

CIFAR10. For the CIFAR10 dataset, we rely on the well-known VGG-16 [33] architecture. This architectures consists of 13 ReLU-activated convolutional layers with $[64, 64, 128, 128, 256, 256, 256, 512, 512, 512, 512, 512, 512]$ channels and a kernel size of 3×3. There are also 2×2 max pooling layers after the 2^{nd}, 4^{th}, 7^{th}, 10^{th}, and 13^{th} convolutional layers with a stride of 2. We do not use the hyperparameters from the original implementation of Simonyan and Zisserman [33] but rather, perform a hyperparameter sweep using the validation set. This is done to achieve as close to current benchmarks on the dataset and architecture as possible. Specifically, we make use of cross-entropy loss in combination with the ADAM optimizer. We use an initial learning rate of 0.001 for the models with BatchNorm, and a smaller learning rate of 0.0001 for those without. The learning rate is multiplied by 0.99 after each epoch, and the momentum and stability terms of ADAM are kept at the default values. We use the same batch size of 64 for both sets of models. We train for 300 epochs using early stopping with a patience of 20% on the validation error.

The accuracy for both sets of models is in Table 1. Both sets of models achieved near interpolation on the training data, with the BatchNorm models significantly outperforming the non-BatchNorm models on the unseen data from both the training and evaluation sets. Additionally, the BatchNorm models exhibited minimal variation between seeds, whereas the accuracy of the non-BatchNorm models showed much greater variability across different seeds on the unseen data.

Table 1. Train, validation, and evaluation classification accuracy for the BatchNorm (BN) and non-BatchNorm (NBN) models on MNIST and CIFAR10. ± indicates the standard deviation across 4 random initialization seeds.

	Training Acc	Validation Acc	Evaluation Acc
CNN MNIST BN	100.000 ± 0.000	99.355 ± 0.010	99.198 ± 0.100
CNN MNIST NBN	100.000 ± 0.000	99.250 ± 0.087	99.150 ± 0.072
VGG-16 CIFAR10 BN	99.988 ± 0.024	87.080 ± 0.315	85.99 ± 0.320
VGG-16 CIFAR10 NBN	99.982 ± 0.021	81.030 ± 1.008	79.435 ± 1.059

3.2 Sparsity

We only use the training data representations during analysis, as we are specifically focused on data that both models have seen and learned. This approach provides an indication of where these two models differ, given that it is data both have memorized. To extract the representations of the samples, we pass each sample through the network and record the activation values of each layer after the ReLU activation function. This produces a representation $\mathbf{x}^l \in [0, \infty]$ for each sample at each layer of the network, and results in the 4D tensor $\mathbf{X}^l_{\mathcal{B}}$ for a batch of samples, as described in Sect. 2.1. When analysing a certain dimension, such as the layer or channels of the network, the tensor is unfolded into a 2D matrix where the one dimension is the size of the element analysed and the other is the product of all other dimensions. For example, analysing the sample dimension gives a matrix of $\mathbf{X}^l_{\mathcal{B}} \in \mathbb{R}^{N \times CHW}$.

We define a sparse element as any element with a value of zero. While activation values close to zero could have minimal impact on the network's decision-making process and could therefore also be considered as values for which the network effectively does not activate [14, p. 251], we do not consider such small values as sparse elements. To calculate sparsity, we first identify the inactive elements – those with activation values of zero. The tensor is then reshaped to focus on the specified axis. For channel sparsity, the resulting 2D matrix is $\mathbf{X}^l_{\mathcal{B}} \in \mathbb{R}^{C \times NHW}$ and the sparsity calculated as:

$$s_c = \frac{1}{N \cdot H \cdot W} \sum_{j=1}^{NHW} \mathbb{I}(\mathbf{x}^l_{c,j} = 0) \tag{3}$$

where $\mathbf{x}^l_{c,j}$ is the element of the matrix at position (c, j) where c is the channel, and $\mathbb{I}(\cdot)$ is the indicator function, which equals 1 if the condition inside is true and 0 otherwise. To calculate the layer sparsity we use the vector $\mathbf{x}^l_{\mathcal{B}} \in \mathbb{R}^{1 \times NCHW}$ as follows:

$$s_l = \frac{1}{N \cdot C \cdot H \cdot W} \sum_{j=1}^{NCHW} \mathbb{I}(\mathbf{x}^l_j = 0) \tag{4}$$

resulting in a single sparsity value per layer (per model and seed).

3.3 Clustered Representations

The same representations ($\mathbf{x}^l$) as before are used when analysing clustering characteristics. Due to the high dimensionality of the representations and the computational complexity associated with the high dimensionality, we reduce the representations to $\mathbf{X}^l \in \mathbb{R}^{N \times C}$. This is achieved by taking the average of the elements in the feature map across the height H and width W for each channel, expressed as:

$$\mathbf{X}^l_{n,c} = \frac{1}{H \cdot W} \sum_{h=1}^{H} \sum_{w=1}^{W} \mathbf{X}^l_{n,c,h,w} \tag{5}$$

where $\mathbf{X}^l_{n,c,h,w}$ denotes the element of the tensor $\mathbf{X}^l$ at position (n, c, h, w), and $\mathbf{X}^l_{n,c} \in \mathbb{R}^{N \times C}$ is the resulting 2D tensor. Each element represents the average value of the feature map over the spatial dimensions H and W for each channel C and sample N. We then normalize each sample representation by dividing it with its Euclidean norm to produce $\mathbf{x^l}$ to be $\in [0, 1]$.

We consider cluster purity from both a class-based and class-agnostic approach. For the class-based approach, we do not apply any clustering techniques to the representations, but instead, we analyze the clusters naturally formed by the network for each labelled class. In this context, we measure the DBI of the layer representations using the labels of the sample classes as the cluster labels, following the method of Natekar and Sharma [28].

For the class-agnostic approach, we first use the k-means clustering algorithm to group similar representations and provide cluster labels for each representation. To select the optimal number of clusters we first use 2 to 15 clusters during an initial clustering step. We then measure the purity of these representation clusters using the DBI metric, as discussed in Sect. 5.2. We then select the optimal number of clusters as the clusters that produced the lowest DBI score.

4 Sparsity Analysis and Results

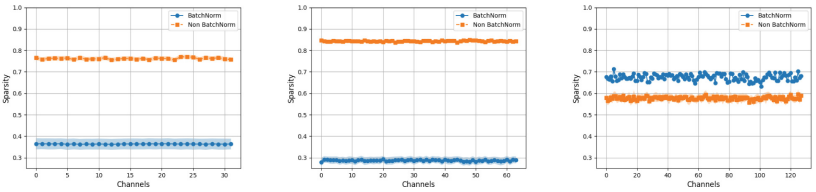
In this section, we analyze and compare the sparsity of networks trained with and without BatchNorm. We consider the sparsity of representations both per channel (Sect. 4.1) and per layer (Sect. 4.2).

4.1 Channel Sparsity

We begin by analyzing the sparsity of individual channels within the network for selected layers. The analysis uses the representations of the hidden layers with dimensions $\mathbf{x}^l \in \mathbb{R}^{C \times NHW}$, as described in Sect. 3.2. The sparsity for each layer is calculated according to Eq. 3, using 5 000 randomly selected training samples. We select a subset of layers spread through the networks and evaluate the same layers per model type.

Figures 1 and 2 illustrate the average channel sparsity for the four models for both BatchNorm and Non-BatchNorm configurations, across the MNIST

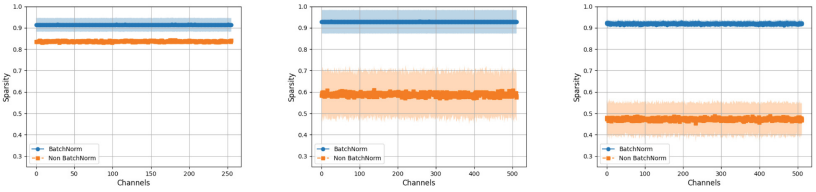
and CIFAR10 datasets, respectively. The standard deviation of metrics across different seeds is indicated with shading. We choose layers 6, 10 and 13 to probe the networks in the deeper layers at consistent intervals.



(a) Layer 2 (b) Layer 3 (c) Layer 4

Fig. 1. Average channel sparsity for BatchNorm (blue) and Non-BatchNorm (orange) models for selected convolutional layers of the standard CNN models trained on MNIST. Error bars show the standard deviation across 4 random initialization seeds. (Color figure online)

For the standard CNN models trained on MNIST, as shown in Fig. 1, we observe that the models trained with BatchNorm generally exhibit lower channel sparsity compared to those trained without BatchNorm. This observation is consistent across all channels and layers of the model, with the exception of the final layer, where channels in the BatchNorm models are more sparse than those in the Non-BatchNorm models. We note that channel sparsity remains consistent across different initialization seeds, and are also relatively uniform within each layer. The one exception is the last layer, where there is a slightly larger degree of variation in channel sparsity, especially for BatchNorm models.



(a) Layer 6 (b) Layer 10 (c) Layer 13

Fig. 2. Average channel sparsity for BatchNorm (blue) and Non-BatchNorm (orange) models in selected VGG-16 convolutional layers trained on CIFAR10, with error bars showing the standard deviation across 4 random initialization seeds. (Color figure online)

However, when we turn our attention to Fig. 2, which shows the sparsity per channel for the VGG-16 models trained on CIFAR10, the opposite behaviour

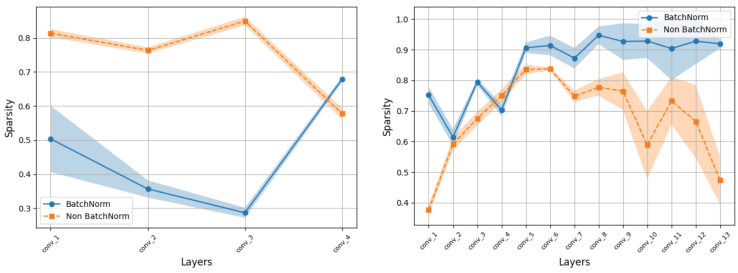
is observed. The VGG-16 models trained with BatchNorm tend to be more sparse than those trained without. We also note greater variation among different initialization seeds, especially for the Non-BatchNorm models. That said, we again note that each channel shows similar sparsity within a layer.

When comparing Figs. 1 and 2, we observe that the channels in the CIFAR10 models trained with BatchNorm are significantly more sparse than their MNIST counterparts. This could potentially be an artefact of the different datasets, as MNIST consists of many 0-valued inputs in itself whereas CIFAR10 consists of more uniformly distributed values larger than 0. It is therefore strange that the CIFAR10 BatchNorm models are *more* sparse than those of MNIST.

4.2 Layer Sparsity

In this section, instead of analysing a single channel individually, we analyse the sparsity of each layer within the networks. This analysis uses the representations of the hidden layers with dimensions $\mathbf{x}^l \in \mathbb{R}^{N \times CHW}$, as described in Sect. 3.2. We once again make use of 5 000 randomly selected training samples. For each layer, the sparsity is computed according to Eq. 4.

Figure 3 illustrates the average layer sparsity for both the MNIST (left) and CIFAR10 (right) architectures with and without BatchNorm. We again include error bars indicating the standard deviation across the 4 random initialization seeds. Figure 3 reveals a pronounced difference in layer sparsity between the standard CNNs and VGG-16 models. For the standard CNNs, the representations of the BatchNorm models are generally less sparse than those of the models trained without BatchNorm, with the exception of the final layer. Conversely, for the VGG-16 models, the BatchNorm models exhibit greater overall sparsity compared to the Non-BatchNorm models. The sparsity values are consistent across



(a) Standard CNN trained on MNIST (b) VGG-16 trained on CIFAR10

Fig. 3. Average layer sparsity for models trained with (blue) and without (orange) BatchNorm. Error bars indicate the standard deviation across 4 random initialization seeds. (Color figure online)

different seeds for both architectures, except for the last layers of the VGG-16 models where the variation is higher between seeds. Notably, the sparsity in the VGG-16 Non-BatchNorm models decreases significantly in the final layers, while the sparsity in the BatchNorm models remains consistent and relatively high.

It has been speculated that sparser representations are more desirable, as they suggest that the network is selectively activating in response to specific features. However, it is evident that BatchNorm neither consistently induces nor prevents sparsity in the representations. Yet, in both cases, the models with BatchNorm generalize better. This points to the notion that the presence or absence of sparse representations does not appear to be a primary factor that determines the generalization ability of a network. Instead, it is likely that other effects of BatchNorm play a more significant role in the network's generalization performance.

5 Clustering Analysis and Results

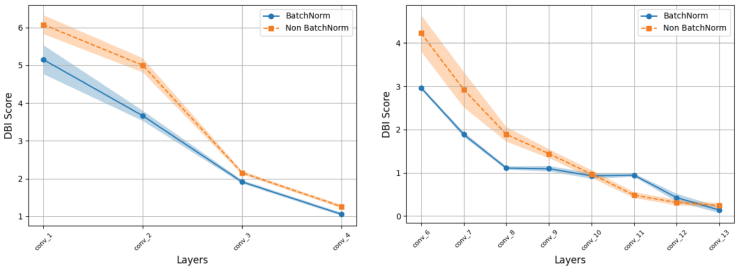
In this section, we analyse the effect of BatchNorm on the purity of representational clusters formed by the network. We consider this from both a class-specific (Sect. 5.1) and class-agnostic (Sect. 5.2) perspective.

5.1 Class-Based Analysis

We first analyze the purity of the clusters formed in each layer of the network, under the assumption that the class of each sample corresponds to the cluster to which it belongs. We use 15 000 randomly selected training samples and preprocess the representations as described in Sect. 3.3. Figure 4 presents the DBI scores for each layer of the networks trained with and without BatchNorm, for both the MNIST and CIFAR10 architectures, respectively. The average DBI scores across the 4 different initialization seeds, along with the corresponding standard deviation, are displayed. For the VGG-16 network, we only show from convolutional layer six onward as the earlier layers are not informative and show unstable behavior among different seeds.

For both network architectures, the models trained with BatchNorm generally exhibit higher cluster purity compared to those trained without. An exception is observed in layers 11 and 12 of the VGG-16 model; however, the overall cluster purity remains poorer in the BatchNorm models. Furthermore, we also observe that the purity values are relatively consistent across the different seeds, with minimal variation. Since only the 10 dataset classes serve as the cluster labels, and each sample can only belong to the cluster corresponding to its class label, both models start off with very low purity when traversing through the layers. However, in the deeper layers, clusters become increasingly pure, ultimately nearing complete purity, as observed in layer 13 of the VGG-16 models.

We suspect that the common features shared among different classes could lead to the formation of either fewer or more clusters than there are classes in the earlier layers. Overall, observations are consistent with the well-known belief



(a) Standard CNN trained on MNIST (b) VGG-16 trained on CIFAR10

Fig. 4. Average class-based DBI cluster purity score per layer for BatchNorm (blue) and Non-BatchNorm (orange) models, with error bars showing standard deviation across 4 random initialization seeds. Lower scores indicate higher purity. (Color figure online)

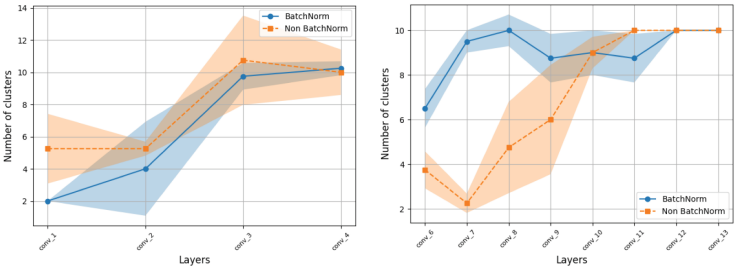
that earlier layers learn more general patterns in CNNs [7]. In deeper layers, we suspect that more class-specific features, with fewer shared features between classes, likely contribute to the higher clustering purity observed.

5.2 Class-Agnostic Analysis

We now turn our attention to the class-agnostic clustering perspective. As described in Sect. 3.3, we assign clusters using the k-means clustering algorithm, and then measure the resulting purity. As with the previous experiment, a total of 15 000 randomly selected training samples are utilized for these calculations.

Optimal Number of Clusters. The optimal number of clusters for each layer is shown in Fig. 5 for both the MNIST (left) and CIFAR10 (right) architectures. The error bars again report the standard deviation across the 4 random initialization seeds. We only show from convolutional layer 6 onward for the VGG-16 models as the initial layers are not informative and unstable between seeds. In Fig. 5 we observe that, for both architectures, the optimal number of clusters initially starts small but then increases. At the final layers, the optimal number converges to 10 clusters, corresponding to the number of classes in the dataset. For the standard CNN models, the increase in the optimal number of clusters occurs in the middle of the network; for the VGG-16 models, this occurs closer to the end. This is true for both the models trained with and without BatchNorm.

For VGG-16 models, the optimal cluster patterns differ across model types In BatchNorm models, the number of optimal clusters increases in the early layers, stabilizing until the last two layers, where it matches the number of classes. For Non-BatchNorm models, the optimal number of clusters is initially reached but significantly decreases in the middle layers, then increases again in the later layers, ultimately matching the number of classes in the final layers.

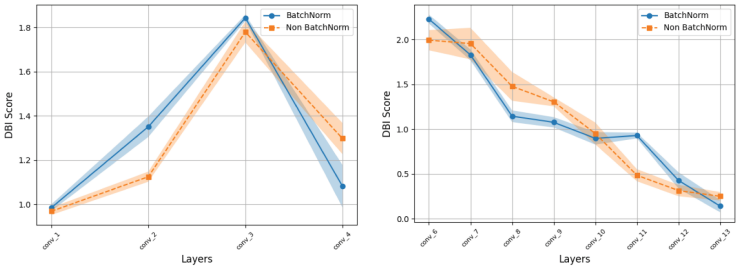


(a) MNIST trained standard CNN (b) CIFAR10 trained VGG-16

Fig. 5. Average optimal number of class-agnostic clusters per layer for BatchNorm (blue) and Non-BatchNorm (orange) models, with error bars showing the standard deviation across 4 random initialization seeds. (Color figure online)

These observations support the hypothesis presented in Sect. 5.1 that optimal class-wise clusters are achieved only in the later layers. It suggests that while different class features contribute to cluster formation, the clusters do not become class-specific until the final layers of the network. These plots also suggest that the improved generalization ability of BatchNorm models is likely due to the greater consistency of clusters formed by the representations throughout the network, compared to those in models trained without BatchNorm.

Cluster Purity. We now consider the DBI score of each optimal cluster found using k-means. This is shown for the MNIST and CIFAR10 models in Fig. 6. We again only show from convolutional layer 6 onward for the VGG-16 models as the initial layers are not informative and unstable between seeds. In Fig. 6 we observe that the standard CNNs, for both model types, follow the same pattern



(a) MNIST trained standard CNN (b) CIFAR10 trained VGG-16

Fig. 6. Average class-agnostic DBI cluster purity score for k-means per layer for BatchNorm (blue) and Non-BatchNorm (orange) models, with error bars indicating standard deviation across 4 random initialization seeds. Lower scores indicate higher purity. (Color figure online)

where the DBI score increases significantly until layer 3 and then decreases. The BatchNorm models also have a slightly higher score than the Non-BatchNorm except for the last layer where the BatchNorm has a lower score. Comparing these results with what was seen in Fig. 4 with Fig. 6 there is a difference: the scores in Fig. 4 only decreases whereas the DBI scores in Fig. 6 increases before decreasing in the last layer. One thing to note is that the DBI scores in Fig. 4 are significantly higher (poorer) than those in Fig. 6.

For the VGG-16 network, the overall DBI score of the BatchNorm models is better compared to the non-BatchNorm models. Variation between seeds is more stable for the BatchNorm models than non-BatchNorm models. This suggests that the clusters are more stable for the BatchNorm models over seeds. Comparing Fig. 6 with Fig. 4, the same pattern emerges for the VGG-16 models where the BatchNorm models have purer clusters except for layers 11 and 12. This suggests better cluster forming in general, and thus better generalization in the BatchNorm models, especially in the last layer.

6 Discussion

Let us summarise the main observations. In the context of convolutional neural networks and the experiments conducted here:

1. BatchNorm does not consistently induce nor prevent sparsity in hidden representations.
2. Sparse representations do not correlate with better generalization.
3. BatchNorm tends to produce purer representational clusters than models trained without, especially in the case of class-based repesentations. For class-agnostic representation, results are less clear, except at the final layer. where BatchNorm clusters are again purer.
4. BatchNorm forms class-agnostic clusters earlier in the network and these early clusters are more consistent with the actual number of classes.

It is challenging to determine the exact cause of the sparsity in the representations studied because we observed significant differences between models trained on MNIST and those trained on CIFAR10. However, a clear distinction exists between models with BatchNorm and those without. For standard CNNs, applying BatchNorm results in much less sparse representations, which aligns with the findings of Pretorius et al. [30]. In contrast, VGG-16 models exhibit the opposite behaviour, where BatchNorm models produce significantly more sparse representations than their non-BatchNorm counterparts.

The difference in sparsity might be influenced by the dataset on which the models are trained, or to the architectural differences between the models. Specifically, VGG-16 models with BatchNorm seem to activate for more specific features, leading to increased sparsity, particularly around layer 5. On the other hand, non-BatchNorm models in VGG-16 also show increased sparsity until layer 5 but revert to activating more general features in deeper layers, causing a reduction in sparsity. Hwever, note the variability in the results across seeds for the deeper layers – this could be indicative of other underlying processes at play.

Generally, BatchNorm models tend to produce purer clusters compared to non-BatchNorm models. In the VGG-16 models, both clustering methods yielded similar results with BatchNorm performing better overall, with the exception of layers 11 and 12. For the layers where non-BatchNorm perform better in class-agnostic clustering, it results in more clusters and a wider range of cluster numbers across seeds compared to the BatchNorm models.

The work also raises additional questions: Do clearer patterns emerge if the experiment is repeated on additional datasets and architectures? Since the representations are highly dimensional, clustering them is computationally expensive, and repeating these experiments over multiple seeds quickly becomes expensive, it remains an avenue we wish to pursue further. Similarly, we used 0 as the threshold for measuring sparsity, even though 'very small' numbers could also be considered sparse. In the current analysis, we avoided re-running experiments across multiple thresholds, although it could be worth exploring in more detail.

7 Conclusion

Studying the characteristics of DNNs' internal representations can provide insight into the underlying mechanisms that contribute to good generalization. In this study, representations were analysed from two convolutional architectures: a small fairly standard CNN architecture and a VGG-16 architecture. We developed models with similar training accuracy but – since the models with BatchNorm generalized better on unseen data – the models represented different generalization gaps. The main question asked was: how does BatchNorm affect the characteristics of the internal representations of these two sets of models?

Representational sparsity did not correlate either with the generalization ability of the networks or could be linked to whether BatchNorm was applied or not. Clearer trends emerged with regard to the consistency of representations, specifically when considering the purity of clusters formed by these representations at different layers in the network. Overall, BatchNorm tended to produce purer representational clusters, both in the case of class-based and class-agnostic analysis. In addition, class-specific clusters formed earlier in the BatchNorm networks, and these were more consistent with the actual number of classes.

Trends observed were complex and pose additional questions that can be explored further. In this work, our objective was to provide initial results and motivate the importance and potential of studying the characteristics of DNN representations to shed light on the underlying mechanisms controlling the generalization ability of these models.

Acknowledgments. This work is based on the research supported in part by the National Research Foundation of South Africa (Ref Numbers PSTD23042296065, RA211019646111).

References

1. Arthur, D., Vassilvitskii, S.: K-means++: the advantages of careful seeding. In: Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms, vol. 8 (2007)
2. Balestriero, R., Baraniuk, R.G.: Batch normalization explained. arXiv preprint arXiv:2209.14778 (2022)
3. Bengio, Y., Courville, A., Vincent, P.: Representation learning: a review and new perspectives. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1798–1828 (2013)
4. Benz, P., Zhang, C., Kweon, I.S.: Batch normalization increases adversarial vulnerability and decreases adversarial transferability: a non-robust feature perspective. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2021)
5. Bjorck, N., Gomes, C.P., Selman, B., Weinberger, K.Q.: Understanding batch normalization. In: Advances in Neural Information Processing Systems NeurIPS, vol. 31 (2018)
6. Bousquet, O., Elisseeff, A.: Algorithmic stability and generalization performance. In: Advances in Neural Information Processing Systems NeurIPS, vol. 13 (2000)
7. Caron, M., Bojanowski, P., Joulin, A., Douze, M.: Deep clustering for unsupervised learning of visual features. In: Proceedings of the European Conference on Computer Vision (ECCV) (2018)
8. Chuang, C.Y., Mroueh, Y., Greenewald, K., Torralba, A., Jegelka, S.: Measuring generalization with optimal transport. In: Advances in Neural Information Processing Systems NeurIPS, vol. 34 (2021)
9. Daneshmand, H., Joudaki, A., Bach, F.: Batch normalization orthogonalizes representations in deep random networks. In: Advances in Neural Information Processing Systems NeurIPS, vol. 34 (2021)
10. Davies, D.L., Bouldin, D.W.: A cluster separation measure. IEEE Trans. Pattern Anal. Mach. Intell. PAMI **1**(2), 224–227 (1979)
11. Dunn, J.C.: A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. J. Cybern. **3**(3) (1973)
12. Foret, P., Kleiner, A., Mobahi, H., Neyshabur, B.: Sharpness-aware minimization for efficiently improving generalization. In: 9th International Conference on Learning Representations, ICLR (2021)
13. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, vol. 15 (2011)
14. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
15. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning ICML, vol. 37 (2015)
16. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall Inc., Upper Saddle River (1988)
17. Jiang, Y., Krishnan, D., Mobahi, H., Bengio, S.: Predicting the generalization gap in deep networks with margin distributions. In: 7th International Conference on Learning Representations, ICLR (2018)
18. Kawaguchi, K., Deng, Z., Ji, X., Huang, J.: How does information bottleneck help deep learning? In: Proceedings of the 40th International Conference on Machine Learning ICML, vol. 202 (2023)

19. Krizhevsky, A.: Learning multiple layers of features from tiny images (2012). https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
20. LeCun, Y., Cortes, C., Burges, C.: The MNIST database of handwritten digits, **2** (1998). http://yann.lecun.com/exdb/mnist/
21. Li, J., Zhou, P., Xiong, C., Hoi, S.: Prototypical contrastive learning of unsupervised representations. In: 9th International Conference on Learning Representations, ICLR (2021)
22. Li, Z., Arora, S.: An exponential learning rate schedule for deep learning. In: 8th International Conference on Learning Representations, ICLR (2019)
23. Liao, R., Schwing, A., Zemel, R., Urtasun, R.: Learning deep parsimonious representations. In: Advances in Neural Information Processing Systems NeurIPS, vol. 29 (2016)
24. MacQueen, J., et al.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1 (1967)
25. Mouton, C., Davel, M.: Exploring layerwise decision making in dnns. Commun. Comput. Inf. Sci. CCIS **1551** (2022)
26. Mouton, C., Theunissen, M.W., Davel, M.H.: Input margins can predict generalization too. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, no. 13 (2024)
27. Nakkiran, P., Kaplun, G., Bansal, Y., Yang, T., Barak, B., Sutskever, I.: Deep double descent: where bigger models and more data hurt. J. Stat. Mech. Theory Exp. **2021** (2019)
28. Natekar, P., Sharma, M.: Representation based complexity measures for predicting generalization in deep learning. arXiv preprint arXiv:2012.02775 (2020)
29. Nielsen, F.: Hierarchical clustering, pp. 195–211 (2016)
30. Pretorius, A.M., Barnard, E., Davel, M.H.: Relu and sigmoidal activation functions. In: Proceedings of South African Forum for Artificial Intelligence Research FAIR (2019)
31. Santurkar, S., Tsipras, D., Ilyas, A., Madry, A.: How does batch normalization help optimization? In: Advances in Neural Information Processing Systems NeurIPS, vol. 31 (2018)
32. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8) (2000)
33. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: 3rd International Conference on Learning Representations, ICLR (2015)
34. Zhan, X., Xie, J., Liu, Z., Ong, Y.S., Loy, C.C.: Online deep clustering for unsupervised representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020)
35. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. Commun. ACM **64**(3) (2021)

Does Simple Trump Complex? Comparing Strategies for Adversarial Robustness in DNNs

William Brooks[1,2,4(✉)], Marelie H. Davel[1,2,3], and Coenraad Mouton[1,2]

[1] Faculty of Engineering, North-West University, Potchefstroom, South Africa
williambrooks499@gmail.com
[2] Centre for Artificial Intelligence Research, Cape Town, South Africa
[3] National Institute for Theoretical and Computational Sciences, Stellenbosch, South Africa
[4] South African National Space Agency, Pretoria, South Africa
http://engineering.nwu.ac.za/must

Abstract. Deep Neural Networks (DNNs) have shown substantial success in various applications but remain vulnerable to adversarial attacks. This study aims to identify and isolate the components of two different adversarial training techniques that contribute most to increased adversarial robustness, particularly through the lens of margins in the input space – the minimal distance between data points and decision boundaries. Specifically, we compare two methods that maximize margins: a simple approach which modifies the loss function to increase an approximation of the margin, and a more complex state-of-the-art method (Dynamics-Aware Robust Training) which builds upon this approach. Using a VGG-16 model as our base, we systematically isolate and evaluate individual components from these methods to determine their relative impact on adversarial robustness. We assess the effect of each component on the model's performance under various adversarial attacks, including AutoAttack and Projected Gradient Descent (PGD). Our analysis on the CIFAR-10 dataset reveals which elements most effectively enhance adversarial robustness, providing insights for designing more robust DNNs.

Keywords: Deep Neural Networks · Adversarial robustness · Margins in input space · Dynamics-Aware Robust Training · Large margin loss · Adversarial attacks

1 Introduction

Deep neural networks (DNNs) have transformed numerous fields and have achieved remarkable success in tasks ranging from image classification to natural language processing. Despite their widespread adoption, DNNs are inherently vulnerable to adversarial perturbations – small perturbations to the input data that lead a deep learning model to misclassify the input, often with high confidence [19]. This vulnerability poses challenges where reliability and security of machine learning models are paramount.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 253–269, 2025.
https://doi.org/10.1007/978-3-031-78255-8_15

The exact reason for the susceptibility of DNNs to these perturbations remains unknown and is an ongoing area of research. Several methods have been developed to generate adversarial examples and evaluate the robustness of DNNs [6,7,15,16]. Popular attack algorithms, such as Projected Gradient Descent (PGD) [15], have become standard tools for testing the robustness of machine learning models. These attack algorithms aim to exploit the vulnerability of neural networks to adversarial examples and current research aims to improve the robustness of DNNs, by building defenses to such perturbations [2,3,17,22].

Popular defences against adversarial attacks, such as Adversarial Training (AT) [10,15], aim to enhance the robustness of these models by explicitly exposing them to adversarial examples during training. Despite being effective, this approach is computationally expensive as it requires generating adversarial perturbations during training and does not guarantee robustness against all types of attacks. Margin-based techniques, while often similarly computationally expensive, have been explored as an alternative due to their ability to offer a better robustness-accuracy trade-off compared to methods like PGD adversarial training [9,26]. Margin-based techniques focus on increasing the margin – the minimum distance between a sample and its decision boundary – in the input space. Large margins in the input space have been shown to be correlated with improved adversarial robustness of DNNs [8].

The work by Xu et al. [26] and Elsayed et al. [9] primarily focused on increasing margins, utilizing innovative approaches to margin maximization (MM) of DNNs. Elsayed et al. introduced a novel loss function based on the first-order Taylor approximation of the margin, which maximizes the margin at any layer, demonstrating improvements in adversarial robustness. Xu et al. built on this concept by identifying the vulnerability of data points with smaller margins and proposing Dynamics-Aware Robust Training (DyART), a technique that prioritizes increasing these smaller margins in the input space. DyART achieved near state-of-the-art performance on benchmarks such as RobustBench [5] improving on recognized baselines such as AT [15], TRADES [27], MMA [8], GAIRAT [28], MAIL [21] and AWP [24] under various perturbation bounds.

Both the approaches by Elsayed et al. [9] and Xu et al. [26] focus on MM and introduce loss functions for increasing margins in the input space to improve the adversarial robustness of DNNs. Despite the progress made by these margin-based techniques, the techniques themselves are complex and it is not clear which elements most contribute to enhancing adversarial robustness. To address this question, we conduct a detailed analysis of the adversarial robustness of models trained using the techniques proposed by Xu et al. and Elsayed et al. We use AutoAttack [7], a state-of-the-art evaluation method, to systematically compare the performance of these methods and identify the key factors contributing to their robustness. The primary contributions of this study are as follows:

1. An evaluation of the adversarial robustness of models trained using Xu et al. and Elsayed et al.'s techniques,
2. Identification of the impactful components of these methods in enhancing adversarial robustness,

3. Insights into the aspects which matter when constructing adversarially robust classifiers.

The remainder of this paper is organized as follows: Sect. 2 provides an overview of adversarial training, margin-based techniques, and the work by Xu et al. [26] and Elsayed et al. [9]. Section 3 describes the approach followed to disentangle the effects of the underlying elements of each technique and describes the individual components analysed. Section 4 describes the experimental setup, while the results of the analysis are presented in Sect. 5.

2 Background

In this section, we provide background information on adversarial training and the two main methods we consider. First (Sect. 2.1) we provide a brief introduction to adversarial training in general, and how the robustness of a model is typically evaluated. Following this, we describe how margins are defined and discuss the commonalities of adversarial training techniques that maximize margins (Sect. 2.2). Finally, we provide an overview of the MM methods of Elsayed et al. and Xu et al. in Sects. 2.3 and 2.4, respectively.

2.1 Adversarial Training and Evaluation

Adversarial examples pose a considerable challenge to the robustness of deep neural networks (DNNs). The vulnerability of DNNs to these adversarial examples has sparked significant research into methods that can mitigate their effect. Although AT is the standard approach for enhancing the robustness of classifiers, it often results in a decrease in clean accuracy (accuracy when considering standard samples, not adversarially manipulated ones), a phenomenon known as the accuracy-robustness trade-off [20]. Therefore, current research aims to develop robust training methods able to provide the best possible robustness to adversarial attacks with as little possible drop in performance on clean examples. This is an active area of research and many methods have been proposed [8,26,27].

Conversely, the goal of adversarial attack algorithms are to fool a classifier by slightly perturbing clean examples. This has led to a research arms race where stronger adversarial attacks are developed to fool robust training methods, examples of which are Projected Gradient Descent (PGD) [15], Fast Adaptive Boundary (FAB) [6], DeepFool [16], Fast Gradient Sign Method (FGSM) [10], and the Carlini&Wagner (C&W) attack [4]. These attacks can be used to generate adversarial examples, but can also be used to evaluate the robustness of DNNs to these attacks. However, training a model using these attack-specific adversarial perturbed examples can lead to overfitting to specific types of attacks. Although a model trained with AT might be robust against PGD attacks, for example, it might not generalize well to other, potentially unknown, attacks [11]. It is therefore important to measure the robustness of models using current state-of-the-art (SOTA) approaches that combine multiple attacks.

AutoAttack [7] has become the standard benchmark for evaluating the adversarial robustness of modern DNNs. AutoAttack consists of an ensemble of four attacks. These four attacks include two variations of the standard PGD attack developed by Madry et al. [15]: an enhanced version of PGD known as Auto-PGD with cross-entropy (APGD-CE) and the targeted Auto-PGD (APGD-t) [7]. Additionally it includes the targeted FAB [6] attack, and the black-box Square attack [1]. Croce et al. established a framework, RobustBench [5], that utilizes AutoAttack with the intention of being a standardized benchmark to measure the robust accuracy of DNNs, and to compare findings and results.

2.2 Margin Maximization

When employing adversarial training, it is necessary to distinguish between techniques that directly utilize adversarial examples – training with both clean samples and adversarial examples – and those that focus on MM. Prominent MM approaches include TRADES [27], the work by Elsayed et al. [9], DyART [26], and MMA [8].

These approaches leverage the concept of a margin, denoted as R from here onwards, defined as the minimum distance of a sample to its nearest decision boundary. Consider a deep classifier f parameterized by θ, with inputs $x \in X$ and corresponding labels $y \in Y = \{1, 2, \ldots, n\}$, that generates a prediction score to classify the input vector x to class y. We can define the prediction of the classifier f_θ as the class y corresponding to the maximum output logit $z_\theta^y(x)$ given by $f_\theta(x) = \arg\max_{y \in Y} z_\theta^y(x)$. To define the decision boundary, we introduce ϕ, which represents the logit margin – the difference between the logit value for the true class y and the highest logit value among all other classes $y' \neq y$:

$$\phi_\theta(x) = z_\theta^y(x) - \max_{y' \neq y} z_\theta^{y'}(x) \tag{1}$$

A sample x is deemed to be correctly classified when $\phi_\theta^y(x) > 0$ and incorrectly classified otherwise. The decision boundary exists where there is a tie, that is, $\phi_\theta^y(x) = 0$. We can then define the margin, R_θ, as the distance of a sample x to the nearest sample $\hat{x}$ on the decision boundary:

$$R_\theta(x) = \min_{\hat{x}} \|\hat{x} - x\|_p \quad \text{s.t.} \quad \phi_\theta^y(\hat{x}) = 0 \tag{2}$$

Here $\|.\|_p$ is some chosen l_p norm. We typically use the l_∞ norm as is convention [15,26].

The core idea of MM is to utilize a loss function that combines traditional cross-entropy loss ($\mathcal{L}_{\text{CE}}$) with a MM loss term ($\mathcal{L}_{\text{MM}}$). This hybrid loss function, as shown in Eq. 3, is designed to increase margins in the input space, thereby improving the model's adversarial robustness during training.

$$\mathcal{L} = \mathcal{L}_{\text{CE}}(y, \hat{y}) + \lambda \cdot \mathcal{L}_{\text{MM}}(\gamma, R) \tag{3}$$

Here y is the true label of a sample and $\hat{y}$ the model's corresponding prediction. The $\mathcal{L}_{\text{MM}}$ term typically includes a user-defined hyperparameter, γ, which

represents the minimum margin that should be maintained, making the model less susceptible to perturbations of an input. In addition, a hyperparameter λ is incorporated to weigh the margin maximization component of the loss function.

We will primarily focus on the MM loss functions proposed by Elsayed et al. and Xu et al., respectively. Both methods utilize hybrid loss functions that aim to maximize margins in the input space whilst minimizing the loss with respect to the inputs for correct classification [9,26].

2.3 Large Margin Loss Approach

The L_{MM} function introduced by Elsayed et al., for a batch $\mathcal{B}$ of size N, is defined as:

$$\mathcal{L}_{MM}(\gamma, R) = \frac{1}{N} \sum_{i \in \mathcal{B}} \mathscr{A}_{y' \neq y} \max \{0, \gamma - R_\theta(x_i)\} \tag{4}$$

Here $\mathscr{A}_{y' \neq y}$ is some aggregation operator, and they experiment with choosing this as either the *max* operator or the *sum* operator. For each training sample x and corresponding label y, the individual loss is computed for each class where $y' \neq y$. The margin of a sample x is therefore calculated to the decision boundary of every other class, $y' \neq y$. These individual losses are then aggregated using the max or sum operator.

Intuitively, this loss function aims to ensure that all samples have at least a margin of γ (the minimum margin) to the nearest decision boundary. Additionally, it penalizes samples with smaller margins more than those with larger margins, and also accommodates both correctly and incorrectly classified samples. Consider a correctly classified sample with a margin smaller than the minimum margin. The margin loss for this sample will then be equal to $\gamma - R$. However, if the samples has a margin larger than the minimum margin, it will simply be assigned a loss of zero. On the other hand, if a sample is incorrectly classified, the margin (R) will be negative and the loss for that sample will be adjusted to account for both the distance the decision boundary must move for it to be correctly classified and have a positive margin of at least γ.

The behavior of this loss function is perhaps made more clear if one considers how Elsayed et al. measures the margin for each sample. While some authors measure the exact distance to an adversarial example to measure the margin, Elsayed et al. opt to rather approximate this distance. This greatly reduces the computational cost, as calculating adversarial examples for each batch of training data can be highly expensive. Specifically, they utilize a first order Taylor approximation, as shown in Eq. 5, to approximate the margin R.

$$R_\theta(x) = \frac{z_\theta^y(x) - z_\theta^{y'}(x)}{\left\| \nabla_{\boldsymbol{x}} z_\theta^y(x) - \nabla_{\boldsymbol{x}} z_\theta^{y'}(x) \right\|_q}, \tag{5}$$

The numerator in Eq. 5 is given by the difference between the output logits of the true class, y, and some other class, y' (i.e. the same as Eq. 1). The denominator computes the difference between the gradients $\nabla_{\boldsymbol{x}}$ of $z_\theta^y(x)$ and $z_\theta^{y'}(x)$ with regard

to to the input $\boldsymbol{x}$, and then calculates the norm $\|.\|_q$. Calculating the gradient terms requires calculating the Jacobian matrix which gives the partial derivatives of the logits with respect to the input x. Here, $\|.\|_q$ denotes the dual norm of $\|.\|_p$ where $\frac{1}{q} + \frac{1}{p} = 1$. Specifically, when $p = \infty$, then $q = 1$. Furthermore, it should be clear that if the sample is incorrectly classified, the numerator will be negative, and the resulting margin will also be a negative value. Since there is a derivative in the MM loss function, it requires calculating the Hessian (second-order gradient) when calculating the gradient of the loss w.r.t. to the model parameters for gradient descent, which is computationally too expensive. They therefore treat the denominator in Eq. 5 as a constant during back-propagation.

Finally, Elsayed et al. also experiment with extending this loss function to the margin of each sample as measured at the hidden layers. Specifically, they replace the input x with its hidden representation h_l at a given layer l. Their goal is to enforce a large margin based on the data representation at each specific layer. We do not consider the maximization of margins measured at the hidden layers in this work.

2.4 DyART

Similar to the approach by Elsayed et al., Xu et al. designed a MM loss function. They identified an issue with other robust training methods where increasing the margin for one datapoint might decrease the margin of another data point. They called this 'conflicting dynamics'. To mitigate this issue, they introduced DyART, an MM loss function designed to prioritize increasing smaller margins [26]. This ensures that all samples have a margin of at least γ. Their MM function is defined as follows:

$$MM(R) = \begin{cases} \frac{1}{\alpha}\exp(-\alpha R), & R < \gamma \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

This function assigns an exponential loss to samples with a margin smaller than γ; otherwise, the loss is zero. The hyperparameter, α, smooths this exponential function. However, unlike the MM function of Elsayed et al., the loss function is only applied when $\phi_\theta^y(x) > 0$, meaning incorrectly classified samples are ignored. Since most samples are still incorrectly classified at the start of training, a burn-in period or training warmup is recommended, where the model is trained naturally using cross-entropy for a certain number of epochs before applying the MM loss function. The loss for a training batch $\mathcal{B}$ of size N, with correctly classified samples $\mathcal{B}_\theta^+$ of size m, is defined as:

$$\mathcal{L}_{MM}(R) = \frac{1}{m} \sum_{i \in \mathcal{B}_\theta^+} MM(R_\theta(x_i)) \tag{7}$$

Xu et al. also use a more accurate estimation of the margin for each sample than Elsayed et al. Instead of relying on the first-order Taylor approximation, DyART utilizes FAB [6] to find the closest boundary point $\hat{x}$ from sample x. The margin

is then given by $||x - \hat{x}||_\infty$. FAB iteratively applies the Taylor approximation to find a point on the decision boundary, and is much more computationally expensive. In addition, if the margin is calculated to each class it needs to calculate the Jacobian at each iteration between the true class and every other class. Therefore, finding the closest point on the decision boundary becomes computationally prohibitive, especially for datasets with a large number of classes, such as CIFAR-100. FAB also requires multiple iterations to produce high quality adversarial examples.

To lessen this computational burden, Xu et al. do not calculate the margin to each non-label class for each samples. Instead, they introduced an alternative method for calculating the margin, known as the soft margin, $R_\theta^{\text{soft}}(x)$, which is the minimum distance of sample to a point on the 'soft decision boundary'. The soft decision boundary is an approximation of the exact, or 'true', decision boundary. More specifically, the soft decision boundary is given by the points where $\phi_\theta^y(x;\beta) = 0$ for a hyperparameter β, and $\phi_\theta^y(x;\beta)$ is defined as:

$$\phi_\theta^y(x;\beta) = z_\theta^y(x) - \frac{1}{\beta}\log\sum_{y'\neq y}\exp\left(\beta z_\theta^{y'}(x)\right) \tag{8}$$

Therefore, instead of calculating the distance to the exact decision boundary, which requires evaluating the logits for all possible pairs of classes, as defined in Eq. 1, the Log-Sum-Exp function gives a weighted average which emphasizes the largest logits without needing to compare each logit. This reduces the number of operations required as it avoids the need to compute the margin to every possible pair of logits. Moreover, this soft decision boundary is controlled by a hyperparameter, β, and the higher this value, the closer the soft decision boundary represents the exact decision boundary.

Finally, recall that Elsayed et al. are unable to calculate the appropriate gradient of L_{MM} w.r.t. the model parameters during training due to computational constraints. Xu et al. encounter the same problem, since the margin calculation (recall Eq. 2), is within itself an optimization problem. Inspired by MMA [8], Xu et al. derive a closed-form expression that allows efficiently calculating the gradient of L_{MM} w.r.t. the model parameters. This approach allows for the model parameters to be updated based on a more accurate representation of how the margin changes, rather than approximating it as done by Elsayed et al. [9] and Ding et al. [8].

3 Approach

As should be clear from Sect. 2, the DyART adversarial training technique can be considered a more complicated version of Elsayed's that introduces several improvements. Our approach is to progressively introduce each of these improvements individually to Elsayed's loss function to isolate and identify design choices that lead to improved adversarial robustness.

In this section we elaborate on the different models we train and the expected behavior of each design choice. We provide more extensive implementation and hyperparameter details in the following section. We first describe our baseline models, in Sect. 3.1. Following this, we describe the different elements we introduce to Elsayed's method in Sect. 3.2. Finally, in Sect. 3.3 we describe how we evaluate each model's performance.

3.1 Baseline Models

We start by implementing three baseline models using the CIFAR-10 dataset, which we partition into 45,000 training samples and 5,000 validation samples: 1) a standard model trained with vanilla cross-entropy loss, 2) a model trained using Elsayed's MM loss function, and 3) a model trained with Xu's DyART MM loss function. This allows us to compare each change we make to the original methods, as well as compare to a 'normally' trained model.

We also make some slight changes to the original implementations to suit our needs. In terms of Elsayed's method, we opt to only approximate the margin to the highest predicted non-label class, instead of calculating the margin for every class and using an aggregation operator. This reduces the computational burden significantly. Additionally, as mentioned in Sect. 2.3, Elsayed et al. experimented with increasing the margins at the hidden layers as well. We only apply his approach in the input space in order to make it comparable with the DyART method.

For the DyART method of Xu et al. we also make some slight changes. They implemented stochastic weight averaging [12] for improved performance. We omit this as to better isolate the effect of each loss function individually. In addition to this, they also train on an additional 10M synthetically generated images to further improve their robustness results. We omit this for the same reason.

3.2 Additions to Elsayed

In this section we elaborate on each of the elements we add to Elsayed's MM method.

Elsayed's MM with Burn-in. Our approach begins with the Burn-in Phase, where we enhance the standard MM approach by Elsayed et al. by incorporating an initial burn-in period. During this phase, the model undergoes natural training for several epochs using standard cross-entropy loss before any MM techniques are applied. This helps the model establish a solid baseline with relatively high clean accuracy and a greater proportion of correctly classified samples. To optimize this phase, we also need to determine the best set of hyperparameters to maximize the number of correctly classified samples before transitioning to the margin-maximization process. The reasoning behind this is that a strong initial baseline may facilitate the MM process, potentially leading to enhanced adversarial performance.

Elsayed's MM with Burn-in and Exponential Loss. Following the burn-in phase, we experiment with the Exponential Loss Function, as defined in Eq. 6. This function is designed to operate on correctly classified samples, which necessitates the burn-in period, and encourages the model to maintain a margin of at least γ. Because this loss function is exponential, it penalizes samples with smaller margins more heavily than those with larger margins. Incorporating this into Elsayed's loss function might be beneficial in improving the adversarial robustness due to the way smaller margins are managed and penalized.

Elsayed's MM with FAB. Next, we replace Elsayed's Taylor approximation Eq. 5 with FAB, aiming to achieve a more precise margin approximation. FAB accurately calculates the closest boundary points, offering a significant improvement over the original first-order Taylor approximation method. This however also requires adapting FAB to function similarly to our adapted Elsayed et al., to only consider the highest predicted class instead of all the classes. This provides some insight into the importance of a much more accurate approximation of the margin, meaning if it is worth the added computational cost and if this is crucial for increasing the adversarial robustness of DNNs.

Elsayed's MM with FAB and Soft Boundary. Finally, we integrate the Soft Boundary technique with our adapted FAB to balance the trade-off between computational cost and the accuracy of margin estimation. Comparing the results of utilizing FAB on its own versus with the soft boundary will shed more light on the importance of a more accurate approximation of the margin and whether reducing the compute by utilizing their soft boundary approach can be beneficial.

By systematically testing the listed components in a carefully considered order: beginning with the Burn-in Phase, followed by the Exponential Loss Function, then Improved Margin Approximation with FAB, and finally the integration of the Soft Boundary, we aim to develop a better understanding of how each element contributes to the overall robustness of the model.

3.3 Evaluating Robustness

For each model, we evaluate its adversarial robustness in several ways. We rely on existing adversarial attack algorithms, and select three specific attacks and two key use cases. First, we use standard PGD as used by Madry et al. [15] to generate adversarial examples for each epoch during training using the validation set. These examples serve as an additional validation set, where we calculate the accuracy and robust loss of our model. This approach helps mitigate robust overfitting – where prolonged training leads to increased robust loss despite decreasing (clean) training loss [23] – by applying early stopping on the adversarial validation set.

The second use case involves evaluating the adversarial robustness of our final models. We rely on two different attack algorithms: APGD-CE [7] – an

improved version of the standard PGD attack [15], and finally AutoAttack [7]. While AutoAttack serves as a strong benchmark, APGD-CE allows us to gain a more in-depth understanding of how our models respond to different gradient-based attacks. This combination provides sufficient information to evaluate the adversarial robustness of our models and gain insights into the contributions of the different components.

4 Experimental Setup

In this section, we elaborate on our implementation and hyperparameter details for the experiments outlined in the previous section. First, in Sect. 4.1, we describe the dataset and architecture we consider, along with the standard training hyperparameters. Following this, in Sect. 4.2, we elaborate on how hyperparameters were tuned for each model individually.

4.1 Architecture and Standard Training Parameters

In this section we explain our standard training setup, including the dataset, architecture, and training procedure we use. We use the CIFAR10 [13] for all of our experiments. This is a popular dataset for investigating adversarial robustness [3,5,9,26]. This is because even though the dataset is considered relatively simple, building a robust CIFAR10 classifier is still an open problem [3]. Furthermore, we specifically focus on robustness to l_∞ attacks with a maximum perturbation bound of $\delta = \frac{8}{255}$, which is also a popular evaluation case to investigate on CIFAR10 [5].

In terms of model architecture, Xu et al. and Elsayed et al. both conducted experiments using larger CNN-based architectures, such as Wide Residual Networks and ResNet-18 s. However, these are very large models that are expensive to train. Instead, to accommodate our computational budget, we rely on the smaller but well known VGG-16 architecture [18] for all experiments.

We train our models using stochastic gradient descent with a momentum of 0.9, employing cosine annealing for the learning rate without restarts and a minimum learning rate decay of 0.001 [14]. Cosine annealing allows us to set an initial learning rate and define a target learning rate to which it should decay or increase. This approach enables us to maintain constant learning rates during training or to implement learning rate schedules with specific patterns, such as increasing or decreasing the learning rates from some initial learning rate to some specified target value. The models are trained for 200 epochs, and we use batch normalization on all layers. Xu et al. also experimented with using group normalization [25], but we find that batch normalization offers better performance.

For model selection and to mitigate robust overfitting [23], we created an additional adversarial validation set consisting of 1,024 samples generated from the original validation set using a 20-step PGD attack [15] with a perturbation bound of $\delta = \frac{8}{255}$. Early stopping is performed based on the model's performance

on this adversarial validation set, and the model with the highest adversarial validation accuracy was selected.

4.2 Hyperparameter Optimization

In this section, we expand on how hyperparameters were chosen for each model. We list each model and describe the hyperparameter optimization process. From each sweep, we select two models: the model with the highest clean validation accuracy, and the model with the highest validation APGD-CE robust accuracy. We report on the results of these hyperparameter optimizations in the following section.

Baseline VGG16. We performed extensive hyperparameter sweeps to identify the best settings for our VGG-16 model. Initially, we explored various batch sizes (64, 128, 256, 512), learning rates (0.1, 0.01, 0.001), and weight decay values (0.001, 0.0005, 0.0002) to determine the optimal configuration. The model with the highest validation accuracy used a batch size of 128, a learning rate of 0.1, and a weight decay of 0.0005. This model achieves a very respectable 89.24% accuracy on the CIFAR10 test set. We also use these hyperparameters as a starting point for further experiments.

DyART. To find optimal hyperparameters for the VGG16 model using the DyART training method, we conduct the hyperparameter sweep in two separate phases. First, we conducted a sweep to optimize the burn-in period. This phase involved experimenting with several learning rates (0.1, 0.01, 0.001) and batch sizes (64, 128, 256, 512, 1024). Since the DyART training method is very computationally expensive (due to the expensive margin-measuring method), we limited the burn-in phase to 25 epochs of standard training followed by only 3 epochs using the DyART loss function. For the hyperparameters specific to the DyART MM loss, we use the exact same hyperparameters as used by Xu et al. [26], which are $\alpha = 3$, $\gamma = \frac{16}{255}$, $\lambda = 1000$, $\beta = 5$, and $\delta = \frac{8}{255}$. For the second phase, we selected the top 10 models' hyperparameters based on their adversarial validation accuracy from the initial sweep. We then proceeded to retrain these models for 25 epochs of natural training followed by 200 epochs with the DyART loss function.

Elsayed's Margin Maximization Loss. For the MM loss function of Elsayed et al., we conducted a full sweep to determine the optimal learning rate (0.1, 0.01, 0.001) and batch size (128, 256, 512). In addition, it was necessary to also search over values of λ (10, 15, 20, 25), as the scale of the margin loss is different than when using Xu's method and therefore requires a different weighting with cross-entropy[1] We found that a batch size of 128, a learning rate of 0.1 and $\lambda = 25$ yielded the best adversarial validation accuracy.

[1] We could also not rely on the λ values chosen by Elsayed et al., as they included the margins of hidden layers which also changes the scale.

Additions to Elsayed For each component that we add to Elsayed's loss function (as described in Sect. 3), we use the hyperparameters that yielded the highest robust validation accuracy from the hyperparameter sweep described above, for the vanilla implementation of Elsayed's margin maximzation loss. However, we also search over different learning rates for each addition to ensure performance. This is with the exception of the addition of Xu's exponential margin loss, which necessitates changing λ as it changes the scale of the margin loss. In the following section we present results using both $\lambda = 25$ (as used for Elsayed's method) and $\lambda = 1000$ (as used for Xu's method).

5 Results

This section presents the outcome of the experiments designed to optimize adversarial robustness for the VGG-16 model using the approaches detailed in Sect. 3 and setup defined in Sect. 4. As outlined in Sect. 4.2, we conducted extensive hyperparameter sweeps across various configurations, including different learning rates, batch sizes, and regularization parameters, to identify the optimal settings for both DyART and MM loss functions. The results of these sweeps are summarized in Table 1.

We first discuss the observations for each result in Table 1 individually, before considering the results as a whole in the following section.

VGG-16 Baseline: The VGG-16 baseline shows no robustness under strong adversarial attacks, with an APGD-CE and AutoAttack accuracy of 0%. This is to be expected, as the model has not undergone any training to withstand adversarial attacks.

Elsayed's MM Function: Introducing Elsayed's MM function drastically improves the model's robustness, with the best APGD accuracy model achieving 14.29% accuracy on APGD-CE, while maintaining a relatively high clean accuracy of 87.35%. This initial comparison shows the effectiveness of Elsayed's MM function in significantly enhancing adversarial robustness compared to the baseline without a substantial drop in clean accuracy. However, we do not observe any gains in accuracy when using AutoAttack.

Elsayed's MM with Burn-in: The burn-in phase to Elsayed's MM function further improves robustness. The model with the best APGD-CE accuracy reaches 27.64%, almost doubling the robustness compared to the MM-only approach. The clean accuracy sees a slight decrease, but this trade-off can be justified by the considerable gain in robustness. This indicates that first allowing the model to learn from clean examples before attempting to maximize the margins is the better strategy for achieving robustness. That said, against the stronger adversary of AutoAttack, we still observe an accuracy of 0%.

Table 1. Performance of VGG-16 models evaluated under a perturbation bound of 8/255. 'Clean Accuracy' is reported for both validation and test sets, while 'Robust Accuracy' includes results from AutoAttack and APGD-CE. For each configuration - DyART and Elsayed's MM function - we report the results for models with the highest clean accuracy and the highest APGD accuracy on the test set.

Model	Robust Accuracy		Clean Accuracy	
	AutoAttack	APGD-CE	Val	Test
VGG-16 Baseline				
Baseline	0	0	90.08	89.24
Elsayed's MM function				
Best Clean Accuracy	0	13.17	88.64	87.62
Best APGD Accuracy	0	14.29	88.70	87.35
Elsayed's MM with Burn-in				
Best Clean Accuracy	0	13.30	88.12	87.00
Best APGD Accuracy	0	27.64	87.34	86.44
Elsayed's MM with Burn-in and Exponential Loss $\lambda = 25$				
Best Clean Accuracy	0	9.22	87.84	87.07
Best APGD Accuracy	0	26.78	86.30	85.15
Elsayed's MM with Burn-in and Exponential Loss $\lambda = 1000$				
Best Clean Accuracy	0	47.38	88.32	87.51
Best APGD Accuracy	0.13	49.26	84.58	83.60
Elsayed's MM with FAB				
Best Clean Accuracy	0	4.46	88.54	87.47
Best APGD Accuracy	0	5.40	85.98	85.24
Elsayed's MM with FAB and Soft Boundary				
Best Clean Accuracy	0	2.80	87.38	86.86
Best APGD Accuracy	0	25.70	86.32	85.26
DyART				
Best Clean Accuracy	0.01	18.77	85.60	84.89
Best APGD Accuracy	33.24	33.52	74.72	72.81

Elsayed's MM with Burn-in and Exponential Loss: The exponential loss with $\lambda = 25$ leads to mixed results. While the best APGD- accuracy model still performs well with an APGD-CE of 26.78%, this is slightly lower than with burn-in alone. However, when λ is increased to 1000, there is a significant boost in robustness, with the APGD-CE jumping to 49.26%, a substantial improvement over previous configurations. This illustrates that the λ parameter must be carefully calibrated to strike an optimal balance between standard cross-entropy loss and the MM loss. We also find a 0.13% robust accuracy on AutoAttack using the larger value for λ, albeit at the cost of a noticeable reduction in clean accuracy,

highlighting the increased trade-off as robustness improves. This points to the notion that the exponential function is effective (to some extent) in mitigating the 'conflicting dynamics' issue identified by Xu et al. (recall Sect. 2.4).

Elsayed's MM with FAB: Utilizing FAB to estimate the margin in Elsayed's MM function results in a decrease in APGD-CE, with the best model achieving only 5.40% accuracy. Even when a soft boundary is added, the maximum APGD-CE accuracy achieved is 25.70%, which is lower than the robustness obtained with the exponential loss approach (which uses the first-order Taylor approximation). This indicates that FAB, while an alternative, might not be as effective in this context when compared to the Taylor approximation. This is especially striking if one considers that FAB is much more computationally expensive.

DyART: DyART shows different behavior compared to variations of Elsayed's method. While it has a APGD-CE of 33.52%, this does not surpass the APGD-CE score of the 'Elsayed's MM with Burn-in and Exponential Loss $\lambda = 1000$' method. However, we do observe a very respectable 33.24% accuracy when considering the stronger AutoAttack adversary, which is a substantial improvement on all of the other methods. We suspect that this stems from DyART's more accurate calculation of the margin gradients. That said, we note that DyART also has a more aggressive trade-off between robustness and clean accuracy.

5.1 Discussion

When considering the results of the previous section as a whole, we note that all configurations with Elsayed's MM function surpass the VGG-16 baseline in terms of robustness. This clearly demonstrates the effectiveness of these techniques in defending against adversarial attacks, but also demonstrates the trade-off with clean accuracy. Furthermore, these models unfortunately do not achieve any robustness improvements when evaluated using AutoAttack.

It is also evident that some of the additions made by Xu et al. are more effective at improving robustness than others. More specifically, we note that incorporating the exponential loss function achieved a 34.97% increase in robust accuracy evaluated using APGD-CE over the standard approach by Elsayed et al. Likewise, we find that incorporating a burn-in period of standard training is also very beneficial to improving robust accuracy. Conversely, and quite surprisingly, we find that a more accurate and more computationally expensive approximation of the margin with FAB did not improve the adversarial robustness of these models.

Finally, when comparing the variations of Elsayed et al.'s. method to DyART, we find that the addition of the margin gradients leads to a very large increase in AutoAttack robust accuracy, but simultaneously, a larger trade-off between the robustness and clean accuracy of the models.

6 Conclusions

While both Elsayed et al.'s and Xu et al.'s margin maximization techniques improve the adversarial robustness of deep neural networks in the input space, the specific elements contributing to this improvement was unknown. Our study aimed to address this by providing a detailed overview of the two techniques and isolating the individual elements of the techniques, evaluating their performance using current SOTA adversarial attack algorithms.

We have identified certain key elements that greatly contribute to adversarial robustness in DNNs, as well as certain design choices that do not have a large effect. More specifically, we find that:

1. A period of standard training before introducing margin maximization (known as a burn-in period) is a simple yet effective trick to increase adversarial robustness.
2. Penalizing samples with smaller margins more is very beneficial, such as through the use of an exponential loss function.
3. Using a more accurate estimation of the margin does not necessarily lead to improved robustness.
4. An accurate calculation of the gradient of the margin with regard to the model parameters greatly improves adversarial robustness.

In conclusion, our study provides insights into which elements positively contribute towards designing more robust image classification models. While the studied techniques initially seem complex, it has been shown that they can be decomposed into a number of fairly simple elements, each of which can be analysed individually. From the observed results, it is expected that future developments that target more accurate and efficient gradient estimations will be more beneficial than techniques aiming to produce more accurate estimations of the margin itself.

Acknowledgments. This work is based on research supported in part by the National Research Foundation of South Africa (Ref Numbers PSTD23042296065, RA211019646111).

References

1. Andriushchenko, M., Croce, F., Flammarion, N., Hein, M.: Square attack: a query-efficient black-box adversarial attack via random search. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12368, pp. 484–501. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58592-1_29
2. Bai, Y., Zhou, M., Patel, V.M., Sojoudi, S.: MixedNUTS: training-free accuracy-robustness balance via nonlinearly mixed classifiers. Trans. Mach. Learn. Res. (2024)
3. Bartoldson, B.R., Diffenderfer, J., Parasyris, K., Kailkhura, B.: Adversarial robustness limits via scaling-law and human-alignment studies. In: International Conference on Machine Learning (2024)

4. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: IEEE Symposium on Security and Privacy (2017)
5. Croce, F., et al.: Robustbench: a standardized adversarial robustness benchmark. In: Neural Information Processing Systems (2021)
6. Croce, F., Hein, M.: Minimally distorted adversarial examples with a fast adaptive boundary attack. In: International Conference on Machine Learning (2020)
7. Croce, F., Hein, M.: Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In: International Conference on Machine Learning (2020)
8. Ding, G.W., Sharma, Y., Lui, K.Y.C., Huang, R.: MMA training: direct input space margin maximization through adversarial training. In: International Conference on Learning Representations (2020)
9. Elsayed, G.F., Krishnan, D., Mobahi, H., Regan, K., Bengio, S.: Large margin deep networks for classification. In: Neural Information Processing Systems (2018)
10. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. CoRR arxiv:1412.6572 (2014)
11. Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B., Madry, A.: Adversarial examples are not bugs, they are features. In: Neural Information Processing Systems (2019)
12. Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D.P., Wilson, A.G.: Averaging weights leads to wider optima and better generalization. In: Conference on Uncertainty in Artificial Intelligence (2018)
13. Krizhevsky, A.: Learning multiple layers of features from tiny images (2012). https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
14. Loshchilov, I., Hutter, F.: SGDR: stochastic gradient descent with warm restarts. In: International Conference on Learning Representations (2017)
15. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018)
16. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
17. Rebuffi, S.A., Gowal, S., Calian, D.A., Stimberg, F., Wiles, O., Mann, T.: Data augmentation can improve robustness. In: Neural Information Processing Systems (2021)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR arxiv:1409.1556 (2014)
19. Szegedy, C., et al.: Intriguing properties of neural networks. CoRR arxiv:1312.6199 (2013)
20. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A.: Robustness may be at odds with accuracy. In: International Conference on Learning Representations (2019)
21. Wang, Q., et al.: Probabilistic margins for instance reweighting in adversarial training. In: Neural Information Processing Systems (2021)
22. Wang, Z., Pang, T., Du, C., Lin, M., Liu, W., Yan, S.: Better diffusion models further improve adversarial training. In: International Conference on Machine Learning (2023)
23. Wong, E., Rice, L., Kolter, Z.: Overfitting in adversarially robust deep learning. In: International Conference on Machine Learning (2020)
24. Wu, D., Xia, S.T., Wang, Y.: Adversarial weight perturbation helps robust generalization. In: Neural Information Processing Systems (2020)

25. Wu, Y., He, K.: Group normalization. In: European Conference on Computer Vision (2018)
26. Xu, Y., Sun, Y., Goldblum, M., Goldstein, T., Huang, F.: Exploring and exploiting decision boundary dynamics for adversarial robustness. In: International Conference on Learning Representations (2023)
27. Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., Jordan, M.: Theoretically principled trade-off between robustness and accuracy. In: International Conference on Machine Learning (2019)
28. Zhang, J., Zhu, J., Niu, G., Han, B., Sugiyama, M., Kankanhalli, M.: Geometry-aware instance-reweighted adversarial training. In: International Conference on Learning Representations (2021)



A Novel Approach To Lion Re-Identification Using Vision Transformers

Boitumelo Matlala, Dustin van der Haar, and Hima Vandapalli(✉)

University of Johannesburg, Johannesburg, South Africa
219120943@student.uj.ac.za, {dvanderhaar,himav}@uj.ac.za

Abstract. In recent years, technology has played a crucial role in wildlife and ecosystem conservation, significantly decreasing the time and effort required for wildlife monitoring, population estimation, poaching prevention, habitat mapping and, specifically, animal re-identification. This study proposes a novel approach that can be used in wildlife research and conservation to track individual animals over time by employing deep learning techniques. The challenges associated with animal re-identification in diverse natural environments such as variability in appearance, species diversity, accuracy and reliability can be addressed by leveraging the capabilities of advanced deep learning models, namely Vision Transformers and Convolutional Neural Networks. When trained on a Lion wildlife dataset, the Vision Transformer demonstrated significantly better performance compared to the Convolutional Neural Network in terms of accuracy, precision, recall and training time. This research contributes towards ongoing ecological initiatives to improve population monitoring, anti-poaching efforts, and habitat protection.

Keywords: Animal Re-identification · Vision Transformer · Tracking

1 Introduction

Animal re-identification is the process of recognizing individual animals and differentiating them from one another across various locations and time periods. It is important in facilitating the understanding of animal behaviour, population dynamics such as size and distribution, while also supporting conservation efforts [1].

In the past, researchers were reliant on using manual methods for re-identifying animals within a particular population [2]. These methods were typically invasive and involved physically tagging or implanting devices into animals for monitoring. Prevalent manual methods, including radio telemetry, GPS tracking and bio-logging, are often expensive to implement, labor intensive, time consuming while also being prone to error [2].

Recent visual based methods for animal re-identification focus on the use of images with the aim of matching new images with known individuals in a database. These algorithms operate by analyzing the distinctive aesthetic traits

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 270–281, 2025.
https://doi.org/10.1007/978-3-031-78255-8_16

such as pelage patterns, coloration, stripes, spots etc. in order to characterize animals. Cameras, drones, and satellite imagery have made the digitization of wildlife data increasingly prevalent and readily available [4]. The vast amount of visual data generated through digitization introduces the challenge of developing efficient and accurate methods to process the data.

The latest deep learning algorithms, such as Vision Transformers and Convolutional Neural Networks have provided us an opportunity to make the process of animal re-identification more efficient [5]. Vision transformers, which are state-of-the-art deep neural networks designed to analyze visual data have demonstrated improved capabilities in image recognition task even out performing conventional Convolutional Neural Networks in certain applications [7].

Our research aims to investigate the application of deep learning and particularly Vision Transformers to improve the process of lion re-identification by utilizing an annotated wildlife lion re-identification data. Benefits that will be derived from the proposed system include enabling more accurate tracking of individual lions as well as providing more insights into their behaviour. This research aims to pave the way for future studies in automated wildlife monitoring and can be adapted for other re-identification tasks in different domains.

2 Related Work

Radio telemetry techniques involve attaching or implanting radio transmitters to animals and using radio signals to monitor them. Very High Frequency (VHF) or Ultra High Frequency (UHF) spectral bands radio waves are required for the purpose of re-identifying remote animals using radio telemetry [11]. Alternatively, GPS tracking devices have also been used to track animals [11]. GPS tracking techniques have been particular successful in understanding the movement of animals and provide accurate information regarding the location of an animal, therefore accurate re-identification. Research utilizing GPS techniques contribute towards building a comprehensive dataset of fine-scale and high-resolution movement data which can later be utilized to advance further research efforts. Both techniques provide real-time data on the location of animals therefore allowing researchers to be able to learn about individual animals over time. However, radio telemetry has limited range and line-of-sight while GPS signals can be obstructed resulting in loss of data [11].

Research has also been conducted on the application of biometric identification methods for animal re-identification using unique traits such as an animal's DNA [8], or through acoustic monitoring based on distinctive vocalizations [17].

The above mentioned methods usually rely on invasive or specialized equipment to re-identify an animal which can be expensive to acquire. The operation of capturing an animal and attaching equipment or extracting biometric feature can be time consuming. These impediments have resulted in modern researchers using image based techniques to re-identify animals [6].

One early work uses Scale Invariant Feature Transforms (SIFT) to extract feature or key points from images such as spots, stripes etc. that are distinct and creates a descriptor for each feature [10]. These descriptors are used to match

the same features in different images by employing nearest-neighbor search. SIFT demonstrated effective re-identification of animals with distinctive patterns such as giraffes and zebras. However, since SIFT was initially intended for rigid objects, its performance can degrade when applied to non-rigid (i.e. changes in shape, appearance etc.) objects such as animals, which make it difficult to match features across different images of the same animal [14].

Another traditional computer vision method for animal re-identification that utilize the feature detection and matching framework is Oriented FAST and Rotated BRIEF (ORB) [15]. Features from Accelerated Segment Test (FAST) algorithm is firstly used to detect features with increased speed and efficiency. Then Binary Robust Independent Elementary Features (BRIEF) is used to compute binary descriptors for each feature.

Shuyuan et al. [16] proposed PPbM-b (Patch-based Part Matching - body). This is a modern computer vision method for animal re-identification that uses body features. It operates by separating the animal's body into multiple patches then after extracting features from each patch using Convolutional Neural Networks. The features which were extracted are then compared to the correctly identify individual animals. When body patterns, textures, or shapes of animals are unique enough the method provide reliable identification cues. PPbM-b was applied to a dataset of 3,649 tiger images across 92 classes achieving a Top-1 accuracy of 88.9%, Top-5 accuracy of 96.6%, and mAP of 71.0%. [16].

3 Methods

3.1 Dataset Description

A dataset was created by using images from the Drakenstein Lion Park, which were sourced from their various social media sites. The dataset was constructed using twelve distinct lions, with each individual captured in approximately five different poses and in at least three distinct settings, resulting in a total of 285 images collected. The dataset encompasses both male and female lions of varying ages. The diversity in poses and environments ensures that the re-identification models are able to recognize lions accurately regardless of their pose or surroundings. This variety helps the models generalize better, preventing overfitting and improving its performance with unseen data.

The created dataset overcomes challenges found encountered when utilizing available wildlife image datasets or data sources such as inconsistent image quality, equipment limitations, annotation difficulties, as well as data imbalance, where some subjects are over-represented. Additionally, access to high-quality images from existing databases are limited by privacy or licensing restrictions (Figs. 1 and 2).

3.2 Data Preparation, Augmentation and Hyperparameter Optimization

Pixel value extraction was initially performed on each image in the dataset. The images were then resized to 384×384 pixels with the aim of reducing computational complexity, and flattened into a one-dimensional array of pixel values.



Fig. 1. Edina Portrait



Fig. 2. Stuart Yawn

Four image augmentation techniques were performed on the dataset to include variability in the dataset and therefore make the models more robust. Initially, random horizontal flipping and rotation up to 20 °C was applied on the images in the dataset. These transformations ensure that the models are trained to recognize features regardless of how the subject is positioned in the image by introducing variability in image orientation. Following random horizontal flipping and rotation, color jittering was applied. This transformation adjusts the brightness, contrast, saturation, and hue to simulate different lighting conditions that may occur in natural environments. This assists in lessening the models' sensitivity to changes in lighting.

Slight variations in camera focus are common in images or videos taken in natural environments and have a significant effect on the prediction capabilities of vision models. A Gaussian blur was therefore applied on the images in the dataset as an attempt to replicate an out-of-focus image scenario often associated with low quality wildlife camera traps used to monitor animals.

The four transformations mentioned above are useful in reducing the risk of overfitting and allow the models to perform better in real world scenario.

Following the augmentation process, we carefully split the dataset into 70% for training, 15% for validation, and 15% for testing. This split ensures that the model has a robust amount of data for learning while retaining enough separate data to accurately evaluate its performance

Random search was used for hyperparameter optimization. It allows for searching the space of possible hyperparameters without exhaustively testing every possibility therefore minimizing the use of our limited compute resources (Fig. 3).

3.3 Convolutional Neural Network

A custom Convolutional Neural Network architecture for lion re-identification was implemented for this research. The network begins with an input layer that processes images of the prior specified size utilizing one channel. It consists of three convolutional layers: the first layer applies 32 filters of size 3×3, the second layer uses 64 filters of the same size, and the third layer utilizes 128 filters of the

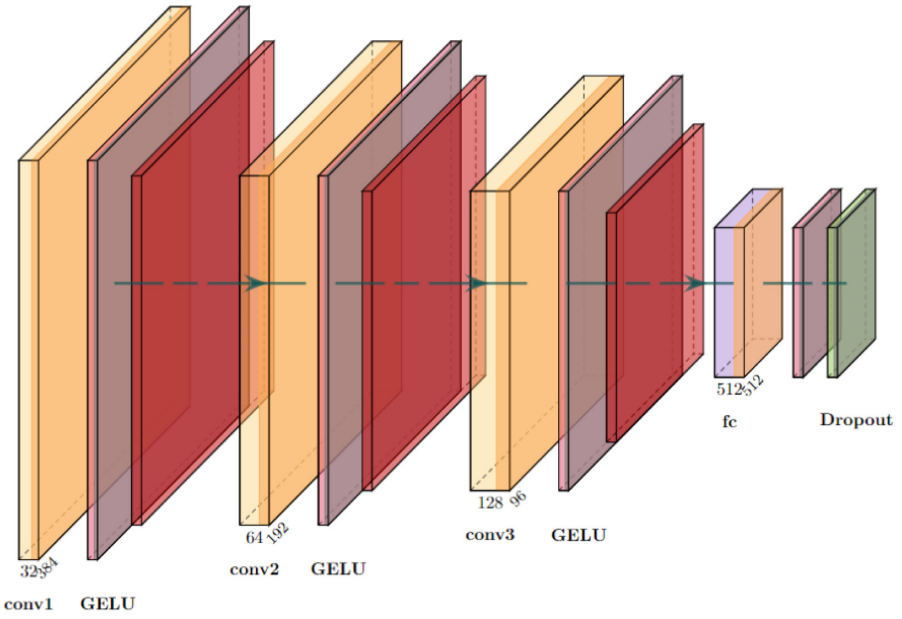


Fig. 3. Convolutional Neural Network Architecture

same size as well. Each convolutional layer is followed by Gaussian Error Linear Unit activation function and a 2×2 max pooling layer, which reduces the spatial dimensions by half. A Gaussian Error Linear Unit activation function is particularly useful for animal re-identification because it propagates more fine-grained information during the forward pass, allowing for more subtle differences in texture and patterns to be learned compared to other activation functions. After the convolutional layers, the output is flattened into a 1D tensor, which is fed into a fully connected layer. This fully connected layer also includes a Gaussian Error Linear Unit activation function and dropout for regularization. This architecture is designed to extract detailed hierarchical features from input images, reduce their dimensionality, and classify the extracted features effectively.

The following configurations were set for the convolutional neural network: A batch size of 32 was used, indicating that 32 samples will be processed at a time before the model's parameters are updated. An epoch of 150 was used, specifying that the entire training dataset will be passed through the model 150 times. A learning rate of 1e-4 was set, determining the step size for updating the model's weights, which were initially set to random values. A total of 12 classes was set, indicating that the model is being trained to classify input data into 12 different categories. The input size was set to 384, denoting that the input images are resized to 384×384 pixels. The in-channel was set to 1 indicating that a grayscale images were used. In real-world conservation, where grayscale images from infrared or low-quality cameras are common, using grayscale makes your model

more adaptable to real-world conditions. A dropout rate of 0.25 was set. This is regularization technique where each neuron is dropped with a probability of 0.25 during training to prevent over-fitting. The Adam optimizer was used with beta setting 0.9 and 0.9999 (Fig. 4).

3.4 Vision Transformer

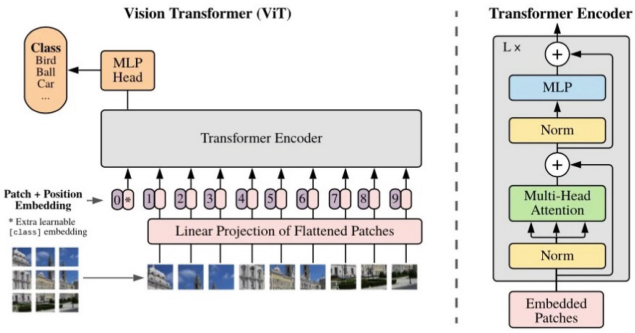


Fig. 4. Vision Transformer Architecture [13]

A Vision Transformer is a deep learning model that is built by modifying the transformer architecture originally designed for natural language processing to classify images. It improves on conventional deep learning models such as the Convolutonal Neural Network by splitting images into fixed patches and treating them as input tokens. These tokens are processed similar to words in a sentence by the transformer model, using self-attention mechanisms to capture the relationships and dependencies between different parts of the image. For this research, a pre-trained 16 patch Vision Transformer model from the timm library, designed for RGB images was used for lion re-identification. The model is modified to accept single-channel (grays-scale) images by replacing the patch embedding layer with a convolutional layer that projects 16×16 patches into the model's embedding dimension.

The Vision Transformer was pre-trained using the ImageNet-21k dataset, which comprises 14 million images across 21,000 classes. The model was trained with an AdamW optimizer, a batch size of 4096, a learning rate of 1e-4, a cosine learning rate schedule with a 5,000-step warmup, and weight decay set to 0.05. Data augmentation involved random horizontal flipping, and color jittering with brightness, contrast, saturation, and hue adjustments [3].

For fine-tuning, we set the number of hidden layers to 12, the number of attention heads to 12, the hidden activation function to GeLU, the hidden dropout probability to 0.25, and the attention probabilities dropout to 0.1.

The Vision Transformer improves on convolution-based models by adding a self-attention mechanism that allows global context understanding and capture long-range dependencies [9]:

Global contextual understanding is a model's capacity to take into account the whole image, or major parts of it, in order to differentiate between animals [13]. Vision Transformers are particularly strong in this respect because their self-attention mechanism allows for processing information from all the regions of an image at once, whereas Convolutional Neural Networks are generally more inclined toward local patterns [13]. This global approach is especially useful when trying to capture the overall shape and position of an animal, as well as the patterns on the fur. It also helps in distinguishing similar animals with respect to local features by taking into account the differences over the entire image, for instance, differences in the body length, pelage patterns, coloration, stripes or the way the animal interacts with the surrounding. Also, Vision Transformers are less sensitive to occlusions and are able to take into account the background context, which allows to increase the re-identification accuracy by evaluating the entire scene instead of the animal alone [13].

Long-range dependencies refers to the capacity of a model to capture interactions and relations between elements and features that are far from each other in the input sequence or image [13]. It is important to be able to capture long-range dependencies because the features that make one lion different from another may be distributed across the body or may involve interactions with the environment. As for Vision Transformers, their self-attention mechanism allows them to capture these long-range dependencies, as they analyze the whole image at a time. As opposed to Convolutional Neural Networks, Vision Transformers are not confined by the size of the convolutional kernels and can directly learn the relationships between distant parts of the image [13]. In this way, Vision Transformers are better at grasping these relationships and distinguishing between individual lions even if the key features are located far apart or some of them are hidden (Table 1).

4 Ablation

4.1 Convolutional Neural Network

We investigated the effect of replacing the GELU activation function with the ReLU activation function for all the layers involved. This change reduced the accuracy drastically, thus supporting the hypothesis that GELU excels at capturing fine details of texture and patterns which are critical for distinguishing between animals of the same species. An increase in the number of epochs from 150 to 250 was necessary using ReLU activation to achieve at least a 70% precision and accuracy.

Table 1. Convolutional Neural Network (CNN) - Factors with Percentage Impact

Component	Modification	Performance Impact
Convolutional Layer	Remove 3rd convolutional layer (128 filters)	–15% across all metrics
Convolutional Layer	Halve the number of filters (16, 32, 64)	–5% across all metrics
Pooling Layer	Replace max pooling with average pooling	–8% performance

Lastly, we evaluated the effect of the fully connected layer. When dropout was removed from the fully connected layer, overfitting became apparent as the validation performance dropped, thus indicating that dropout is an effective way of preventing overfitting of the model to the training data (Table 2).

4.2 Vision Transformer

Table 2. Vision Transformer - Factors with Percentage Impact

Component	Modification	Performance Impact
Transformer Layers	Reduce transformer layers from 12 to 8	–22% across all metrics
Positional Encoding	Remove positional encoding	–17% across all metrics

First, we tested the results of changing the size of the patch from the standard 16×16 to 32×32, and found that a increase in the size of the patch resulted in the model capturing less details, but at the same time, the computations time taken was shorter, while a decrease of the patch increased the amount of time for computations, but the model attained all important information. A 16×16 patch required at least 30 epochs to train which is nearly three times as many epochs to train before stoppage was triggered due to convergence.

We then changed the number of attention heads from 12 to 6 and found that reducing the number of heads made it impossible for the model to attend to different parts of the image at the same time and thus decrease its global context understanding capabilities whereas increasing the number of heads made the model to be very computational intensive but with little improvements in performance.

5 Results

From Table 3 below we can see that the Vision Transformer significantly outperforms the convolutional neural network. An F1 score of 0.9744 indicates that

the model is highly accurate in identifying positive cases but is also efficient in minimizing false positives and negatives. A precision score of 0.9787 and recall score of 0.9747 indicates that the model is almost entirely correct in predicting all positive cases correctly.

The Convolutional Neural Network produced an F1 score of 0.8558 which is less accurate than the Vision Transformer. Although we can see that the Vision Transformer does outperform the state-of-the art Convolutional Neural Networks mentioned earlier. A precision score of 0.8820 indicates that the Convolutional Neural Network is capable of predicting a majority of positive cases correctly but this is still significantly less compared to the Vision Transformer.

Table 3. Comparison of Vision Transformer and CNN

Metric	Vision Transformer	Convolutional Neural Network
F1 Score	0.9744	0.8558
Precision	0.9787	0.8820
Recall	0.9747	0.8552
ROC AUC	1.0000	0.9973
Log Loss	0.0762	0.6075
Mean Absolute Error	0.9327	3.3872

The Receiver Operating Characteristic curve (Fig. 5) provides a complete picture of a model's ability to distinguish among distinct identities of lions. For animal re-identification, the closer an Receiver Operating Characteristic curve is to the top-left corner, the better it performs in terms of correctly recognizing individual lions and keeping false identifications low. The Area Under Curve summarizes overall effectiveness with a higher Area Under Curve indicating a better performance.

Another graph of interest presented below (Fig. 6) is the Precision-Recall curve. This plot represents the relationship between precision and recall. It is a way of understanding the precision-recall trade-off in imbalanced datasets common in animal re-identification tasks. This occurs because there are different numbers of sightings for different animals such as in the dataset used, where some lions are sighted more than others. High values for precision and recall close to the top right corner means that our model is able to tell apart individual lions even if they are poorly represented within the data. Together, these graphs offer a detailed analysis of a model's performance in the task of animal re-identification (Figs. 5 and 6).

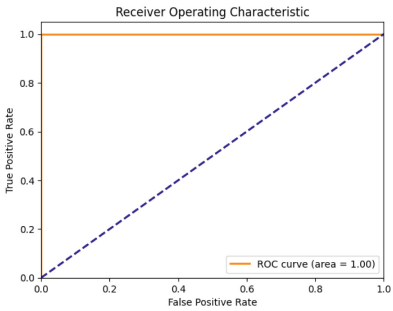


Fig. 5. Vision Transformer ROC

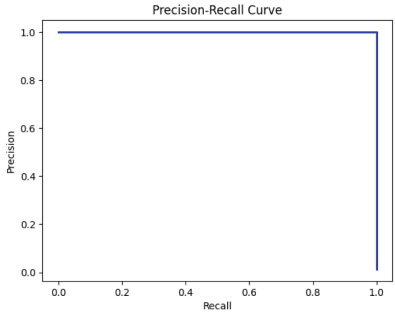


Fig. 6. Vision Transformer Precision-Recall Curve

6 Discussion

It is clear from the results of this study that the Vision Transformer outperforms Convolutional Neural Network (CNN) in the lion re-identification task. These results are consistent with the growing literature attesting to the efficacy of Transformer architectures on vision tasks when global context and long range dependencies are essential, as in the example of identifying individual lions across pose, lighting conditions, and environment.

The key advantage of using the Vision Transformer lies in its self attention mechanism, which is an important property and key to identifying animals with subtle features that spread across the animal body. On the other hand, CNNs solve this problem by concentrating more on local patterns, which limit their effectiveness.

Vision Transformer are more expensive computationally compared to CNNs. In real world applications, particularly in resource constrained wildlife monitoring environment, this could be a challenge. Additionally, while our dataset contained a sufficient number of images from different lions, the limited sample size may restrict the model's generalizability to other species or more diverse environments. Future work should aim to test the Vision Transformer's effectiveness on larger datasets from existing wildlife image datasets to confirm its generalizability. Furthermore, a standardized approach to data collection, ensuring diversity in poses, settings, and lighting conditions should be taken. Incorporation of multi-perspective imaging is essential in wildlife monitoring to accurately reflect real-world variability in animal sightings.

The distinction between this work and real-time re-identification circumstances such as live drone imagery are mainly in the computational requirements and environmental changes. The performance of re-identification in real time differs with pre-processed images where real-time images from drone feeds are generally of less quality containing motion blurs, inconsistent illumination or occlusion. Furthermore, Vision Transformers may also experience high latency which is a disadvantage for real-time applications.

For replication it is recommended to implement the architecture of the CNN and Vision Transformer as explained in the Method section and train using an A100 GPU available on Google Colab. The dataset used is available upon request on and should be prepared as explained in the Method section as well.

7 Conclusion

In this study, we investigated the effectiveness of a Vision Transformer and a Convolutional Neural Network in lion re-identification tasks by comparing their performances using a variety of performance metrics. The Vision Transformer is an enhancement of Convolutional Neural Networks as it handles global context and learns long-range dependencies within images through its self-attention mechanism. It may also be faster at inference time, particularly with large datasets, as the images are processed in parallel and do not require as many convolution operations as Convolutional Neural Networks, making it well-suited for handling large datasets. All evaluated aspects showed that Vision Transformer outperforms the Convolutional Neural Network. These results are beneficial to the development of deep learning, especially in the specified application of animal re-identification. Due to the improvement in the methods of identification of these species, this research contributes to ecological work in the future, such as the monitoring of population, anti-poaching measures, and the protection of habitats. In conclusion, such developments are vital for the purpose of wildlife conservation and management. The experiments were carried out using a Lion Wildlife dataset, hence, the results may not be generalized to all classification tasks. Further studies should consider applying Vision Transformers on diverse datasets, especially multi-species datasets.

References

1. Moskvyak, O., Maire, F., Dayoub, F., Baktashmotlagh, M.: learning landmark guided embeddings for animal re-identification. arXiv:2001.02801 (2020). https://arxiv.org/abs/2001.02801
2. Burghardt, T., Campbell, N.: Past, present, and future approaches using computer vision for animal re-identification from camera trap data. arXiv:1811.07749 (2018). https://arxiv.org/abs/1811.07749
3. Steiner, A.P., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., Beyer, L.: How to train your ViT? Data, augmentation, and regularization in vision transformers. Transactions on Machine Learning Research (2022). https://openreview.net/forum?id=4nPswr1KcP
4. Wang, D., Shao, Q., Yue, H.: Surveying wild animals from satellites, manned aircraft, and unmanned aerial systems (UASs): a review. Remote Sens. **11**(1308) (2019). https://doi.org/10.3390/rs11111308
5. Nguyen, H., et al.: Animal recognition and identification with deep convolutional neural networks for automated wildlife monitoring. In: 2017 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 40–49 (2017). https://doi.org/10.1109/DSAA.2017.31

6. Lennox, R.J., Blouin-Demers, G., Rous, A.M., Cooke, S.J.: Tracking invasive animals with electronic tags to assess risks and develop management strategies. In: Proceedings of Springer International Publishing Switzerland (2016)
7. Stassin, S., Corduant, V., Mahmoudi, S.A., Siebert, X.: Explainability and evaluation of vision transformers: an in-depth experimental study. Electronics**13**(1), 175 (2024). https://www.mdpi.com/2079-9292/13/1/175
8. Bugge, C.E., et al.: Biometric methods of animal identification. This paper is a compilation of groupwork provided by participants on a course in Laboratory Animal Science at the Norwegian School of Veterinary Science, February 2011. Compiled by Adrian Smith, Norecopa (www.norecopa.no)
9. Tuia, D., et al.: Perspectives in machine learning for wildlife conservation. Nat. Commun. **13**(1), 792 (2022). https://doi.org/10.1038/s41467-022-27980-y
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vision **60**(2), 91–110 (2004). https://doi.org/10.1023/B:VISI.0000029664.99615.94
11. Gutema, T.M., Nassar, M.: Wildlife radio telemetry: use, effect and ethical consideration with emphasis on birds and mammals. Mohammad Nassar for Researches (MNFR) (2015). oai:ojs.gssrr.org:article/4119
12. Bugge, C.E., et al.: Biometric methods of animal identification. This paper is a compilation of groupwork provided by the following participants on a course in Laboratory Animal Science at the Norwegian School of Veterinary Science, February 2011. https://www.google.com
13. Dosovitskiy, A., et al.: An image is worth 16×16 words: transformers for image recognition at scale. arXiv:2010.11929 (2021). https://arxiv.org/abs/2010.11929
14. Crall, J.P., Stewart, C.V., Berger-Wolf, T.Y., Rubenstein, D.I., Sundaresan, S.R.: HotSpotter - patterned species instance recognition. In: 2013 IEEE Workshop on Applications of Computer Vision (WACV), pp. 230–237 (2013). https://doi.org/10.1109/WACV.2013.6475023
15. Kaur, A., Kumar, M., Jindal, K.: Cattle identification system: a comparative analysis of SIFT, SURF, and ORB feature descriptors. Multimedia Tools Appl. **82**, 1–23 (2023). https://doi.org/10.1007/s11042-023-14478-y
16. Shuyuan, L., Jianguo, L., Weiyao, L., Hanlin, T.: Amur tiger re-identification in the wild. arXiv preprint arXiv:1906.05586 (2019)
17. Yeo, C., Al-Haddad, S.A.R., Ng, C.K.: Animal voice recognition for identification (ID) detection system. In: Proceedings of the 2011 IEEE 7th International Colloquium on Signal Processing and Its Applications (CSPA 2011), March 2011. https://doi.org/10.1109/CSPA.2011.5759872

Assessing Data-Driven of Discriminative Deep Learning Models in Classification Task Using Synthetic Pandemic Dataset

Sunday Adeola Ajagbe[1,2(✉)], Pragasen Mudali[1], and Matthew O. Adigun[1]

[1] Department of Computer Science, University of Zululand, Kwadlangezwa 3886, South Africa
saajagbe@pgschool.lautech.edu.ng, {MudaliP, adigunm}@unizulu.ac.za

[2] Department of Computer Engineering, Abiola Ajimobi Technical University, Ibadan 200255, Nigeria

Abstract. Deep learning models' exploration of synthetic data has been comparatively understudied compared to other areas of research**.** This paper proposes an assessment of discriminative deep learning (DL) models for classification task based on synthetic pandemic dataset to identify a robust and efficient model for a relevant application. In a specific term, it provided an empirical assessment performance of three different discriminative DL models: CNN (Convolutional Neural Network), RNN (Recurrent Neural Network), and MLP (Multilayer Perceptron) for investigation. The study architecture depicts a complete classification and detection procedure of the presence COVID-19 pandemic synthetic dataset. It consists of four phases: (i) dataset collection and description phase, (ii) preprocessing phase; (iii) discriminative DL modelling phase; (iv) classification and evaluation phase. The research outcome shows that RNN achieved the highest classification report, it is observed that RNN achieved the highest classification report values of 0.9881 for recall, precision, F1 Score and accuracy metrics and a little higher ROC-AUC where it was measured 0.9995. The RNN model achieves perfect scores on all metrics, indicating an outstanding performance. This could be due to a variety of factors, such as the nature of the dataset.

Keywords: Data-Driven AI · Discriminative deep learning · Classification task · Synthetic dataset · Pandemic detection

1 Introduction

The COVID-19 pandemic was deemed a pandemic by the World Health Organization [1], which also suggested several tools and techniques for quickly identifying patients that pose the greatest risk of disease and death. Due to the uncontrolled proliferation of several SARS-CoV2 variants, the coronavirus pandemic has presented a challenge to international health systems, resulting in a sharp increase in the testing and hospitalization requirements for the diagnosis and treatment of infected patients. Even though COVID-19 affects people in different ways, over 80% of instances with mild-to-moderate

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 282–299, 2025.
https://doi.org/10.1007/978-3-031-78255-8_17

sickness get better without going to the hospital. Fever, dry cough, and fatigue are the most prevalent indications and symptoms of this illness, and they all start off mildly in infected individuals. Furthermore, some patients may exhibit additional symptoms as exhaustion and chest pain [2]. Accurate COVID-19 diagnosis has been achieved through the application of artificial intelligence (AI) machine learning (ML) and deep learning (DL) algorithms, which can be regarded as a helpful technology for enhancing standard diagnostic techniques like computed tomography (CT), reverse transcription polymerase chain reaction (RT-PCR), and chest X-ray. It is desirable to develop technologies such as drone-based non-human thermal screening as a crucial method of quickly identifying infection [3, 4].

One type of ML model that focuses on differentiating between several classes or categories in the data is called discriminative DL models [5]. Discriminative models learn the decision border between classes directly, in contrast to generative models that make an effort to describe the underlying distribution of data [6]. The following are a few popular and significant discriminative DL models: CNNs, MLP, RNNs which has three variances; Long Short-Term Memory Networks, Bi-Long Short-Term Memory Networks and Gated Recurrent Units [7]. Discriminative DL models have the advantages over DL models due to there being simpler probabilistic models (simplify learning by not requiring modelling the complete data distribution), efficiency in training (often use less data and resources than generative models in order to perform well on classification tasks) and direct focus on the classification (specifically tailored to the objective of class distinction, frequently resulting in improved performance in classification tasks). However, usage of discriminative learning for COVID-19 classification task is under investigated. Figure 1 depict DL taxonomy in accordance with [7] study.

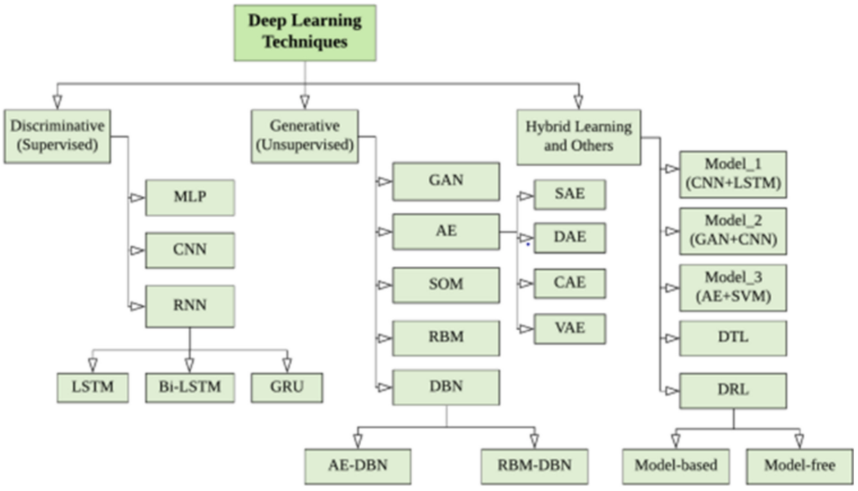


Fig. 1. The DL Taxonomy [7]

Motivation for the Study

There are a number of theoretical, practical, and developmental needs in the fields of AI, ML and DL that drive the comparison of DL models and discriminative learning. Essentially, the DL like many other model selections is governed basically by the performance optimisation, task-specific requirements, benchmarking and standards, resource allocation, theoretical insights, innovation and experimentation and practical development. In order to select the most appropriate, efficient, and effective model for a particular task and to further the general growth of science and technology in the field of AI, it is imperative to compare the discriminative DL models to know the strengths and weaknesses of these models for efficient use in future tasks. Despites the contributions of discriminative DL models in classification tasks, the empirical study of these models for classification task based on the synthetic dataset remain an open issue, hence, this study is informed.

Objectives of the Study

i. The study empirically compares discriminative DL model, namely, CNN, RNN and MLP in the classification task using synthetic pandemic dataset.
ii. The performance evaluation of the proposed techniques was carried out using recall, precision, F1-score, accuracy and ROC-AUC as the state-of-the-art metrics for comparison technique.
iii. The current study is also validated with existing studies that used synthetic and non-synthetic dataset.

Study Organisation

This is how the remaining research is arranged. A review of related studies is presented in Sect. 2. The technique, dataset source and collection, preprocessing, modeling, and model evaluation are all covered in Sect. 3. Results and discussion in Sect. 4. The conclusion and next steps for implementing an effective DL-based strategy to combat the COVID-19 pandemic are covered in Sect. 5.

2 Review of the Related Studies

The current section presents the review of discriminative DL-based related studies with respect to pandemic classification tasks. we also reviewed studies that focuses on the usage of synthetic dataset that employed AI techniques for classification purposes. These seek to expand on the scope of earlier research and open up new avenues for investigation.

Hasan, et al. [8] developed artificial intelligence (AI)-based algorithms that were incorporated into wearable Internet of Things (IoT) devices. These devices were very successful in detecting COVID-19 and forecasting insights that could be put into practice through widespread use. The findings demonstrate that AI is a promising field of study that has applications in drug discovery, disease prediction, disease forecasting, and the growth of the healthcare industry globally.

Krohling and Krohlin, [9] in their research, attempted how well the SARS-COV2 virus could be classified using spectral data obtained from spectroscopy, the study presented a 1D convolutional neural network (1D-CNN). A number of experiments were conducted, including a comparison between 1D-CNN and the DL model and standard algorithms used in chemometrics for the most well-known datasets. The results show that 1D-CNN performed better than the DL model and standard model, achieving an average accuracy of 96.5%, specificity of 98%, and sensitivity of 94%. Only accuracy was used to evaluate the model performance, whereas, other state-of-the-art metrics that access the robustness of the models needed to be used for the evaluation.

Mohammedqasem, et al., [10] created a novel DL based model for medical datasets with high missing values. In order to improve the deep predictive training model for patients with COVID-19, the Data Missing Care Framework was utilized to address the issue of high missing data in medical datasets. Additionally, Grid Search optimization was employed to tune assessments and set multiple hyperparameters on three AI models: Artificial Neural Network (ANN), CNN, and RNN. For every deep-learning classifier, the evaluation metrics were nearly perfect, yielding the best evaluation for the COVID-19 dataset in terms of accuracy 98%, precession 98.5%, F1-score 98.6%, and ROC Curve 95%. Although, the study used accuracy, precision, F1 score, and ROC but MLP which share the same architecture with CNN and RNN was not among the model engaged in the comparative study.

Salama, et al. [11] recommended using hybrid models for COVID-19 detection that are based on machine learning and deep learning models in order to detect the virus early on and lower the chance of patient death. The dataset, which consists of 2481 CT pictures split into COVID-19 and non-COVID-19 categories, is the outcome of using ten distinct deep CNN network models to extract characteristics from CT scans.

Bhosale and Patnaik [12] presented a pandemic investigation utilizing both public and private data and certain deep learning techniques like CNN and RNN. A total of 64 bespoke, pre-trained DL models were chosen based on taxonomies related to imaging modalities. Sample selection, network design, training using a minimally annotated database, and security concerns are the factors that limit the use of DL-based approaches. Examining the pathophysiology, immunological responses, causative agents, and epidemiological illnesses are all included in this.

Effati and Nejat, [13] gave the initial performance analysis that contrasts several CNN architectures for the independent preliminary. The classification performance of COVID-19 identification ResNet (CIdeR), densely connected networks (DenseNet), squeeze neural networks (SqueezeNet), visual geometry groups (VGGs), residual networks (ResNets), and Alex neural networks (AlexNet) architectures is investigated through comparison and analysis.

Gürsoy and Kaya [14] proposed that automated computer-aided methods are required for COVID-19 scan and medical data, which calls for the use of clinical markers, MRIs, CT scans, X-rays, cough sounds, and RT-PCR or analysis of medical data DL and ML in order to detect COVID-19.

Manettas, et al. [15] opined that large amounts of annotated datasets are needed for computer vision or classification task training when using deep learning algorithms like

CNN. Nevertheless, in highly complex and dynamic production environments in particular, this kind of approach is costly, labor-intensive, error-prone, and time-consuming. When appropriate training datasets are created, synthetic datasets can be used to expedite the DL training phase. This study offers a method for creating data sets by using a series of simulation tools in succession. The framework creates artificial images of components that are created. To support a manufacturing activity, a computer vision system must identify the states of the parts, such as their rotation in various rotation axes. The artificially created images are used to retrain a number of previously trained CNNs. CNNs were tested using real images of components that have been fabricated. Various CNN models' performances were shown, contrasted, and commented. The rotation accuracy per model for up-oriented images shown 96.5%, 85.5% and 73.8% for VGG16, Mobilenet and DenseNet121 respectively. The rotation accuracy per model for down-oriented images shown 92.0% 71.7% and 65.9% Mobilenet and DenseNet121 respectively. The rotation accuracy per model for mixed orientation shown 95.6% 89.8% and 74.5% Mobilenet and DenseNet121 respectively. The findings suggested that CNNs trained on artificially created datasets might function satisfactorily when applied to support manufacturing operations. Only accuracy was used to evaluate the model performance, whereas, other state-of-the-art metrics that access the robustness of the models needed to be used for the evaluation.

Rizaev and Achim, [16] unveiled a 46,080 pictures synthetic SAR imaging dataset featuring ship wakes for ten distinct genuine vessel models. There are sixteen ship heading directions, six ship velocities, eight wind directions, two wind velocities, and three incidence angles among the many simulation parameters. The classification performance for ship wake sceneries that were noisy, noisy-free, and denoised was thoroughly examined in this work. It employed training from scratch and made use of the conventional AlexNet architecture. Hyperparameters were determined using Bayesian optimization to attain optimal classification performance. The findings showed that vessel classes can be classified very effectively using their SAR signatures, with maximum accuracies of 96.12%, 92.7%, and 93.6% when training with noisy, denoised, and noise-free datasets, respectively. The study came to the conclusion that using CNN to train on denoised SAR datasets would be the most effective approach in real-world applications. The findings indicate that the SAR simulator's adaptability can lead to new opportunities for applying machine learning to a range of SAR platforms. Only accuracy was used to evaluate the model performance, whereas other state-of-the-art metrics that access the robustness of the models needed to be used for the evaluation.

AlMohimeed, et al., [17] developed a stacking ensemble model utilising CXR pictures, two datasets linked to COVID-19, and specific symptoms associated with the disease. The stacking ensemble framework is comprised of a number of models that have already been put through their training. The MLP, RNN, LSTM, and GRU are some examples of these types of neural networks. In ensemble stacking, a meta-learner, more specifically a support vector machine (SVM), is utilized in order to make an educated guess as to what the ultimate outcomes will be. The evaluation makes use of two datasets, namely COVID-19-symptoms-2 and COVID-19-symptoms-1, both of which are accessible through the website GitHub. Not only fever, coughing, shortness of breath, and a chest infection are included in the datasets that are being examined, but there are

also other symptoms. Additionally, they include activities such as traveling to a foreign country, wearing a mask, and coming into contact with an infected individual. GRU, LSTM, MLP, and RNN are the techniques that are utilized in order to evaluate the first architecture that has been proposed using these two datasets. The second stacking ensemble learning model was developed by combining the DL algorithms DenseNet121, InceptionV3, VGG16, and ResNey50. This model was utilised to determine the outcome. A face-to-face comparison between two models is conducted using two datasets, each of which has several classes. 99.30% accuracy was attained by the first model, and 99.62% accuracy by the second.

Mirza and Samak [18] reported the context of medical image classification tasks, the combination of a hybrid CNN-LSTM structure and neuromorphic fuzzy logic offers an efficient method for dealing with uncertainty. This strategy results in an increase in the precision of the COVID-19 predictions. All of the CNN models, including Deep Feature + SVM, COVID CAPS, and Bayesian CNN, are not as good as the CNN-LSTM model with Neutrosophic Fuzzy logic. The level of accuracy as of right now is 98.6%, which is an increase of 4.4%.

Rayan, et al., [19] considered the identification of COVID-19 was accomplished by this framework through the utilization of a wide variety of airborne sound data, including sounds of breathing and coughing, amongst others. For the purpose of developing a classification model that can be utilized for testing COVID-19, DL techniques are utilized. For the MIT-Covid-19 dataset, the Weiner filter application was responsible for performing the preprocessing necessary. For the purpose of classification, the CNN-LSTM method was utilized, and the Mel-frequency cepstral coefficient feature extraction technique is utilized. In order to evaluate the effectiveness of the framework that we developed for this study, the authors utilized CNN, GRU, and LSTM. CNN-LSTM was found to be 97.7 times more effective than other methods that were being utilized, as demonstrated by the results.

Akl, et al., [20] used the CT dataset to evaluate a suggested framework, and five different metrics were used to evaluate the results but the study considered some DL and ML architectures which make it different from this current study that focus mainly on discriminative DL models. The investigation shown that the proposed CNN model outperforms the widely-known DL model in terms of feature extraction. When ML models were used for classification and DL models were used for feature extraction in COVID-19 CT scan image recognition, there was an improvement over a fully-fledged DL model of the same type. The success rate of the first method, which was 99.39%, became truly remarkable when ensemble learning models were used instead of conventional ML models.

In a recent study by Kuzinkovas and Clement, [21] it was possible to train a model with the help of the 33,920 chest X-ray images that were included in the COVID-QU-Ex dataset. Images of COVID-19, pneumonia caused by viruses other than COVID, and normal images were all included in the dataset in the same quantity regardless of their respective types. The model is constructed using GLCM textural features in conjunction with pre-trained CNNs, with a particular emphasis on ResNet50, VGG19, and VGG16. It was determined that the model was able to differentiate between cases of COVID-19 and those that did not contain the virus 98.34% of the time when it was tested on 6581

chest X-rays. The model achieved a score of 94.68% when it was put through a test to differentiate between normal chest X-rays, pneumonia caused by viruses other than COVID, and COVID-19.

Wong et al., [22] worked on the development of a computer vision system that was capable of identifying and differentiating between a variety of grocery items that are stored in a warehouse. It outlines the steps that must be taken in order to construct a complete DL pipeline using physical objects, beginning with the collection of data and ending with the training of models. The creation of three-dimensional models of real-world objects can be accomplished through the use of photogrammetry, which allows for the creation of an artificial dataset. One hundred thousand fabricated images were used to train a CNN known as Inception V3, which was trained using ten different categories. A success rate of 96% was achieved when it was tested with actual images of food items that were provided by a third party.

In the PRISMA framework was utilized in order to investigate the feasibility of employing ML for the purpose of COVID-19 prediction. As of January 24, 2023, a comprehensive search of five databases was carried out in order to locate studies that were pertinent to the current topic. A total of 1,824 unique papers were discovered throughout the course of the search. Within the scope of this meta-analysis, a total of 204 studies were taken into consideration, all of which made an effort to forecast the outcome of COVID-19 by utilizing machine learning models. The primary factors that were investigated throughout this study were automated diagnosis (18.1%), severity classification (31.9%), and outcome prediction (50%) respectively. The study were able to identify 34 unique characteristics and then place them into five distinct categories. It was found that there were twenty-one different kinds of machine learning models, which were found in six different categories. X-rays of the chest, CT scans, and imaging of older adults were the most frequently used imaging modalities. XGB, CNN, and RF were the three models that were featured the most frequently in the game. All of the neural networks (ANN, MLP, and DNN), ensemble approaches (XGB), regression models (PLS-DA), and distance-based methods (kNN) had high AUC values, which indicated that they performed exceptionally well [23].

In essence, previous studies typically utilized one or two metrics to evaluate the robustness of the models; however, this study added more metrics, including recall, precision, and F1 score. Furthermore, earlier research employed CNN variations and other generative learning models without taking into account how well other discriminative DL models performed for real-world decisions and efficiency, particularly in a synthetic dataset where computational load might be lower.

3 Methodology

The decision-making process benefits greatly from the use of DL models, even in cases when the volume of data is very vast [6]. In order to apply data processing techniques for certain sectors, it is necessary to define the different categories of data, including volume, variety, and velocity. The modelling of normal data analysis includes the ANN model, the classification and clustering process model, and the use of effective algorithms [24, 25]. Specific forms of data can be generated from various sources, and it

is crucial to develop systems that can handle the features of data. Figure 2 show the developed architecture discriminative DL models classification task investigation based on synthetic pandemic dataset. The developed architecture depicts a complete classification and detection procedure of the presence of COVID-19 using synthetic dataset. It consists of four phases: (i) dataset collection and description phase, (ii) preprocessing phase; ((iii) discriminant DL modelling phase; (iv) classification and evaluation phase.

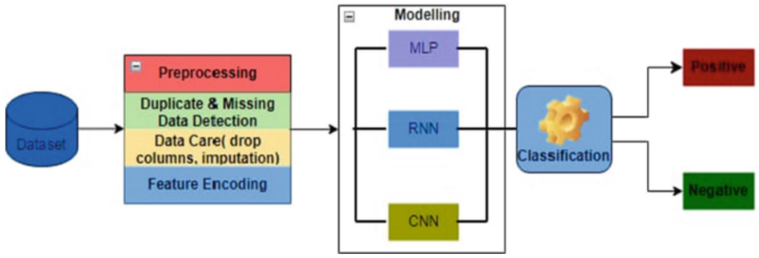


Fig. 2. Developed architecture discriminant DL models classification task investigation

3.1 Data Source and Description

The COVID-19 synthetic dataset that was used in this investigation was taken from a repository, specifically, the Kaggle database. The dataset was accessed from this link (https://www.kaggle.com/datasets/rishanmascarenhas/covid19-temperatureoxygenpulse-rate). Sample datasets distribution is shown in Fig. 3 where the values from 0–9 were displayed. The details contain the ID, Oxygen, Pulse rate, Temperature and the results (±). In all, the dataset has 5010 negative and 4990 positive cases in 248.5 kB sized dataset. Figure 4 shows the distribution of the datasets by classes. There is no ethical concern over this dataset as it was sourced from opensource and it has been deanonymized to ensure privacy and security.

3.2 Dataset Preprocessing

Posing the research as a categorization challenge: Let S $= \{(x_1,y_1),(x_2,y_2),\cdots(x_n,y_n)\}$ be the set of training instances of dimension d. Y $= \{y_1,y_2\cdots,y_n\}$ be the set of labels where x_i is a feature with corresponding y_i label is a set of features according to classification task definition. The classification model's initial step is feature extraction. This is important when the input data, or the derived characteristics, are large and difficult to handle in their raw form. Duplication and missing values were handle by imputation technique, where the average of such column were fund and imputed in such missing value. This issue will be resolved by the identification of key aspects, which was carried out in accordance with [25–27].

	ID	Oxygen	PulseRate	Temperature	Result
0	0	98	65	95	Negative
1	1	96	92	95	Negative
2	2	95	92	99	Negative
3	3	97	56	96	Negative
4	4	88	94	98	Positive
5	5	94	100	103	Positive
6	6	88	81	104	Positive
7	7	91	79	95	Negative
8	8	93	59	101	Positive
9	9	86	117	99	Positive

Fig. 3. Sample datasets distribution

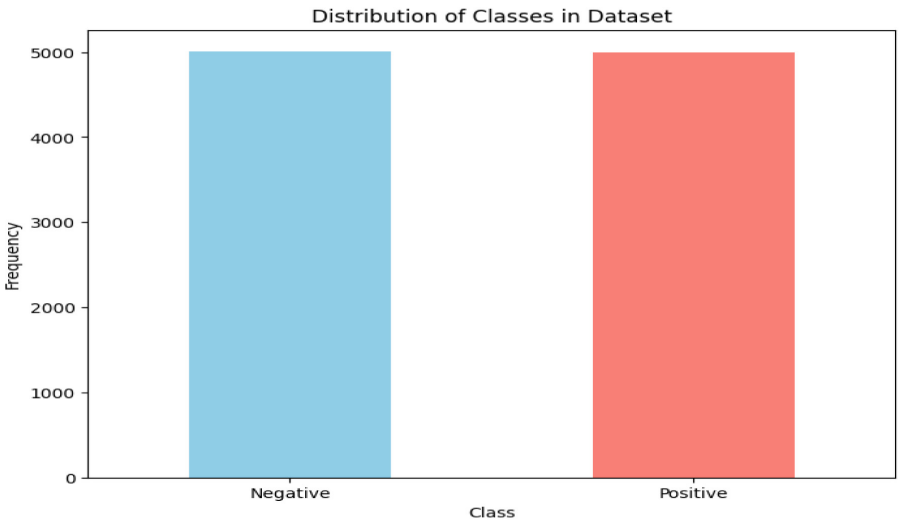


Fig. 4. Distribution of the datasets by classes

3.3 Modelling

This section describes the modelling, and the chrematistics of discriminative DL models investigated in this study, their parameters, and metrics used for the classification experiment. It is important to mention that based on the design on DL models, the neural network selects the best and suitable for in most classification tasks and this is the case in this study. The approach employed in this research is a 90:10 dataset splitting

ration, meaning that 90% of the information was utilized for training while 10% were utilized for testing. The task was performed with three classifiers namely: CNN, RNN and MLP as the main discriminative DL models under study. Using a Jupyter notebook and the sklearn [28] libraries on the Anaconda platform, the experiment was carried out. An Intel® CoreTM i5–7200 CPU running at 2.50 GHz to 2.70 GHz on a Pentium Windows PC with 8 GB RAM was used for all trials. It was running the experiments in Google colab RAM:12.67 GB, Disk:107.72 GB and running Python 3. As mentioned in Sect. 3.1, the dataset was cleaned, and additional preparation was carried out. The train-test split ratio of the dataset was 90:10.

3.4 Performance Metrix

Table 1 shows the confusion that denotes the parameters used in the performance matrix evaluation. The True Positive Rate, also known as Recall, is calculated by dividing the total number of positive occurrences (P) by the number of optimistic TP that are correct. 1.0 is the true positive rate (TP Rate) that is considered to be best, while 0.0 is the TP Rate that is considered to be the lowest feasible, as shown in Fig. 5 [29].

Table 1. Confusion matrix table

TP	*FP*
TN	*FN*

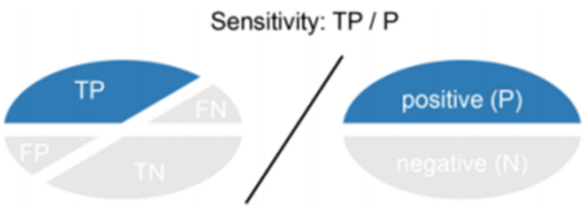


Fig. 5. Ellipses for recall calculation [29].

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{P} \tag{1}$$

One way to evaluate the effectiveness of a machine learning or deep learning model, or how well the model makes positive predictions, is through precision. The precision is calculated by taking the total number of positive predictions and dividing it by the number of genuine positives observed. In Fig. 6 [29], the calculation of precision is displayed for your perusal.

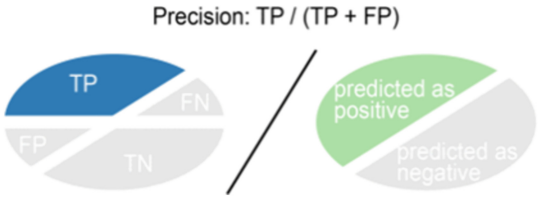


Fig. 6. Ellipses for precision calculation [29]

$$Precision = \frac{TP}{TP + FN} \tag{2}$$

An F1 score is determined by the harmonic mean of the recall and precision aspects of the test. A single metric that incorporates both precision and recall is referred to as the F1 score. With regard to both binary and multi-class categorization, it is frequently utilized as an assessment parameter for both LLM evaluation and classification. A representation of the F1 score calculation can be found in Fig. 7 [29].

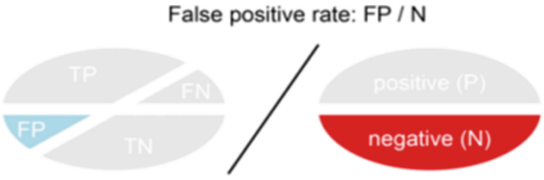


Fig. 7. Ellipses for F 1 score calculation [29]

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{3}$$

The term "**accuracy"** refers to a statistic that is calculated by dividing the total number of data sets by the sum of genuine positive (TP) and genuine negative (TN) predictions. From what can be seen in Fig. 7, the highest level of accuracy is represented by a value of 1.0, while the lowest level is represented by a value of 0.00 [24]. An illustration of the calculation of accuracy can be found in Fig. 8 [29].

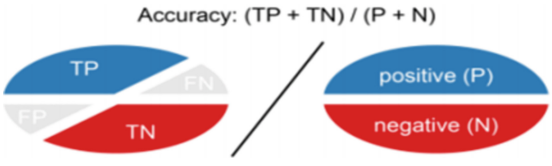


Fig. 8. Ellipses for accuracy calculation [29]

$$ACC = \frac{TP + TN}{TP + TN + FN + FP} = \frac{TP + TN}{P + N} \tag{4}$$

4 Result and Discussion

The findings of this study are presented and discussed in this part. The results of comparing the three distinct deep learning models' performances on the extracted and reduced dataset were shown, as was covered in Sub-section B of the result and discussion Section. Each and every metric has a value between 0 and 1. The model is better the closer the measures' values are near 1.

4.1 Results of Discriminant DL Models in Classification Task Using Synthetic Pandemic Dataset

Table 2 presents the performance comparison of the three different classifiers based on recall, precision, F1-Score, and accuracy. It is observed that RNN achieved the highest classification report values of 0.9881 for recall, precision, F1 Score and accuracy metrics and a little higher ROC-AUC where it was measured 0.9995. Compared to other models, RNN performance value is more constant and higher across all metrics. Using F1- score as for comparing all models, CNN obtained 0.8566, 0.8590, 0.8562 and 0.9048 for recall, precision, F1 Score and accuracy metrics respectively. MLP obtained the least value of 0.9115, 0.9134, 0.9114 and 0.9115 for recall, precision, F1 Score and accuracy metrics respectively. CNN follows RNN in recall while MLP follows RNN in precision, F1 score accuracy and ROC AUC. Therefore, CNN is making it the least performing model. Table 2 displayed the comparative results of the discriminative DL models investigated using synthetic dataset.

Figure 9 shows the graphical performance of discriminative DL-based models evaluated based on synthetic dataset in this study. The bar chart you provided evaluates the performance of three different discriminative DL models: CNN, RNN, and MLP. The scores were plotted against the models. The performance metrics displayed are recall, precision, F1-score accuracy and ROC-AUC. The RNN model achieves perfect scores on all metrics, indicating an outstanding performance. This could be due to a variety of factors, such as the nature of the dataset. Both models perform similarly and very well, with scores just above 0.9 for precision, recall, and F1-score, and slightly below 0.9 for few metrics. This indicates consistent and reliable performance but slightly less than the RNN. The RNN model outperforms the CNN and MLP models significantly, achieving perfect scores on all metrics. This result suggests that RNNs are well-suited for this particular dataset and task. The high performance of CNN and MLP indicates they are also strong contenders but may not capture the underlying patterns in the data as effectively as the RNN. Further investigation into the potential overfitting of the RNN model and the nature of the data might be necessary to ensure the robustness and generalizability of the results.

Table 2. Results of the discriminative DL models investigated using synthetic dataset.

Models	Recall	Precision	F1-Score	Accuracy	ROC AUC
CNN	0.8566	0.8590	0.8562	0.9048	0.9805
RNN	**0.9881**	**0.9881**	**0.9881**	**0.9881**	**0.9995**
MLP	0.9115	0.9134	0.9114	0.9115	0.9798

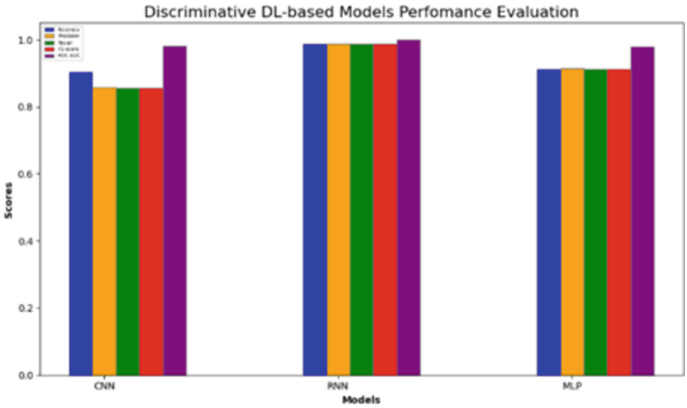


Fig. 9. Discriminative DL-based Models Performance Evaluation based on Synthetic Dataset.

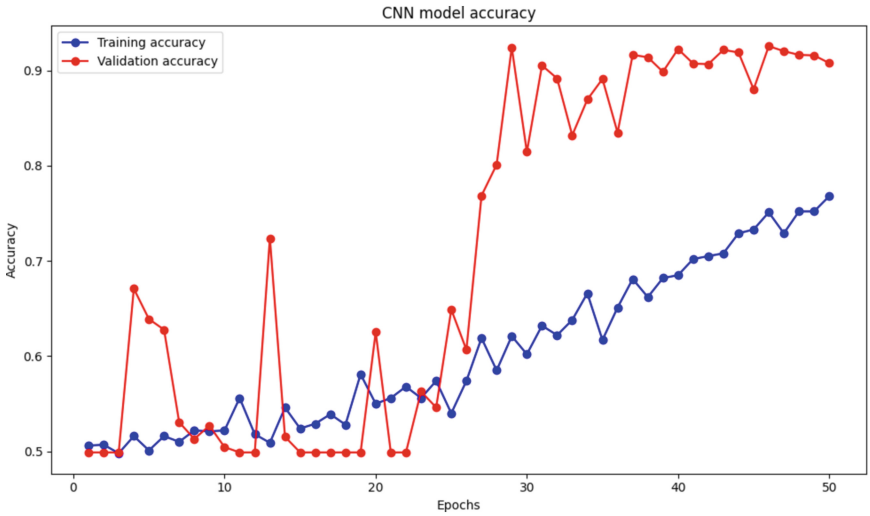


Fig. 10. CNN accuracy vs epochs curve

4.2 Discussion

The suggested RNN model outperformed the other models when employing different metrics, according to the findings of this empirical study. The highest classification report values across recall, precision, F1 Score, accuracy ROC-AUC metrics by RNN make RNN considered best method. One of the method's limitations is the amount of dataset used in this research as it is small. RNNs surpass other models on synthetic datasets exhibiting sequential or temporal attributes owing to its intrinsic architecture for sequence processing, capacity to capture long-term dependencies, adaptability to variable-length inputs, and proficiency in extracting temporal properties. Their specific architecture enables the retention of contextual information throughout time steps, facilitating more precise and contextually informed predictions than models that consider inputs as independent or necessitate considerable feature engineering to capture sequential dependencies. Utilizing these abilities, RNNs and their advanced variations, such as LSTMs and GRUs, are especially adept at tasks where the sequence and timing of data points are critical, rendering them the optimal selection for circumstances illustrated by your synthetic dataset. Overfitting is a prevalent issue with RNNs, particularly when the model grows excessively intricate and begins to memorize the training data instead of generalizing to novel data. Various solutions can be employed to mitigate overfitting in RNNs. Early stopping was implemented to mitigate model overfitting.

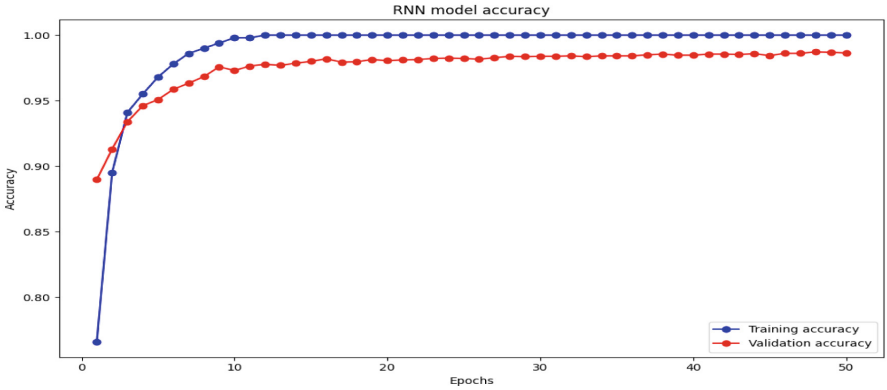


Fig. 11. RNN accuracy vs epochs curve

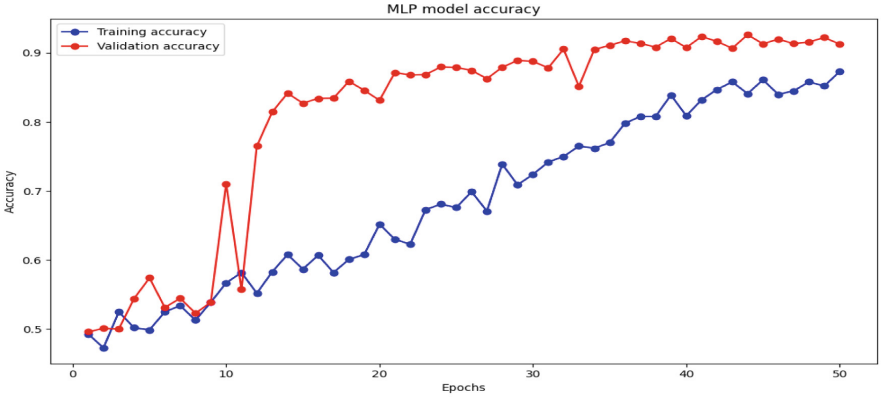


Fig. 12. MLP accuracy vs epochs curve

Table 3 is comparison of the current study with the existing state-of-the-art studies. Figures 9, 10 and 11 shows the training and validation accuracies vs epoch curve of CNN, RNN and MLP respectively. While RNN reflect a stable and steady in both in training and validation after attaining the peak accuracy CNN and produce unstable and unsteady curve all through. The reason for this performance might be unconnected the use of synthetic data as the results contravene [30]. Although, Zheng, [30] did not calibrate RNN and CNN in their study but MLP was considered to be outstanding among the models investigated (Fig. 12).

Table 3. Comparison of the current study with the existing state-of-the-art.

Studies	Model	Recall	Precision	F1 score	Accuracy	ROC AUC	Data type
[23]	LGBM	0.96	0.97	0.96	0.97	–	X-ray Images
[15]	VGG16	–	–	–	0.96	–	Synthetic manufacturing
[16]	AlexNet	–	–	–	96.16%	–	SAR imagery
[22]	Inception V3				96%		Synthetic pandemic
Our study	**RNN**	**0.9881**	**0.9876**	**0.9881**	**0.9881**	**0.9995**	**Synthetic pandemic**

5 Conclusion and Future Considerations

This paper proposes a comparative analysis of discriminative DL models for classification task based on synthetic dataset to identify a robust and efficient model for a relevant application. The study calibrated CNN, RNN and MLP which are baseline discriminative DL models for investigation. It is anticipated that training models with synthetic datasets would help determine the advantages and disadvantages of DL for the necessary task and improve performance, particularly in pandemic detection and control. The RNN model outperforms the CNN and MLP models significantly, achieving perfect scores on all metrics. This result suggests that RNNs are well-suited for this particular dataset and task.

The high performance of CNN and MLP indicates they are also strong contenders but may not capture the underlying patterns in the data as effectively as the RNN. Further investigation into the potential overfitting of the RNN model and the nature of the data might be necessary to ensure the robustness and generalizability of the results. If the number of cases and the need for preventive actions become too great for the resources to handle, this might also prove to be very helpful in the case of contagion. In addition to body temperature and knowledge of the frequency of chronic illnesses. Picture data can also be combined with these features to improve the system's accuracy and efficiency for medical professionals. A few examples of the medical applications for the proposed study are tumor identification, breast cancer screening, and more. Researcher may apply optimization techniques to achieve optimum performance of this model and validate this results in the near future. Furthermore, there is need to confirm the performance of RNN with datasets other than synthetic dataset. The synthetic nature of the dataset raises questions about the generalizability of the results, which should be further investigated thoroughly. Effect of metaheuristic optimisation algorithms for hyperparameter turning may also be investigated on discriminative DL models.

Acknowledgments. Authors acknowledge the Centre of Excellence, University of Zululand for the support received for this work.

References

1. Harapan, H., et a.: Coronavirus disease 2019 (COVID-19): a literature review. J. Infect Public Health **13**(5), 667–673 (2020)
2. Jiehao, C., et al.: A case series of children with 2019 novel coronavirus infection: clinical and epidemiological features. Clin. Infect. Dis. **71**(6), 1547–1551 (2020)
3. Manigandan, S., Wu, M.T., Ponnusamy, V.K., Raghavendra, V.B., Pugazhendhi, A., Brindhadevi, K.: A systematic review on recent trends in transmission, diagnosis, prevention and imaging features of COVID-19. Process Biochem. **98**, 233–240 (2020)
4. Mohammedqasem, R., Mohammedqasim, H., Ata, O.: Real-time data of COVID-19 detection with IoT sensor tracking using artificial neural network. Comput. Electr. Eng. **100**, 107971 (2022)
5. Ajagbe, S.A., Adegun, A.A., Mudali, P., Adigun, M.O.: Performance of machine learning models for pandemic detection using COVID-19 dataset. In: 2023 IEEE AFRICON, Nairobi, Kenya (2023)
6. Ajagbe, S.A., Adigun, M.O.: Deep learning techniques for detection and prediction of pandemic diseases: a systematic literature review. Multimedia Tools Appl. **83**, 5893–5927 (2023)
7. Sarker, I.H.: Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. SN Comput. Sci. **2**(6), 420 (2021)
8. Hasan, M.M., Islam, M.U., Sadeq, M.J., Fung, W.-K., Uddin, J.: Review on the evaluation and development of artificial intelligence for COVID-19 containment. Sensors **23**(1), 527 (2023)
9. Krohling, B.A., Krohlin, R.A.: 1D Convolutional neural networks and machine learning algorithms for spectral data classification with a case study for COVID-19, ArXiv, pp. 1–24 (2023)
10. Mohammedqasem, R., et al.: Multi-objective deep learning framework for COVID-19 dataset problems. J. King Saud Univ. Sci. **35**(3), 102527 (2023)
11. Salama, G.M., Mohamed, A., Khaled, M.: COVID-19 classification based on a deep learning and machine learning fusion technique using chest CT images. Neural Comput. Appl. **36**(10), 5347–5365 (2024)
12. Bhosale, Y.H., Patnaik, K.S.: Application of deep learning techniques in diagnosis of covid-19 (Coronavirus): a systematic review. Neural Process. Lett. **55**(3), 3551–3603 (2023)
13. Effati, M., Nejat, G.: A performance study of CNN architectures for the autonomous detection of COVID-19 symptoms using cough and breathing. Computers **12**(2), 44 (2023)
14. Gürsoy, E., Kaya, Y.: An overview of deep learning techniques for COVID-19 detection: methods, challenges, and future works. Multimedia Syst. **29**, 1603–1627 (2023)
15. Manettas, C., Nikolakis, N., Alexopoulos, K.: Synthetic datasets for Deep Learning in computer-vision assisted tasks in manufacturing. In: 9th CIRP Global Web Conference – Sustainable, Resilient, and Agile Manufacturing and Service Operations: Lessons from COVID-19 (2021)
16. Rizaev, I.G., Achim, A.: SynthWakeSAR: a synthetic SAR dataset for deep learning classification of ships at sea. Remote Sens. **14**(16), 3999 (2022)
17. AlMohimeed, A., Saleh, H., El-Rashidy, N., Saad, R.M.A., El-Sappagh, S., Mostafa, S.: Diagnosis of COVID-19 using chest x-ray images and disease symptoms based on stacking ensemble deep learning. Diagnostics (Basel) **13**, 1–23 (2023)
18. Mirza, O.M., Samak, A.H.: Neutrosophic fuzzy logic-based hybrid CNN- LSTM for accurate chest x-ray classification in COVID-19 prediction. Appl. Math. Inf. Sci. Int. J. **18**(1), 139-152 (2024)
19. Rayan, A., Holyl, A.S., Alaerjan, A.S., Alanazi, S., Taloba, A.I., Shahin, O.R., Salem, M.: Utilizing CNN-LSTM techniques for the enhancement of medical systems. Alexandria Eng. J. **72**, 323–338 (2023)

20. Akl, A.A., Hosny, K.M., Fouda, M.M., Salah, A.:A hybrid CNN and ensemble model for COVID-19 lung infection detection on chest CT scans. PLoS One **18**(3), e0282608 (2023)
21. Kuzinkovas, D., Clement, S.: The detection of COVID-19 in chest x-rays using ensemble CNN techniques. Information **14**(7), 370 (2023)
22. Wong, M.Z., Kunii, K., Baylis, M., Ong, W.H., Kroupa, P., Koller, S.:Synthetic dataset generation for object-to-model deep learning in industrial applications. PeerJ. Comput. Sci. **14**(5), e222 (2019)
23. Najjar, R., Hossain, Z., Ahmed, K.A., Hasan, R.: Exploring machine learning strategies in COVID-19 prognostic modelling: a systematic analysis of diagnosis, Classification and Outcome Prediction, BMJ Yale, pp. 1–24 (2024)
24. Oguntoye, J.P., Awodoye, O.O., Oladunjoye, J.A., Faluyi, B.I., Ajagbe, S.A., Omidiora, E.O.: Predicting COVID-19 from chest X-ray images using optimized convolution neural network. LAUTECH J. Eng. Technol. **17**(2), 28-39 (2023)
25. Awotunde, J.B., et al.: An improved machine learnings diagnosis technique for COVID-19 pandemic using chest x-ray images. In: International Conference on Applied Informatics (ICAI 2021). Communications in Computer and Information Science (2021)
26. Taiwo, G.A., Akinwole, T.O., Ogundepo, O.B.: Statistical analysis of stakeholders perception on adoption of AI/ML in sustainable agricultural practices in rural development. In: Proceeding Proceedings of Ninth International Congress on Information and Communication Technology. ICICT 2024 2024. Lecture Notes in Networks and Systems (2024)
27. Taiwo, G.A., Saraee, M., Fatai, J.:Crime prediction using twitter sentiments and crime data. Informatica **48**(6), 35–42 (2024)
28. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
29. Jooda, J.O., et al.: Multiple instances fingerprint image data acquisition. In: 2023 International Conference on Electrical, Computer and Energy Technologies (ICECET), Cape Town, South Africa (2023)
30. Zheng, Q., et al.: Deep tabular data modeling with dual-route structure-adaptive graph networks. IEEE Trans. Knowl. Data Eng. **35**(9), 9715–9727 (2023)



Deep Neural Network Compression for Lightweight and Accurate Fish Classification

Daanyaal Salie(✉), Dane Brown, and Kenneth Chieza

Department of Computer Science, Rhodes University, Makhanda, South Africa
daanieson@gmail.com, d.brown@ru.ac.za

Abstract. This research demonstrates the effectiveness of compression of deep learning models in a fish species classification system. In a bid to enhance model efficiency, response-based knowledge distillation was employed, balancing model compression and accuracy retention. The underwater footage collected in the Seychelles was annotated in collaboration with the South African Institute for Aquatic Biodiversity (SAIAB), and named the Aldabra data set. The accuracies of the MobileNet and DenseNet models were 89.26% and 86.78%, respectively, in the Aldabra dataset. Although the student model (MobileNet) achieved a lower accuracy, it enabled a three-fold reduction in parameters. In particular, in the much larger Fish4Knowledge dataset, MobileNet trained via knowledge distillation exceeded ResNet's accuracy, reaching 96.64% with an 18.7-fold size reduction. This performance improvement was attributed to efficient learning facilitated by knowledge distillation. The edge computing-friendly MobileNet model assimilates knowledge from the much larger ResNet output with information gain.

Keywords: Classification · Deep learning · Knowledge distillation · Model compression · Computer vision · Marine biodiversity

1 Introduction

Accurate classification of fish species is essential for the conservation of biodiversity, the sustainable management of fisheries, and food security [13]. The transition from labour-intensive manual methods to advanced AI-driven systems represents a transformative shift within aquatic ecological research. Convolutional neural networks (CNNs) are at the forefront, excelling in feature extraction and classification tasks.

Despite advancements, the deployment of machine learning models on edge devices for real-time analysis poses significant challenges. Resource-intensive models need to be optimised for under-represented classes of species and

This work was undertaken in Distributed Multimedia CoE at Rhodes University.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 300–318, 2025.
https://doi.org/10.1007/978-3-031-78255-8_18

low-quality data, dealing with issues such as illumination and complex backgrounds. Knowledge distillation (KD) and model selection are crucial for creating lightweight, efficient fish classification systems adapted for edge devices, common in this hyperconnected era [3].

CNNs are the most widely used machine learning models in the identification of fish species, excelling in feature extraction, detection, and classification tasks [2]. Using convolutional layers, CNNs can automatically learn and extract hierarchical features by dividing images into receptive fields, enabling the model to recognise intricate patterns [19]. Popular robust pretrained CNN architectures such as Xception, InceptionResNetV2, and ResNet101V2 have demonstrated remarkable accuracy, each offering unique strengths for modelling image data [1]. For example, Xception employs a depth-first structure and utilises separable convolutions in-depth, while InceptionResNetV2 distinguishes itself by integrating residual connections. Furthermore, ResNet101v1 addresses the challenge of the problem of vanishing gradients with version two modifying mappings in deep residual networks, which allows direct proportion between blocks. Together, these models demonstrate the varied strategies distinguished pretrained CNNs utilise in addressing the complexities of fish classification. However, the deployment of these models on edge devices for real-time analysis remains a resource-intensive problem. This necessitates the development of a streamlined and lightweight fish classification approach that is tailored for a substantial reduction in parameters.

Lightweight architectures such as MobileNet and DenseNet are designed for efficiency and minimal computational demands [8]. MobileNet, tailored for mobile and edge platforms, optimises efficiency by integrating depthwise separable convolution to apply a single filter per channel and standard 1 *times* 1 pointwise convolutions to create a linear combination of the depthwise layer-reducing parameters by pooling. The lower number of parameters also reduces memory usage and reduces computational costs while preserving essential features [17]. DenseNet addresses the problem of vanishing gradients with unique connectivity patterns, where each layer connects to every other layer in a feed-forward fashion in the model architecture, forming dense connections within blocks for a low number of parameters [12]. This innovative approach ensures a continuous gradient flow during backpropagation, and beyond overcoming this challenge, DenseNet consistently outperforms benchmark datasets, showcasing its reliability and superior performance.

Despite current advances, the implementations of existing machine learning models for fish classification become resource-intensive when deployed on edge devices for real-time analysis. This is mainly due to the models' need to analyse low-quality data and overcome challenges inherent in underwater fish classification, such as illumination issues and complex background analysis leading to misclassification [4]. Therefore, there is a growing need for a more streamlined and lightweight fish classification approach explicitly tailored for edge devices.

Leveraging compression techniques such as KD and efficient model selection is crucial to achieving this goal. Response-based KD involves the transfer of

knowledge from a larger, complex model (the teacher) to a smaller, lightweight model (the student) [3]. This process synergises with the CNN models chosen, enabling efficient deployment in resource-constrained environments.

Therefore, this paper aims to create an efficient, lightweight fish classifier using response-based KD. The contributions include technical progress and insights into deep learning methods and their potential for aquatic biodiversity monitoring and conservation initiatives through computer vision on edge devices for fish classification systems. The expected outcome is a practical solution that provides the foundation for automated aquatic ecology monitoring and understanding, widely applicable to marine and environmental science.

The structure can be outlined as follows. Section 3 provides detailed information on the proposed system, including the algorithms and datasets implemented. Section 4 analyses the experimental results. Section 5 explains the key findings, while Sect. 6 summarises the paper and offers insights into future research.

2 Related Work

Mathur and Goel [16] constructed a fish species classification system for underwater images, which are often hindered by poor visibility and low contrast due to light scattering and absorption. To overcome the limitations posed by small and low quality data sets, the authors propose a transfer learning-based method using the ResNet50 network with 25.6 M parameters. This method focuses on fine-tuning the last few layers of the ResNet50 network to improve the accuracy of the classification.

The model was tested on the Fish4Knowledge dataset, which comprises 27,370 images, using a 70:30 split. The method achieved a validation accuracy of 98.44%, also demonstrating high precision, recall, and F1-score values of 0.94, 0.85, and 0.89, respectively.

Chen et al. [7] addressed the issue of computational costs in complex deep-learning models through response-based KD. The experimental results demonstrate that response-based KD allows a smaller MobileNet (3.5 M parameters) to mimic DenseNet100 (7.2 M parameters). Refer to Sect. 3.2 for more details on response-based KD.

The experimental results showed that response-based KD is capable of reducing the parameters by more than 2-fold while maintaining an accuracy loss of less than 1% of 77.94% to 77.20% on CIFAR-100 datasets. They found that student models are typically up to six times faster and smaller in size compared to their teacher models. These are promising results that will be explored with underwater fish data.

3 Materials and Methods

This section describes the collected and public underwater image datasets, implementation of reduced parameters deep learning models, along with their associated hyperparameters and the evaluation metrics.

3.1 Datasets

Two datasets used for the evaluation of the models are our Aldabra and the publicly available Fish4Knowledge dataset.

The Aldabra dataset, created in collaboration with the South African Institute for Aquatic Biodiversity (SAIAB), consists of 1910 images taken in 2013 at the Aldabra Atoll in the Indian Ocean near the Seychelles. It includes 1206 annotations across 18 different species of fish, as outlined in Table 1.

Table 1. SAIAB 2013 Aldabra dataset showing the number of images and annotations available for each fish species

Fish Species	Image count	Annotation count
Acanthurus triostegus	4	33
Caesio lunaris	4	33
Heniochus diphreutes	16	34
Chromis weberi	10	37
Heniochus acuminatus	10	37
Siganus luridus	2	37
Caesio caerulaura	9	38
Lethrinus rubioperculatus	5	39
Naso hexacanthus	7	42
Amphiprion akallopsis	14	43
Dascyllus reticulatus	11	43
Chromis viridis	3	44
Lutjanus monostigma	7	44
Dascyllus aruanus	10	46
Thalassoma amblycephalum	13	121
Pterocaesio tile	7	127
Caesio xanthonota	15	189
Pseadanthias squamipinnis	26	219

Fish4Knowledge is a publicly available underwater fish data set compiled from 700,000 unconstrained underwater video footage captured on the coral reefs of Taiwan [10]. These videos have been transformed into an image dataset of 27,370 images, classified into 23 classes, as detailed in Table 2. Widely used as a benchmark in various studies, this data set is extensively used for fish classification purposes [16].

3.2 Response-Based Knowledge Distillation

To accomplish response-based KD, soft targets are employed, along with introducing a “temperature” factor [5]. These soft targets represent the probability

Table 2. Distribution of image counts per fish species in the Fish4Knowledge dataset, illustrating the range of available data for each category

Fish Species	Image count
Neoglyphidodon nigroris	16
Zanclus cornutus	21
Siganus fuscescens	25
Pempheris vanicolensis	29
Balistapus undulatus	41
Scolopsis bilineata	49
Scaridae	56
Zebrasoma scopas	90
Abudefduf vaigiensis	98
Hemigymnus fasciatus	127
Hemigymnus melapterus	127
Canthigaster valentini	147
Pomacentrus moluccensis	181
Chaetodon trifascialis	190
Lutjanus fulvus	206
Acanthurus nigrofuscus	218
Neoniphon sammara	299
Myripristis kuntee	450
Chaetodon lunulatus	2534
plectroglyphidodon dickii	2683
Chromis chrysura	3593
Amphiprion clarkii	4049
Dascyllus reticulatus	12112

distribution of the teacher model for classification, computed through the softmax function with a modifiable temperature factor. A high temperature, like 10, produces a smoother, more uniform distribution, indicating less confidence in specific predictions, while a lower temperature, such as 1, yields a sharper distribution, reflecting greater confidence in the teacher model's predictions. It is common practice to use hyperparameter tuning to find the optimal values.

The distillation loss function plays a crucial role in evaluating the variance between the soft targets of the teacher and the prediction of the student models [6]. This is controlled via a parameter called the alpha coefficient, which controls the balance between classification and distillation loss. This parameter effectively allows adjustment of the proportion of the training of the student model by the soft predictions of the teacher or traditional hard labels through classification loss. Improves the ability of the student model to understand predictions using

the correct predictions of the teacher model and addresses the incorrect classifications of the teacher model through classification loss. Achieving a balance between these two loss functions, adjusted by the alpha coefficient, ensures the most effective training of the student model.

In contrast to traditional pruning methods, where computational overhead reduction is achieved by removing non-essential components, response-based KD offers a nuanced approach [8]. It simplifies computational efficiency and empowers the student model to emulate the teacher's behaviour, making it essential in our pursuit of the deployment of efficient fish classification models.

However, the fish classification domain has challenges, such as class imbalance, a common issue arising from habitat variations due to environmental conditions and food source availability [20]. This results in datasets with unequal species sample sizes, as shown in Sect. 3.1. This imbalance in class samples challenges the model's ability to classify minority classes accurately. To mitigate this, machine learning models employ techniques such as data augmentation, rotations, flips, and colour adjustments, to create more samples for minority classes.

Explored in [9], oversampling and undersampling techniques are essential to address class imbalance. Undersampling randomly trims samples from the majority class to achieve balance, yet it comes with the trade-off of potential information loss. On the contrary, oversampling, often executed through data augmentation, aims to increase the number of minority class samples, but requires vigilance against overfitting due to the introduction of altered copies. These two techniques allow balanced models that carefully manage potential information loss and protect against overfitting arising from augmented copies.

3.3 Response-Based Knowledge Distillation Methodology

Figure 1 outlines the practical methodology of response-based KD, highlighting each step used in this research.

The proposed approach starts with selecting teacher models: Xception, InceptionResNetV2, ResNet101V2, and DenseNet121-due to their known ability to discern complex patterns in fish images, supported by benchmarks in image classification [1]. Subsequently, the models are evaluated using a framework from previous research [13] to find the best one for classifying fish species.

The focus transitions to selecting a suitable student model, with MobileNetV2 chosen for its ability to balance model size and computational efficiency for practical deployment in resource-constrained edge devices [8]. Subsequently, response-based KD is applied as a compression technique during the training of the student model, drawing insights from studies emphasising its efficacy in creating resource-efficient but effective classification models [14,22].

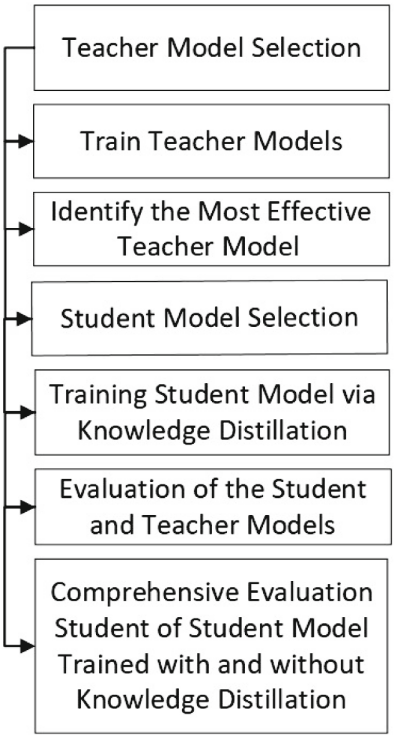


Fig. 1. Process diagram illustrating the methodology employed for the development of a lightweight fish classification system

A performance assessment phase was initiated, which involved a comparative analysis of the student and teacher models distilled with knowledge, evaluating the efficacy of knowledge transfer in fish classification. Additionally, a training session without response-based KD was conducted to assess its impact on student model performance and efficiency, providing insight into the implications of this compression technique.

Lastly, a comprehensive evaluation considered both training approaches for the student model, providing a nuanced understanding of the strengths and limitations of response-based KD within the domain of fish species classification. This structured approach to the methodology establishes a robust foundation for subsequent discussions.

3.4 Implementation

A 70:20:10 split was used for training, validation, and testing during implementation. Oversampling was applied to the Aldabra data set to address imbalances using data enhancement techniques such as flipping, rotations, and colour adjust-

ments. Due to memory constraints, no class imbalance measures were applied to the Fish4Knowledge dataset.

Pre-trained teacher models, including InceptionResNetV2, ResNet101V2, Xception, and DenseNet121, were selected to create an efficient fish classification system through response-based KD, with a MobileNetV2 serving as the student model.

Before response-based KD was applied, hyperparameter tuning was critical, exploring parameters such as dropout, L2 regularisation, and the choice of optimiser. Dropout and L2 regularisation values are essential in preventing overfitting, while the choice between Adam and SGD optimisers significantly impacts convergence speed and model performance on unseen data. The `KerasTuner` library was used with the random search algorithm to effectively transverse the parameter space. During training, dynamic tuning of other parameters was performed by monitoring the loss, including early stoppage and reduction in learning rate on a plateau.

Trained with derived hyperparameters from tuning, the teacher models InceptionResNetV2, ResNet101V2, Xception and DenseNet121 underwent a comprehensive evaluation. Metrics such as accuracy, precision, recall, and F1 were meticulously documented in the test split, allowing the selection of the best performing teacher model for subsequent use.

In the process of response-based KD, `Keras` distiller class was employed to optimize distillation hyperparameters, specifically temperature and alpha, using the `KerasTuner` random search algorithm. This adjustment identified the ideal temperature, which adjusts the 'softness' of the probability distribution of the teacher model to reveal. Inter-class relationships and teacher confidence between classes, and alpha, the distillation coefficient, which determines the balance between standard classification loss and distillation loss, for each student-teacher model pairing.

Following this, the student model underwent training on the optimised hyperparameters for response-based KD. A comparative analysis was conducted for teacher models, knowledge-distilled student models, and student models trained from scratch.

Therefore, this implementation mitigates class imbalance, employs pre-trained teacher and student models and fine-tunes hyperparameters. Using response-based KD, we present a streamlined fish classification system. The analysis highlights the efficacy of knowledge-distilled student models, emphasising their ability to compress and optimise models.

3.5 Classification Metrics

This study uses a comprehensive set of metrics to evaluate the fish classification model. Macro variants of precision, recall, and F1 score are used to provide a balanced assessment across all classes, crucial to address the class imbalance seen in the data sets used in this study [21].

Accuracy measures overall performance, while precision focusses on accurate classification. Recall assesses the model's ability to identify relevant instances. F1 is the harmonic mean of precision and recall, ensuring a balanced evaluation. Aggregated into macro precision, macro recall, and macro F1, these metrics collectively offer a holistic overview of performance.

Loss values indicate the classification performance and generalisation ability of a model, where lower values correspond to better performance [18]. This helps to identify underfitting and overfitting issues.

4 Results

The experiments analysed the stages required to build a streamlined fish species classification system. Due to response-based KD, experiments were performed independently on the Aldabra and Fish4Knowledge datasets, with our Aldabra dataset being the primary interest and the Fish4Knowledge dataset being the benchmark.

4.1 Validation of Model Parameters

`KerasTuner` was utilized to finetune hyperparameters based on validation loss, the analysis of Tables 3 and 4 highlights the standout performance of DenseNet121 on the Aldabra dataset and MobileNetV2 on the Fish4Knowledge dataset.

Table 3. Parameters denote the model size and complexity while the hyperparameter optimization values determined for deep learning models on the Aldabra dataset.

Model	Parameters	val_loss	Dropout	L2 Reg.	Optimiser
InceptionResnetV2	54 m	0.6196	0.3	0.0012	SGD
ResNet101V2	42 m	0.4764	0.3	1.11×10^{-5}	SGD
Xception	23 m	0.5683	0.3	1.99×10^{-5}	Adam
DenseNet121	7 m	0.3544	0.2	9.17×10^{-5}	Adam
MobileNetV2	2 m	0.3606	0.2	3.81×10^{-4}	Adam

The findings in Table 3 were derived from optimal combinations of dropout rates, L2 regularisation, and optimiser choices. It revealed that DenseNet121 achieved the lowest validation loss of 0.3544 on Aldabra with a dropout of 0.2 and L2 regularisation value of 9.17×10^{-5}, using the Adam optimiser.

Table 4. Parameters denote the model size and complexity while the hyperparameter optimisation values determined for deep learning models on the Fish4Knowledge Dataset

Model	Parameters	Best val_loss	Dropout	L2 Reg.	Optimiser
InceptionResnetV2	54 m	0.1584	0.2	0.0001	SGD
ResNet101V2	42 m	0.1264	0.2	0.0002	SGD
Xception	23 m	0.1977	0.3	1.48×10^{-5}	Adam
DenseNet121	7 m	0.1465	0.2	1.16×10^{-5}	Adam
MobileNetV2	2 m	0.1248	0.2	2.79×10^{-5}	Adam

Tables 4 similarly showed that MobileNetV2 produced the best validation loss of 0.1248 on Fish4Knowledge with the same dropout rate of 0.2, an L2 regularization value of 2.79×10^{-5}, and also utilizing Adam. These configurations highlight the critical role of hyperparameter tuning in improving model performance, positioning DenseNet121 and MobileNetV2 as the leading choices for the respective datasets based on their superior classification capabilities in validation data.

4.2 Experiment 1 Suitable Teacher Model

Experiment 1 identifies the most effective teacher model for response-based KD in Aldabra and Fish4Knowledge. The results of the Aldabra dataset are illustrated in Fig. 2, where the DenseNet121 teacher model demonstrates superior performance. It achieves the highest accuracy at 89.26% and Macro F1 at 83.14%, indicating its balanced precision and recall. Moreover, this model exhibits robustness, as evidenced by its minimal validation loss, elaborated in Sect. 4.1. The consistency across these metrics suggests that DenseNet121 is the most suitable candidate for the teacher model in the response-based KD process applied to the datasets in question.

Overfitting is evident by the third epoch, shown in Fig. 3. To combat this, early stopping is used when validation losses stop dropping. Overfitting persists despite using data augmentation and regularization Sects. 3.4. This issue is common in small, imbalanced datasets, as mentioned in [9]. DenseNet121, with fewer parameters, identifies patterns more effectively without memorizing them. It is a better choice for the Aldabra dataset, proving effective as a teacher model.

In the Fish4Knowledge dataset, as shown in Fig. 4, ResNet101V2 achieves 96.53% accuracy and 92.45% precision, surpassing DenseNet121. However, given the dataset's class imbalance, DenseNet121 proves more effective for minority classes with a superior Macro F1 of 83.97%. Since ResNet101V2 excels in classifying major classes, it is the favoured teacher model for this dataset.

Experiment 1 can be summarized as DenseNet121 being the optimal teacher model for the Aldabra dataset, while ResNet101V2 was more suitable for the

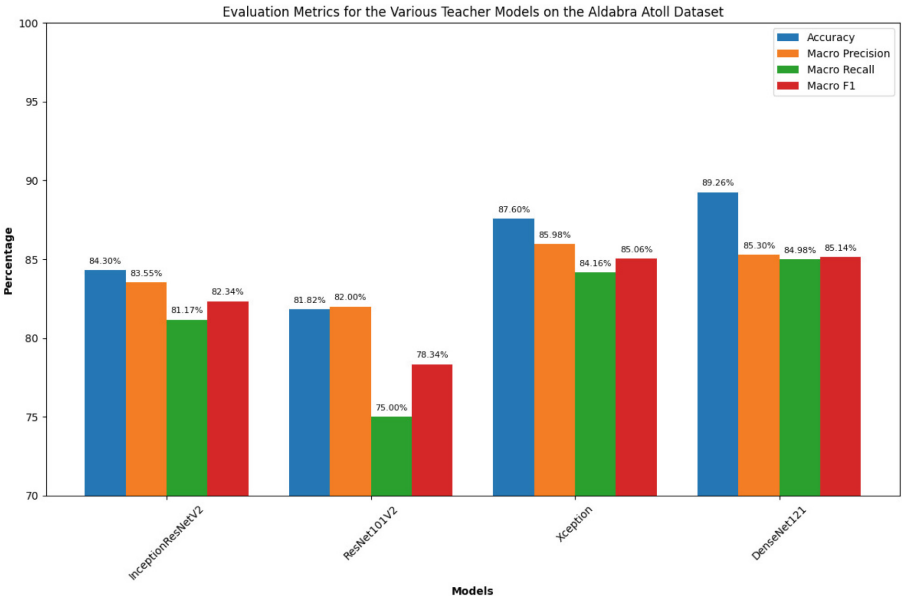


Fig. 2. Comparative analysis of teacher models on the Aldabra dataset

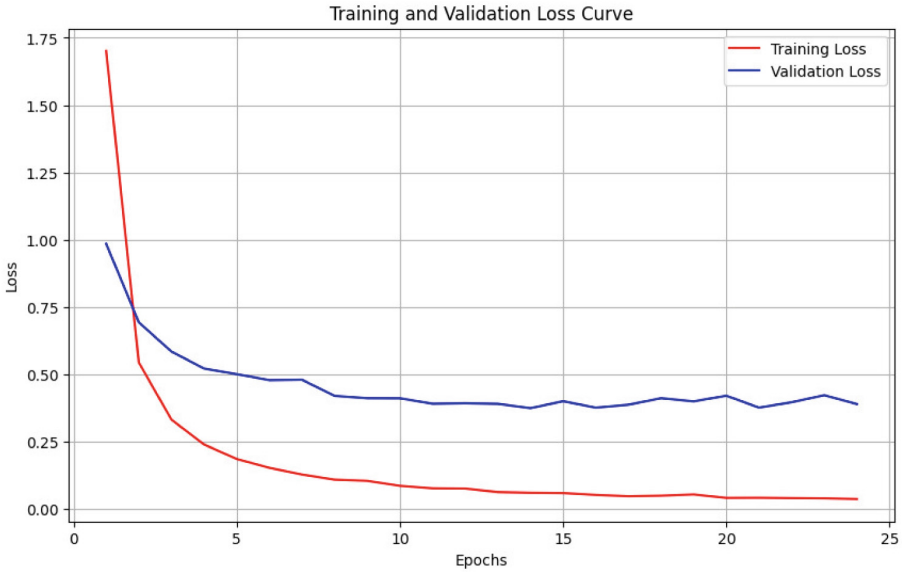


Fig. 3. DenseNet121 Model's Training and Validation Loss Curves Over 25 Epochs, Emphasizing Effective Learning and Generalization on the Aldabra Dataset

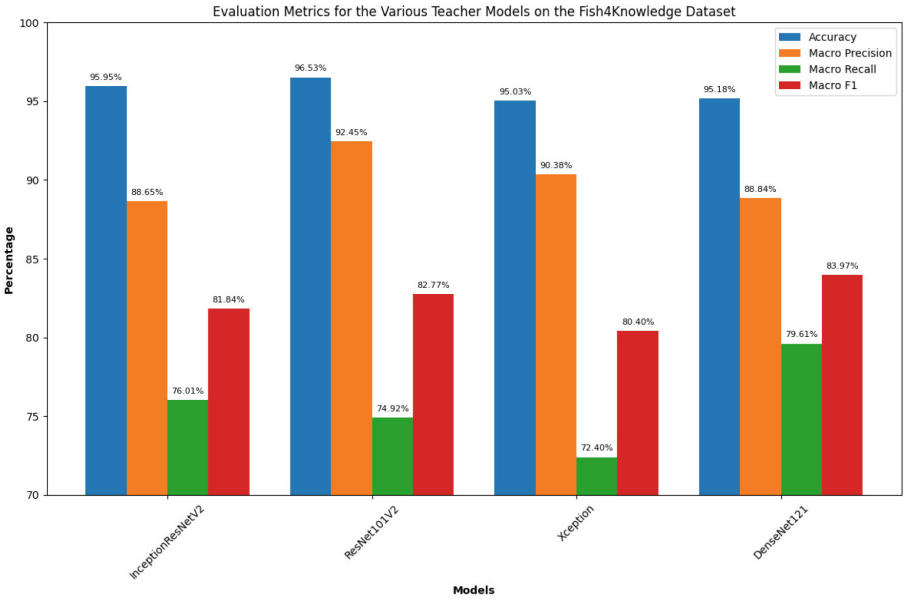


Fig. 4. Comparative analysis of teacher models on the Fish4Knowledge dataset

Fish4Knowledge dataset, with notable class imbalances offering avenues for further research.

4.3 Optimizing of Knowledge Distillation Parameters

Section 3.4 introduces the use of hyperparameters in response-based KD: temperature, which softens the teacher model's output probabilities and alpha, balancing between classification and distillation loss. The optimal values for these hyperparameters were determined using `KerasTuner`, leveraging effective teacher models from Experiment 1 and the chosen student model, MobileNetV2. Figure 5 displays the optimal values for each dataset's student-teacher model pair.

Table 5. Final hyperparameters for knowledge distillation across the Aldabra and Fish4Knowledge datasets

Dataset	Distillation Loss	Alpha	Temperature
Aldabra Atoll	3.6684888×10^{-8}	0.9	10.0
Fish4Knowledge	1.0349675×10^{-8}	0.6	4.0

For the Aldabra dataset pairing of DenseNet121 and MobileNetV2, a high temperature of 10.0 indicates significant softening of the teacher model's output, potentially revealing confounding variables in classes for the student model.

An alpha of 0.9 shows a strong emphasis on distillation loss. Conversely, the Fish4Knowledge pairing ResNet101V2 and MobileNetV2 achieves a temperature of 4.0, suggesting outputs with a balance between the teacher and original probabilities. The alpha of 0.6 also indicates a balance between distillation and classification losses.

4.4 Experiment 2 Teacher Vs Student (Response-Based KD)

Experiment 2 evaluates whether a student model can achieve comparable accuracy to its teacher. The student model, MobileNetV2, was thus trained using response-based KD. Model hyperparameters from Sect. 4.1 and distillation hyperparameters from Sect. 4.3 were applied to each dataset and their respective teacher-student model pairs.

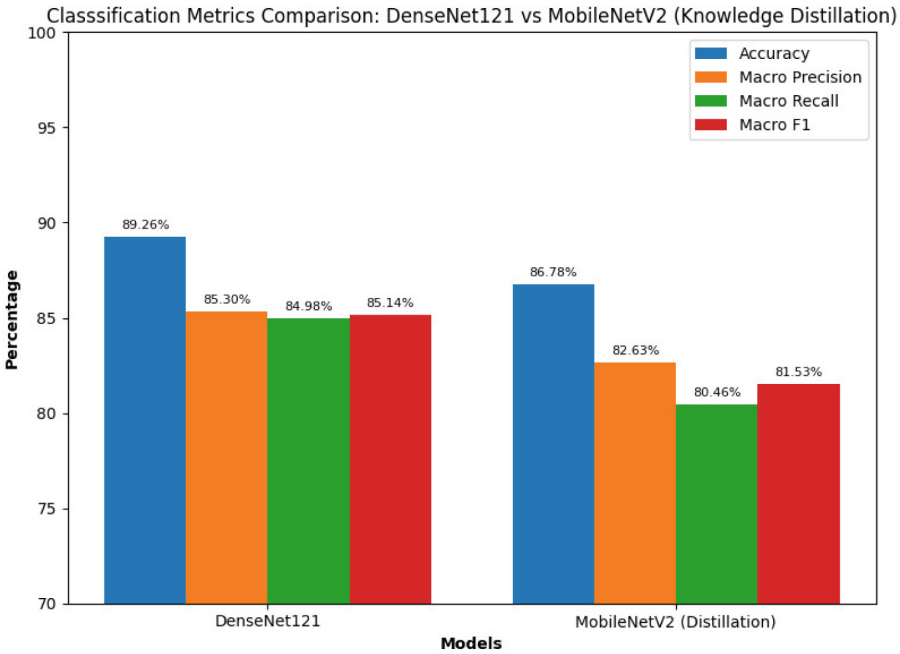


Fig. 5. Knowledge Distillation on the Aldabra Dataset: Teacher Model ResNet101V2 vs Student Model MobileNetV2

As a reminder for the Aldabra dataset, the teacher model DenseNet121 has 89.26% accuracy as shown in Fig. 5. In contrast, the less complex student model MobileNetV2 reached 86.78%. Given their architectural differences, this 2.48% difference in accuracy underscores the power of response-based KD. DenseNet121 showed slightly better precision at 85.30% compared to MobileNetV2's 82.63% and a recall of 84.9% against MobileNetV2's 80.46%. MobileNetV2's F1 stood at

81.53%, trailing DenseNet121's 85.14%. The student model's performance, especially considering the high temperature setting of 10, allows it to learn nuanced details from the teacher's predictions as cited in [3].

Shifting focus to the Fish4Knowledge dataset, represented in Fig. 6, the teacher model ResNet101V2 achieved an accuracy of 96.53%. The student model, MobileNetV2, slightly exceeded it with an accuracy of 96.64%, credited to a temperature setting of 4 and a distillation coefficient of 0.6, suggesting that the student model balanced its own learning with the teacher. The student model also outperformed the teacher in precision, registering 97.75% against ResNet101V2's 92.45%, and in recall, marking 82.72% over ResNet101V2's 74.92%. Consequently, the student achieved a superior F1.

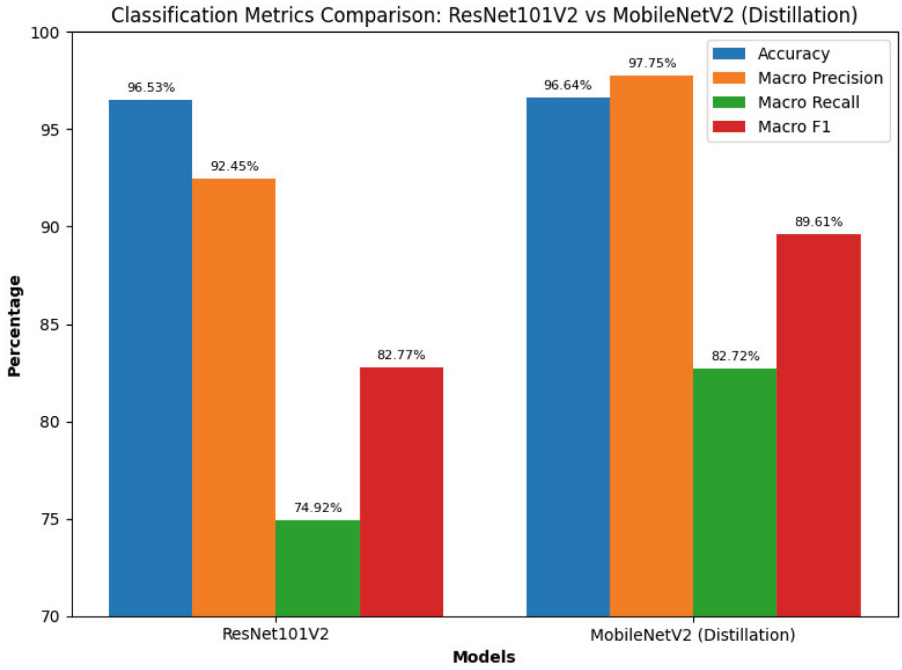


Fig. 6. Knowledge Distillation on the Fish4Knowledge Dataset: Teacher Model ResNet101V2 vs Student Model MobileNetV2

Experiment 2 showed that response-based KD does not only reduce model sized due to substantial parameter reduction but can also improve accuracy. Specifically, on the Aldabra dataset, the student model closely mirrored the teacher and resulted in similar accuracies but with the benefit of reduced size and improved inference speed. However, on Fish4Knowledge it exceeded its teacher.

The efficacy of response-based KD can allow for lightweight models such as MobileNetV2 to achieve comparable performance to more complex models as

possible. The next experiment will measure the extent response-based KD has on improving smaller models.

4.5 Experiment 3 Student Response-Based KD Vs Student Non-Response-Based KD

Experiment 3 highlights the difference between the students own learning ability and passing information of the teacher to it via response-based KD (Fig. 7).

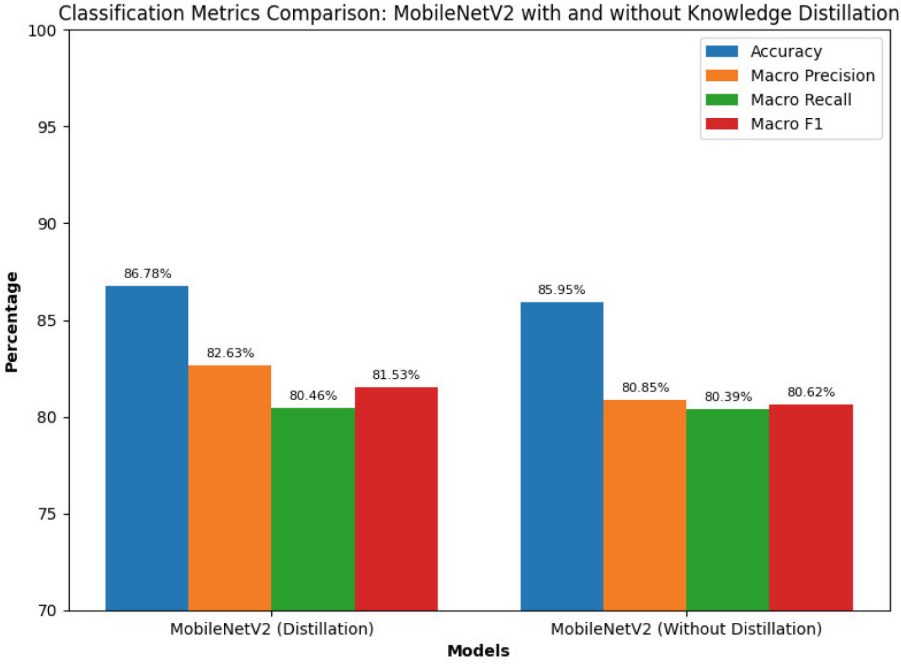


Fig. 7. Knowledge Distillation vs without on the Aldabra Dataset using MobileNetV2

Analysis of the MobileNetV2 models revealed that the non-response-based KD MobileNet achieved 85.95% while response-based KD achieved a slightly higher accuracy of 86.78. In precision, the response-based KD-trained model scores 82.63%, higher than the non-response-based KD model's 80.85%. However, recall rates for both models are closely matched. The response-based KD model's F1, at 81.51%, suggests better balance and generalization than the non-response-based KD model's 80.62%. Data imbalance, detailed in Table 1, affects performance, but response-based KD appears to offer mitigation.

The performance of student models on the Fish4Knowledge dataset is depicted in Fig. 8. MobileNetV2 (response-based KD) achieved 96.64% accuracy, a marginal improvement over MobileNetV2 (non-response-based KD) at

96.06%. The response-based KD-trained model achieved a precision of 97.75%, A 7.14% increase over the non-response-based KD-trained model. Recall achieved was consistent on both models. The F1-score shows response-based KD's overall improvement to the lightweight MobileNetV2 model.

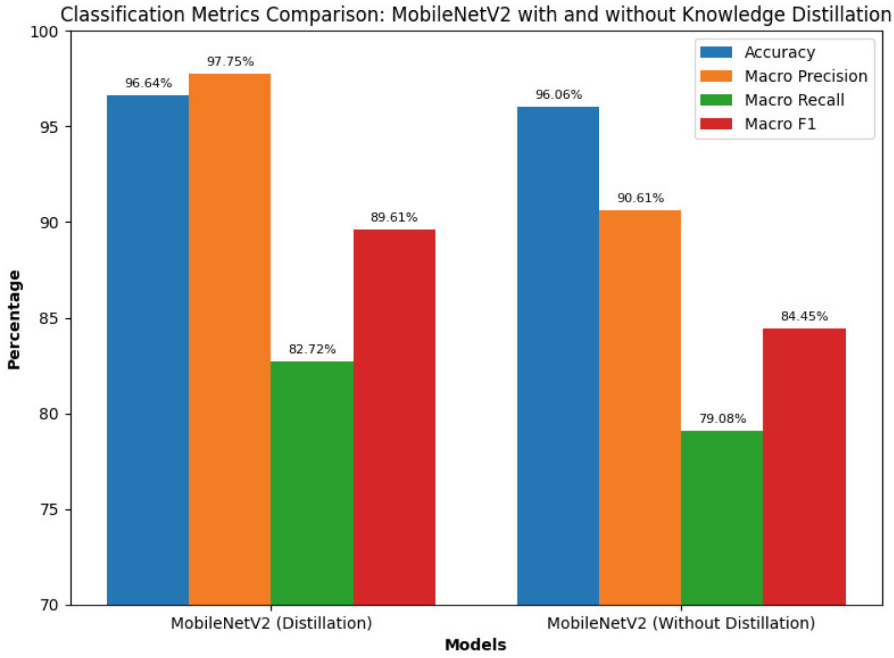


Fig. 8. Knowledge Distillation vs without on the Fish4Knowledge Dataset using MobileNetV2

5 Discussion on Knowledge Distillation

response-based KD emerged as a highly effective technique for creating a lightweight yet accurate fish classifier. This effectiveness is evident in the substantial reductions in model parameter sizes [3].

In the experiments conducted on the Aldabra dataset, a DenseNet121 served as the teacher model, while a MobileNetV2 was employed as the student model. This approach resulted in a significant parameter size reduction of approximately 3-fold, with exact values from 7,061,079 to 2,281,042 for DenseNet121 and MobileNetV2, respectively [12].

Likewise, on the Fish4Knowledge dataset, a ResNet101V2 functioned as the teacher model, with a MobileNetV2 as the student model, leading to an impressive compression of approximately 18.7 times in parameter size with exact values from 42,673,687 to 2,287,447 [11]. This underscores the potential of response-based KD in producing lightweight fish classifiers, reducing susceptibility to class imbalance issues.

Throughout the experiments, it became evident that response-based KD not only substantially reduced model complexity but also enhanced the accuracy of the student model, MobileNetV2, on the Fish4Knowledge dataset compared to training without distillation. Although the popular teacher model ResNet achieved high accuracy, the student model outperformed it, particularly in precision and F1 [15]. These results highlight the successful knowledge transfer from the teacher model to the student, enabling the latter to make more accurate predictions.

However, when comparing the results on the Aldabra dataset, which were significantly smaller than Fish4Knowledge, the improvements were limited to model size reduction for efficiency. This suggests that the size of the dataset may influence the effectiveness of response-based KD. In cases where the dataset is relatively small, as in the Aldabra dataset, the teacher model (DenseNet121) performed exceptionally well, and the student model (MobileNetV2) needed more room for improvement through distillation.

It is essential to recognize that diligent efforts were made to address the class imbalance issue during data collection and annotation. Employing techniques such as response-based KD, data augmentation, dropout, and L2 regularization to mitigate overfitting on the training dataset. This aims to uncover the underlying patterns of fish and, hopefully, improve the performance of minority classes. However, it remains a multifaceted challenge that necessitates comprehensive solutions beyond the scope of this study [18]. The consequences of class imbalance are far-reaching as they potentially introduce bias into models and compromise the performance of minority classes. This has significant real-world implications, particularly in domains like environmental monitoring and biodiversity studies, where several species are underrepresented in nature.

6 Conclusion

This study embarked on a comprehensive exploration through hyperparameter tuning to develop an efficient fish species classification system employing Knowledge Distillation (response-based KD) for model compression. The selected teacher models based on the tuning were DenseNet121 for the Aldabra dataset and ResNet101V2 for the Fish4Knowledge dataset. This played a pivotal role in optimizing deep learning via response-based KD, which transferred learnt weights more effectively than directly training the student model without it.

The chosen student model, MobileNetV2, thus proved to be a successful way to implement response-based KD. This resulted, optimally trained, lightweight model not only sustained similar accuracies as the larger (teacher) model but also demonstrated a significant reduction in parameter size, affirming the success of the compression objective.

In the future of algorithmic and data-driven AI, efficient and accurate fish classification systems will be vital for ecological research and environmental monitoring response-based KD showed potential for model compression, reducing parameters by several fold that can allow for real-time analysis on edge devices.

This work is a foundational step in advancing the field, emphasizing the ongoing need for research and innovation in developing efficient fish classification models for practical applications. The future holds promising possibilities for contributing to more effective environmental conservation and biodiversity monitoring efforts.

References

1. Ahmed, M.A., Hossain, M.S., Rahman, W., Uddin, A.H., Islam, M.T.: An advanced bangladeshi local fish classification system based on the combination of deep learning and the internet of things (IoT). J. Agric. Food Res. **14**, 100663 (2023). ISSN 2666-1543, https://doi.org/10.1016/j.jafr.2023.100663
2. Ali, A.M., Benjdira, B., Koubaa, A., Shafai, W.E., Khan, Z., Boulila, W.: Vision Transformers in Image Restoration: A Survey. Sensors **23**(5), 2385 (2023). https://doi.org/10.3390/s23052385
3. Alzubaidi, L., et al.: Review of deep learning: concepts, CNN architectures, challenges, applications. Future Directions. J. Big Data **8**(1), 53 (2021). https://doi.org/10.1186/s40537-021-00444-8
4. Barbedo, J.G.A.: A review on the use of computer vision and artificial intelligence for fish recognition, monitoring, and management. Fishes **7**(6), 335 (2022)
5. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum Learning. In: Danyluk, A.P., Bottou, L., Littman, M.L. (eds.) Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, 14–18 June 2009, ACM International Conference Proceeding Series, vol. 382, pp. 41–48, ACM (2009). https://doi.org/10.1145/1553374.1553380,
6. Chen, L., Yang, Y., Wang, Z., Zhang, J., Zhou, S., Wu, L.: Lightweight underwater target detection algorithm based on dynamic sampling transformer and knowledge-distillation optimization. J. Marine Sci. Eng. **11**(2) (2023). ISSN 2077-1312, https://doi.org/10.3390/jmse11020426
7. Chen, W.-C., Chang, C.-C., Lee, C.-R.: Knowledge distillation with feature maps for image classification. In: Jawahar, C.V., Li, H., Mori, G., Schindler, K. (eds.) ACCV 2018, Part III. LNCS, vol. 11363, pp. 200–215. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20893-6_13
8. Choudhary, T., Mishra, V.K., Goswami, A., Sarangapani, J.: A comprehensive survey on model compression and acceleration. Artif. Intell. Rev. **53**(7), 5113–5155 (2020). https://doi.org/10.1007/s10462-020-09816-7
9. Cui, S., Zhou, Y., Wang, Y., Zhai, L.: Fish detection using deep learning. Appl. Comput. Intell. Soft Comput. **2020**, 3738108:1–3738108:13 (2020), https://doi.org/10.1155/2020/3738108,
10. Fisher, R.B., Chen-Burger, Y.-H., Giordano, D., Hardman, L., Lin, F.-P. (eds.): Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data. ISRL, vol. 104. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30208-9
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016, pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90,

12. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017, pp. 2261–2269 (2017). https://doi.org/10.1109/CVPR.2017.243,
13. Iqbal, M.A., Wang, Z., Ali, Z.A., Riaz, S.: Automatic fish species classification using deep convolutional neural networks. Wirel. Pers. Commun. **116**(2), 1043–1053 (2021). https://doi.org/10.1007/s11277-019-06634-1
14. Khan, M.S., Alam, K.N., Dhruba, A.R., Zunair, H., Mohammed, N.: Knowledge distillation approach towards melanoma detection. Comput. Biol. Med. **146**, 105581 (2022). https://doi.org/10.1016/j.compbiomed.2022.105581
15. Liang, D., Chen, X., Xu, W., Zhou, Y., Bai, X.: TransCrowd: weakly-supervised crowd counting with transformers. Sci. China Inf. Sci. **65**(6), 1–14 (2022). https://doi.org/10.1007/s11432-021-3445-y
16. Mathur, M., Goel, N.: FishResNet: automatic fish classification approach in underwater scenario. SN Comput. Sci. **2**(4), 273 (2021). https://doi.org/10.1007/s42979-021-00614-8
17. Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., Terzopoulos, D.: Image segmentation using deep learning: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **44**(7), 3523–3542 (2022). https://doi.org/10.1109/TPAMI.2021.3059968
18. Montesinos López, O.A., Montesinos López, A., Crossa, J.: Overfitting, model tuning, and evaluation of prediction performance. In: Multivariate Statistical Machine Learning Methods for Genomic Prediction, pp. 109–139. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-89010-0_4
19. Scherer, D., Müller, A.C., Behnke, S.: Evaluation of pooling operations in convolutional architectures for object recognition. In: Diamantaras, K.I., Duch, W., Iliadis, L.S. (eds.) Artificial Neural Networks - ICANN 2010 - 20th International Conference, Thessaloniki, Greece, 15–18 September 2010, Proceedings, Part III, LNCS, vol. 6354, pp. 92–101 (2010). https://doi.org/10.1007/978-3-642-15825-4-10,
20. Sidumo, B., Sonono, E., Takaidza, I.: An approach to multi-class imbalanced problem in ecology using machine learning. Ecol. Inf. **71**, 101822 (2022). https://doi.org/10.1016/j.ecoinf.2022.101822
21. Sun, G.Q., et al.: Impacts of climate change on vegetation pattern: mathematical modelling and data analysis. Phys. Life Rev. **43**, 239–270 (2022), ISSN 1571-0645, https://doi.org/10.1016/j.plrev.2022.09.005, https://www.sciencedirect.com/science/article/pii/S1571064522000653
22. Zhang, J., Wang, W., Lu, C., Wang, J., Sangaiah, A.K.: Lightweight deep network for traffic sign classification. Ann. des Télécommunications **75**(7–8), 369–379 (2020). https://doi.org/10.1007/s12243-019-00731-9

Pre-training a Transformer-Based Generative Model Using a Small Sepedi Dataset

Simon Phetole Ramalepe[1(✉)], Thipe I. Modipa[1,3,4], and Marelie H. Davel[2,3,4]

[1] Department of Computer Science, University of Limpopo, Polokwane, South Africa
{simon.ramalepe,thipe.modipa}@ul.ac.za
[2] Faculty of Engineering, North-West University, Potchefstroom, South Africa
[3] Centre for Artificial Intelligence Research, Pretoria, South Africa
[4] National Institute for Theoretical and Computational Sciences, Stellenbosch, South Africa

Abstract. Due to the scarcity of data in low-resourced languages, the development of language models for these languages has been very slow. Currently, pre-trained language models have gained popularity in natural language processing, especially, in developing domain-specific models for low-resourced languages. In this study, we experiment with the impact of using occlusion-based techniques when training a language model for a text generation task. We curate 2 new datasets, the Sepedi monolingual (SepMono) dataset from several South African resources and the Sepedi radio news (SepNews) dataset from the radio news domain. We use the SepMono dataset to pre-train transformer-based models using the occlusion and non-occlusion pre-training techniques and compare performance. The SepNews dataset is specifically used for fine-tuning. Our results show that the non-occlusion models perform better compared to the occlusion-based models when measuring validation loss and perplexity. However, analysis of the generated text using the BLEU score metric, which measures the quality of the generated text, shows a slightly higher BLEU score for the occlusion-based models compared to the non-occlusion models.

Keywords: Transformers · Text generation · Pre-training · Occlusion-based training · Datasets

1 Introduction

Low-resourced languages are characterised by small training datasets, as well as limited tools and linguistic analyses [12]. The Sepedi language, one of the official languages in South Africa falls in this category. It is a highly disjunctively written language which is mostly characterised by 2 character words as discussed by the authors in [26]. Previous attempts to develop text-based language models include the use of recurrent neural networks (RNNs) and long-short memory networks

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 319–333, 2025.
https://doi.org/10.1007/978-3-031-78255-8_19

(LSTMs) [29]. These techniques could perform well on short sentences but their performance would degrade as the sentence length increases. The introduction of the transformer architecture [31] has since gained much attention in developing transformer-based models that can generate longer coherent sentences. This technique uses a multi-head self-attention mechanism to compute the contextual representation of each word by considering the entire input sequence. Complementary approaches when developing language models for low-resourced languages include transfer learning techniques and data augmentation. Transfer learning is a machine learning approach that leverages knowledge gained from pre-trained models to improve the performance of a low-resourced model in a related task [25,32]. In the context of text generation, this approach transfers knowledge from a pre-trained language model to a low-resourced language through the fine-tuning process [12,14]. Transfer learning has emerged as an approach that effectively addresses data scarcity in low-resourced languages. Data augmentation is another approach that has been explored in several studies [4,7,22,28,30], with various approaches proposed for generating artificial data. The technique has also been shown to improve the performance of a language model.

The purpose of this study is to analyse the implications of pre-training a Sepedi generative pre-trained model (GPT) when only a relatively small dataset is available. Specifically, we aim to understand how well a semi-matched dataset (from the same language but a different context) can be used as the seed model for the new context, and whether the type of pre-training used makes a difference. The target context we are interested in is the Sepedi radio news dataset obtained from radio broadcasts, and the semi-matched dataset contains a mix of different styles. The study contributes:

- Sepedi monolingual (SepMono) dataset curated from several South African resources and Sepedi radio news (SepNews) dataset curated from radio news domain.
- A comparison of two different pre-training techniques for a small dataset, before and after fine-tuning.
- Sepedi transformer generative pre-trained models (SepGPT) and (SepGPT-OCC), trained on the newly curated Sepedi corpus using the standard GPT training approach and occlusion-based technique.

2 Background

We provide a brief overview of language models and the use of pre-training, before discussing selected closely related studies.

2.1 Pre-trained Language Models

Language models are applied to many computational linguistic problems including text generation. Language modeling remains the most fundamental task in natural language processing (NLP) and automatic speech recognition systems

[34]. It assigns a joint probability distribution over a sequence of linguistic words [32,34]. In its simplest form, the joint probability of an entire sequence of words can be estimated by multiplying the number of conditional probabilities using the chain rule.

$$\begin{aligned} P(w) &= P(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} P(w_i \mid w_{<i}) \\ &= P(w_1)P(w_2 \mid w_1) \ldots P(w_i \mid w_1, w_2, \ldots, w_{i-1}) \end{aligned} \tag{1}$$

Pre-trained language models (PLMs) are models that were initially trained on a large corpus to capture broad representations of a language's syntactic and semantic knowledge [32]. The primary pre-training objective for generative language models is to predict the missing token in a given sentence. Alternative objectives like "next-sentence prediction" are also used for specific purposes, such as understanding sentence relationships. In the "predicting the missing token" approach, auto-regressive models like GPT-2 [24] predict the next token based on previous tokens, while Bidirectional encoder representations from transformers (BERT) [6], uses the masked language model (MLM) and next-sentence prediction objectives to predict the masked tokens. The MLM objective is achieved by randomly masking the input tokens and training the model to predict the masked tokens using the rest of the context. The next sentence prediction task is used to train the model to predict the relationship of the next sentence given the previous sentence. We use the terms "occlusion" and "non-occlusion" to refer to pre-training objectives where some input tokens are occluded or masked and the training process where input tokens are not occluded respectively.

The occlusion-based pre-training is a common pre-training objective in computer vision [3], however, it has been used in generative models [6]. This technique helps the model to be robust to unseen tokens [10]. Harbecke et al. [10], show the effectiveness of combining occlusion and language modeling for a classification task. In general, occlusion is regarded as an explanation method where the difference in prediction when removing an input feature is seen as an indicator of the importance of the feature [36]. On the other hand, the non-occlusion pre-training technique is commonly used in autoregressive models like GPT-2 [24].

Subsequently, pre-trained models are commonly fine-tuned on smaller datasets to perform several downstream tasks using the learned knowledge. To avoid catastrophic forgetting during the standard fine-tuning approach, Howard et al. [11] proposed discriminative fine-tuning (which uses different learning rates for different layers), slanted triangular learning rates (which varies the learning rates using the defined pattern for better convergence), and gradual unfreezing of layers (which unfreezes layers from top to bottom during training). The authors show that these approaches can improve the model's performance.

2.2 Closely Related Studies

Several studies have adopted pre-training before fine-tuning their models on downstream tasks. We discuss some of the most relevant studies in this section,

with a specific focus on the size of the dataset and evaluation metrics used to evaluate the performance of the models.

Niculescu et al. [21] developed a Romanian text generation model (RoGPT-2) using the GPT-2 architecture. The authors pre-trained their model from scratch using 17 GB of monolingual data and fine-tuned it on several tasks for news article generation. The performance of their model was evaluated using perplexity as a metric and obtained a score of 34.37 on the validation set and 33.74 on the test set for the natural language generation (NLG) task on their base model. Experimenting with news generation, they obtained a BLEU score of 35.90 between the reference text and the generated text. Similarly, Buzea et al. [1] developed a smaller Romanian text generation model (MCBGPT-2) using a standard GPT-2 architecture. The model was trained on a small dataset of 24k news items crawled from online news portals. The performance of the model was monitored using the sparse categorical cross-entropy loss function.

In another study, Martin et al. [17] developed SwaBERT, a low-resourced language model for the Swahili language using the BERT architecture. The authors trained their model with monolingual data collected from news websites, forums, Wikipedia and popular social media websites with a total size of 105 MB. The original BERT's architecture was followed when pre-training their model. In their approach, the authors trained a model using both MLM and next sentence prediction as used in [6]. However, during training, they experimented by varying the vocabulary size and the number of training and warm-up steps. The performance of their model was analysed on several downstream tasks including news classification, emotions detection, sentiment analysis and named entity recognition (NER).

In the study by Wongso et al. [35], the authors developed several low-resourced models for the Sudanese language. The authors pre-trained GPT-2 Model, BERT and RoBERTa architectures using 785 MB monolingual data and evaluated them for emotional classification through fine-tuning. The authors observed that both Sudanese RoBERTa and BERT surpassed or performed comparably to larger multilingual models. Perplexity and validation loss were used to evaluate their models during pre-training. The study by Martin et al. [18] also demonstrated that it is possible to pre-train a large language model using a relatively small dataset. The authors pre-trained a GPT-2 model using 4 GB of French data. They used the masked language modeling approach as used by RoBERTa as their training objectives. However, instead of fixing the masked token, they introduced dynamic token masking which seemed to improve the model variability and made it robust during training. Their model was evaluated on several downstream tasks including natural language inference.

Finally, in another low-resource language context, Salim et al. [27] developed a BanglaGPT generative model using the GPT architecture. They trained their model from scratch using 26.24 GB corpus, scraped from several websites. The model outperformed multilingual GPT (mGPT) and LSTM models with an optimal perplexity score of 2.86.

3 Experimental Setup

In this section, we start by describing the datasets used in this study, the model development process, and lastly the evaluation techniques used.

3.1 Data Collection

SepMono Dataset. Table 1 lists the available Sepedi monolingual text datasets used to develop the pre-trained models. We indicate the specific size of each dataset to show the relative size and scarcity of the available Sepedi textual datasets. We combine all these datasets to create a Sepedi monolingual dataset referred to as "SepMono". The SepMono dataset consists of 432,970 sentences with 11,360,000 tokens after cleaning. All of these data sets are freely available for research purposes.

The National Centre for Human Language Technology (NCHLT) corpus [23], was curated from a collection of several South African government entities crawled from gov.za websites. The corpus was collected from various language units from 2007 to 2011. The Autshumato dataset, developed by McKellar [19] was collected from several sources (magazines, policies, newsletters, translation works) and documents crawled from the government domain. The News headlines dataset [16], is another Sepedi text corpus based on radio news headlines. It was crawled from one of the South African national Sepedi radio stations between 2018 and 2020. The Sepedi newspaper dataset [15], was collected from the Vukuzenzele newspaper between 2011 and 2022. The dataset has both monolingual and translated parallel data from English to other South African low-resource languages. The Leipzig dataset [8] consists of newspaper texts and texts randomly collected from the web while Web crawl dataset [5,33] consists of general data extracted from Common Crawl for various languages including Sepedi.

Table 1. Existing monolingual Sepedi datasets curated to create the SepMono dataset

Dataset (monolingual)	#Data size
NCHLT	12 MB
Autshumato1	19 MB
Autshumato2(from bilingual)	16 MB
Headlines News	125 KB
Vukunzenzele	600 KB
Web crawl	8.5 MB
Leipzig	9 MB

We split the dataset into 80% training, 10% validation and 10% testing. We show the size of the partitioned dataset in the first row of Table 2.

Table 2. Datasets partitions used during training and evaluation. The SepMono is used during pre-training process while SepNews-1 and 2 are used for fine-tuning.

SepMono Dataset	Training	Validation	Testing	Total data
#Sentences	346,380	43,290	43,299	432,970
#Tokens	9,020,000	859,110	1,480,000	11,360,000
#Unique tokens	105,090	30,320	39,720	125,04
SepNews Dataset				
SepNews-1				
#Sentences	12,668	3,168	-	15,836
#Tokens	357,431	81,896	-	439,327
#Unique tokens	14,966	5,612	-	16,772
SepNews-2				
#Sentences	-	-	3,520	3,520
#Tokens		-	101,758	101,758
#Unique tokens		-	6,420	6,420

SepNews Dataset. The last row of Table 2 shows the SepNews dataset which represents our target context. The dataset was curated in two phases (June 2022 to April 2023 and May 2023 to November 2023). We named these datasets "SepNews-1" and "SepNews-2" respectively. The SepNews-1 dataset is split into 80% training and 20% validation sets. For testing, we used the SepNews-2 dataset, which comes from a different time period and is therefore a strong proxy for unseen contexts.

Data Cleaning and Pre-processing. To prepare the data for model training, we remove special characters, repeated full stops, and forward and backward slashes. We also break the sentences at full stops. We use the GPT2TokenizerFast[1] to tokenized the data. The tokenizer uses the Byte Pair Encoding (BPE) technique which breaks the input text into sub-word tokens based on the most common byte pairs in the dataset. This technique is well suited for morphologically rich languages (such as the Sepedi language) which use a combination of smaller morphological units to construct individual words [20]. It also helps to control the vocabulary size during model training. We first normalise the data to lowercase and fine-tune the GPT2TokenizerFast using the SepMono dataset to capture the morphological representation of the Sepedi language.

3.2 Models

Model Architecture. We adopt the GPT-2 architecture [24] to develop the Sepedi generative model (SepGPT). The GPT-2 architecture uses the decoder part of the transformer architecture [31] with the pre-training objective of predicting the next word given the previous words. The GPT-2 architecture consists

[1] https://huggingface.co/transformers/v3.0.2/model_doc/gpt2.html#gpt2tokenizerfast.

of a series of decoder blocks, each incorporating a masked self-attention block which helps to identify the relevant words the model should focus on. The feed-forward neural network block within the hidden layer, on the other hand, establishes the relationships between the input tokens. At the lower level, it has the token and positional embedding layer to map each token in the vocabulary to a high-dimensional vector representation and to capture the positional embedding of the tokens.

Model Training. The experiments are conducted on Google Colab's cloud development environment[2] with NVIDIA T4 GPUs. We start training the model using the standard pre-training objective of the GPT-2 model and later add the occlusion-based technique to help the model learn different structural representations that capture the semantics of the sentence. To optimise hyperparameters, the Weights and Biases[3] random sweep configuration is utilized, with validation loss monitored throughout. The AdamW optimizer is employed, starting with a learning rate of 1e-4, and the model convergence is enhanced using a learning rate scheduler with a warm-up. With a patience of 5, early stopping is applied to control overfitting. Due to limited computational resources, the hyperparameter search is randomized to 20 counts over 100 epochs, exploring the optimal learning rate, number of transformer layers, number of attention heads, and dropout rate (see experimental details in the Appendix for more information). The optimal validation loss and perplexity are recorded at each epoch, while the test loss from the best-performing models is also logged. The non-occlusion model obtained its optimal hyperparameters using a higher number of attention heads and layer blocks, as shown in Table 4 in the Appendix.

Occlusion-Based Training. Occlusion-based techniques have been used extensively in computer vision and classification tasks as discussed in Sect. 2.1. In this study, we experiment with this technique for a text generation task. To train the model using the occlusion-based technique, we use the same experimental setup as described above to find the optimal hyperparameters. However, we now combine the generative approach of predicting the next word given the previous word with an occlusion-based technique to help the model learn different structural representations that capture the semantics of the sentence. In addition to the standard generative approach, the occlusion probability is added as a hyperparameter to the model. We experiment with probabilities of 0.1, 0.3, and 0.5 and record the model's performance per epoch during training. This technique introduces noise to the input text by randomly occluding some tokens in the input sequence based on a given probability. The model is then trained to predict the occluded tokens and recover them using the surrounding non-occluded tokens.

[2] https://colab.google/notebooks/.
[3] https://wandb.ai/site.

Fine-Tuning Models. To evaluate the performance of the trained models, we fine-tune them for a text generation task using the radio news (SepNews) dataset. To fine-tune the models, the same experimental setup as in the pre-training process is applied. The optimal hyperparameters obtained for each model are used, and the models are initialized with the respective pre-trained weights. The models are then fine-tuned for 50 epochs with early stopping. Specifically, we experiment with the gradual unfreezing fine-tuning technique as discussed and experimented in [11,25]. This technique works by gradually unfreezing model layers (from top to bottom during training). Importantly, the approach helps the model to retain previous knowledge and avoid catastrophic forgetting during fine-tuning. Initially, only the top 2 layers are unfrozen. We set the unfreeze interval to 2 epochs, meaning that at every 2 epochs, additional layers are progressively unfrozen. This interval, basically specifies how often layers are unfrozen during the training process.

3.3 Evaluation Metrics

Although human evaluation is the most standard method for evaluating text generation language models, it is expensive to execute and the results are difficult to reproduce [2]. Model evaluation can either be done intrinsically or extrinsically. Intrinsic evaluation methods that have been used for language modeling include the Bilingual Evaluation Understudy (BLEU) score and perplexity [1,9,21]. Perplexity (PPL), remains the most preferred metric to evaluate the performance of a language model [13,34]. It is the exponentiation of the entropy, which is the average negative log-likelihood of the true word sequence. That is, for a language model that assigns probabilities to sequences of words, the perplexity PPL of a sequence of words W where $W = w_1,w_2,...w_n$ can be computed as:

$$\mathrm{PPL}(W) = \exp\left(\frac{1}{n}\sum_{i=1}^{n} -\log P(w_i \mid w_1, \ldots, w_{i-1})\right) \tag{2}$$

where $P(w_i|w_1, w_2, ...w_{i-1})$ is given by a language model and is the probability assigned by the model to the word given the previous words in the sequence. We also use the categorical cross-entropy loss function to monitor the model's performance during training.

The BLEU score metric is used to evaluate the quality of the text generated by a model. It does this by comparing the generated text to the reference text. Its computation is based on the precision of n-grams between the generated text and the reference text, along with a brevity penalty to handle shorter text. It is calculated as follows:

$$\mathrm{BLEU} = BP \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{3}$$

where BP is the brevity penalty, which adjusts the score based on the length of the generated text compared to the reference length. The brevity penalty is calculated as follows:

$$BP = \begin{cases} 1 & \text{if } c > r \\ \exp\left(1 - \frac{r}{c}\right) & \text{if } c \leq r \end{cases} \tag{4}$$

where c is the length of the generated text, and r is the length of the reference text.

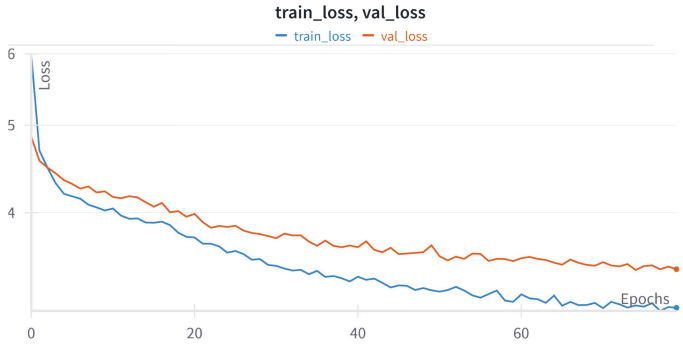
4 Results

In this section, we discuss the performance of the 4 developed models by analysing their performance using validation loss and perplexity. We start with the analysis of the pre-training results, before demonstrating the effects of fine-tuning.

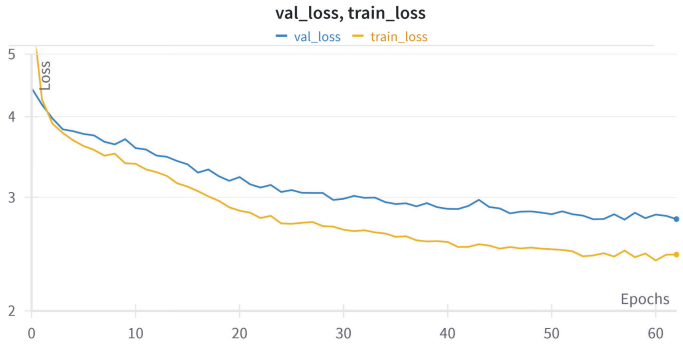
Figure 1a and 1b show the training curves of the optimal scores recorded during the pre-training process of the occlusion and non-occlusion models respectively. Although the maximum number of epochs was set to 100, the occlusion-based model achieved its optimal validation loss of 3.46 at epoch 79, while the non-occlusion model reached its optimal validation loss of 2.78 at epoch 61. Based on the size of the dataset, and the complexity of occlusion-based training, the SepGPT-OCC obtained the optimal validation loss which is 0.68 higher compared to the non-occlusion model (SepGPT). However, the fine-tuning process (the last row of Table 3) significantly closed this gap to just 0.11. Similar observations are noted in validation perplexity, test loss and test perplexity. Notably, the fine-tuning process improved the validation perplexity of the occlusion-based model by almost 50%. Of importance to note is that the test loss and test perplexity are relatively high compared to the pre-trained models. The introduction of a completely new test dataset (same news domain but from a different time period) presumably affected the test performance of our fine-tuned models.

We also compare our models with other low-resourced models trained from scratch using the transformer-based technique (the first row of Table 3). While systems evaluated on different datasets and for different languages are not directly comparable, this comparison provides an indication of the level of performance achieved. We note that the validation perplexity score of the SepGPT-OCC pre-trained model outperformed both the Sudanese GPT-2 and RoGPT-2 base models by a validation perplexity score of 5.08 and 2.48 respectively. Furthermore, the test perplexity score of the SepGPT-OCC model also outperformed the RoGPT-2 base model. Although the optimal test loss obtained from the SepGPT is 2.30 higher compared to the standard BanglaGPT model, the general performance of all our models and the training processes used in this study were satisfactory.

We further generated text from the trained models and computed a BLEU score to measure the quality of the generated text from the standard SepGPT and the SepGPT-OCC models. Although BLEU score is used mostly in translation tasks, it gives a good indication of similarities between the generated and the reference text. Without human evaluation, the pre-trained SepGPT-OCC model obtained a BLEU score of 5% higher compared to the SepGPT model



(a) Occlusion-based training curve.



(b) Non-occlusion training curve.

Fig. 1. Comparison of training curves: (a) Occlusion-based and (b) Non-occlusion.

while the BLEU score of the fine-tuned SepGPT-OCC model was 3.86% higher. These scores show how robust the occlusion model can be to unseen text. We show an example of the generated text from the SepGPT-OCC (FT) model Table 5 in the Appendix section. Although further analysis of the generated text is necessary we note that the sentences are mostly grammatically correct.

Even though the occlusion-based technique introduces noise to the training data, it is observed that both models (pre-trained and fine-tuned) could still produce a competitive result compared to a non-occlusion model. Further observations from experiments show that although the performance of the occlusion-based technique is low compared to standard training, the technique can provide valuable insight, especially in building robust code mixed text generation models.

Table 3. Model comparison with other low-resourced languages. We focus on validation loss, perplexity, and BLEU score. (FT = Fine-tuned)

Model	Size	Train Loss	Validation Loss	Validation Perplexity	Test Loss	Test Perplexity	BLEU Score
Sudanese GPT-2	26 GB	2.43	3.61	36.97	-	-	-
Sudanese BERT	26 GB	2.80	2.84	17.20	-	-	-
Sudanese RoBERTa	26 GB	1.96	1.95	7.04	-	-	-
BanglaGPT	785 MB	-	-	2.86	0.45	-	-
RoGPT-2 base	17 GB	-	-	34.37	-	33.74	35.90%
SepGPT-standard	63 MB	2.44	2.78	16.04	2.75	15.71	24.19%
SepGPT-occlusion	63 MB	3.13	3.46	31.89	3.42	30.85	29.48%
SepGPT-standard (FT)	63 MB	2.28	2.69	14.87	4.02	56.00	44.98%
SepGPT-occlusion (FT)	63 MB	2.42	2.80	16.48	4.03	56.74	48.84%

5 Conclusion

The purpose of this study was to evaluate the performance of a generative transformer-based model on a relatively small low-resourced Sepedi dataset. We curated and cleaned 2 datasets, the Sepedi monolingual dataset (SepMono) from various available Sepedi resources and the radio news dataset (SepNews). We used the SepMono dataset to train 2 transformer-based generative models using the standard GPT-2 training objective and the occlusion-based technique. The SepNews dataset was used to fine-tune our 2 pre-trained models using the gradual unfreezing technique. In this process, 2 additional Sepedi text generation models were further developed. Although the transformer-based models are data-intensive, we successfully used the hyperparameter search to obtain the optimal parameters for our pre-trained models (SepGPT and SepGPT-OCC). Compared to other low-resourced GPT models, our models obtained higher and comparable results. We further used the BLEU score metric to evaluate the performance of the generated text from our trained models and obtain the first and new optimal BLEU score of 44.98% (SepGPT) and 48.84% (SepGPT-OCC) for the Sepedi language. The scores obtained in this study create a new baseline for the Sepedi language and other South African low-resourced languages. Although the occlusion-based technique approach obtained in general a higher BLEU Score compared to the non-occlusion model, the non-occlusion model performed fairly better in both validation and perplexity loss which could mean that the model is more reliable in generating coherent text. In future work, we aim to experiment with other fine-tuning techniques and also, to analyse the performance of these techniques and their impact on generating code-switched text.

Acknowledgements. We would like to acknowledge the Telkom Centre of Excellence for Speech Technology at the University of Limpopo and the MUST deep learning research group at the Northwest University (Potchefstroom) for their continued support.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Appendix: Experimental Details

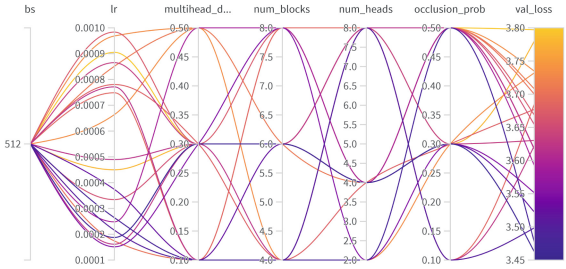


Fig. 2. Occlusion hyperparameter search

In Fig. 2, we show the optimal sweep curves for the hyperparameter search using the occlusion-based approach. For our experiment, the batch size is fixed at 512, and we search for the optimal learning rate, number of transformer layers, number of attention heads, and the dropout rate. The goal is to minimize the validation loss. At each epoch, we record the optimal validation loss and perplexity. Additionally, we include occlusion probabilities as a hyperparameter to evaluate model performance at different occlusion levels. These probabilities are set to 0.1, 0.3, and 0.5.

We observe from Fig. 2 that when the model becomes too complex, its performance deteriorates (i.e., the lower the validation loss, the better). This could be attributed to the introduction of noise in the training set. Interestingly, the model performed best with an occlusion probability of 0.3.

Table 4. Optimal hyperparameters for each model

Hyperparameter	Occlusion	Non-occlusion
Batch size	512	512
Learning rate	2e−4	2e−4
Dropout	0.3	0.3
Vocab size	50,225	50,225
Weight decay rate	1e−2	1e−2
Block layers	6	8
Attention heads	4	8
Max epochs	100	100
Occlusion probability	0.3	-

Table 5. Translation of Reference and Generated Texts from Sepedi to English

Sepedi Text	English Translation
Reference Text: moragonyana mothopo woo o akareditše le thuto ya barutwana ya ka sekolong go bala direto tše di ngwadilwego ka maleme a mangwe, ka kudukudu seisimane ga ešita le seafrikaanse. moranang ke ye tee ya dikgwedi tše nne tša goba le matšatši a. morekiši wa dikenywa o ile a...	Reference Text: Later on, that source also included a lesson for students at school to read poems written in other languages, particularly English and Afrikaans. 'Moranang' is one of the four months, having only a few days. A fruit vendor went and...
Generated Text: moragonyana mothopo woo o tsenago kgahlanong le maemong a gagwe ba tsenetše thoko kgopelo ya sekolo kudu gape go šomišana le mašeleng ao a amantšhwago lefaseng ka bophara kudu ke moeta pele wa eff, ka gore mongwaledi phare wa peleng wa naga e lemošitše mošomo:	Generated Text: Later on, that source went against his position, and they joined the request for the school, particularly to collaborate with funds associated worldwide, as led by the former EFF leader, as the former secretary warned of the work.

References

1. Buzea, M.C., Truan-Matu, S., Rebedea, T.: Automatic Romanian text generation using GPT-2. UPB Sci. Bull. **84**(4) (2022)
2. Celikyilmaz, A., Clark, E., Gao, J.: Evaluation of Text Generation: A Survey. arXiv (2020)
3. Chandel, H., Vatta, S.: Occlusion detection and handling: a review. Int. J. Comput. Appl. **120**(10), 33–38 (2015)
4. Chang, C.T., Chuang, S.P., Lee, H.Y.: Code-switching sentence generation by generative adversarial networks and its application to data augmentation. In: Interspeech 2019, pp. 554–558. International Speech Communication Association (2019). https://doi.org/10.21437/Interspeech.2019-3214
5. Conneau, A., et al.: Unsupervised cross-lingual representation learning at scale. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 8440–8451. Online (2020)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North, pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (2019)
7. Gao, Y., Feng, J., Liu, Y., Hou, L., Pan, X., Ma, Y.: Code-switching sentence generation by BERT and Generative Adversarial Networks. In: Interspeech 2019, pp. 3525–3529. International Speech Communication Association (2019). https://doi.org/10.21437/Interspeech.2019-2501
8. Goldhahn, D., Eckart, T., Quasthoff, U.: Building large monolingual dictionaries at the leipzig corpora collection: from 100 to 200 languages. In: Calzolari, N., et al. (eds.) Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012), pp. 759–765. European Language Resources Association (ELRA), Istanbul, Turkey (2012)
9. Gupta, D., Ekbal, A., Bhattacharyya, P.: A semi-supervised approach to generate the code-mixed text using pre-trained encoder and transfer learning. In: Cohn, T., He, Y., Liu, Y. (eds.) Findings of the Association for Computational Linguistics:

EMNLP 2020, pp. 2267–2280. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.findings-emnlp.206
10. Harbecke, D.: Explaining Natural Language Processing Classifiers with Occlusion and Language Modeling (2021)
11. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Melbourne, Australia, pp. 328–339 (2018)
12. Howcroft, D.M., Gkatzia, D.: Most NLG is low-resource: here's what we can do about it. In: Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM), pp. 336–350. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid) (2022)
13. Lee, N., Bang, Y., Madotto, A., Fung, P.: Towards few-shot fact-checking via perplexity. In: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1971–1981. Association for Computational Linguistics, Online (2021)
14. Li, J., Tang, T., Zhao, W.X., Wen, J.R.: Pretrained language model for text generation: a survey. **5**, 4492–4499 (2021)
15. Marivate, V., Njini, D., Madodonga, A., Lastrucci, R., Dzingirai, I., Rajab, J.: The Vuk'uzenzele South African multilingual corpus (2023)
16. Marivate, V., et al.: Investigating an approach for low resource language dataset creation, curation and classification: Setswana and Sepedi. In: Mabuya, R., Ramukhadi, P., Setaka, M., Wagner, V., van Zaanen, M. (eds.) Proceedings of the first workshop on Resources for African Indigenous Languages. pp. 15–20. European Language Resources Association (ELRA), Marseille, France (2020)
17. Martin, G., Mswahili, M.E., Jeong, Y.S., Woo, J.: SwahBERT: language model of Swahili. In: Carpuat, M., de Marneffe, M.C., Meza Ruiz, I.V. (eds.) Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 303–313. Association for Computational Linguistics, Seattle, United States (2022). https://doi.org/10.18653/v1/2022.naacl-main.23. https://aclanthology.org/2022.naacl-main.23
18. Martin, L., et al.: CamemBERT: a tasty French language model. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 7203–7219. Association for Computational Linguistics, Online (2020). https://doi.org/10.18653/v1/2020.acl-main.645
19. McKellar, C.: Autshumato Monolingual Sepedi Corpus (2022)
20. Mielke, S.J., Eisner, J.: Spell once, summon anywhere: a two-level open-vocabulary language model. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, pp. 6843–6850 (2019)
21. Niculescu, M.A., Ruseti, S., Dascalu, M.: RoGPT2: Romanian GPT2 for text generation. In: 2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI), pp. 1154–1161. IEEE, Washington, DC, USA (2021). https://doi.org/10.1109/ICTAI52525.2021.00183
22. Pratapa, A., Bhat, G., Choudhury, M., Sitaram, S., Dandapat, S., Bali, K.: Language modeling for code-mixing: the role of linguistic theory based synthetic data. In: Gurevych, I., Miyao, Y. (eds.) Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1543–1553. Association for Computational Linguistics, Melbourne, Australia (2018). https://doi.org/10.18653/v1/P18-1143
23. Puttkammer, M., Schlemmer, M., Pienaar, W., Bekker, R.: NCHLT Sepedi text corpora (2014)

24. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
25. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**(140), 1–67 (2020)
26. Ramalepe, S.P., Modipa, T.I., Davel, M.H.: The development of a Sepedi text generation model using transformers. In: Proceedings of South Africa Telecommunication Networks and Applications Conference (SATNAC), pp. 51–56. Fancourt, Western Cape, South Africa (2022)
27. Salim, M., Murad, H., Das, D., Ahmed, F.: BanglaGPT: A Generative Pretrained Transformer-Based Model for Bangla Language (2023). https://doi.org/10.1109/ICICT4SD59951.2023.10303383
28. Samanta, B., Reddy, S., Jagirdar, H., Ganguly, N., Chakrabarti, S.: A deep generative model for code switched Text. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, pp. 5175–5181 (2019)
29. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, NIPS 2014, vol. 2, pp. 3104–3112. MIT Press, Cambridge (2014)
30. Tarunesh, I., Kumar, S., Jyothi, P.: From Machine Translation to Code-Switching: Generating High-Quality Code-Switched Text (2021)
31. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017)
32. Wei, C., Wang, Y.C., Wang, B., Kuo, C.C.J.: An overview of language models: recent developments and outlook. APSIPA Trans. Signal Inf. Process. **13**(2) (2024). https://doi.org/10.1561/116.00000010
33. Wenzek, G., et al.: CCNet: extracting high quality monolingual datasets from web crawl data. In: Calzolari, N., et al. (eds.) Proceedings of the Twelfth Language Resources and Evaluation Conference, pp. 4003–4012. European Language Resources Association, Marseille, France (2020). https://aclanthology.org/2020.lrec-1.494
34. Winata, G.I.: Multilingual Transfer Learning for Code-Switched Language and Speech Neural Modeling (2021)
35. Wongso, W., Lucky, H., Suhartono, D.: Pre-trained transformer-based language models for Sundanese. J. Big Data **9**(1), 39 (2022). https://doi.org/10.1186/s40537-022-00590-7
36. Zintgraf, L.M., Cohen, T.S., Adel, T., Welling, M.: Visualizing deep neural network decisions: prediction difference analysis. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017, Conference Track Proceedings. OpenReview.net (2017)



Automated Fish Detection in Underwater Environments: Performance Analysis of YOLOv8 and YOLO-NAS

Kenneth Chieza(✉), Dane Brown, James Connan, and Daanyaal Salie

Department of Computer Science, Rhodes University, Makhanda, South Africa
chiezakenneth@gmail.com, {d.brown,j.connan}@ru.ac.za

Abstract. Accurately determining fish populations is crucial for marine science, aiding in species tracking, fish behaviour studies, and sustainable aquaculture practices. This study evaluates the effectiveness of automated fish detection using You Only Look Once (YOLO) models, particularly YOLOv8 and YOLO-NAS, using benchmark datasets (DeepFish and OzFish) and two novel datasets from the South African Institute for Aquatic Biodiversity (SAIAB). The results indicate that YOLOv8 models consistently outperform YOLO-NAS models, effectively addressing challenges like poor image quality and varying environmental conditions. This study concludes that YOLOv8 models are better suited for underwater fish detection, offering robust performance and highlighting the potential of advanced deep learning models in enhancing ecological studies and sustainable fishery management.

Keywords: Fish Detection · Computer Vision · Deep Learning

1 Introduction

Determining the size of fish populations is a fundamental part of marine science, providing essential insights for tracking species abundance, studying fish behaviour, and informing sustainable aquaculture practices [12]. These insights are also crucial for addressing global fish stock exploitation, as initiating long-term conservation and management projects without adequate knowledge of fish species and their sub-populations is a significant challenge [5]. Furthermore, accurate population data is important for informing national policies to ensure the long-term health of oceans, which are both a major tourist attraction and food source in South Africa [1]. Effective fish detection methods are an important first step in this process, as accurate counts are impossible without first locating and identifying the fish.

Underwater video footage analysis has become the primary method for detecting fish, addressing the drawbacks of traditional techniques such as studying captured fish and visual inspections by divers. This approach provides significant benefits, such as generating large amounts of reusable data, ease of

This work was undertaken in Distributed Multimedia CoE at Rhodes University.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 334–351, 2025.
https://doi.org/10.1007/978-3-031-78255-8_20

application, and the ability to cover a large area [3]. However, manual review of the substantial data involved is time-consuming and prone to errors. As a result, the use of automated fish detection methods has become increasingly important to address these challenges, with the ultimate aim of improving the efficiency of ecological studies, including spatial distribution, predator-prey relationships, over-exploitation, and evolutionary processes [5,8].

Accurately detecting fish in their natural underwater habitats is challenging. This difficulty primarily arises from the poor quality of images captured underwater. These images are often degraded due to several factors, including the extreme environment in natural bodies of water, random fish movement against coral and sand, low-resolution capturing devices, varying luminosity, colour bias, water murkiness, occlusions, and light attenuation [9,16,23]. These challenges highlight the need for further research on effective methods to overcome them.

Recent advancements in computer vision have led to the development of machine learning models with the potential to overcome the difficulties associated with identifying and localising fish in underwater videos. In particular, deep learning-based object detection algorithms have shown the most promise with large potential for AI-driven applications [9,12,20]. These algorithms can automatically extract patterns and features in complex scenarios that are often difficult to handle using traditional non-deep-based techniques [13]. Within this domain, the You Only Look Once (YOLO) architectures have emerged as some of the most efficient and accurate methods for fish detection.

Several studies have made use of YOLO architectures for the fish detection task [2,9,11,17,22]. While earlier versions have demonstrated strong performance across different datasets and detection scenarios, more recent versions, such as YOLOv8 and YOLO-NAS, promise further enhancements in accuracy. However, there is limited research specifically evaluating YOLOv8 and YOLO-NAS in underwater environments. This study aims to comprehensively assess these two versions of YOLO architecture to address this research gap and make several contributions outlined below:

1. addressing the limited research on the application of YOLOv8 and YOLO-NAS in underwater environments, filling a crucial gap in the literature regarding using these models as fish detection methods
2. introduces two new fish detection datasets, one from Protea Banks and another from Lake Tanganyika, enhancing the diversity of data available for training and testing YOLO models, contributing to the generalisation capabilities of the models across different aquatic ecosystems.
3. providing a qualitative analysis of YOLOv8 and YOLO-NAS to visualise specific challenges faced when detecting fish underwater.

The remainder of this paper is organised as follows: Sect. 2 provides an overview of the YOLO architectures used in this study, detailing their core components and differences. Section 3 outlines the methodology, including dataset preparation, training, and evaluation procedures. In Sect. 4, the results are presented and discussed, focusing on the performance comparison between the models. Finally, Sect. 5 concludes the paper by summarising the key findings and suggesting potential areas for future research.

2 YOLO Architectures

YOLO models are deep learning models for real-time object detection. They provide an end-to-end solution for region proposal and classification, as they can directly predict bounding boxes and their corresponding class probabilities with a single feed-forward network [18]. This is in contrast to previous detection models, such as RCNN, which require separate networks for identifying potential object locations and classifying those objects [7]. There have been several iterations of YOLO over the years.

Several studies have made use of YOLO architectures for the fish detection task. Wang et al. [22] used YOLOv2 to detect out-of-water fish, achieving a mean Average Precision (mAP) of 91.3%. YOLOv3 is the most commonly used detection model for fish detection. Raza and Song [17] proposed an enhanced YOLOv3 architecture for fish detection with three significant modifications. First, a downsampling scale with a stride of 4 was added to the base architecture to produce a finer feature map that better characterises smaller fish objects. A k-means++ clustering algorithm was also applied to the fish detection dataset to determine optimal anchor box sizes. Additionally, the default loss function was altered to reduce the impact of varying object sizes within the same class. These enhancements resulted in a 4.13% increase in mAP compared to the baseline YOLOv3 architecture on a dataset containing six classes of fish of varying sizes. Jalal et al. [9] used YOLOv3 in combination with a Gaussian Mixture Model (GMM) and Optical flow for additional detections. This approach achieved F1-scores of 95.47% and 91.2% on two datasets. Similarly, Knausgård et al. [11] used YOLOv3 as part of a fish detection system, achieving a peak absolute precision of 86.96% on a temperate fish dataset. Al Muksit et al. [2] performed a comparative analysis of YOLOv3 and two approaches for fish detection that extend YOLOv3 named YOLO-FISH-1 and YOLO-FISH-2. These models were trained on the OzFish and DeepFish datasets that are also utilised in this study. YOLO-FISH-1 performed better on the OzFish dataset, while YOLO-FISH-2 achieved the best results on the DeepFish dataset. Al Muksit et al. [2] performed a further analysis by adding YOLOv4 to the set of models and training them on a dataset made by merging DeepFish and OzFish. YOLOv4 performed optimally for this analysis. Gašparović et al. [6] assessed the performance of YOLOv4, YOLOv5, YOLOv6, YOLOv7, and YOLOv8 on an underwater object detection dataset. YOLOv5 resulted in the highest mAP of 97.10%, while YOLOv4, YOLOv6, YOLOv7, and YOLOv8 achieved mAP scores of 94.21%, 95.30%, 96.30%, and 95.10% respectively. Ouis and Akhlouf [15] evaluated the performance of YOLOv7 and YOLOv8 models on the fish detection task using underwater sonar images. YOLOv8 produced better results with an average precision of 72.47%.

Although research has been conducted to evaluate YOLOv8 on underwater object detection, these studies have primarily focused on underwater objects other than fish [2] or have utilised sonar images of fish [15]. To the best of our knowledge, no studies have yet explored the application of YOLO-NAS for fish detection, leaving a gap in the current research. While earlier YOLO versions,

such as YOLOv2 and YOLOv3, have demonstrated strong performance across various datasets and detection scenarios, this study focuses specifically on two recent versions: YOLOv8 and YOLO-NAS, which may offer further improvements in accuracy.

2.1 YOLOv8

YOLOv8, released in January 2023, is an addition to the YOLO family of object detection models developed by Ultralytics as a successor to YOLOv5 [10]. It is made up of a three-part architecture consisting of the backbone, neck, and head. The backbone, a modified version of the CSPDarknet53 architecture, extracts features from input images and uses the cross-stage partial connections (CSP) technique to pass information through the network. The neck combines these features at multiple scales through a series of downsampling and upsampling convolutional layers. This multi-scale feature combination allows the model to detect objects of various sizes. Finally, the head processes these fused features to perform the object localisation and classification tasks to pinpoint the location and identity of objects within an image.

YOLOV8 uses an anchor-free approach instead of the anchor box approach used in previous YOLO iterations (YOLOv5 and earlier). This allows the model to directly predict the centre of an object as an alternative to predicting an offset from a predefined anchor box. This allows for better generalisation and handling of irregularities. The anchor-free approach reduces the amount of bounding box predictions, simplifying the Non-Maximum Suppression (NMS) process, as there are fewer detections to filter through post-inference.

A number of architectural changes were introduced in YOLOV8 to create a more efficient model structure. C3, the primary building block in previous YOLO versions, is replaced with an improved version called C2f. Additionally, there is a change from a 6×6 to a 3×3 convolution in the steam (initial backbone layers). This improved design allows for the concatenation of bottleneck outputs as opposed to previous architectures, which only used outputs from the last bottleneck. These adjustments enable the model to capture a richer set of features, leading to better performance.

YOLOV8 caters for a diverse number of applications by providing different model sizes: nano (n), small (s), medium (m), large (l), and extra large (x). There is often a speed and accuracy tradeoff when moving from smaller to larger models, with smaller models being faster while larger models often provide better accuracy.

2.2 YOLO-NAS

YOLO-NAS (You Only Look Once Neural Architecture Search), released in May 2023, is a state-of-the-art real-time object detection model developed by Deci AI[1]. In contrast to previous versions of YOLO, which relied on human expertise

[1] https://www.supergradients.com/.

and a manual design, YOLO-NAS employs a Neural Architecture Search to automate neural network design. This is done to balance accuracy, complexity, and size.

During the development of YOLO-NAS models, AutoNAC, Deci's specialised Neural Architecture Search engine, was used to optimise the size and structures of different parts of the network architecture, including the block type, amount of blocks, and the number of channels per stage [21]. AutoNAC considered modules of the inference stack, such as compilers and quantisation, to assess 10^{14} different architecture configurations to find an 'efficiency frontier' which hosts the optimal-performing models. All YOLO-NAS models are products of this area.

A notable feature introduced by YOLO-NAS within the YOLO family is the integration of Quantisation-Aware RepVGG (QA-RepVGG) blocks into the model's architecture [4]. These specialised blocks comprise the model's 'QSP' and 'QCI' modules, supporting quantisation and reparameterisation. The QA-RepVGG blocks are designed to reduce accuracy loss even when model weights and activations are quantised to lower bit widths, enabling efficient deployment on edge devices without significant performance effects.

3 Methodology

The methodology adopted for this study as illustrated in Fig. 1 aims to provide a comprehensive assessment of the performance of different YOLO architectures in the context of fish detection. This approach focuses on the latest YOLO versions, which have been minimally explored or yet to be applied to this task. The process commences with the collection of data from both publicly available benchmark datasets and collaboration with the South African Institute for Aquatic Biodiversity[2] (SAIAB) to create our own dataset. After converting all datasets into

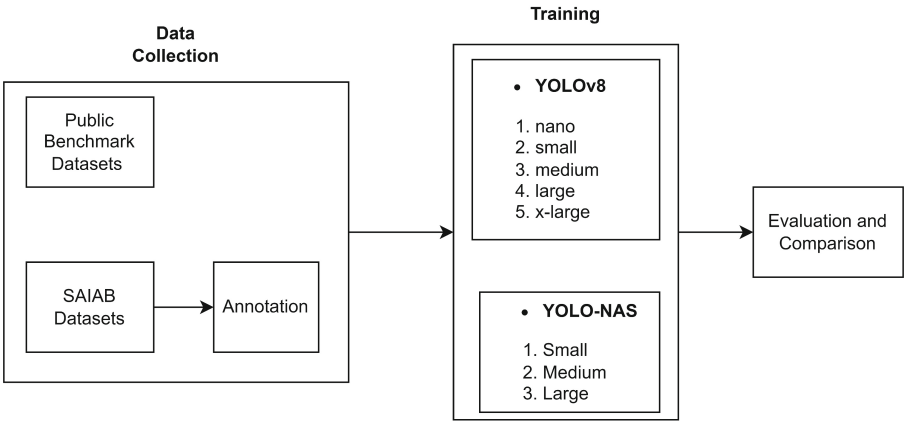


Fig. 1. An outline of the methodology used to assess YOLO models.

[2] https://saiab.ac.za/.

the YOLO format, the YOLOv8 and YOLO-NAS models are trained and evaluated. The evaluation phase involves a quantitative assessment using standard metrics.

3.1 Public Datasets

The Australian Research Data Commons Data Discoveries program developed OzFish[3], a dataset to promote machine learning research for automated fish detection [14]. OzFish is made up of 1,754 images with 43,572 bounding box annotations. This dataset is particularly challenging as it contains a large number of fish objects per image, averaging 25 fish objects per image. Additionally, many of the OzFish images feature very small fish, making detection even more difficult.

Another commonly used dataset for fish detection is DeepFish[4]. Saleh et al. [19] created the DeepFish dataset to benchmark various computer vision tasks, including classification, segmentation, and localisation. The classification subset of the dataset was converted by Al Muksit et al. [2] to make it suitable for the detection task. This resulted in 4,505 images with 15,463 fish objects from 20 different underwater environments. The DeepFish dataset averages 4 fish objects per image, making it less challenging than OzFish.

3.2 SAIAB Datasets

This study utilises two datasets obtained from SAIAB, with videos captured in the regions of Protea Banks and Tanganyika. These datasets were annotated using the Roboflow[5] annotation tool. During the annotation process, each instance of fish was labelled by drawing a bounding box around it and assigning a corresponding fish label. The novel application of these datasets for fish detection presents an opportunity to gain new insights, which may have been possible to observe when utilising benchmark datasets.

The Tanganyika dataset features diverse fish species from Lake Tanganyika in Zambia. This dataset includes 5,444 images, each annotated to make up a total of 18,789 fish instances. The Tanganyika dataset offers a unique setting for fish detection as it is the only freshwater dataset considered in this study. This dataset could provide valuable insights into how YOLO models perform in detecting fish within freshwater ecosystems.

The Protea Banks dataset was captured from a submerged reef in KwaZulu-Natal, South Africa. This dataset includes 2,059 images with 20,472 fish annotations of the same species, averaging ten fish objects per image. The species homogeneity provides an opportunity to assess the performance of the YOLO models in a controlled environment where species variation does not influence performance. Insights gained from experiments with this dataset could help to understand how model performance might vary when dealing with single-species versus multi-species datasets.

[3] https://github.com/open-AIMS/ozfish.
[4] https://alzayats.github.io/DeepFish/.
[5] https://roboflow.com.

3.3 Training and Evaluation

The study trains and evaluates all YOLOv8 and YOLO-NAS models across all datasets using the same structured approach. YOLOv8 models are evaluated in five size variations: nano (n), small (s), medium (m), large (l), and extra-large (xl), while YOLO-NAS models are assessed in three sizes: small (s), medium (m), and large (l). Each dataset is split into training, validation, and test sets using a 70:20:10 ratio. Subsequently, data augmentation is applied to the training data using the default augmentations provided by each YOLO implementation. Following augmentation, each model-dataset combination is trained for a fixed number of 300 epochs, ensuring a comprehensive training period to learn the features in the datasets. The models are continuously validated during training to monitor performance and adjust hyperparameters if necessary. After completing the training phase, the best-performing model is selected for further evaluation on an unseen test set. A limitation of the study is that YOLOv8 and YOLO-NAS do not provide identical sets of parameters for tuning. While efforts were made to ensure fair and comparable evaluation, differences in available hyperparameters between the two architectures may have affected their performance. All the models were trained and evaluated on the same computer with the following hardware:

- **CPU:** Intel Core i7-13700 (13th Gen), 16-Core, 3.4 GHz Base, 5.2 GHz Boost
- **GPU:** NVIDIA RTX A6000 48 GB VRAM
- **RAM:** 128 GB

4 Results and Discussion

This section provides a quantitative and qualitative analysis of the performance of YOLOv8 and YOLO-NAS models on various underwater datasets. The evaluation includes multiple size variants of each model to determine their effectiveness and robustness across different aquatic environments. The quantitative analysis examines precision, recall, and mean Average Precision (mAP) values achieved by each size variant of YOLOv8 and YOLO-NAS at the 50% Intersection over Union (IoU) threshold. The qualitative analysis compares the best-performing size variant of YOLOv8 and YOLO-NAS on each dataset to provide insight into the specific strengths and weaknesses of each model in challenging underwater scenarios.

4.1 Model Performance on Benchmark Datasets

The YOLO models were trained separately on the benchmark datasets to determine how they compare to previous studies that utilised the same datasets. The results achieved by each model on the Deepfish dataset are shown in Table 1 below.

Table 1. Performance results on Deepfish

Model	Precision (%)	Recall (%)	mAP (%)
YOLOv8n	94.29	93.58	96.96
YOLOv8s	**94.59**	**94.80**	**97.74**
YOLOv8m	95.24	94.54	97.59
YOLOv8l	95.38	93.81	97.47
YOLOv8x	95.22	94.29	97.46
YOLO-NASs	81.89	92.96	93.53
YOLO-NASm	**81.48**	**93.81**	**94.21**
YOLO-NASl	79.99	93.92	93.98
YOLOv3	94	94	96.01
YOLO-Fish-1	95	93	96.15
YOLO-Fish-2	95	93	96.75

YOLO-NAS models performed below the baseline set by Al Muksit et al. [2] (YOLO-Fish-1 and YOLO-Fish-2) on DeepFish, while YOLOv8 models surpassed this baseline. This suggests that the neural architecture search of YOLO-NAS may have been less effective for this specific dataset than the general improvements in YOLOv8. This is further supported by the difference of over 10% in precision results between the YOLOv8 and YOLO-NAS models, indicating that YOLO-NAS struggled with the Deepfish dataset, producing more false positives.

Interestingly, increasing the model size had little effect on performance for the Deepfish dataset, as indicated by the minimal variations in precision, recall, and mAP across different YOLOv8 and YOLO-NAS model sizes. This suggests that for datasets similar to Deepfish, utilising larger models with more parameters does not significantly enhance performance. Therefore, smaller, more efficient models may be preferable for such datasets.

A qualitative analysis was conducted on the best-performing YOLO-NAS (Medium) and YOLOv8 (Small) models. Figure 2 illustrates examples of each model's detections. The red bounding boxes represent ground truth labels, and the blue boxes represent model predictions. Detections were considered valid only if the model's confidence exceeded 0.5.

There were notable differences in detection performance between YOLO-NAS and YOLOv8, reinforcing the quantitative findings. Figure 2a demonstrates a reoccurring error of YOLO-NAS, where it incorrectly misidentifies a portion of a fish as a separate individual, while YOLOv8 accurately represents the fish as a single object. In Fig. 2b, YOLO-NAS struggles with the complex background, misinterpreting parts of the boulder as multiple fish, whereas YOLOv8 effectively handles the background. Another example of YOLO-NAS producing a false positive is provided in Fig. 2e, where a water bubble is detected as a fish. In Fig. 2c, YOLO-NAS fails to detect a camouflaged fish, a challenge YOLOv8 overcomes in this instance. YOLO-NAS generally struggles with identifying smaller fish in

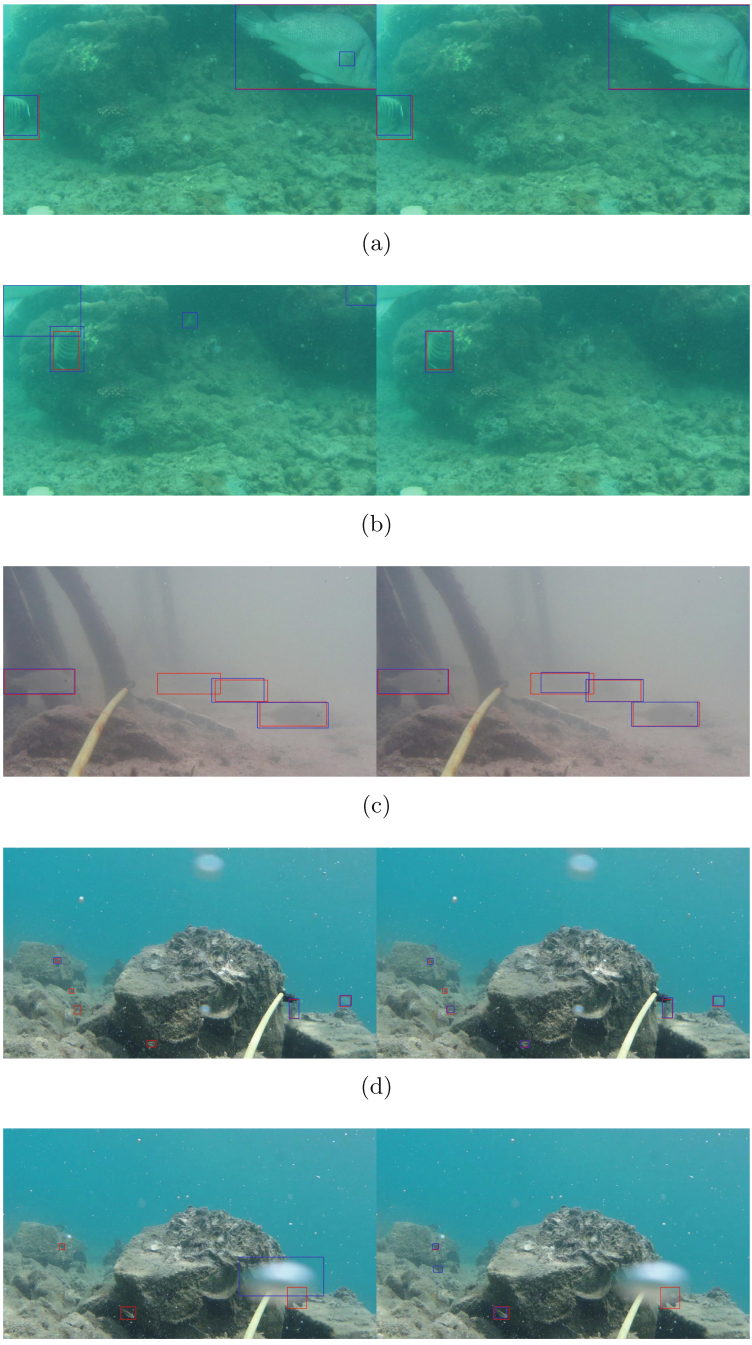


Fig. 2. Comparison of YOLO-NAS (left) and YOLOv8 (right) predictions on the DeepFish dataset, highlighting key differences in performance. Ground truth predictions are shown in red, and model predictions are in blue. (Color figure online)

the DeepFish dataset, this is demonstrated in Figs. 2d and 2e, where YOLONAS fails to identify multiple instances of fish, while YOLOv8 fails to detect only one fish. These findings emphasise YOLOv8's superior ability to handle diverse and challenging conditions, suggesting it is more effective than YOLO-NAS for this dataset.

The results achieved by each model on the OzFish dataset are summarised in Table 2 below.

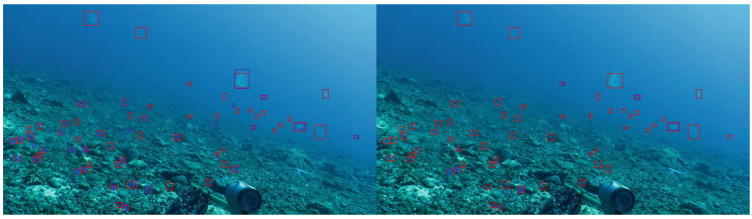
Table 2. Performance results on Ozfish

Model	Precision (%)	Recall (%)	MAP (%)
YOLOv8n	81.97	64.78	72.68
YOLOv8s	84.26	63.49	73.71
YOLOv8m	81.45	66.96	74.43
YOLOv8l	83.85	65.57	74.66
YOLOv8x	82.23	64.38	73.65
YOLO-NASs	85.66	61.79	72.57
YOLO-NASm	84.93	62.37	71.86
YOLO-NASl	86.60	62.98	72.61
YOLOv3	80	65	69.13
YOLO-Fish-1	82	64	69.9
YOLO-Fish-2	83	64	70.34

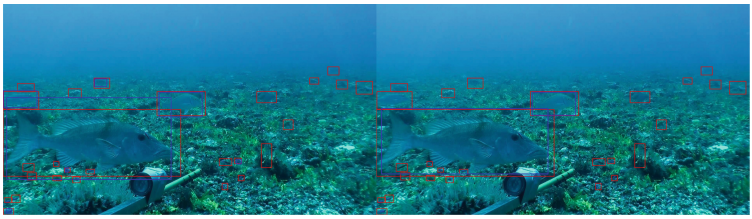
All YOLOv8 and YOLO-NAS variants surpassed the performance baseline set by Al Muksit et al. [2] on OzFish, particularly regarding precision and mAP. This demonstrates the significant advancements in the YOLO architecture since YOLOv3.

Models trained on the OzFish dataset showed a trend of increased performance with larger YOLOv8 models, as expected. However, YOLO-NAS models did not follow the same trend. This discrepancy further emphasises how YOLO-NAS and YOLOv8 models perform when trained on benchmark underwater datasets, with YOLOv8 variants demonstrating superior performance.

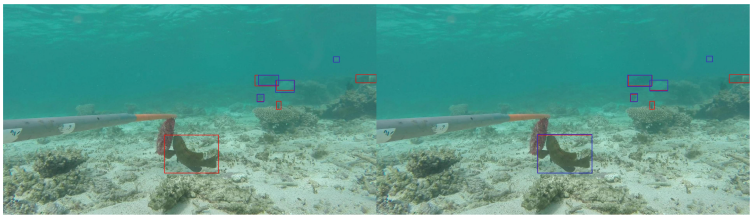
A qualitative analysis was performed using the best-performing YOLO-NAS (Large) and YOLOv8 (Large) model sizes from each architecture to complement the quantitative results. Figure 3 contains the examples considered. As shown in Figs. 3a and 3b, both YOLO-NAS and YOLOv8 struggle to detect smaller and blurry fish in the background. However, YOLO-NAS demonstrates a slight advantage by identifying some of these smaller and background fish. This suggests that while YOLOv8 may excel in clear and prominent detections, YOLO-NAS can sometimes capture finer details, potentially making it more robust in cluttered or complex scenes. In Figs. 3c and 3e, a different trend is observed. YOLO-NAS fails to detect clearly visible fish, which YOLOv8 successfully identifies. This discrepancy may be attributed to the representation of fish species



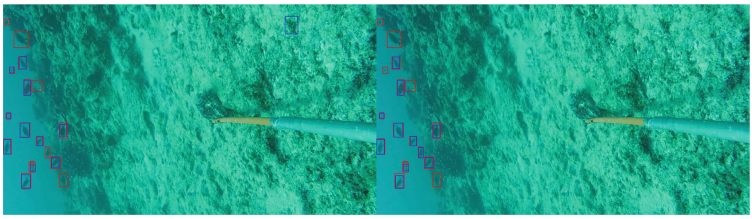
(a)



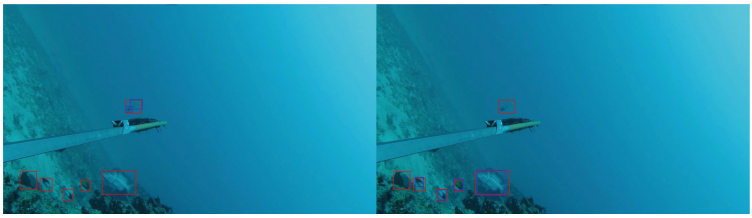
(b)



(c)



(d)



(e)

Fig. 3. Comparison of YOLO-NAS (left) and YOLOv8 (right) predictions on the OzFish dataset, highlighting differences in detection performance. Ground truth predictions are shown in red, and model predictions are in blue. (Color figure online)

in the dataset. YOLO-NAS struggles with underrepresented species, whereas YOLOv8 can generalise better, even with fewer examples. Figure 3d illustrates a recurring challenge for YOLO-NAS: misdetection of the background as fish. This issue aligns with observations from the DeepFish dataset, indicating a consistent challenge for the YOLO-NAS and further emphasising YOLOv8's robustness to such scenarios.

In general, the YOLO models performed better on the Deepfish dataset than on the OzFish dataset. This discrepancy is likely caused by the different characteristics between the two datasets. While OzFish contains a more significant number of fish instances to learn from, it includes many small and blurry fish objects, making it challenging for the models to detect these objects accurately. In contrast, the Deepfish dataset has clearer and larger fish instances, contributing to better model performance. The performance difference between the two datasets highlights the importance of data quality in fish detection tasks.

4.2 Model Performance on SAIAB Datasets

To assess their performance in different underwater environments, the YOLO models were evaluated on two SAIAB datasets: Tanganyika and Protea Banks. Table 3 below summarises the results achieved by each model on the Tanganyika dataset.

Table 3. Performance results on Tanganyika dataset

Model	Precision (%)	Recall (%)	MAP (%)
YOLOv8n	89.45	91.40	92.50
YOLOv8s	94.03	96.00	96.36
YOLOv8m	96.85	95.96	97.25
YOLOv8l	97.71	96.12	97.89
YOLOv8x	97.14	95.82	97.57
YOLO-NASs	69.89	86.35	81.47
YOLO-NASm	69.73	84.79	81.30
YOLO-NASl	64.16	87.51	80.55

Similar to the results observed on the benchmark datasets, YOLOv8 models demonstrated superior performance on the Tanganyika dataset. However, the performance gap between YOLOv8 and YOLO-NAS models was much wider on this dataset. This suggests that the freshwater environment of Lake Tanganyika presents unique challenges that the YOLOv8 architecture addresses more effectively. Furthermore, YOLOv8 consistently resulted in high performance metric values, with YOLOv8l achieving the highest mAP, reinforcing that the YOLOv8 architecture offers more robust performance across diverse underwater environments than YOLO-NAS.

While there was a significant improvement between the YOLOv8n model and the YOLOv8s model on the Tanganyika dataset, increasing the YOLOv8 model size and beyond a certain point mostly yielded minimal performance improvements. This observation is consistent with the results on the Deepfish dataset, suggesting that, at this point, the models are limited by dataset characteristics rather than model size. On the other hand, no trend was observed for the YOLO-NAS models; in some cases, larger models reduced performance. This further supports the idea that the architecture optimisations in YOLO-NAS might be less effective for underwater fish detection tasks.

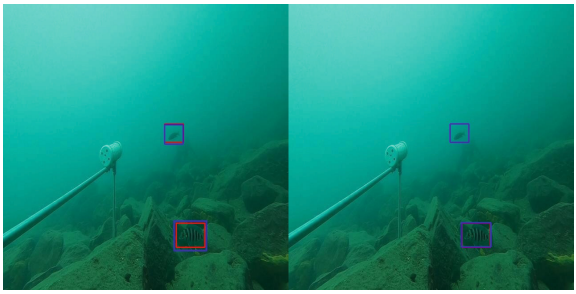
A qualitative analysis was performed using the best-performing YOLO-NAS (Small) and YOLOv8 (Large) model sizes for each architecture on the Tanganyika dataset to understand the quantitative results further. This analysis is visualised in Fig. 4, where predictions from YOLO-NAS (left) and YOLOv8 (right) are compared against the ground truth. In Fig. 4a, YOLOv8 misses a fish further in the background. This type of false negative is common among YOLOv8 predictions and mirrors similar observations made on the OzFish dataset. The consistent difficulty in detecting fish in the background suggests a limitation in YOLOv8's ability to generalise to different depth cues in underwater environments. Figures 4b, 4c, and 4d illustrate the tendency of YOLO-NAS models to identify non-fish objects as fish incorrectly. Specifically, the Baited Remote Underwater Video (BRUV) system is consistently misidentified as a fish across multiple images. This consistent pattern of misidentification significantly contributes to the lower precision observed in YOLO-NAS models, as indicated in the quantitative results.

Table 4 summarises the results for each model on the Protea Banks dataset.

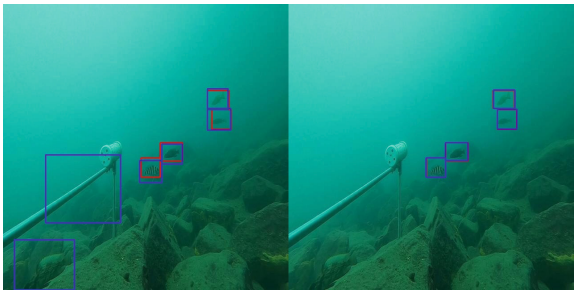
Table 4. Performance results on Richards Bay dataset

Model	Precision (%)	Recall (%)	MAP (%)
YOLOv8n	95.37	94.14	96.97
YOLOv8s	95.88	96.39	97.66
YOLOv8m	95.93	95.76	97.56
YOLOv8l	96.52	95.39	97.70
YOLOv8x	96.33	95.22	97.70
YOLO-NASs	91.41	90.44	94.21
YOLO-NASm	82.38	92.23	94.09
YOLO-NASl	90.35	92.14	94.98

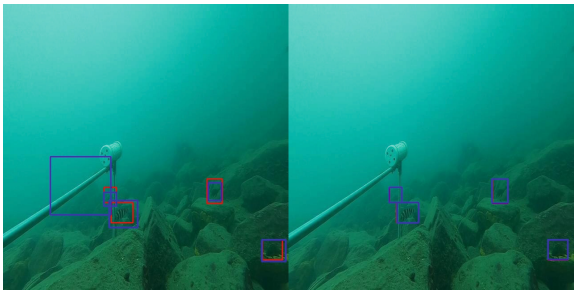
YOLOv8 models once again demonstrated superior performance compared to the YOLO-NAS models. While YOLO-NAS models performed reasonably well, they did not match the precision, recall, and mAP achieved by the YOLOv8 models. This consistency in performance further supports the robustness of the YOLOv8 architecture in varying underwater environments.



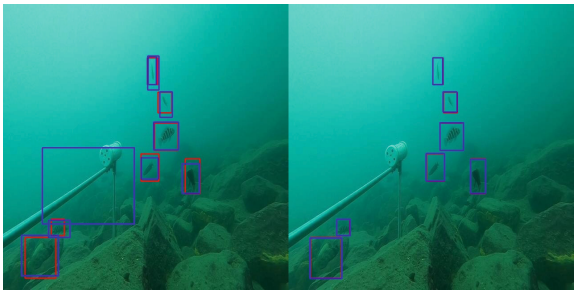
(a)



(b)



(c)

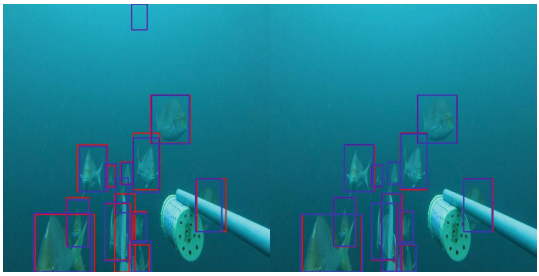


(d)

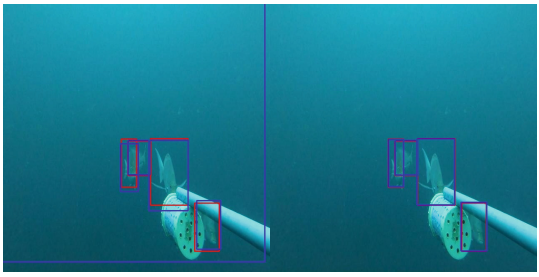
Fig. 4. Comparison of YOLO-NAS (left) and YOLOv8 (right) predictions on the Tanganyika dataset, highlighting differences in detection performance. Ground truth predictions are shown in red, and model predictions are in blue. (Color figure online)

Similar to the Tanganyika dataset results, increasing the model size in the YOLOv8 series provided little improvement, reinforcing the idea that dataset characteristics significantly limit model performance beyond a certain point.

The qualitative analysis for the Protea Banks dataset was conducted using the best-performing model sizes from both the YOLOv8 (large) and YOLO-NAS (large) architectures. Figure 5 provides visual examples of predictions versus ground truth for both models, with YOLO-NAS results on the left and YOLOv8 results on the right. This figure highlights several qualitative differences between the models. Figures 5a, 5b, and 5c demonstrate instances where YOLO-NAS mistakenly identifies background elements as fish. This type of error



(a) Figure 1a



(b) Figure 1b



(c) Figure 1c

Fig. 5. Comparison of YOLO-NAS (left) and YOLOv8 (right) predictions on the Richards Bay dataset, highlighting differences in detection performance. Ground truth predictions are shown in red, and model predictions are in blue. (Color figure online)

was observed across all the other datasets. In Fig. 5a, YOLO-NAS incorrectly detects noise in the background as a fish. In Figs. 5b and 5c, the model mistakenly identifies a large portion of the image as a fish. Figure 5c also shows a specific misidentification by YOLO-NAS, where it detects a water bubble as a fish. This issue was similarly observed in the DeepFish dataset, indicating a pattern in the model's behaviour across different environments.

5 Conclusion

The preservation of marine biodiversity is a global concern, with numerous fish species facing the threat of extinction due to environmental challenges such as overfishing, climate change, and pollution. This necessitates the assessment of state-of-the-art methods to monitor and protect fish populations. Automated fish species detection is crucial in this context to provide accurate and efficient fish identification without the time and cost of manual appraisal by an expert.

This study compared the performance of YOLOv8 and YOLO-NAS models for fish detection in underwater environments. YOLOv8 consistently outperformed its predecessors and the YOLO-NAS models, indicating substantial improvements in the YOLO architecture's ability to handle diverse underwater environments, effectively managing clear and challenging image conditions. This performance gap supports the conclusion that YOLOv8 is better suited for the fish detection task than YOLO-NAS.

While this study has provided valuable insights into the performance of YOLOv8 and YOLO-NAS for automated fish detection, several avenues for future research remain. Investigating the impact of various data augmentation techniques on model performance could be beneficial in enhancing the robustness of detection models in challenging underwater environments. Additionally, exploring the use of combined datasets from multiple underwater environments spanning different aquatic ecosystems, such as freshwater and seawater, could lead to the development of more generalised models capable of performing well across diverse conditions. Future research could also examine ensemble methods, combining multiple models to leverage their strengths, potentially leading to more accurate and reliable fish detection systems. These directions offer promising opportunities to improve automated fish detection systems.

References

1. Action, S.: World fisheries and aquaculture. Food Agric. Organ. **2020**, 1–244 (2020)
2. Al Muksit, A., Hasan, F., Emon, M.F.H.B., Haque, M.R., Anwary, A.R., Shatabda, S.: YOLO-fish: a robust fish detection model to detect fish in realistic underwater environment. Ecol. Inform. **72**, 101847 (2022)
3. Ben Tamou, A., Benzinou, A., Nasreddine, K.: Live fish species classification in underwater images by using convolutional neural networks based on incremental learning with knowledge distillation loss. Mach. Learn. Knowl. Extraction **4**(3), 753–767 (2022)

4. Casas, E., Ramos, L., Bendek, E., Rivas-Echeverría, F.: Assessing the effectiveness of YOLO architectures for smoke and wildfire detection. IEEE Access (2023)
5. Fischer, J.: Fish identification tools for biodiversity and fisheries assessments review and guidance for decision-makers. Food and Agriculture Organization of the United Nations (2013). https://doi.org/10.13140/RG.2.2.12490.93125
6. Gašparović, B., Mauša, G., Rukavina, J., Lerga, J.: Evaluating YOLOv5, YOLOv6, YOLOv7, and YOLOv8 in underwater environment: is there real improvement? In: 2023 8th International Conference on Smart and Sustainable Technologies (SpliTech), pp. 1–4. IEEE (2023)
7. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
8. Heithaus, M.R., Dill, L.M.: Food availability and tiger shark predation risk influence bottlenose dolphin habitat use. Ecology **83**(2), 480–491 (2002)
9. Jalal, A., Salman, A., Mian, A., Shortis, M., Shafait, F.: Fish detection and species classification in underwater environments using deep learning with temporal information. Eco. Inform. **57**, 101088 (2020)
10. Jocher, G., Chaurasia, A., Qiu, J.: YOLO by ultralytics (2023). https://github.com/ultralytics/ultralytics
11. Knausgård, K.M., et al.: Temperate fish detection and classification: a deep learning based approach. Appl. Intell. 1–14 (2022)
12. Labao, A.B., Naval, P.C., Jr.: Cascaded deep network systems with linked ensemble components for underwater fish detection in the wild. Eco. Inform. **52**, 103–121 (2019)
13. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
14. Australian Institute of Marine Science (AIMS), U.o.W.A.U., University, C.: Ozfish dataset - machine learning dataset for baited remote underwater video stations (2019). https://doi.org/10.25845/5e28f062c5097
15. Ouis, M.Y., Akhloufi, M.: YOLO-based fish detection in underwater environments. Environ. Sci. Proc. **29**(1), 44 (2023)
16. Qin, H., Li, X., Liang, J., Peng, Y., Zhang, C.: Deepfish: accurate underwater live fish recognition with a deep architecture. Neurocomputing **187**, 49–58 (2016)
17. Raza, K., Song, H.: Fast and accurate fish detection design with improved YOLO-v3 model and transfer learning. Int. J. Adv. Comput. Sci. Appl. **11**(2) (2020)
18. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
19. Saleh, A., Laradji, I.H., Konovalov, D.A., Bradley, M., Vazquez, D., Sheaves, M.: A realistic fish-habitat dataset to evaluate algorithms for underwater visual analysis. Sci. Rep. **10**(1), 14671 (2020). https://doi.org/10.1038/s41598-020-71639-x
20. Salman, A., et al.: Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system. ICES J. Mar. Sci. **77**(4), 1295–1307 (2020)
21. Terven, J., Córdova-Esparza, D.M., Romero-González, J.A.: A comprehensive review of YOLO architectures in computer vision: from YOLOv1 to YOLOv8 and YOLO-NAS. Mach. Learn. Knowl. Extraction **5**(4), 1680–1716 (2023)

22. Wang, M., Liu, M., Zhang, F., Lei, G., Guo, J., Wang, L.: Fast classification and detection of fish images with YOLOv2. In: 2018 OCEANS-MTS/IEEE Kobe Techno-Oceans (OTO), pp. 1–4. IEEE (2018)
23. Zhang, M., Xu, S., Song, W., He, Q., Wei, Q.: Lightweight underwater object detection based on YOLO v4 and multi-scale attentional feature fusion. Remote Sens. **13**(22), 4706 (2021)

Socio-Technical and Human-Centred AI (Information Systems)

The Effect of Generative Artificial Intelligence on Cognitive Thinking Skills in Higher Education Institutions: A Systematic Literature Review

Harry Moongela, Machdel Matthee, Marita Turpin(✉), and Alta van der Merwe

Department of Informatics, University of Pretoria, Pretoria 0001, South Africa
marita.turpin@up.ac.za

Abstract. Generative Artificial Intelligence (GenAI) tools have gained popularity in Higher Education Institutions (HEIs), and tools such as ChatGPT have been reported to impact students' cognitive thinking skills. This study addresses a gap in the literature regarding the effect of GenAI on the cognitive thinking skills of students and scholars in HEIs. A Systematic Literature Review (SLR) was conducted that yielded relevant research articles, on which a thematic analysis was performed. Cognitive thinking skills identified in the articles include critical thinking, creative thinking, reflective thinking, computational thinking and problem-solving. The study summarised teaching interventions reported in the various studies, which aimed to engage with and enhance cognitive thinking skills. The findings revealed positive and negative impacts of GenAI on cognitive thinking skills. The findings indicated that GenAI such as ChatGPT can help improve, boost, and develop cognitive thinking skills and enhance learning and the overall higher education experience under well-planned guidance. The study subsequently provides guidelines on how students and scholars can use GenAI to improve cognitive thinking skills in HEIs. HEIs are encouraged to create special programs to teach students how to correctly use GenAI tools like ChatGPT. It is worth noting that students need to use their knowledge, thinking skills, and good judgment to use GenAI properly.

Keywords: Cognitive Thinking · Critical Thinking · Creative Thinking · Reflective Thinking · Computational Thinking · Problem Solving · Generative AI · Higher Education Institutions

1 Introduction

Generative Artificial Intelligence (GenAI) is a technology that can generate human-like responses and use deep learning models to answer complex queries [1, 2]. GenAI has been reported to be revolutionising the field of Higher Education Institutions (HEIs) and has become popular, especially with the introduction of Chat Generative Pre-Trained Transformer (ChatGPT) in HEIs since its release in November 2022 [1, 3, 4]. ChatGPT

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 355–371, 2025.
https://doi.org/10.1007/978-3-031-78255-8_21

is designed using deep learning techniques to generate human-like text based on the input of requests received from users, and it is initially pre-trained using a large database of text from the Internet [3]. GenAI applications such as ChatGPT are also known to be commonly adapted and used in HEIs [1, 3, 4].

Like many technologies, GenAI has been reported to have numerous benefits and also, shortcomings in teaching and learning in the HEIs [1, 3–5]. The benefits of GenAI involve using AI to support teaching and learning, facilitating cognitive thinking skills such as critical thinking, facilitating idea generation, creating lesson plans and creating openness in education settings [2, 3, 6]. Shortcomings and concerns of GenAI include the rise of plagiarism, compromise of academic integrity, and the possible negative impact that GenAI has on students' cognitive thinking skills [1, 3, 6]. At the end of the day, like many other AI systems, the effectiveness of GenAI, and in this case ChatGPT, depends on how it is applied and used in different ways [3].

According to the World Economic Forum Future of Jobs report, the ability to think critically and solve problems is becoming the most sought-after skill in today's workplace environment, with cognitive skills seeing the fastest growth in importance [4, 7]. Therefore, fostering critical thinking and problem-solving in higher education seems important so students can apply these skills in the workplace. Businesses further reveal that coming up with new ideas (which is defined as creative thinking), is becoming more important than figuring things out (which is defined as analytical thinking) [4, 5, 7]. Companies also report prioritising AI and big data, which are currently trending and deemed important to the workforce [7]. As of 2023, two cognitive skills are considered the top core skills in companies: analytical thinking in first place and creative thinking in second place [7]. This shows that cognitive skills play an important role in the workplace and hence why thinking skills need to be fostered in HEI.

2 Background and Significance

Researchers have highlighted the need to carefully create clear frameworks, guidelines, and policies to integrate GenAI in HEIs [1, 4, 5]. It is also important to understand how users perceive and experience using GenAI in HEIs [1]. This study will focus on ChatGPT, a chatbot that GenAI drives. The use of ChatGPT has been gaining traction in the teaching and learning space and has been reported to have positive and negative uses in HEIs [1, 3, 5]. The positive uses of ChatGPT are that it can help translate languages, generate conversations, complete texts and draw up sample documents [1]. The negative use of ChatGPT is that it can used to write and answer questions from students' assignments, tests and exams; ChatGPT can also be used to write essays, research articles, and conduct academic research without always being detected sometimes by plagiarism checkers [1, 3].

Since it seems GenAI is here to stay, HEIs have inevitably decided to embrace the use of GenAI by providing students and scholars with information on how to use GenAI responsibly, ethically, effectively and efficiently to benefit the users and all stakeholders involved [1–3, 6]. Integrating technology, such as GenAI, into educational settings, such as HEIs, can provide resourceful and meaningful ways to achieve the best learning outcomes [3, 5, 6].

As of late, most of the GenAI body of literature has highlighted and covered the positives and negatives of GenAI and shed some light on how AI can stimulate cognitive thinking among students [1, 2]. Moreover, synthesizing and analysing the existing body of research studies is imperative in order to explore and understand the effect of GenAI on the cognitive thinking skills of students and scholars in HEIs. Cognitive skills, in this case, refer to presenting a problem using words, coming up with precise conceptual ideas, setting goals, developing strategies, suggesting a course of action to a colleague, reporting on one's actions, tracking progress toward the objective, and keeping an eye on the presence of teammates [8, 9]. This topic is important because the effect of GenAI on cognitive skills has been questioned regarding the positive and negative effects, the AI data accuracy, limitations, biases, reliability, validity and integrity [1, 3, 6, 8, 9]. The concern about using GenAI is that it can negatively influence students' creativity, problem-solving, and critical thinking skills [1, 2].

Since students and academics are the primary actors that use GenAI tools in HEI, it is important to explore and understand the effect of the use of GenAI on the cognitive thinking skills of students and scholars in HEIs [2]. Therefore, this study aims to fill the knowledge gap that exists in the field of GenAI and cognitive thinking, particularly in assessing and analysing the impact that the use of GenAI has on the cognitive thinking skills of students and scholars in HEIs; the study also aims to contribute to the discussion and body of knowledge of GenAI and cognitive thinking skills.

The structure of this paper is as follows: Sect. 3 highlights the research method, research objective and research questions. Section 4 highlights the findings; Sect. 5 highlights the discussion and Sect. 6 presents the conclusion.

3 Research Method

3.1 Research Objective

This paper aims to examine the effect of GenAI on the cognitive thinking skills of students and scholars in HEIs. A Systematic Literature Review (SLR) was carried out, which involves identifying and assessing the relevant primary articles in the existing literature that might address the research question [10, 11]. This section presents the research question that guides the study, the search terms used, the selection criteria and the source selection.

3.2 Research Question

In recent years, integrating GenAI into HEIs has sparked considerable interest and debate. While GenAI offers diverse functionalities ranging from language translation to document generation, its impact on cognitive skills, particularly critical thinking, remains a topic of significant inquiry. This study seeks to address the pivotal question: *What effect does the use of Generative AI have on the cognitive thinking skills of students and scholars in HEIs?* Despite its potential to enhance productivity and efficiency, concerns have been raised about how GenAI might influence creativity, problem-solving abilities, and the overall quality of cognitive engagement among users. By exploring this question,

the research aims to contribute valuable insights into the nuanced relationship between GenAI tools and the cognitive thinking skills essential for academic and professional success in educational settings.

3.3 Search Terms

The selection of search terms was motivated by the study's focus on understanding how GenAI impacts critical thinking skills among students and scholars in HEIs. Terms such as "Critical thinking," "Problem-solving," and "cognitive thinking" were chosen to explore the multifaceted effects of AI technologies like GenAI and large language models on cognitive abilities essential for academic success. By incorporating specific AI-related terms like "ChatGPT," the study aims to comprehensively examine the potential benefits and challenges these technologies pose in educational contexts.

The search was conducted in June-July 2024, by the first author and assisted by the other authors. The specific search terms used are:

("Critical thinking" OR "Problem solving" OR "analytical thinking" OR "thinking skills" OR "analytical skills" OR "cognitive thinking" OR "reasoning skills") AND ("Generative AI" OR "large language models" OR ChatGPT) AND ("Higher Education Institutions" OR Universities).

3.4 Selection Criteria

Inclusion Criteria. Studies that adhered to the following criteria: (1) studies published between 2020 and 2024 to ensure the most up-to-date research was evaluated since the field of AI is rapidly evolving; (2) studies published in English; (3) articles with the main focus on GenAI and cognitive skills; (4) articles that are published in IT journals or any technology related journal articles as well as conference proceedings.

Exclusion Criteria. Studies that adhered to the following criteria: (1) studies not published between 2020 and 2024; (2) studies not presented in journal articles (e.g. in the form of a slide show or overhead presentation); (3) studies not relating to technology or IT; (4) articles whose main focus is not GenAI and cognitive skills; (5) articles with only the abstract but not the full text available; (6) duplicate papers (same paper retrieved from different databases); (7) duplicate report of the same study (when several reports of a study exist in different databases, the complete version of the study was included in the review).

3.5 Source Selection

This study employed a comprehensive approach by utilising prominent sources and databases such as Scopus, SpringerLink, Science Direct, and Web of Science. These platforms were selected to ensure a thorough exploration of scholarly literature on the impact of GenAI on critical thinking skills within HEIs. The study aimed to gather robust evidence and insights into the evolving relationship between AI technologies and cognitive abilities in educational settings by accessing a diverse range of peer-reviewed articles from these databases.

3.6 Flowchart of Search Process

The PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) flow chart [11] (Fig. 1) shows the whole search process of identifying the relevant articles, from the chosen databases, the screening process, the selection process and the process of including and excluding articles, using PRISMA reporting guidelines.

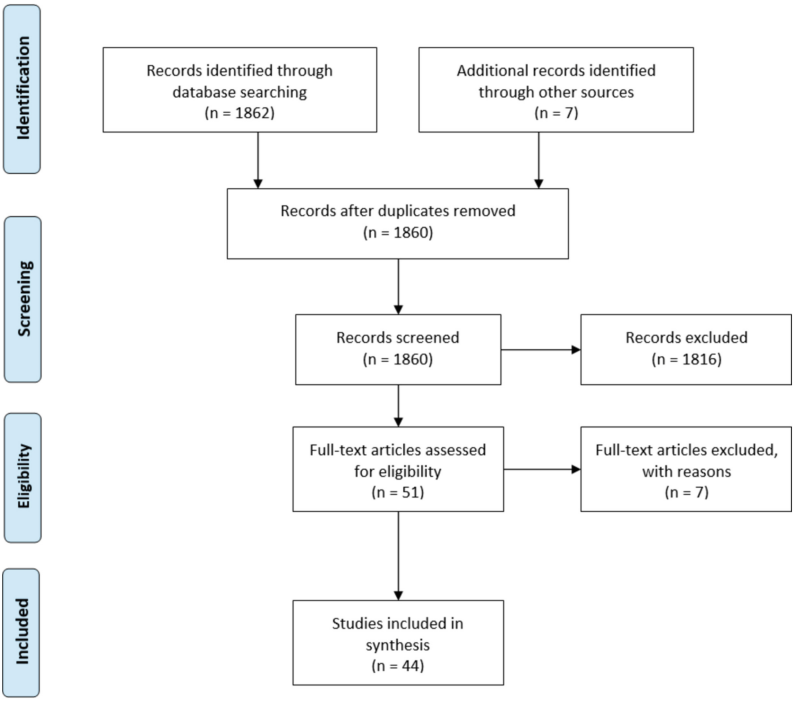


Fig. 1. PRISMA Flow Chart

After the 44 relevant studies were identified, a quality assessment was carried out to examine the quality of each study. The quality assessment criteria involved including studies that could answer the research questions and were relevant to the study. Even though all 44 articles mentioned some key search terms, only 18 articles were included in the review since they mentioned all the key and relevant themes needed to answer the research question. The other 26 articles did not highlight precisely the impact of the use of GenAI on the cognitive thinking skills of students and scholars in HEIs.

4 Findings

The PRISMA Flow Chart facilitated the transparent selection and inclusion of pertinent articles, supporting a rigorous analysis of insights and findings to advance understanding in this critical area of research [11]. We discuss the types of cognitive thinking skills and the effect of GenAI on these cognitive thinking skills.

The following types of thinking skills were identified from the literature:

(1) Critical thinking, (2) Creative thinking, (3) Reflective thinking, (4) Computational thinking, (5) Problem-solving.

The descriptive bar chart below (Fig. 2) shows the frequency of the types of thinking identified from the relevant articles. It is worth noting that high order thinking and cognitive thinking skills were not extensively discussed in findings since they are umbrella terms that are used interchangeably and encompass critical thinking, creative thinking, reflective thinking, computational thinking and problem-solving. Most of the articles were published from various countries; five were published in China, two from the USA and one from each country in South Africa, Ghana, France, Switzerland, Finland, Poland, Indonesia, UAE, Qatar, Taiwan and Ecuador. This indicates heterogeneity in the spread of countries where research on the topic was performed.

Although articles that met the criteria for this study were published between 2020 and 2024, it is no surprise that most of the articles considered for findings were published between 2023 and 2024.

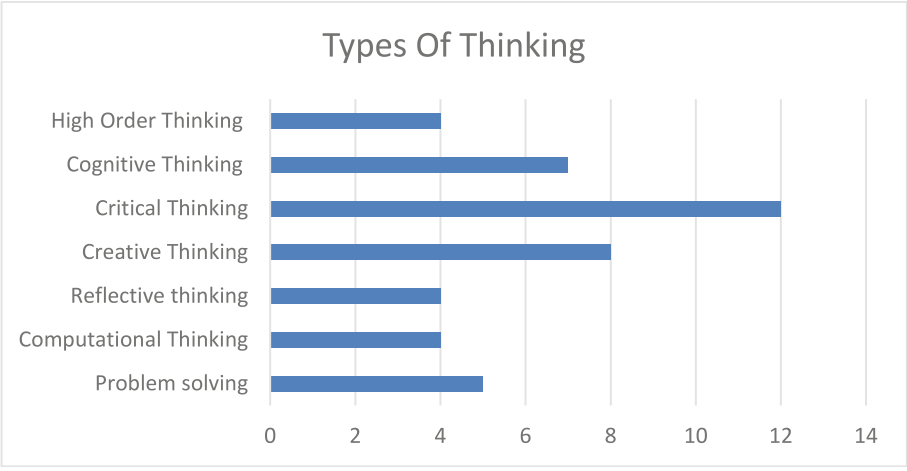


Fig. 2. Types of Thinking

4.1 Critical Thinking

Critical thinking in AI education refers to the ability to analyse information or a problem from multiple viewpoints, evaluate multiple perspectives, and formulate reasoned arguments within the context of AI-driven environments [12, 13]. Critical thinking can also be described as the process and method used to analyse and evaluate data, provide a comparative analysis, and use it for problem-solving and decision-making [9, 14].

Literature reveals that the most used AI tool in HEIs is ChatGPT [9, 15]. ChatGPT has been used to examine critical thinking in different disciplines, including social sciences, business, health, psychology, law, music, information systems, and research methodology [9, 12–15]. In this case, most of the studies focused on the impact of ChatGPT (GenAI) on critical thinking and reported various outcomes. According to lecturers and students, ChatGPT is reported to have the potential to strengthen and promote critical thinking in HEIs [4, 12, 13]. ChatGPT can return answers that can be differentiated from answers given by human beings, and students can also use the ChatGPT answers to help them with critical thinking [4, 13]. Through prompt engineering, the use of ChatGPT has led to the enrichment of students' educational experiences [12]. Essel, Vlachopoulos, Essuman and Amankwa [9] revealed that incorporating the use of ChatGPT for tasks in class positively influences the student's critical thinking and improves the development of critical thinking. The study suggested that the use of ChatGPT and AI chatbot models can be valuable in HEIs and help lecturers develop to bolster students' critical thinking skills [9].

Students reported that ChatGPT generates straightforward responses and generally accurate solutions, but they lack depth, include prejudices, have limitations and leave out some important information [15–18]. ChatGPT responses might not encourage critical thinking as students are less likely to dig deep or question things as much as they should [16, 19]. As a result, students might accept answers from ChatGPT without fact-checking and considering different perspectives [4, 16]. This could lead to students becoming less active in their learning and simply taking in information without thinking about it [16, 19].

Another limitation of using AI tools like ChatGPT for critical thinking tasks is that it could prevent students from developing their independent thinking abilities [4, 16]. The overreliance on AI tools can weaken students' critical thinking, making it harder for students to learn important lessons from their mistakes and overcome tough challenges [4, 20]. There is a risk that students could depend too heavily on AI to think and solve problems for them, which might hold back their ability to come up with creative ideas and analyse problems on their own [4, 16].

4.2 Creative Thinking

Creative thinking in higher education refers to thinking outside the box and being able to solve problems, create new ideas, create innovate outputs and develop effective new concepts [9, 21]. Most of the literature that examined creative thinking was reported to use ChatGPT as an AI tool [4, 9, 21]. The studies were conducted in HEI disciplines such as chemistry, education, technology and research methodology courses [9, 19, 21]. In a study conducted by Lee, Chen, Wang, Huang and Wu [21], the students used the Guidance-based ChatGPT-assisted Learning Aid (GCLA), which is a way of using ChatGPT by allowing students to first try and solve problems by themselves before they get help from using ChatGPT. GCLA aims to guide students by offering hints rather than giving them direct answers [21]. GCLA also aims to create an appropriate environment for enhancing Self-Regulated Learning (SRL) and higher-order thinking skills (HOTS). GCLA ultimately offers guidance for students to develop their creative thinking and improve the repetitive process of students' responses [4, 21].

GLCA has encouraged students to be more actively involved in their educational practices, leading to more effective learning experiences in HEIs [9, 21]. The use of ChatGPT in GLCA has also had a positive effect on students' cognitive skills, particularly on their creative thinking skills, when compared to the traditional use of ChatGPT [4, 19, 21]. The fact that GCLA provides students with guidance rather than giving students direct answers has positively affected students' creation of knowledge and has made their learning more efficient [21]. GCLA can help students maintain reliable support from lecturers, participate in complex academic tasks, take charge of their learning and improve their independent thinking [4, 21]. Another study by Li [19] that used ChatGPT-FLGA (ChatGPT-based Flipped Learning Guiding Approach) with students also reported that students' creative thinking was boosted with the assistance of ChatGPT. Additionally, the fact that the AI educators kept reminding students to be creative also played a role in this improvement by encouraging students to explore new ideas [19, 20].

4.3 Reflective Thinking

Reflective thinking in higher education refers to the ability to self-reflect and integrate learnt knowledge, subsequently increasing the quality and depth of learning practices [9]. A study by Essel, Vlachopoulos, Essuman and Amankwa [9] examined reflective thinking and focused on a research methodology course using a flipped classroom approach. The study reported that using ChatGPT for educational activities positively impacts students' reflective thinking and improves its development [9]. Using ChatGPT in HEIs allows lecturers to develop, support and enhance students' reflective thinking skills; this fosters an environment that improves the learning and teaching experience in HEIs [9, 22].

4.4 Computational Thinking

Computational thinking (CT) refers to a high-level ability to use basic computer science ideas to solve problems and is related to knowing how to program computers [23–26]. CT is also defined as a process, skill or model which allows students to use computers to enhance their thinking and solve problems in a similar way that computers would solve them [25, 26]. Learning to program is regarded as the best way for students to improve their CT skills, and this involves being able to automate solutions and choose the best, most effective, and most efficient solution [24, 25].

CT involves defining issues clearly, focusing on core ideas, thinking step-by-step, using set procedures, examining and putting solutions into practice, applying lessons to broader situations, and adapting approaches to new contexts [24–26]. In summary, the CT framework highlights the following important concepts: abstraction of core ideas, decomposition of problems, algorithmic thinking of the plan, evaluation of the solution, and generalisation [24, 25]. CT combines several thinking skills, such as problem-solving, critical, logical, creative, communication, and collaborative skills [24, 25]. All these skills come together in how a person approaches and solves problems.

Similar to previous studies, scholars used ChatGPT to examine how AI tools impact students' CT skills; the studies focused on academic courses such as programming, data

structures and algorithms and English as a foreign and writing language [23–26]. A study by Shanshan and Sen [23] tested out debugging using Artificial Intelligence–Generated Content (AIGC) by ChatGPT and compared it to traditional debugging carried out by students. The study revealed that AIGC generated by ChatGPT produced better CT and learning experiences than traditional debugging, which suggested that ChatGPT improves CT [23]. The use of ChatGPT to debug was also reported to encourage clear two-way communication and improve students' understanding of key ideas, leading to better learning outcomes, stronger problem-solving, and overall CT [23]. It is worth mentioning that the benefits of AIGC usually depend on the quality and accuracy of the ChatGPT-generated content [18, 23].

A study by Liao, Zhong, Zhe, Xu, Liu and Xie [24] introduced a proposed an Intelligent Programming Scaffolding System (IPSSC) using ChatGPT to students in a HEI. The IPSSC using ChatGPT was implemented with guidance from the theoretical framework of computational thinking and scaffolding [24]. The perceptions from students reveal that they had a positive attitude towards IPSSC, and the use of ChatGPT in IPPC effectively enhanced students' CT [24]. In contrast, the study reported that students' problem-solving skills decreased significantly [24].

Other studies have researched a different context and looked at the impact that CT has on AI [25, 26]. The study conducted by Celik [25] used CT and AI literacy scales to examine the impact of CT on AI; the study revealed that CT has a positive effect on and increases AI literacy, which helps with using, identifying and evaluating AI tools [25]. The logic behind it is that if students employ critical, algorithmic, and creative thinking, they are more likely to engage with AI tools [25]. Another study conducted by Tang and Ma [26] that studied the impact of CT on teaching English as a foreign language revealed that CT can enhance and assist students who speak a different language to pick up new English words more quickly and effectively. This validates that CT used in conjunction with AI plays a significant positive and inspirational role in foreign language teaching and education [26].

4.5 Problem-Solving

Problem-solving is the process of collecting, understanding and examining all the relevant information and sharing knowledge, expertise, and efforts to find the most effective and efficient solution [8, 21, 24]. As mentioned in Sect. 4.2, Lee, Chen, Wang, Huang and Wu [21] carried out a study that focused on a foundational chemistry course; the researchers were testing the impact of ChatGPT on problem-solving. Researchers set up an experiment where they split students in a chemistry class into two groups; one group used GCLA with ChatGPT, while the other group used ChatGPT in the usual way [21]. By comparing these two groups, the researchers could see if the guided ChatGPT method was better than using ChatGPT normally [21].

The GCLA method allows students to try to solve problems on their own first, and then if they need help, they use ChatGPT in a guided way. GCLA gives students hints instead of direct answers [21]. The GCLA method aims to help students learn to study on their own, improve and boost their problem-solving skills and make students' answers less repetitive [21]. The researchers found that GCLA made students more active in their learning, leading to better learning experiences in higher education, helping students create knowledge more effectively, making learning more efficient, and improving their problem-solving skills [4, 21, 24]. Overall, GCLA is considered helpful because it guides students instead of giving them answers, provides support from educators, helps students with difficult class tasks and encourages students to manage their learning [21, 24].

In contrast, a study conducted by Liao, Zhong, Zhe, Xu, Liu and Xie [24] found that when students used ChatGPT, they became too dependent on ChatGPT, which led to a decrease in their ability to solve problems on their own, instead of getting better at it. This shows that the excessive use of ChatGPT can harm and diminish students' problem-solving abilities, contrary to what might have been expected or hoped for [4, 24]. Although using ChatGPT makes it easy for students to get answers or information, this can harm their ability to figure things out independently and hamper their ability to learn from mistakes and challenges [4, 24].

4.6 Cognitive Thinking Skills Relationships

The five important types of cognitive thinking skills that were identified from the literature include (1) Critical thinking, (2) Creative thinking, (3) Reflective thinking, (4) Computational thinking, and (5) Problem-solving. The diagram below (Fig. 2) explains how each type of cognitive thinking skill relates to another kind of cognitive thinking skill and the logical relationships shared between different cognitive thinking skills.

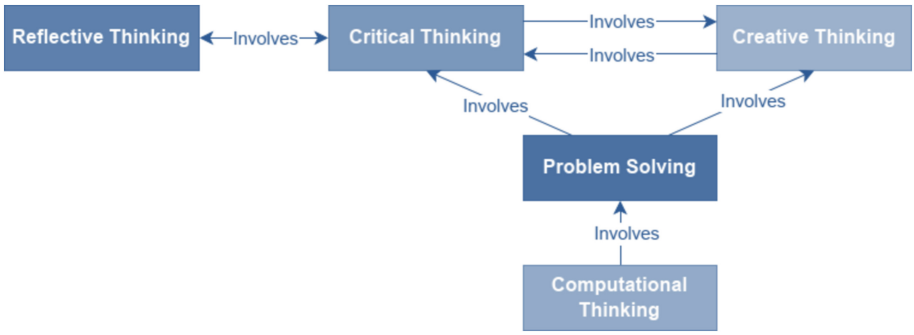


Fig. 3. Cognitive Thinking Skills Relationships

The diagram above (Fig. 3) shows the relationship between reflective, critical, creative, problem-solving, and computational thinking. Reflective thinking involves critical thinking, for instance, when an individual is given a problem to solve, they need to be able to think critically about how to solve the problem and then be able to reflect, contemplate and take a step back throughout the thinking process. Critical thinking involves creative thinking; when an individual is given a problem to solve and can come up with a new solution, they must combine critical thinking and creative thinking [15]. Problem-solving involves critical and creative thinking to come up with the best solutions; an individual may need to be creative and innovative and think outside the box to find a solution to a problem [4, 21]. Computational thinking also involves problem-solving; an individual can solve a problem without using computational thinking, while computational thinking requires problem-solving. The combination of critical thinking, creative thinking, reflective thinking, problem-solving and computational thinking forms a shared relationship, where each thinking skill relates to every other type of thinking skill and falls under the umbrella of cognitive thinking skills that are impacted when using AI in HEIs.

Table 1 below presents the summary of the cognitive thinking skills experiments.

Table 1. Cognitive Thinking Skills Experiments Summary

Effect on Cognitive Thinking Skills	Interventions	References
Critical Thinking		
Positive and Negative	• ChatGPT is used for prompt engineering • Evaluating essays created by ChatGPT • Human facilitation is needed to mark ChatGPT essays using a specific criterion, for the experimental group, for the control group and for interviews • An experimental group using ChatGPT to solve tasks and a control group using traditional databases and search engines for similar tasks • Critical Thinking Scale, Creative Thinking Scale, Reflective Thinking Scale, and student interviews (semi-structured) were conducted • Experiment involving students critiquing responses (provided by ChatGPT) to answering a specific question • The experiment analysed the level of criticality that the students demonstrated in working with ChatGPT and assessed the quality of ChatGPT outputs • ChatGPT is used to generate and produce a rubric for critical thinking and assess the students' assignments against the rubric produced • Human intervention, engagement, expertise, critical thinking and judgement are needed in working with ChatGPT	[9, 13, 15]

(*continued*)

Table 1. (*continued*)

Effect on Cognitive Thinking Skills	Interventions	References
Creative Thinking		
Positive	• An experimental group and a control group were used to examine the impact of GCLA compared to traditional ChatGPT use within a blended learning setting • Participants in the groups attempt to solve problems independently first before using ChatGPT (through GCLA) for assistance • GCLA provides hints rather than direct answers, creating an environment conducive to the development of Self-Regulated Learning (SRL) and higher-order thinking skills • ChatGPT and learning scaffolding theory used to develop a ChatGPT-based Flipped Learning Guiding Approach (ChatGPT-FLGA) based on the analysis, design, development, implementation and evaluation model • A Quasi-experiment was conducted to investigate the effectiveness of ChatGPT-FLGA • An experimental group and a control group were used and each group received flipped classroom instructions and conducted discussions through Tencent QQ applications, but only the experimental group learned with ChatGPT-FLGA • An experimental group using ChatGPT to solve tasks and a control group using traditional databases and search engines for similar tasks • Critical Thinking Scale, Creative Thinking Scale, Reflective Thinking Scale, and student interviews (semi-structured) were conducted • Human facilitation is needed for the experimental group, control group and for interviews	[9, 19, 21]
Reflective Thinking		
Positive	• An experimental group using ChatGPT to solve tasks and a control group using traditional databases and search engines for similar tasks • Critical Thinking Scale, Creative Thinking Scale, Reflective Thinking Scale, and student interviews (semi-structured) were conducted • Human facilitation needed for experimental and control groups, and as interviewees	[9]

(*continued*)

Table 1. (*continued*)

Effect on Cognitive Thinking Skills	Interventions	References
Computational Thinking		
Positive	• A debugging experiment with 3 interfaces was integrated with Artificial Intelligence–Generated Content (AIGC) using ChatGPT • Experiment conducted based on 3 levels of AIGC integration design • Analysis of covariance and a structural equation model was used to examine the effects • Human facilitation is needed for experimental groups, control groups, survey groups and interviews • An experiment was conducted to investigate the affordance of the scaffolding system using ChatGPT • A research model built based on literature and different theoretical frameworks • Data collection tools methods used include: **AI literacy scale** for evaluating AI-based technologies used in everyday life and **Computational thinking scale** to analyse perceptions in computational thinking sub-skill areas • Computational thinking increases and leads to better AI literacy, which facilitates in using AI-based technologies	[23–25]
Problem-Solving		
Positive and Negative	• An experimental group and a control group were used to examine the impact of the GCLA compared to traditional ChatGPT use within a blended learning setting • Participants in the groups attempt to solve problems independently first before using ChatGPT (through GCLA) for assistance • GCLA provides hints rather than direct answers, creating an environment conducive to the development of self-regulated learning (SRL) and higher-order thinking skills • Human facilitation is needed for experimental and control groups	[21]

5 Discussion

Considering that using GenAI tools have a positive and negative impact on students' and scholars' cognitive thinking skills in HEIs, it is imperative and useful to provide guidelines for HEIs to implement when using GenAI. HEIs play an important role in implementing the use of AI tools such as ChatGPT to improve students' cognitive thinking skills and support the process of teaching and learning [4, 14, 16, 22]. HEIs must create programs that evaluate, provide training and enhance cognitive thinking skills [4, 14, 16]. HEIs also need to develop, encourage and facilitate the proper use of ChatGPT in education [4, 14, 16]. ChatGPT and other AI tools help facilitate teaching for lecturers and learning for students. However, they still need human engagement, expertise, analysis, judgment, and critical thinking to evaluate the responses and to use the AI tool efficiently [4, 15].

The literature in the studies also reveal that ChatGPT in HEIs still requires participation from human beings, human experiences, and human analysis to effectively utilise GenAI tools [9, 13, 15]. Such an approach would be beneficial for educational disciplines such as humanities, education, and social sciences, and it has the potential to improve a wide range of courses that require cognitive thinking skills, such as research methodology classes [8, 9]. Evidence from the literature shows that if HEIs were to replicate the experiments of enhancing cognitive thinking skills, human facilitation and guidance would be needed [9, 13, 15, 19]. Educators and students will have to contribute and put in a lot of effort, time and work to effectively pull off the experiments to enhance students' cognitive thinking skills [9, 19, 21]. The most important thing is planning on how to use AI tools in HEIs; good planning is what decides whether the AI tools will be helpful or detrimental to learning in HEIs [12, 24].

Integrating GenAI like ChatGPT into education, especially to develop cognitive thinking skills, requires careful consideration [17–19]. GenAI can offer valuable information and support learning and should not replace human teaching [17, 19, 22]. To foster cognitive thinking skills, GenAI should be used as a starting point rather than the final answer [5, 19, 21]. Students should analyse GenAI-generated information, question its reliability, and conduct further research [17, 19, 22]. Educators must teach students to critically evaluate GenAI content, identify biases, and compare it to other sources [18, 19, 22]. Through modelling critical thinking and guiding discussions, educators can show students how to use GenAI effectively [18, 19]. HEIs, educators, technologists, policymakers and students must collaborate to maximise GenAI's benefits while minimising risks, particularly regarding cognitive thinking skills [5, 18, 19, 22]. Although GenAI has limitations, it can enhance cognitive thinking skills when used to support, not replace, human-driven learning and intellectual growth [17, 19, 22].

Using GenAI effectively in HEIs will require creating new social GenAI systems specifically meant for educational purposes rather than using generic GenAI [27]. These GenAI systems should be able to conversate with humans and other AI entities, contribute to online resources, and function in educational roles like educators, students, advisors, and mentors [17, 27]. These GenAI systems should be programmed to recognise their limitations, understand their obligations to students, maintain the integrity of online information, and show respect for human educators and subject matter experts [17, 27].

It is crucial to work out and know how to design and limit these educational social GenAI systems [27].

6 Conclusion

In this paper, we set out to investigate the effect of GenAI on the cognitive thinking skills of students and scholars in HEIs. A systematic literature review was performed and the findings show that using GenAI, such as ChatGPT in HEIs by students and lecturers, can help improve cognitive thinking skills such as critical thinking, creative thinking, reflective thinking, computational thinking and problem-solving skills. GenAI can boost cognitive skills as students can use answers from tools such as ChatGPT to foster and practice their critical, creative, reflective, computational and problem-solving skills. Using ChatGPT for class tasks has been noted to develop cognitive thinking skills, enhance learning and augment the overall higher education experience. GenAI shows promise for developing cognitive thinking skills in higher education, but it needs to be used carefully and in conjunction with human skills and judgment. The contribution for this study is focused on showing the effect of GenAI on the cognitive thinking skills (critical thinking, creative thinking, reflective thinking, computational thinking, problem-solving skills) of students and scholars in HEIs.

However, this study has limitations; most of the literature considered focused on the impact of ChatGPT on students' cognitive thinking skills; future studies should focus on other GenAI tools and their impact on students' cognitive thinking skills. Future research can also investigate the long-term outcomes or effects of GenAI on students' cognitive thinking skills as opposed to short-term experiments reported in this study. This study revealed five common types of cognitive thinking skills, and future studies should try to discover more types of cognitive thinking skills used in HEIs that may emerge. This study focused on four research article databases, which might have limited the number of articles found; future research should search in more databases and expand their research criteria. Another limitation of this study is that this research could be outdated and considered obsolete in the coming months since the field of GenAI is constantly being updated and is liable to change rapidly due to new research.

References

1. Michel-Villarreal, R., Vilalta-Perdomo, E., Salinas-Navarro, D.E., Thierry-Aguilera, R., Gerardou, F.S.: Challenges and opportunities of generative AI for higher education as explained by ChatGPT. Educ. Sci. **13** (2023)
2. Thiga, M.M.: Generative AI and the development of critical thinking skills. IRE J. **7**, 83–90 (2024)
3. van den Berg, G., du Plessis, E.: ChatGPT and generative AI: possibilities for its contribution to lesson planning, critical thinking and openness in teacher education. Educ. Sci. **13** (2023)
4. Abulibdeh, A., Zaidan, E., Abulibdeh, R.: Navigating the confluence of artificial intelligence and education for sustainable development in the era of industry 4.0: challenges, opportunities, and ethical dimensions. J. Clean Prod. **437**, 140527 (2024)
5. George, A.S.: Preparing students for an AI-driven world: rethinking curriculum and pedagogy in the age of artificial intelligence. Partners Univ. Innov. Res. Publ. **1**, 112–136 (2023)

6. Shanto, S.S., Ahmed, Z., Jony, A.I.: Enriching learning process with generative AI: a proposed framework to cultivate critical thinking in higher education using chat GPT. Tuijin Jishu/J. Propulsion Technol. **45**, 3019–3029 (2024)
7. Di Battista, A., et al.: Future of Jobs Report 2023, p. 295. World Economic Forum, Geneva (2023)
8. Samadi, M.A., Jaquay, S., Lin, Y., Tajik, E., Park, S., Nixon, N.: Minds and machines unite: deciphering social and cognitive dynamics in collaborative problem solving with AI. In: ACM International Conference Proceeding Series, pp. 885–891. Association for Computing Machinery (2024)
9. Essel, H.B., Vlachopoulos, D., Essuman, A.B., Amankwa, J.O.: ChatGPT effects on cognitive skills of undergraduate students: receiving instant responses from AI-based conversational large language models (LLMs). Comput. Educ. Artif. Intell. **6**, 100198 (2024)
10. Maramba, G., et al.: Healthcare supply chain efficacy as a mechanism to contain pandemic flare-ups: a South Africa case study. Int. J. Inf. Syst. Supply Chain Manag. (IJISSCM) **17**, 1–24 (2024)
11. Boland, A., Dickson, R., Cherry, G.: Doing a Systematic Review: A Student's Guide. SAGE Publications Ltd., London (2017)
12. Walter, Y.: Embracing the future of Artificial Intelligence in the classroom: the relevance of AI literacy, prompt engineering, and critical thinking in modern education. Int. J. Educ. Technol. High. Educ. **21** (2024)
13. Royer, C.: Outsourcing humanity? ChatGPT, critical thinking, and the crisis in higher education. Stud. Philos. Educ. (2024)
14. Sánchez-Guerrero, J., Miranda-López, X., Buenaño, H., Lozada-Miranda, D.: ChatGPT to motivate critical thinking in the teaching-learning process of first semester students at the technical university of Ambato. In: Rocha, Á., Fajardo-Toro, C.H., Rodríguez, J.M.R. (eds.) MICRADS 2023, vol. 380, pp. 169–179. Springer, Singapore (2024). https://doi.org/10.1007/978-981-99-8894-5_15
15. Huang, C.W., Coleman, M., Gachago, D., Van Belle, J.P.: Using ChatGPT to encourage critical AI literacy skills and for assessment in higher education. In: Van Rensburg, H.E., Snyman, D.P., Drevin, L., Drevin, G.R. (eds.) SACLA 2023, vol. 1862, pp. 105–118. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-48536-7_8
16. Wu, Y.: Critical thinking pedagogics design in an era of ChatGPT and other AI tools—shifting from teaching "What" to teaching "Why" and "How." J. Educ. Dev. **8**, 1 (2024)
17. Darwin, Rusdin, D., Mukminatien, N., Suryati, N., Laksmi, E.D., Marzuki: Critical thinking in the AI era: an exploration of EFL students' perceptions, benefits, and limitations. Cogent Educ. **11**, 2290342 (2024)
18. Walczak, K., Cellary, W.: Challenges for higher education in the era of widespread access to generative AI. Econ. Bus. Rev. **9** (2023)
19. Li, H.F.: Effects of a ChatGPT-based flipped learning guiding approach on learners' courseware project performances and perceptions. Australas. J. Educ. Technol. **39**, 40–58 (2023)
20. Ahmed, Y.A., Sharo, A.: On the education effect of CHATGPT: is AI CHATGPT to dominate education career profession? In: 2023 International Conference on Intelligent Computing, Communication, Networking and Services, ICCNS 2023, pp. 79–84 (2023)
21. Lee, H.Y., Chen, P.H., Wang, W.S., Huang, Y.M., Wu, T.T.: Empowering ChatGPT with guidance mechanism in blended learning: effect of self-regulated learning, higher-order thinking skills, and knowledge construction. Int. J. Educ. Technol. High. Educ. **21** (2024)
22. Chan, C.K.Y.: A comprehensive AI policy education framework for university teaching and learning. Int. J. Educ. Technol. High. Educ. **20**, 38 (2023)

23. Shanshan, S., Sen, G.: Empowering learners with AI-generated content for programming learning and computational thinking: the lens of extended effective use theory. J. Comput. Assisted Learn. (2024)
24. Liao, J., Zhong, L., Zhe, L., Xu, H., Liu, M., Xie, T.: Scaffolding computational thinking with ChatGPT. IEEE Trans. Learn. Technol. (2024)
25. Celik, I.: Exploring the determinants of artificial intelligence (AI) literacy: digital divide, computational thinking, cognitive absorption. Telemat. Inform. **83** (2023)
26. Tang, Y., Ma, X.: Enhancing EFL vocabulary acquisition through computational thinking. In: Zhao, F., Miao, D. (eds.) AIGC 2023, vol. 1946, pp. 70–82. Springer, Singapore (2024). https://doi.org/10.1007/978-981-99-7587-7_6
27. Sharples, M.: Towards social generative AI for education: theory, practices and ethics. Learn. Res. Pract. **9**(2), 159–167 (2023). https://doi.org/10.1080/23735082.2023.2261131



Large Language Models for Democratizing Business Process Modeling: BPMN Model Generation and Style Guide Adherence

Seline Wenger[1], Maja Spahic-Bogdanovic[1,2(✉)], and Andreas Martin[1]

[1] School of Business, FHNW University of Applied Sciences and Arts Northwestern Switzerland, Riggenbachstrasse 16, 4600 Olten, Switzerland
{maja.spahic,andreas.martin}@fhnw.ch
[2] University of Camerino, Via Madonna delle Carceri 9, 62032 Camerino, MC, Italy

Abstract. Business Process Management (BPM) is important for organizations, with process modeling being a key component. However, creating accurate and consistent models is challenging and time-consuming. As large language models (LLMs) offer potential solutions, this article explores using LLMs to generate business process models that follow company internal style guides. Interviews with process engineers at a company provided insights into current practices and challenges. A prototype using OpenAI's GPT-4 API was developed based on these insights. This tool enables non-experts to generate business process models in BPMN 2.0 notation through a user-friendly interface. Evaluation using the PET dataset demonstrated that the LLM-generated models are comparable to those created by human experts regarding recall and precision.

Keywords: Large Language Models · Business Process Modeling · Business Process Model and Notation · Modeling Style Guides · Model Generation · Business Process Management

1 Introduction

Business Process Management (BPM) helps organizations achieve reliable results and identify improvement opportunities [9]. A key stage in BPM is process discovery, where the current business process model is created using event data, documents, and implicit knowledge from personnel. This stage can consume up to 60% of the time spent on BPM projects [11], requiring collaboration between domain experts and process analysts [9]. While Business Process Model and Notation (BPMN) 2.0 is the standard for modeling [22], its complexity often requires formal training and adherence to style guides to ensure clear and consistent models [2,16,27]. Research has focused on automating process modeling using algorithms [1,25] and rule-based Natural Language Processing (NLP) techniques [11], but these approaches often lack generalization [4]. The emergence

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 372–389, 2025.
https://doi.org/10.1007/978-3-031-78255-8_22

of Large Language Models (LLMs), such as OpenAI's GPT model family, accessible over chat interfaces like OpenAI's ChatGPT, offers new opportunities for generating business process models directly from natural language descriptions. Recent studies have highlighted the potential of LLMs in this field [10,13,18,32], yet the quality and reliability of their outputs require careful handling and verification [29]. LLMs can advance business process modeling by enabling non-experts to create accurate models through natural language prompts. However, limited research remains on the ability of LLMs to create models that consistently adhere to internal style guidelines, revealing a gap in current understanding. This article addresses this gap by developing an LLM-based tool to assist non-BPMN experts in business process modeling, demonstrating its practical applications, and identifying areas for further research. The developed LLM-based tool leverages the capabilities of LLMs to generate BPMN 2.0 models through a user-friendly interface. The evaluation of this tool demonstrated that LLM-generated models are comparable to those created by human experts in terms of recall and precision. Additionally, the tool's usability was rated highly in qualitative assessments. While the potential of LLMs in BPM is evident, it is important to recognize their limitations and the need for iterative prompt refinement to achieve high-quality results.

This paper is structured as follows: Sect. 2 reviews the state-of-the-art literature. Section 3 outlines the methodology. Section 4 introduces the practical use case. Section 5 presents the proposed artifact design. Section 6 details the development of the artifacts. Section 7 concludes with a discussion of further research directions.

2 Related Work

This section provides background information on the state-of-the-art in business process management and modeling and the role of LLMs and NLP in this field.

2.1 Purpose of Business Process Modeling

Companies rely on processes as core assets, making BPM important for organizations of different sizes. According to Dumas et al. [9], BPM orchestrates work to ensure reliable results and identify improvement opportunities, with business process modeling at its core. BPMN 2.0 serves as the international standard for modeling, and style guides help create clear, complete, and consistent models [27]. Modeling is represented throughout the entire BPM lifecycle, such as in the creation of as-is, to-be, and executable models. Investing in process modeling helps manage complexity and change by simplifying reality and illustrating relationships [5,34]. It enhances communication since visual models are clearer than text, reducing ambiguity [5,24]. Additionally, it supports knowledge management by converting implicit knowledge into explicit knowledge, preventing incorrect assumptions and defects [5,34]. Several business process modeling languages have evolved, with BPMN 2.0 being the most widely adopted due to its

international standard status. While BPMN 2.0 ensures model correctness, Silver [27] emphasizes the importance of semantically correct, clear, complete, and consistent models. BPMN can be enhanced with style rules focused on labeling, presentation, and modeling principles [27], ensuring compliance with modeling standards and guidelines throughout the company. Within the context of style rules, labeling principles are to use labels and icons for clarity [27], follow a verb-noun naming convention for activities [9], and ensure unique labeling for activities and end events [27]. Presentation principles include modeling processes hierarchically and limiting activities per diagram [27]. Modeling principles include representing externals in a black-box pool and using lanes for organizational units, avoiding XOR gateways unless necessary [27], and minimizing routing paths [9]. Tools like Signavio, Trisotech BPMN Modeler, and bpmnlint support style rule validation.

Despite the benefits, many BPM projects fail in practice, facing challenges such as integrating modeling with organizational priorities and routine operations [2], and lacking a standardized modeling environment [16]. Process discovery, consuming up to 60% of project time, requires collaboration between domain experts and process analysts [9,11]. As domain experts often lack modeling expertise, analysts translate their knowledge into structured models [9]. Promoting stakeholder collaboration and formal training is vital for building modeling expertise [2,16], but most practitioners lack formal BPMN education [24].

2.2 Large Language Models for Business Process Modeling

Business process modeling is time-consuming [11], prompting research into automating the extraction of relevant process data to create models more efficiently. Efforts in process discovery using event logs from information systems [1] often failed due to unrealistic expectations and insufficient digital traces [25]. In the early stages of NLP, the research investigated generating process models from textual input using rule-based approaches [11]. Despite advancements, these methods remain specialized and lack generalization [4]. The transformer architecture, introduced by Vaswani et al. [31], advanced NLP by improving the handling of long-range dependencies in text. This innovation shifted the focus from classical supervised learning to a "pre-train, fine-tune" paradigm [20], which has since evolved into pre-training with raw text in self-supervised fashion, supervised fine-tuning using instruction data, and preference alignment using reinforcement learning from human feedback (RLHF) or direct preference optimization (DPO) [21]. Consequently, LLMs such as OpenAI's GPT-4 can now perform various tasks, including business process modeling. Techniques like zero-shot, few-shot, and chain-of-thought prompting enhance LLM performance, allowing for more accurate and contextually appropriate responses [26,33]. Vidgof et al. [32] highlighted LLMs' potential for structuring large volumes of text, suggesting automatic BPMN model generation. Fill et al. [10] and Grohs et al. [13] explored using OpenAI's ChatGPT, noting improved results with GPT-4. Fill et al. [10] used few-shot prompting with examples, achieving valid outputs in

a custom JSON format. Grohs et al. [13] adopted zero-shot prompting, instructing the LLM to output in a predetermined natural language notation, focusing on recall without considering false positives. Recent research by Zirnstein [35] used GPT-4 to generate BPMN diagrams by creating artificial event logs for a split miner algorithm, producing SAP Signavio-compliant JSON models. However, this method restricts interchangeability with other BPMN tools. Kourani et al. [19] transformed natural language input into Partially Ordered Workflow Language (POWL), later converted to BPMN or Petri Nets, demonstrating high-quality model generation and efficient feedback integration with GPT-4.

Despite these advancements, LLMs have limitations, such as producing overconfident and inaccurate information, known as "hallucinations" [17] or "bullshit" [15]. Evaluation metrics, like HumanEval [8], assess an LLM's ability to generate accurate, coherent, and contextually appropriate responses, providing insights into the capabilities of the selected LLM. Another challenge is the frequent misalignment between LLM behavior and human intentions. [17]. To mitigate these issues, RLHF emerged, and fine-tuning models were used to align with human preferences. RLHF involves pre-training a model, creating a reward model from human preferences, and optimizing the model's outputs based on this reward [23].

Previous research has explored LLMs in BPM [32], prompt engineering [7], and initial attempts at generating BPMN models with LLMs [10,13,18,19,35]. While BPMN's standardized format is XML-based, most studies have focused on alternative formats like JSON or POWL, limiting interchangeability. Practical approaches that align models with internal style guides and offer conversation-based creation with immediate visual representation in BPMN 2.0 are still underdeveloped.

3 Methodology

Hevner and Chatterjee [14] describe design science research (DSR) as a methodology that develops innovative artifacts that address practical problems and contribute to knowledge. This involves iterative development and evaluation, known as the design cycle, enriched by the relevance cycle (practical focus) and rigor cycle (scientific research). This research employs DSR to generate actionable solutions to both academic and practical issues. Vaishnavi and Kuechler [30] extends the design cycle to include five steps: awareness, suggestion, development, evaluation, and conclusion, which structure artifact development and ensure rigorous evaluation of real-world problems. A literature review was conducted to understand the research problem and establish the body of knowledge. Furthermore, to cover the practical perspective in the awareness phase, semi-structured interviews were conducted with BPMN experts at the company cablex AG[1], located in Switzerland. The interviewees included the individual responsible for process modeling and the Head of the Environmental & Quality Management department. These interviews focused on how business process

[1] https://www.cablex.ch/.

modeling is applied at cablex AG, the presence and enforcement of internal style guides, and the relevance of literature-identified challenges in the company's context. In parallel, a survey of employees gathered insights into non-BPMN experts' difficulties and identified potential areas where LLM support could be useful. The interview and survey feedback informed the design principles for an LLM-based application to support business process modeling. Additional input was gathered from the Chief Information Security Officer (CISO) to meet privacy and security requirements of the company cablex AG. The design progresses through iterative development, incorporating continuous feedback [30]. The application aims to align with internal style guides and provide immediate visual representation in BPMN 2.0 format. The artifact is evaluated through a quasi-experiment, comparing BPMN models created by the LLM, the cablex AG BPMN expert, and the article author. The PET[2] dataset [3] serves as the input for this task. Additionally, a focus group interview with non-BPMN experts assesses the prototype's usability using the System Usability Scale (SUS) by Brooke [6]. Feedback from these evaluations refines the artifact. The conclusion phase consolidates and documents results, classifying firm knowledge and identifying areas for future research [30]. The findings are discussed, compared with previous research, and communicated to technology-and management-oriented audiences [14].

4 Application Scenario

This section explores a real-life application use case of cablex AG, serving as the basis for developing the artifact. An overview of the current business process modeling at cablex AG offers insights into its organization. In addition, the challenges identified in the literature analysis are examined for their practical relevance.

4.1 Business Process Modeling at Cablex AG

cablex AG, a subsidiary of Swisscom Group, specializes in ICT and network infrastructure solutions and smart infrastructure projects. The company comprises five business units: Field Services, Wireline Construction, Wireline Rollout, Wireless, and Infrastructure & Business Solutions. With over 2,500 employees in 50 locations, cablex AG strives to be a leading Swiss technology innovator, enhancing customer experiences and benefiting society [28]. cablex AG's focus on clear, organized processes aligns with its vision. The current state of business process modeling at cablex is examined to understand its organization and identify existing issues. This assessment determines if challenges like the shortage of proficient modelers and maintaining standardized models, highlighted in the literature, apply to cablex AG. Interviews were conducted with business analysts from the Environmental & Quality Management (EQM) department to

[2] https://pdi.fbk.eu/pet-dataset/.

capture the current state at cablex AG. At cablex AG, business process modeling focuses on knowledge management and communication, guiding operations, providing information, and facilitating training through graphical representations. These high-level models are used company-wide but are not intended for automation or complete process mapping. Examining the organizational structure of business process modeling at cablex AG reveals four key roles. The Process Owner sponsors the overall process, secures resources, and has strategic control. The Process Manager is the contact point for process details, ensuring accuracy, implementation, education, and training, and takes operational control. The Process Engineer is responsible for modeling business processes, supporting analysis, and optimizing processes. The Process Employee implements specific activities within the process. Business process modeling at cablex involves collaboration between the process manager and the process engineer. A release workflow ensures accurate representation, with both the process manager and process owner reviewing the model before publication. Major updates trigger this workflow, while minor changes do not. Processes are audited every three years to ensure they remain current.

4.2 Modeling Environment and Style Guides

cablex AG's process landscape is organized into management, core, and support processes across four modeling layers: Layer 0 (process landscape titles), Layer 1 (process phases), Layer 2 (process groups), and Layer 3 (actual processes). Managed with IMS Integrierte Managementsysteme AG software, high-level models link to detailed procedural information on the company wiki or specific file shares. The modeling language is BPMN 2.0, simplified to a subset of elements, including start and end events, tasks, XOR and parallel gateways, and sequence flows, with pools and lanes auto-generated based on task assignments. The EQM department has established principles for standardizing models: using the defined BPMN 2.0 subset, limiting processes to ten activities, assigning one business role per activity, modeling processes forward with common exceptions, and avoiding abbreviations unless widely recognized. cablex AG adheres to Swisscom's data confidentiality categories, which are organized into four levels: C1 (public), C2 (general internal information), C3 (confidential), and C4 (strictly confidential). Business processes in IMS are typically C2, accessible to all employees, and free of confidential data. Linked task-specific information follows the "need-to-know" principle to restrict access to potentially confidential information.

4.3 Verification of Challenges at Cablex AG

Three statements were discussed during interviews with process engineers and supported by survey data from cablex employees in various roles to identify common perceptions and differences. The first statement, that domain experts are usually not trained in modeling business processes, was confirmed by interviews and survey results. Domain knowledge at cablex is divided between domain experts (process managers) and process analysts (process engineers). Domain

experts collaborate with process analysts because they are not trained in modeling with IMS, and only EQM division employees are authorized for this task. According to the survey, 70% of respondents have not received any BPMN 2.0 training, and those trained did so through external programs. Consequently, untrained employees perceive their BPMN 2.0 expertise as low, while trained employees rate it as high. This verifies that domain experts are rarely trained in modeling business processes, typically gaining such expertise through individual external training. The second statement, that non-BPMN experts are overwhelmed by the usage of BPMN 2.0, is supported by the literature suggesting using style guides to assist users. At cablex, design principles aim for simplicity by using a subset of BPMN 2.0 elements to reduce complexity. Interviews reveal that process engineers know BPMN 2.0's potential to overwhelm non-experts. Survey data supports this, showing that most individuals without extensive BPMN 2.0 expertise perceive their skill level as low, indicating they may feel overwhelmed by the standard's complexity. This statement is verified through the gathered data. The third statement, that business process modeling often happens in silos leading to a lack of a standardized modeling language and environment, was partly verified. At cablex, business process modeling is managed by the EQM department, but employees sometimes model detailed or case-specific processes themselves using different tools for ad hoc modeling. The survey reveals that there is no standardized tool for ad hoc modeling, with Visio being the most used, followed by draw.io and PowerPoint, none of which fully support BPMN 2.0 notation. While BPMN 2.0 is used in IMS for high-level process models, only 25% of the respondents are familiar with cablex's design principles for modeling business processes. Consequently, models created without EQM supervision often lack standardization. Verification of challenges at cablex AG reveals that domain experts are generally not trained in business process modeling, with BPMN 2.0 training typically obtained through external programs. Non-experts are often overwhelmed by BPMN 2.0, leading cablex to use a simplified subset of BPMN 2.0 elements to reduce complexity. While BPMN 2.0 is the standard modeling language, the lack of a standardized tool and missing awareness of design principles results in inconsistencies in ad hoc modeling.

5 LLM Application Design

This section examines the design suggestions for the artifact, starting with the technical requirements and progressing through the functional requirements and prompt templates.

5.1 Technical Requirements

Based on an interview with the CISO of cablex AG, specific requirements must be met to introduce a new LLM application authorized to process internal knowledge, including data processing criteria and general software release standards.

Confidentiality and Data Classification: Public data (C1) is generally allowed. Internal data (C2) may be processed if data protection agreements and customer-specific contracts are adhered to. Confidential (C3) and strictly confidential (C4) data, including employee, third-party, and military data, are not allowed. *Data Integrity:* Business-critical information must remain unchanged and unmanipulated. The correctness of entered information must be maintained. *System Availability:* Ensure information-processing systems are accessible and protected from loss or inaccessibility. Require multi-factor authentication for system access. *Data Localization:* Data must be stored in Switzerland or a country compliant with the EU General Data Protection Regulation (GDPR). The data entered must not be used to train the LLM.

Of the 49 LLMs with over 10 billion parameters, only 16 are capable of coding. To process internal data (C2) while ensuring data security, two options are considered: local execution and cloud execution. Local execution involves downloading and running the LLM locally, ensuring data privacy but requiring substantial computational power, which may not be feasible. Cloud execution involves using cloud-hosted LLMs with server location control to ensure compliance with data protection regulations. Examples include Google Cloud's Vertex AI and Azure OpenAI Services, though access to Azure may be restricted. Given the hardware limitations for local execution, the cloud-hosted option is preferred. Google's Gemini and OpenAI's GPT models are evaluated for their suitability in generating BPMN 2.0 process models. The LLMs were tested using three simple process descriptions with the same prompt template, focusing on their ability to generate .bpmn files without further instructions on style guides. LangChain was used to build the LLM application. The models tested include GPT-4, GPT-3.5-turbo-16k, and Gemini-pro. The evaluation was based on the annotation schema for process model elements and relations introduced by Bellan et al. [3], calculating recall to measure identification accuracy. GPT-4 outperformed the other LLMs, particularly in modeling complex processes with pools and lanes, making it the preferred choice for the artifact. To comply with confidentiality requirements, a confidentiality disclaimer ensures users enter only data within the allowed classification. Implementing a warning to label AI-generated content promotes user awareness and caution. The application will operate locally, bypassing the need for identity and access management. For future releases, ensure system availability and data integrity by incorporating robust access controls and multi-factor authentication. By meeting these technical and functional requirements, the LLM application can securely and effectively process business process models at cablex AG, aligning with internal standards and compliance regulations.

5.2 Functional Requirements

Based on interviews with process engineers and a survey of cablex employees, several functional requirements were identified for the LLM application. These requirements were captured through user stories, articulated in simple language. Table 1 introduces the user stories and corresponding design suggestions for the LLM application.

Table 1. User Stories and Corresponding Design Suggestions

User Story	Design Suggestion
US1: As a process engineer, I want to ensure that all process models are modeled in a single modeling language, so that interpretability is enhanced.	DS1: Define the output format in the prompt template as a .bpmn file to ensure models are generated in BPMN 2.0, including both the BPMN process and diagram elements
US2: As a process engineer, I want to ensure that every process is modeled as simply as possible so that everyone can understand it.	DS2: Provide a clear list of the permissible subset of BPMN 2.0 elements to be used in the prompt template
US3: As a process engineer, I want to ensure that process models consist of a maximum of ten activities so that readability is enhanced.	DS3: Define in the prompt template that outputs should not exceed ten activities. For longer processes, use collapsed sub-activities
US4: As a process engineer, I want to ensure that process models are modeled forward so that readability is enhanced.	DS4: Clearly state in the prompt template that outputs should be modeled forward
US5: As a process engineer, I want to ensure that every process activity is mapped to a cablex business role, so that responsibilities are clarified.	DS5: Connect the LLM to the database containing cablex business roles. Define in the prompt template that each task's lane must include these roles. If a role is not found, prompt the user with possible examples
US6: As a process engineer, I want to ensure that abbreviations – unless they are widely recognized – are avoided, so that models are understandable for everyone.	DS6: Specify in the prompt template that labeling should not contain abbreviations
US7: As a process engineer, I want to ensure that process models are written in a uniform language so that interpretability is enhanced.	DS7: Define in the prompt template that activities should be labeled using "verb in active form + noun," events as "noun + verb in past form" [27]. Pools should be labeled in "noun + gerund" format, and exclusive gateways should be questions with outcomes labeled

The user stories reflect the style guides at cablex, focusing mainly on labeling and presentation principles, ensuring process models are simple, standardized, and clear, aligning with cablex's internal guidelines. Survey findings revealed interest in three specific actions due to varying levels of BPMN proficiency among employees and the lack of formal training. These actions are creating a process description that generates a .bpmn file, generating a textual description from a .bpmn file, and updating an existing model, which accepts a .bpmn file as input and provides an updated .bpmn file as output. To support these actions, the process modeling tool must be capable of importing and exporting .bpmn files. While some tools at cablex, such as IMS, do not support this capability, alternatives like bpmn.io and Camunda Modeler do. Three prompt templates are proposed to address the user stories and actions identified. These templates will guide the LLM in producing outputs that meet the functional requirements and adhere to the style guides, ensuring the generated process models are standardized and easy to understand.

5.3 Prompt Templates

Prompt templates are essential for guiding LLMs in generating the desired outputs. Giray [12] suggests including instructions, context, input data, and output indicators. Below are the proposed templates for the three key actions: generat-

ing BPMN from text, generating text from BPMN, and updating BPMN models according to style guides.

Generating BPMN from Text

- *Goal:* Generate a BPMN 2.0 model from a natural language process description.
- *Instruction:* Create a Business Process Model using BPMN 2.0.
- *Context:* You are a BPMN modeling expert who can create .bpmn files. The process should be forward-modeled, with each activity assigned to a role in a lane.
- *Input Data:* {process_description}
- *Output Indicator:*
  - Provide the output in .bpmn file format with both BPMN process and diagram elements.
  - Use only basic BPMN 2.0 elements: Start Event, Task, Exclusive Gateway, Parallel Gateway, Sequence Flow, Pool, Lane, and End Event.
  - Avoid other elements like intermediate events and user tasks.
  - Label all components consistently and avoid abbreviations.
  - Use the label formats "noun + verb in past tense" for events, "verb in active form + noun" for tasks, "noun + gerund" for pools, and questions for exclusive gateways.
  - Limit the process to ten activities, using collapsed subprocesses for longer processes.

Generating Text from BPMN

- *Goal:* Create a process description from a BPMN model.
- *Instruction:* Describe a Business Process Model using natural language.
- *Context:* You are a BPMN interpretation assistant who can read .bpmn files.
- *Input Data:* {bpmn_file}
- *Output Indicator:*
  - Provide a continuous text process description, avoiding enumerations and abbreviations.
  - Do not name BPMN elements, ensuring non-experts can understand the text.

Updating BPMN Models

- *Goal:* Update a BPMN model to adhere to cablex style guides.
- *Instruction:* Update a Business Process Model using BPMN 2.0.
- *Context:* You are a BPMN modeling expert who can create .bpmn files. The process should be forward-modeled, with each activity assigned to a role in a lane.
- *Input Data:* {bpmn_file}
- *Output Indicator:*
  - Provide the output in .bpmn file format with both BPMN process and diagram elements.

- Use only basic BPMN 2.0 elements: Start Event, Task, Exclusive Gateway, Parallel Gateway, Sequence Flow, Pool, Lane, and End Event.
- Avoid other elements like intermediate events and user tasks.
- Label all components consistently and avoid abbreviations.
- Use the label formats "noun + verb in past tense" for events, "verb in active form + noun" for tasks, "noun + gerund" for pools, and questions for exclusive gateways.
- Limit the process to ten activities, using collapsed subprocesses for longer processes.

The conceptual dialogue flow of the LLM application involves the user selecting an action ("From Text to BPMN," "From BPMN to Text," or "Update BPMN"), submitting the relevant input, executing the LLM with the appropriate prompt template, and generating the desired output. This ensures the LLM application supports the functional requirements and adheres to cablex's style guides, providing clear and standardized outputs for business process modeling tasks.

6 LLM Application Prototype

This chapter presents the developed artifact, beginning with an overview of its architecture and primary components. It details the implemented prototype, focusing on its three main features: Text to BPMN, BPMN to Text, and BPMN to BPMN. Integrating individual user stories is explained, demonstrating the artifact's effectiveness through practical examples. The chapter concludes with a discussion of the artifact's limitations.

6.1 Architecture of the Prototype

The system architecture has three main components: a frontend interface using Streamlit[3], a backend processing engine with LangChain[4], and the OpenAI language model, as presented in Fig. 1. Streamlit provides a web interface for user interactions. LangChain, chosen for its features and support, handles input preprocessing, template selection, and LLM interactions. Users select actions and provide input through the Streamlit web app, triggering LangChain to process the input and send prompts to the LLM via the OpenAI API. The LLM's response is then relayed back to Streamlit for display.

Due to security constraints, the prototype uses synthesized data. For a production release, transitioning to Azure OpenAI API is necessary, involving Azure Active Directory Authentication and updating to AzureChatOpenAI. The prototype has three features: Text to BPMN, BPMN to Text, and BPMN to BPMN. Each feature integrates user stories to meet functional requirements. Text to BPMN converts natural language descriptions into BPMN 2.0 models, ensuring

[3] https://streamlit.io/.
[4] https://www.langchain.com/.

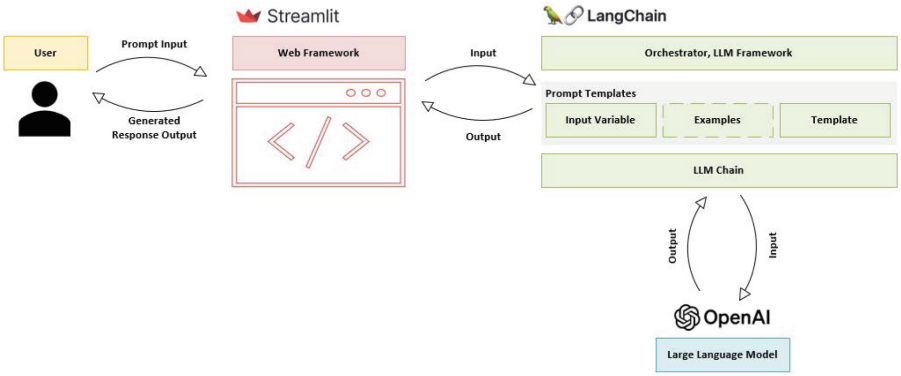


Fig. 1. LLM Prototype: Architecture Overview

simplicity, role mapping, and uniform labeling. BPMN to Text generates understandable text from BPMN models. BPMN to BPMN updates existing models to conform to style guides. The source code is available on Zenodo[5]. The application features a primary page, as shown in Fig. 2, with a dropdown menu for action selection and a sidebar for instructions and API key entry. A modal window provides confidentiality policies. The artifact's effectiveness is demonstrated through practical examples, such as converting a process description into a compliant BPMN 2.0 model, showcasing its capability in process modeling tasks.

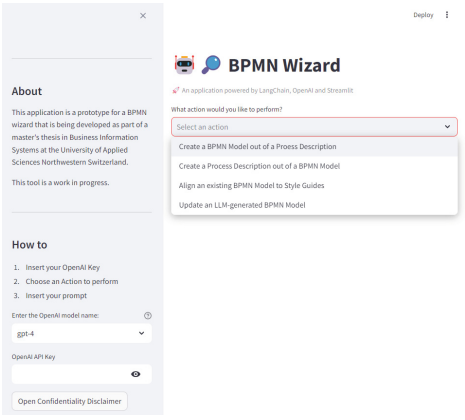


Fig. 2. LLM Prototype: Landing Page

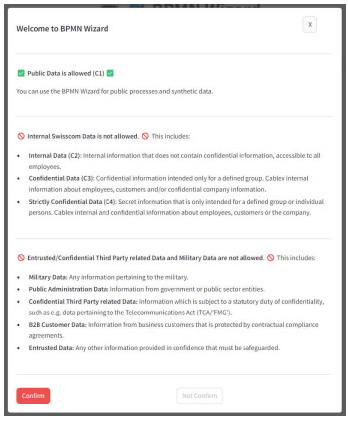


Fig. 3. LLM Prototype: Wizzard Page

[5] https://doi.org/10.5281/zenodo.11657507.

6.2 Prototype Realization

An example demonstrates the practical use of the artifact's "From Text to BPMN" feature. Users select "Create a BPMN Model out of a Process Description" and enter their description, with guidance available via an icon and expandable instructions, as shown in Fig. 4. The description should be procedural, depict the happy path, avoid abbreviations, and allocate tasks to business roles. The backend processes the input, extracting relevant actors via the LLM and converting them into a JSON array, flagging non-compliant roles for correction. The process description is then used to generate a BPMN model in XML format, see Fig. 5. The model adheres to style guides by using BPMN 2.0, limiting elements and activities, modeling forward, representing roles, avoiding abbreviations, and maintaining consistent labeling. The output, an XML code, is displayed and visualized using bpmn.io, with a warning for AI-generated content.

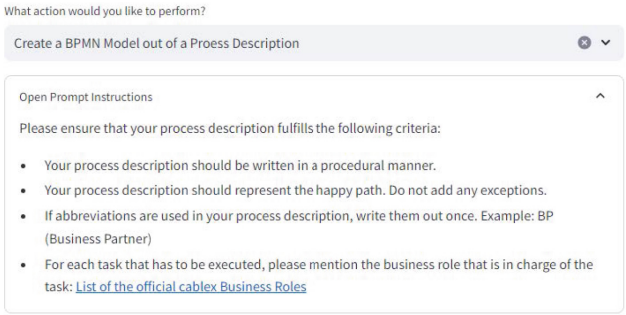


Fig. 4. Input Screen

If the users select "Create a Process Description out of a BPMN Model" and upload a BPMN file, the uploaded model is parsed, and a zero-shot prompt template generates a natural language description that is free from BPMN-specific terminology. The output is displayed with a disclaimer. To align a BPMN model with style guides, users select "Align an existing BPMN Model to Style Guides" and upload a BPMN file. The LLM extracts and verifies business roles, aligns the model with style guides, and displays the output as XML code, visualized using bpmn.io. Ultimately, a non-compliant BPMN process is corrected and updated, adhering to style guides. By integrating these functions and requirements, the LLM application effectively generates, interprets, and updates BPMN models according to internal style guides and user inputs.

6.3 Evaluation

The developed prototype is evaluated for its usability and the accuracy of the produced business process models. Usability is assessed using the SUS score,

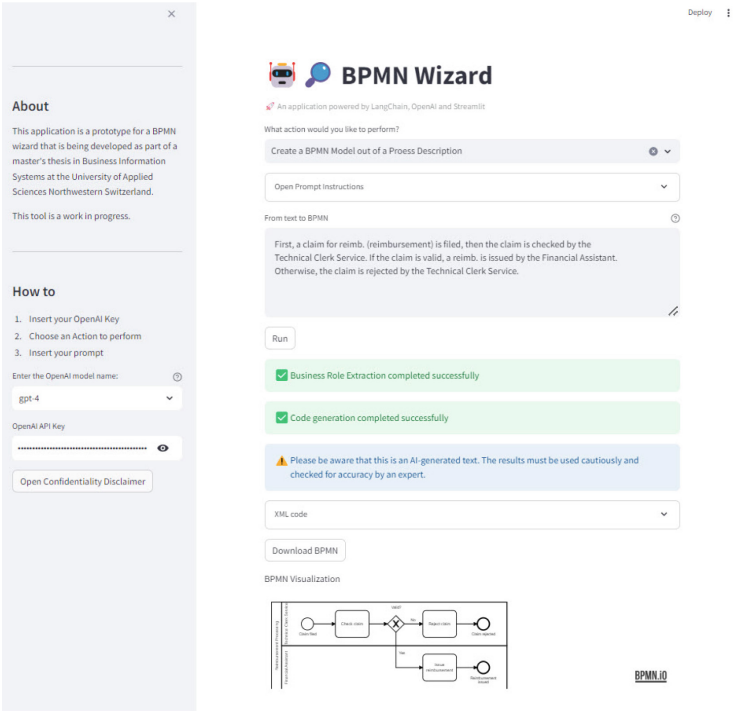


Fig. 5. Output Visualization

and the accuracy is evaluated by calculating recall, precision, and F1-score for Text to BPMN, BPMN to Text, and BPMN to BPMN actions. A focus group with non-BPMN experts demonstrated and discussed the prototype, followed by a short survey based on the SUS scale. An average SUS score of 85 was obtained from six respondents, indicating above-average usability. Feedback led to improvements such as more prominent instructions and considering future multimodal model capabilities. The LLM application's ability to generate business process models (From Text to BPMN) adhering to internal style guidelines was tested using five processes from the PET dataset. Adapted process descriptions were fed into the LLM to generate BPMN XMLs. The LLM achieved an overall F1-score of 96%, matching the performance of a cablex process analyst. The evaluation highlighted the LLM's challenges with start elements and XOR gateways when input descriptions lacked explicit details. The LLM-generated process descriptions (From BPMN to Text) were evaluated using the annotation schema of Bellan et al. [3]. The LLM identified all entities and relationships from the original description, achieving an average recall of 100% and an overall precision of 87%. The F1-Score was 92%, showing the LLM's capability to generate detailed and accurate process descriptions. The LLM's capability to align BPMN files with internal style guides (From BPMN to BPMN) was

assessed using original process models from Friedrich [11]. Despite token limitations affecting three processes, the LLM achieved an average F1-Score of 93% for the assessable processes. The LLM effectively added or adapted labels and wrote out abbreviations, although it struggled with pool labels and gateway contexts. The evaluations indicate that the LLM application effectively generates, interprets, and updates BPMN models according to internal style guides and user inputs.

6.4 Limitations

During development, certain limitations of OpenAI's API impacted the prototype, specifically regarding context length in GPT-4 models. These models have restricted context windows, which limit the total number of input, output, and control tokens in a single request. GPT-4 supports a context window of 8,192 tokens. This context length limitation restricted prototype development to prompt templates with minimal or no examples. Zero-shot templates risk producing inconsistent answers with varying output formats, while few-shot templates yield more consistent formats. Further, token restrictions confined the artifact to short BPMN processes, preventing evaluating processes with more than ten activities using collapsed sub-processes. Additionally, for outputs requiring extensive tokens like BPMN XML, GPT-4 often omits necessary BPMN diagram components. Achieving complete BPMN XML output requires iterative prompt engineering and precise instructions. The behavior of OpenAI's LLMs varies, where effective prompts for GPT-4 may not work with GPT-4-Turbo. Prompt templates were optimized for GPT-4 but may need refinement for other LLMs. It is also important to note that responses can vary between queries within the same LLM despite using consistent prompt templates.

7 Conclusion

Through expert interviews and a survey of non-BPMN experts, key challenges in business process modeling were confirmed: most domain experts lack formal BPMN training, and standardization is absent. The CISO of cablex emphasized that only public data can be processed by an LLM, with warnings advising peer review for AI-generated text, aligning with Swisscom's principles. Handling internal knowledge requires a GDPR-compliant secure cloud. The prototype used the OpenAI API but can transition to Azure OpenAI API. The study formulated user stories and design suggestions for cablex's style guide requirements through structured interviews. These suggestions were incorporated into prompt templates combining expert prompting and few-shot prompting using natural language. This research demonstrates that LLMs can generate valid BPMN 2.0 models. Effective prompt design was crucial, with few-shot prompting yielding consistent responses, in line with Fill et al. [10] and Kourani et al. [19]. The LLM processes natural language inputs directly, unlike Zirnstein [35], who used an intermediary step. Providing the output in BPMN 2.0 format was essential, as

manual transfers to visual diagrams, as done by Grohs et al. [13], hinder usability. The intermediate process representation by Kourani et al. [19] also did not meet style guide requirements. Evaluation with five models from the PET dataset showed the LLM generates semantically correct BPMN 2.0 models adhering to internal style guides. The LLM's performance matched human modelers, achieving high F1 scores, though expert review is advised. LLMs can help non-BPMN experts create business process models following internal guidelines, improving usability. This study highlights key challenges and the need for uniform models adhering to internal style guides. The LLM prototype supports BPMN 2.0 modeling for non-experts, facilitating standardized model creation. It addresses generating BPMN models from descriptions, converting BPMN models to text, and updating models to conform to style guides. Using the XML standard for BPMN 2.0 ensures proper storage and sharing of process models. Limitations include the current token constraints of OpenAI's API, which restricted the prototype to short processes. Future models with larger context windows could handle more complex processes. The output should be reviewed and adjusted as necessary. The cablex AG style guides use only basic BPMN 2.0 elements to reduce complexity, ensuring models facilitate communication and training. Injecting additional BPMN 2.0 knowledge into the LLM was unnecessary due to the simplified models. Future research should expand style guides to other contexts, explore integrating multimodal models, and enable in-app modifications for iterative improvements.

References

1. van der Aalst, W.: Process mining: overview and opportunities. ACM Trans. Manag. Inf. Syst. **3**(2), 1–17 (2012). https://doi.org/10.1145/2229156.2229157
2. Bandara, W., Gable, G.G., Tate, M., Rosemann, M.: A validated business process modelling success factors model. Bus. Process Manag. J. **27**(5), 1522–1544 (2021). https://doi.org/10.1108/BPMJ-06-2019-0241
3. Bellan, P., van der Aa, H., Dragoni, M., Ghidini, C., Ponzetto, S.P.: PET: an annotated dataset for process extraction from natural language text tasks. In: Cabanillas, C., Garmann-Johnsen, N.F., Koschmider, A. (eds.) Business Process Management Workshops. BPM 2022. LNBIP, vol. 460, pp. 315–321. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-25383-6_23
4. Bellan, P., Dragoni, M., Ghidini, C.: A qualitative analysis of the state of the art in process extraction from text. In: 19th International Conference of the Italian Association for Artificial Intelligence, vol. 2776, pp. 19–30. CEUR Workshop Proceedings, Roma (2020)
5. Bridgeland, D., Zahavi, R.: Business Modeling: A Practical Guide to Realizing Business Value, vol. 01. Morgan Kaufmann (2009)
6. Brooke, J.: SUS: A 'Quick and Dirty' Usability Scale. In: Usability Evaluation in Industry, pp. 189–194. Taylor & Francis, Milton Park (1996). https://doi.org/10.1201/9781498710411
7. Busch, K., Rochlitzer, A., Sola, D., Leopold, H.: Just tell me: prompt engineering in business process management. In: van der Aa, H., Bork, D., Proper, H.A., Schmidt, R. (eds.) Enterprise, Business-Process and Information Systems Modeling. BPMDS

EMMSAD 2023 2023. LNBIP, vol. 479, pp. 3–11. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-34241-7_1
8. Chen, M., et al.: Evaluating Large Language Models Trained on Code (July 2021). https://doi.org/10.48550/arXiv.2107.03374
9. Dumas, M., La Rosa, M., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management. Springer, Cham (2018). https://doi.org/10.1007/978-3-662-56509-4
10. Fill, H.G., Fettke, P., Köpke, J.: Conceptual modeling and large language models: impressions from first experimentswith ChatGPT. Enterp. Modell. Inf. Syst. Archit. **18**(3), 1–15 (2023). https://doi.org/10.18417/emisa.18.3
11. Friedrich, F., Mendling, J., Puhlmann, F.: Process model generation from natural language text. In: Mouratidis, H., Rolland, C. (eds.) CAiSE 2011. LNCS, vol. 6741, pp. 482–496. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21640-4_36
12. Giray, L.: Prompt engineering with ChatGPT: a guide for academic writers. Ann. Biomed. Eng. **51**(12), 2629–2633 (2023). https://doi.org/10.1007/s10439-023-03272-4
13. Grohs, M., Abb, L., Elsayed, N., Rehse, J.R.: Large language models can accomplish business process management tasks. In: De Weerdt, J., Pufahl, L. (eds.) Business Process Management Workshops. BPM 2023. LNBIP, vol. 492, pp. 453–465. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-50974-2_34
14. Hevner, A., Chatterjee, S.: Design Research in Information Systems, Integrated Series in Information Systems, vol. 22. Springer, Boston, MA (2010). https://doi.org/10.1007/978-1-4419-5653-8
15. Hicks, M.T., Humphries, J., Slater, J.: ChatGPT is bullshit. Ethics Inf. Technol. **26**(2), 38 (2024). https://doi.org/10.1007/s10676-024-09775-5
16. Indulska, M., Recker, J., Rosemann, M., Green, P.: Business process modeling: current issues and future challenges. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) Advanced Information Systems Engineering. CAiSE 2009. LNCS, vol. 5565, pp. 501–514. Springer, Berlin, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02144-2_39
17. Kaddour, J., Harris, J., Mozes, M., Bradley, H., Raileanu, R., McHardy, R.: Challenges and Applications of Large Language Models (July 2023). https://doi.org/10.48550/arXiv.2307.10169
18. Klievtsova, N., Benzin, J.V., Kampik, T., Mangler, J., Rinderle-Ma, S.: Conversational process modelling: state of the art, applications, and implications in practice. In: Di Francescomarino, C., Burattin, A., Janiesch, C., Sadiq, S. (eds.) Business Process Management Forum. BPM 2023. LNBIP, vol. 490, pp. 319–336. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-41623-1_19
19. Kourani, H., Berti, A., Schuster, D., van der Aalst, W.M.P.: Process modeling with large language models. In: Enterprise, Business-Process and Information Systems Modeling, vol. 511, pp. 229–244. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-61007-3_18
20. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: a systematic survey of prompting methods in natural language processing. ACM Comput. Surv. **55**(9), 1–35 (2023). https://doi.org/10.1145/3560815
21. Martin, A.: Ensuring trustworthy dialogue systems (2024). https://doi.org/10.5281/zenodo.11530754
22. OMG: Business Process Model and Notation (BPMN), Version 2.0.2 (2014). https://www.omg.org/spec/BPMN, publisher: Object Management Group

23. Ouyang, L., et al.: Training language models to follow instructions with human feedback. Adv. Neural Inf. Process. Syst. **35**(NeurIPS) (2022). http://arxiv.org/abs/2203.02155
24. Recker, J.: Opportunities and constraints: the current struggle with BPMN. Bus. Process Manag. J. **16**(1), 181–201 (2010). https://doi.org/10.1108/14637151011018001
25. Reinkemeyer, L.: How to Get Started. In: Process Mining in Action, pp. 11–14. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-40172-6_2
26. Reynolds, L., McDonell, K.: Prompt programming for large language models: beyond the few-shot paradigm. In: Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1–7. New York, NY, USA (May 2021). https://doi.org/10.1145/3411763.3451760
27. Silver, B.: BPMN Method and Style: With BPMN Implementer's Guide, 2nd edn. Cody-Cassidy Press, Altadena (2011)
28. Swisscom: Swisscom Group Vision, strategy and goals. https://www.swisscom.ch/en/about/company/purpose-vision-values-and-goals.html. Accessed 21 Mar 2024
29. Teubner, T., Flath, C.M., Weinhardt, C., van der Aalst, W., Hinz, O.: Welcome to the era of ChatGPT et al. Bus. Inf. Syst. Eng. **65**(2), 95–101 (2023). https://doi.org/10.1007/s12599-023-00795-x
30. Vaishnavi, V.K., Kuechler, W.: Design Science Research Methods and Patterns, 2 edn. CRC Press, Boca Raton (May 2015). https://doi.org/10.1201/b18448
31. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017). https://doi.org/10.48550/arXiv.1706.03762
32. Vidgof, M., Bachhofner, S., Mendling, J.: Large language models for business process management: opportunities and challenges. In: Di Francescomarino, C., Burattin, A., Janiesch, C., Sadiq, S. (eds.) Business Process Management Forum. BPM 2023. LNBIP, vol. 490, pp. 107–123. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-41623-1_7
33. Wei, J., et al.: Chain-of-thought prompting elicits reasoning in large language models. Adv. Neural Inf. Process. Syst. **35**(NeurIPS), 1–14 (2022)
34. Zachman, J.: Enterprise Architecture: The Issue of the Century, October 2015. https://tube.switch.ch/videos/73efd81a
35. Zirnstein, B.: Extraction of BPMN process models from unstructured textual descriptions Bruno Zirnstein. Technical report, Berlin School of Economics and Law (2024). https://doi.org/10.13140/RG.2.2.24412.19841/1

The Role of AI in Modern Language Translation and Its Societal Applications: A Systematic Literature Review

Samuel Ssemugabi(✉)

University of South Africa, Pretoria, South Africa
ssemus@unisa.ac.za

Abstract. Artificial intelligence (AI) has greatly transformed language translation, especially over the past five years. The main aim of this research was to explore the role AI has played in enhancing language translation technologies and its societal implications to provide an understanding of its current state, usage and future prospects. A systematic review was performed using the PRISMA protocol and the Scopus database, resulting in the identification of 351 publications. After applying strict exclusion and inclusion criteria, 52 articles were closely examined, and 16 were included in the final review. The findings demonstrate that AI, mainly through the adoption of neural machine translation (NMT) and transformer-based models, has greatly improved the accuracy and efficiency of translations. These advancements have led to innovations in fields such as education, healthcare and business that facilitate communication in different languages. However, the review also identified important societal implications, including ethical concerns, possible job displacement and the risk of cultural homogenisation. The study identified the need for future AI language translation technologies to focus on culturally sensitive models, universal language frameworks and ethical AI systems that prioritise transparency and fairness. This research provides valuable information for researchers, industry professionals and policymakers on how to harness AI to maximise social benefits while addressing the challenges associated with language translation.

Keywords: Artificial intelligence · Language translation · Systematic literature review · Society · Applications · Language · Translation · AI · SLR

1 Introduction and Background

Language translation has a rich history, beginning with the painstaking manual efforts of human translators [1]. These early methods of machine translation relied heavily on the deep cultural and linguistic knowledge of the individuals. The primary approach involved word-for-word translation, which often failed to capture the nuances and idiomatic expressions of the source language. This process was labour-intensive and time-consuming, but it was the only available method until the advent of computational approaches in the mid-20th century [2]. For several years, translation was a

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 390–404, 2025.
https://doi.org/10.1007/978-3-031-78255-8_23

labour-intensive task, heavily dependent on human translators with extensive linguistic expertise and cultural knowledge. Traditional methods were not only time-consuming but also prone to errors, particularly when handling complex texts or languages with fewer resources [3]. However, the rise of AI has significantly transformed the field, introducing machine learning and neural networks that now play a pivotal role in translating languages with remarkable accuracy and speed [4].

Artificial intelligence (AI) has greatly transformed language translation, especially over the past five years [4]. The journey of AI in language translation began with rule-based machine translation (RBMT) in the 1950s. RBMT relied on predefined grammatical rules and bilingual dictionaries to translate the text. Although groundbreaking at the time, it struggled with idiomatic expressions and context, resulting in rigid and often unnatural translations [5]. The 1990s introduced statistical machine translation (SMT), which marked a significant improvement by using large bilingual text to generate translations based on statistical probabilities. Although SMT offered better consideration of context, it still faced limitations with idiomatic and culturally specific language nuances [6]. The most transformative shift occurred when neural machine translation (NMT) occurred in the 2010s. NMT use deep learning techniques, significantly improving fluency and naturalness in translations by capturing a broader context [3]. In recent years, continued innovations in AI, such as transfer learning techniques, and the availability of large datasets have further improved translation accuracy, especially for languages with low resources which traditionally lacked enough data for training [7]. Although much literature has been produced in this study area, a significant gap remains in the synthesis and consolidation of existing research. This study forms part of the effort to close that gap. The main aim of this research was to explore the role AI has played in enhancing language translation technologies and its societal implications to provide an understanding of its current state, usage and future prospects.

2 Current Technologies in AI Translation

2.1 Machine Learning and Generative AI

In simple terms, machine learning is a method that enables machines to analyse data and make decisions in a way that mimics human behaviour [8]. Generative AI (GenAI) is a specialised subset of machine learning that focuses on creating new data or content based on patterns learnt from existing data. It is a category of artificial intelligence that focuses on creating new content similar to that of humans in text, images, audio and other forms of content [3]. In the context of translation, GenAI has revolutionised the field of AI by enabling machines to generate human-like text and perform complex language-related tasks. For example, GenAI can use large language models (LLMs) such as GPT (Generative Pre-trained Transformer), such as ChatGPT, which are trained on extensive datasets using deep learning techniques [7, 9]. These models can produce text that closely resembles natural human language [10]. They can continuously learn from new data and improve their performance over time. This means that they can handle increasingly complex tasks, including understanding and generating language that accommodates nuances [10]. As GenAI evolves, it is being applied in more diverse and sophisticated ways, addressing challenges in communication and automation that were previously

difficult to solve with traditional AI methods. These developments highlight the growing importance of GenAI in shaping the future of technology in multiple industries [11]. GenAI plays a crucial role in the development of neural machine translation (NMT) systems, improving the ability of these models to produce more accurate and contextually relevant translations using advanced deep learning techniques [4].

2.2 Common AI Techniques in Language Translation

Deep Learning. Deep learning forms the backbone of modern AI translation systems. It can enable these systems to learn from vast amounts of multilingual data with high accuracy. This has significantly advanced the field of translation, moving beyond simple word-for-word translation to a more holistic understanding of the text, such as resolving ambiguities of complex sentences [9, 12].

Transfer Learning. Transfer learning involves pre-training a model on a large dataset and then fine-tuning it on a smaller, task-specific dataset. This approach is particularly effectively applied in practical language translation, where models benefit from the extensive knowledge gained during pre-training [13]. Google's BERT and BERT and OpenAI's GPT models utilise transfer learning to improve their performance on specific tasks, including translation, in order to achieve greater accuracy and better generalisation [14].

Attention Mechanisms. Attention mechanisms play a critical role in making neural translation models more accurate. First introduced in neural machine translation (NMT), these mechanisms help the models to focus on the most important parts of the input sentence, ensuring that keywords and phrases are given the right emphasis during translation. It employs dynamic contextual understanding, meaning that as the model generates each word in an output sentence, it continuously looks back at the input sequence and determines which words to focus on to produce a more accurate and clear sentence in the target language. This leads to translations that are more accurate and more appropriate for the context [13, 15].

3 Popular AI Translation Tools

3.1 Google Translate

Google Translate is one of the most widely used AI translation tools that uses NMT to provide translations in numerous languages. The system uses deep learning to understand context and generate more accurate translations compared to earlier phrase-based models [16]. Google has continuously improved its NMT system by including feedback and expanding its language support [17]. Recent updates have further enhanced its ability to handle idiomatic expressions and contextual nuances, making it a powerful tool for both laypeople and professionals for example, Google Translate can accurately translate colloquial expressions in some languages [16].

3.2 DeepL

DeepL is renowned for its high-quality translations, which are often considered superior to other translation tools. It uses deep neural networks that are optimised specifically for translation tasks. DeepL models are trained on a large amount of multilingual data, allowing them to produce fluent and natural-sounding translations. The system also excels at maintaining the context and tone of the original text, making it a preferred choice for users who require precise and nuanced translations, such as industry, academic and legal professionals [18].

3.3 ChatGPT

ChatGPT, developed by OpenAI, is a flexible language model that can handle various tasks, including language translation. Based on the Transformer architecture, it is pre-trained on varied datasets and fine-tuned for specific applications [19, 20]. ChatGPT excels in conversational translation, understanding context and maintaining the flow of dialogue. This makes it particularly useful for real-time communication and situations where nuanced understanding is crucial [21]. Its ability to generate contextually accurate translations highlights the potential of AI-driven models to improve multilingual communication. ChatGPT's language models have set new standards in conversational AI, significantly impacting machine translation and natural language processing (NLP) [10]. Future AI translation tools are likely to build on advances in large language models, continuing the trend set by technologies such as ChatGPT [10].

4 Methodology

4.1 A Systematic Literature Review

A systematic literature review (SLR) is a methodology that identifies, evaluates and synthesises all previous research relevant to a particular research question, topic area, or phenomenon of interest since it allows researchers to have a reliable method to gain clear, reasonable and unbiased information on the topic [22]. The primary objective of an SLR is to provide a reliable and consolidated overview of existing research, identify gaps, and suggest directions for future research [23]. The significance of an SLR lies in its ability to deliver evidence-based conclusions that can influence both academic research and practical applications. This is especially relevant in fields such as AI and language translation, where rapid developments and their role in society are significant [24, 25]. This study uses an SLR to examine the role of AI in modern language translation and its broader social applications. The structured and systematic nature of an SLR is particularly suitable for exploring this intersection, allowing an in-depth examination of the topic for this study [26].

4.2 Research Design

The research design followed a systematic review protocol based on the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) checklist, which

ensured a structured and replicable approach to the study. This process involved several key stages: defining the research question, developing a search strategy, conducting the literature search, screening the identified studies, assessing the quality of the studies, extracting data and synthesising the findings [23, 24]. The main aim of this research was to explore the role AI has played in enhancing language translation technologies and its societal implications to provide an understanding of its current state, usage and future prospects. To address this, these research questions guided this review: (i) What are the recent advances in AI-driven language translation technologies and how have they improved the accuracy and efficiency of translation tasks? (ii) What are the societal implications and usage of AI-based language translation technologies? (iii) What are the future trends and potential developments in AI language translation?

Search Strategy for Inclusion and Exclusion Criteria for Selection of Publications. A well-defined search strategy was used to identify studies relevant to the application of AI in language translation and its societal implications. The search was conducted using the Scopus database, chosen for its extensive coverage of peer-reviewed literature. The search string in Table 1 was used:

Table 1. Search string used to identify records

TITLE-ABS-KEY ("Artificial Intelligence" OR "AI") AND TITLE-ABS-KEY ("Translation") AND TITLE-ABS-KEY ("Society" OR "Community") AND TITLE-ABS-KEY ("Application" OR "Impact" OR "Effect" OR "Language")

This search string was carefully crafted to encompass the broad spectrum of recent developments in AI technology with respect to its applications in language translation while focussing on social implications [18, 27].

Search Results and Screening Process. The search for articles was carried out in mid-July 2024. The initial search based on the search string yielded 351 publications. The selection process was refined through the following steps [28]: (i) Publication period: Publications from 2020 to 2024 were selected to focus on recent developments narrowing the selection to 239 papers. (ii) Full-text review: Publications that were not full-text were excluded, resulting in 221 papers. (iii) Selection of article type: Only peer-reviewed journal articles were considered in English, further reducing the selection to 81 articles. (iv) Access Filtering: Articles that were accessible in full text were included, leading to a selection of 52 articles. (v) Relevance Check: Finally, the remaining articles were scrutinised for their relevance to the study objectives or methodologies, resulting in 16 articles being selected for the final review. Publication Period: Publications from 2020 to 2024 were selected to focus on recent developments, narrowing the selection to 239 papers.

Inclusion and Exclusion Criteria for the Selection of Publications. To ensure that only the most relevant and high-quality studies were included, specific inclusion and exclusion criteria were established. Inclusion Criteria: Peer-reviewed journal articles, Publications focusing on AI in language translation, Articles published in English and

available in full text, Open-access articles and research published between 2010 and 2024 to capture recent advancements. Exclusion Criteria: Non-peer-reviewed articles, book chapters, editorials and opinion pieces.; Studies not directly related to AI or translation technologies; Abstract-only publications; Articles not available in full text; Articles not published in English. These criteria were rigorously applied at multiple stages of the selection process to ensure that the final selection of articles was relevant and of high quality [5].

Quality Assessment. The quality of the selected studies was assessed using the Critical Assessment Skills Programme (CASP) checklist [29]. This tool facilitated a comprehensive evaluation of the rigour, credibility and relevance of each study. The checklist covered several critical aspects, including (i) Clarity of the research objectives, (ii) Appropriateness of the methodology, (iii) Rigour in data collection and analysis, (iv) Validity and reliability of the findings, and (v) Relevance to the research question. Only studies that met the quality threshold were included in the final synthesis, ensuring that the conclusions drawn from the review were based on reliable and high-quality evidence [30].

PRISMA Flow Diagram. The selection process is summarised in the PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses [31] flow diagram Fig. 1), which visually represents the stages of identification, screening, eligibility and inclusion.

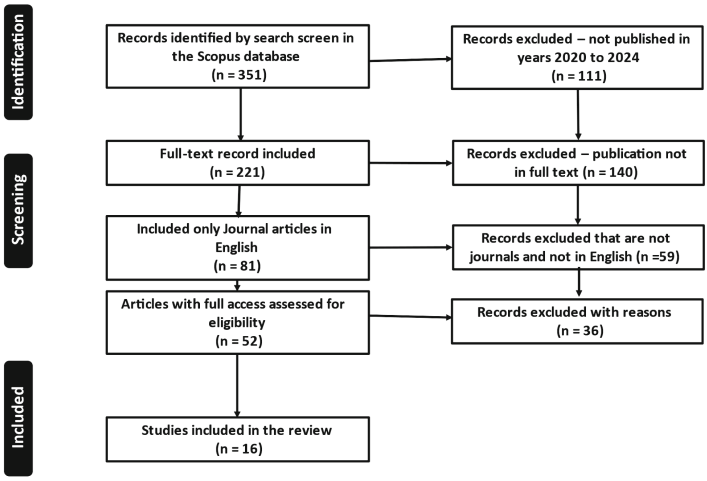


Fig. 1. PRISMA flow flowchart for the selection process of the study

The diagram illustrates the progression from an initial pool of 351 records to the final 16 studies included in the review. Each stage of the screening process was meticulously documented to ensure transparency and reproducibility [32].

5 Results and Discussion

This section presents the results of the 16 articles analysed, organised according to the research questions. First, the section presents the criteria for the journal articles, namely, year of publication, reference, objective (s), sector/domain, approach used and method (s) used (see Table 2). The main aim of this research was to determine how AI is improving language translation technologies, what it means for society and its societal implications to determine a proper understanding of its current state and future. The discussion is structured around three key research questions: (i) What are the recent advances in AI-driven language translation technologies and how have they improved the accuracy and efficiency of translation tasks? (ii) What are the societal implications of AI-based language translation technologies? (iii) What are the future trends and potential developments in AI language translation?

Table 2. The summary of the characteristics of the reviewed studies

Year	Reference	Objectives	Sector/ Domain	Approach	Method
2022	[33]	To explore expert opinions on the future of translation technologies, particularly neural machine translation (NMT) To identify key research areas and challenges in the development of translation technologies To assess the implications of AI in translation for professional practices and training	Translation Studies, AI Technology	Qualitative	Expert Opinion Survey
2021	[34]	To compare the accuracy and efficiency of human translation versus machine translation To assess the strengths and weaknesses of machine translation in various linguistic contexts To identify areas where human expertise is still crucial in translation tasks	Language Translation, AI Technology	Qualitative	Contrastive Analysis

(*continued*)

Table 2. (*continued*)

Year	Reference	Objectives	Sector/ Domain	Approach	Method
2020	[35]	To improve the effectiveness of machine translation by developing a statistical model based on deep neural networks To address issues such as insufficient bilingual data and feature representation in machine translation	AI Technology, Translation	Quantitative	Model development and experimental testing
2024	[36]	To evaluate the effectiveness of ChatGPT in translating kidney transplant FAQs from English to Spanish To assess the linguistic accuracy and cultural sensitivity of AI translations in medical contexts To explore the potential of AI-driven translations in promoting health equity among Spanish-speaking populations	Healthcare, AI Technology, Translation Studies	Experimental	Evaluation of AI-generated translations (GPT-3.5 and GPT-4.0)
2023	[37]	To explore how AI identifies and interprets emotions from text during communication To assess the impact of AI emotion identification on the quality of interpersonal communication To analyse the potential risks and benefits of AI-mediated emotion recognition in social interactions	Technology, AI Development, Communication	Experimental	Emotion Recognition Analysis
2024	[38]	To examine how AI chatbots reproduce biases in global conservation efforts To assess the potential risks of AI-driven content in shaping conservation strategies	Environmental Conservation, AI Technology	Qualitative	Analysis of AI-generated responses
2021	[39]	To investigate public attitudes towards artificial intelligence and its applications To identify factors that influence people's perceptions of AI technologies To assess the implications of public attitudes on AI adoption and policy-making	Technology, Society, Public Policy	Quantitative	Surveys, Statistical Analysis

(*continued*)

Table 2. (*continued*)

Year	Reference	Objectives	Sector/ Domain	Approach	Method
2024	[40]	To explore Hungarian higher education students' perceptions of AI literacy To identify the key components of AI literacy as a critical skill in the 21st century To analyse the challenges and benefits of AI tools in educational settings	Education, AI Technology, Education	Qualitative	Semi-structured Interviews
2024	[41]	To develop a deep learning framework for recognising Arabic Sign Language To evaluate the framework's accuracy and efficiency in translating Arabic Sign Language into text To explore the potential of deep learning in enhancing communication accessibility for Arabic-speaking deaf communities	Technology, Accessibility, AI Development	Experimental	Framework Development and Testing
2022	[42]	To develop a real-time AI-based translator that converts speech-to-text and text-to-sign language for South African official languages To improve communication accessibility for the deaf and hard-of-hearing communities To integrate AI with linguistic and cultural nuances specific to the South African language	Healthcare, Technology, Accessibility	Experimental	System Development and Testing
2024	[43]	To explore how AI technologies are reshaping the field of language translation To assess the impact of AI on translation accuracy, efficiency and accessibility	Technology, Language Translation, AI Development	Analytical	Case Studies, Comparative Analysis
2024	[44]	To investigate the effects of AI-mediated communication on language use and social relationships To assess the potential benefits and drawbacks of using AI in everyday communication	Technology, Communication, Social Science	Experimental	Randomised Experiments, Surveys

(*continued*)

Table 2. (*continued*)

Year	Reference	Objectives	Sector/ Domain	Approach	Method
2020	[45]	To evaluate the presence of gender bias in machine translations generated by Google Translate To analyse how gender biases in machine translation affect the translation outputs and their implications for users	Technology, Gender Studies	Qualitative	Case study analysis
2021	[46]	To develop a spelling correction algorithm for real-time American Sign Language (ASL) recognition systems To improve the accuracy of ASL-to-text translations in real-time communication To evaluate the effectiveness of the algorithm in reducing errors in ASL recognition	AI Technology, Accessibility, Sign Language, Language Translation	Experimental	Algorithm Development and Testing
2020	[47]	To investigate the public discourse around Artificial Intelligence and its translation across different media and contexts To identify common myths and justifications used in AI discussions and how they are conveyed in translations	Media, Technology, Communication Studies, Media	Qualitative	Discourse analysis
2020	[48]	To reassess the claim of human-machine parity in translation, propose better evaluation designs, and offer recommendations for human-machine parity assessments To reassess human-machine parity, propose better evaluation designs, recommend assessment practices	AI Technology, Human-computer interaction	Qualitative	Expert evaluation

5.1 Recent Advancements in AI-Driven Language Translation Technologies

Recent advances in AI-driven language translation technologies have significantly improved both the accuracy and efficiency of translation tasks. One of the most significant developments is the adoption of Neural Machine Translation (NMT), which has greatly improved translation accuracy by understanding context, semantics and syntax more effectively than previous models [33]. This is supported by advances such as transformer-based models, including GPT and BERT (Bidirectional Encoder Representations from Transformers), which have established new benchmarks in translation quality [25]. Advanced NMT models deep encoding information has improved translation efficiency [45]. Advances in deep learning and self-attention techniques have helped NMT models better understand grammar and sentence structure, leading to improved

translations of longer texts [41, 45]. However, challenges remain, especially in handling complex linguistic nuances [34]. In sign language translation, the Swin Transformer architecture has significantly improved recognition accuracy across various tasks, demonstrating AI's potential in specialised language domains [35]. In addition, advances in AI in real-time translation have made live communication more accessible and efficient, particularly in multilingual contexts [36, 41, 46]. Despite these advances, the need for continuous improvement remains, especially in addressing the cultural relevance and emotional accuracy of translations [37].

5.2 Societal Implications of AI-Based Language Translation Technologies

The societal implications of AI-based language translation technologies are profound and affect various aspects of global communication, cultural exchange, and ethical considerations. AI-driven translation technologies have bridged language barriers, facilitating international business, diplomacy and cultural exchange [30]. A new American Sign Language alphabet translator uses AI models like YOLO ((You Only Look Once) to make real-time translation more accurate. It includes spelling correction and achieves an average accuracy of 98.07% [46]. AI-based real-time speech-to-text sign language translators have also been suggested as a good way to reduce communication barriers, especially in the medical field, such as the times during the COVID-19 pandemic[42] However, these technologies also pose significant risks, such as the erosion of linguistic diversity as AI tends to favour more dominant languages, potentially leading to cultural homogenisation [38]. Furthermore, while AI translations have improved access to information for non-English speakers, promoting inclusivity, they have also raised concerns about job displacement among human translators, especially in standard translation tasks [25, 34]. Ethical considerations extend to issues of fairness, accountability and the reinforcement of Western-centric narratives, which may marginalise diverse global perspectives [38]. Current ethical discussions are often shaped by how AI is presented and justified in specific contexts rather than focusing on clear technical solutions. This indicates the need for balanced and better collaboration between ethics and technical fields, ensuring that ethical considerations are not overshadowed or used strategically without being integrated into the technical aspects of AI development [47]. In the medical field, AI translations have improved health equity by improving access to crucial information for non-English speakers, although cultural sensitivity remains a critical area for improvement [36]. In the education sector, especially in higher education, AI translation tools have started to be utilised to efficiently perform academic tasks, which benefits multicultural institutions. However, they still face risks such as generalisation of responses and producing inaccurate information. Therefore, they need to be critically evaluated before being fully implemented [40].

5.3 Future Trends and Potential Developments in AI Language Translation

The future of AI language translation is likely to involve several significant trends and developments. There will be a push towards more culturally sensitive AI models that incorporate nuances specific to different languages and cultures, particularly in idiomatic expressions and culturally specific content [37]. The development of universal language

models that can understand and translate any language, including low-resource and endangered languages, is another key trend [25]. Furthermore, AI translation is expected to become increasingly integral to augmented and virtual reality experiences, providing seamless multilingual interactions in immersive environments [30]. Ethical considerations will also play a crucial role, with a growing focus on developing AI systems that are transparent, fair and accountable, addressing biases and promoting responsible technology use [39]. Recent studies have emphasised how the emphasis on contextualised translation methodologies should be considered for complex discursive and performative differences that arise during the cultural application of AI technologies[47]. Integrating AI in clinical settings to support diverse linguistic communities and other contexts further highlights the potential of AI to revolutionise language translation in the future [35, 36]. This indicates a growing need for balanced collaboration between ethics and technical fields to ensure that ethical considerations are fully integrated into AI translation developments rather than being treated as a supplementary aspect [47]. It has been proposed that higher education institutions should systematically include AI translation literacy and capabilities to prepare them for a multicultural future of AI-driven technologies [40].

6 Limitations and Future Research

Like many studies, several limitations were identified. In the field of information technology, advancements occur at a rapid pace. Due to this, many new developments are presented at conferences, as journal articles often take longer to publish. The SLR could have included more recent developments if both journal articles and conference proceedings were considered, rather than relying solely on journals. Another limitation is that only one database, Scopus, was used to source the literature. A more comprehensive approach would include other well-established databases, such as Web of Science and ScienceDirect. Lastly, the study could have been strengthened by identifying a greater number of relevant articles that were qualified for the review. These limitations should be considered in future research.

7 Conclusion

The systematic literature review aimed to investigate the role of artificial intelligence in improving language translation technologies and their social implications between 2020 and 2024, the last five years. The results show many advancements in AI-driven translation, particularly with the adoption of neural machine translation (NMT) and transformer-based models. These have substantially improved accuracy and efficiency, as well as the ability to handle idiomatic expressions and cultural nuances. However, the societal implications of these technologies remain an issue, which involves ethical concerns, potential job displacement and the risk of cultural homogenisation. In the future, it is expected that AI language translation technology development will consider culturally sensitive models, universal language frameworks, and ethical AI systems that prioritise transparency and fairness. This will likely result in more culturally sensitive AI models that better capture language-specific content. Addressing these challenges will ensure

that AI-driven translation technologies contribute positively to global communication and culture.

References

1. Aithal, P., Aithal, S.: User invented applications of AI-based GPTs in teaching-learning and research-publications. Int. J. Appl. Eng. Manag. Lett. (IJAEML) **8**(1), 39–55 (2024)
2. Hutchins, J.W., Somers, H.L.: An Introduction to Machine Translation. Academic Press, London (1992). https://doi.org/10.1007/978-0-387-72774-5
3. Alsohybe, N.T., Dahan, N.A., Ba-Alwi, F.M.: Machine-translation history and evolution: survey for Arabic-English translations. arXiv preprint arXiv:1709.04685 (2017)
4. Carl, M.: Translation, artificial intelligence and cognition. In: Schwieter, J.W., Ferreira, A. (eds.) The Routledge Handbook of Translation and Cognition, pp. 500–516. Routledge, London (2020)
5. Al Riyami, M., Abdullah, M.: Systematic Literature Review about Examining the Factors Affecting Oman's Implementation of e-Government. Islamic Finance: New Trends in Law and Regulation, pp. 313–321 (2024)
6. Mareček, D., Celikkanat, H., Silfverberg, M., Ravishankar, V., Tiedemann, J.: Are multilingual neural machine translation models better at capturing linguistic features? Prague Bull. Math. Linguist. **115**, 143–162 (2020)
7. Shehata, B., Whelan, E., Abdellatif, M., Campbell, A.: How are we doing with student-centered learning facilitated by educational technologies? A systematic review of literature reviews. Educ. Inf. Technol. **29**(7), 7813–7854 (2024). https://doi.org/10.1007/s10639-024-11612-4
8. Ramachandran, P., Ninan, S., Kumar, S., Vaidya, A.: Role of artificial intelligence in automatic segmentation of brain metastases for radiotherapy. In: M. Begum, T. Subashini, R. John (eds.) Artificial Intelligence in Cancer Diagnosis and Prognosis, Volume 3: Brain and Prostate Cancer, pp. 4–1–4–23. IOP Publishing, Bristol, UK (2022)
9. Son, J., Kim, B.: Translation performance from the user's perspective of large language models and neural machine translation systems. Information **14**(10), 574 (2023). https://doi.org/10.3390/info14100574
10. Farina, M., Yu, X., Lavazza, A.: Ethical considerations and policy interventions concerning the impact of generative AI tools in the economy and in society. AI Ethics 1–9 (2024)
11. Zaghlool, Z.D., Khasawneh, M.A.S.: Incorporating the impacts and limitations of AI-driven feedback, evaluation, and real-time conversation tools in foreign language learning. Migration Letters **20**(7), 1071–1083 (2023)
12. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
13. Bornea, M., Pan, L., Rosenthal, S., Florian, R., Sil, A.: Multilingual transfer learning for QA using translation as data augmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 14, pp. 12583–12591 (2021)
14. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. arXiv preprint arXiv:1801.06146 (2018)
15. Vaswani, A., Huang, L., Chiang, D.: Smaller alignment models for better translations: Unsupervised word alignment with the l0-norm. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers, pp. 311–319 (2012)
16. Prates, M.O., Avelar, P.H., Lamb, L.C.: Assessing gender bias in machine translation: a case study with google translate. Neural Comput. Appl. **32**, 6363–6381 (2020). https://doi.org/10.1007/s00521-019-04144-6

17. Turovsky, B.: Found in translation: More accurate, fluent sentences in Google Translate. Google Blog, 15 November 2016
18. Lai, W., Mesgar, M., Fraser, A.: LLMs Beyond English: Scaling the Multilingual Capability of LLMs with Cross-Lingual Feedback. arXiv preprint arXiv:2406.01771 (2024)
19. Hassani, H., Silva, E.S.: The role of ChatGPT in data science: how AI-assisted conversational interfaces are revolutionizing the field. Big Data Cogn. Comput. **7**(2), 62 (2023)
20. Yenduri, G., et al.: GPT (Generative Pre-trained Transformer) – A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. IEEE Access (2024)
21. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI Blog **1**(8), 9 (2019)
22. Amaro, V., Pires, M.J.: Found in translation, lost in education: Artificial intelligence's impacts on translation tertiary education in Macao. Asian Educ. Dev. Stud. 1–15 (2024)
23. Kitchenham, B., Charters, S.: Guidelines for performing systematic literature reviews in software engineering. Technical report (2007), UK
24. Pan, H., et al.: Comprehensive evaluation methods for translating BCI into practical applications: Usability, user satisfaction, and usage of online BCI systems. Front. Hum. Neurosci. **18**, 1429130 (2024)
25. Masuma, F., Chakraborty, P., Islam, A.A.U., Talukder, P.C., Roy, P., Yousuf, M.A.: Measuring bias in generated text using language models—GPT-2 and BERT. In: Fortino, G., Kumar, A., Swaroop, A., Shukla, P. (eds.) Proceedings of Third International Conference on Computing and Communication Networks. ICCCN 2023. LNNS, vol. 917, pp. 495–598. Springer, Singapore (2024). https://doi.org/10.1007/978-981-97-0892-5_39
26. Aruleba, K., Jere, N.: Exploring digital transforming challenges in rural areas of South Africa through a systematic review of empirical studies. Sci. Afr. **16**, e01190 (2022)
27. Zuhri, R.S., et al.: Information communication technologies education in elementary school: a systematic literature review. J. Educ. Learn. (EduLearn) **18**(3), 1078–1090 (2024)
28. Ramadhan, A., Warnars, H.L.H.S., Razak, F.H.A.: Combining intelligent tutoring systems and gamification: a systematic literature review. Educ. Inf. Technol. **29**(6), 6753–6789 (2024)
29. Long, H.A., French, D.P., Brooks, J.M.: Optimising the value of the critical appraisal skills programme (CASP) tool for quality appraisal in qualitative evidence synthesis. Res. Methods Med. Health Sci. **1**(1), 31–42 (2020)
30. Xia, Y.: Research on statistical machine translation model based on deep neural network. Computing **102**(3), 643–661 (2020)
31. Sarkis-Onofre, R., Catalini, R., Knackmuss, A., Borges, G.A.: How to properly use the PRISMA statement. Syst. Rev. **10**, 1–3 (2021)
32. Rethlefsen, M.L., Page, M.J.: PRISMA 2020 and PRISMA-S: common questions on tracking records and the flow diagram. J. Med. Libr. Assoc. (JMLA) **110**(2), 253–263 (2022)
33. Sánchez-Gijón, P.: What experts say about increasingly relevant translation technologies. Tradumàtica Tecnologies de la Traducció **20**, 295–301 (2022)
34. Briva-Iglesias, V.: Human Translation vs. Machine Translation: A Contrastive Analysis and Factors Involving Machine Translation Use for Legal Translation. Technical Report (2021)
35. Kumar, A., Verma, S., Gupta, R., Singh, K.P.: Next-Gen mulsemedia: virtual reality haptic simulator's impact on medical practitioner for higher education institutions. Electronics **12**(2) (2023)
36. Garcia Valencia, O.A., Mejia, S.P., Solis, J., Escalante, P.: AI-driven translations for kidney transplant equity in Hispanic populations. Sci. Rep. **14**(1), 8511 (2024)
37. Hohenstein, J., O'Sullivan, D., Shepherd, H., Masek, C.: Artificial intelligence in communication impacts language and social relationships. Sci. Rep. **13**(1), 5487 (2023)
38. Urzedo, D., Binkowski, L.J., Marshall, K., Abrahams, M.: AI chatbots contribute to global conservation injustices. Humanit. Soc. Sci. Commun. **11**(1), 1–8 (2024)

39. Sindermann, C., Simon, S., Montag, C.: Assessing the attitude towards artificial intelligence: introduction of a short measure in German, Chinese, and English language. KI-Künstliche Intelligenz **35**(1), 109–118 (2021)
40. Folmeg, M., Fekete, I., Koris, R.: Towards identifying the components of students' AI literacy: an exploratory study based on Hungarian higher education students' perceptions. J. Univ. Teach. Learn. Pract. (2024)
41. Abdul Ameer, R.S., Ahmed, M.A., Al-Qaysi, Z.T., Salih, M.M., Shuwandy, M.L.: Empowering communication: a deep learning framework for Arabic sign language recognition with an attention mechanism. Computers **13**(6) (2024)
42. Madahana, M., Kabiru, S., Wekesa, C., Mwaura, G.: A proposed artificial intelligence-based real-time speech-to-text to sign language translator for South African official languages. South African J. Commun. Disorders **69**(2) (2022)
43. Mohamed, Y.A., Bashir, M., Mohamed, A.H.H., Khanan, A., Adiel, M.A., Elsadig, M.A.: The impact of artificial intelligence on language translation: a review. IEEE Access **12**, 25553–25579 (2024)
44. Wang, L., Smriti, D., Yuan, H., Huh-Yoo, J.: Artificial intelligence systems for supporting informal caregivers of people living with Alzheimer's disease or related dementias: a systematic review. In: Extended Abstracts of the CHI Conference on Human Factors in Computing Systems, pp. 1–11 (2024). https://doi.org/10.1145/3613905.3650846
45. Duan, G., Yang, H., Qin, K., Huang, T.: Improving neural machine translation model with deep encoding information. Cogn. Comput. **13**(4), 972–980 (2021)
46. Rivera-Acosta, M., Ruiz-Varela, J.M., Ortega-Cisneros, S., Rivera, J., Parra-Michel, R., Mejia-Alvarez, P.: Spelling correction real-time American sign language alphabet translation system based on YOLO network and LSTM. Electronics **10**(9), 1035 (2021)
47. Roberge, J., Senneville, M., Morin, K.: How to translate artificial intelligence? Myths and justifications in public discourse. Big Data Soc. **7**(2) (2020). https://doi.org/10.1177/2053951720919968
48. Läubli, S., Ott, M., Pilla, S., McCarthy, A.: A set of recommendations for assessing human-machine parity in language translation. J. Artif. Intell. Res.Intell. Res. **69**, 177–196 (2020)

Deep Learning Forecasting of Photovoltaics Output Using Digital Twin Data

Nomfundo Vilakazi and Terence van Zyl(✉)

Institute for Intelligent Systems, University of Johannesburg, Johannesburg, South Africa
200713297@student.uj.ac.za, tvanzyl@uj.ac.za

Abstract. Solar power is an accessible form of renewable energy, especially in South Africa, where solar radiance is concentrated. However, this source of energy is not constant in supply. Its stochastic nature can lead to fluctuations in the voltage and frequency in the energy grid. Accurate forecasting of the availability of solar power is crucial for operational efficiency and distribution of power. Machine learning models provide a powerful forecasting tool for solar photovoltaic power. Due to the inaccessibility of accurate and measured solar irradiance data, this study uses a digital twin to generate meteorological attributes to predict the solar output power of the University of Johannesburg solar photovoltaic plant based in Auckland Park, Johannesburg. The synthetic dataset contains hourly data over a calendar year. The deep learning algorithms that were used for prediction are the Long-Short Term Memory (LSTM) and Recurrent Neural Network. The findings showed that the LSTM is the best predictor using the Root Mean Squared Error (RMSE), Mean Squared Error (MSE) and Mean Absolute Error (MAE) performance metrics. This study is crucial as it supports the Sustainable Development Goals (SDG) aim to reduce carbon emissions, provide clean energy and improve access to electricity.

Keywords: photovoltaics · digital twin · power prediction

1 Background of Study

1.1 Introduction

The introductory chapter of this study outlines the motivation for this study endeavour and what it aims to achieve. It is divided into five (5) key sections. The first section delves into existing literature; the second section discusses the current challenges; the third section highlights the significance of the study; the fourth section outlines the study's objectives; and the fifth section highlights the current and preempted limitations.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 405–419, 2025.
https://doi.org/10.1007/978-3-031-78255-8_24

1.2 Literature Review

The Sustainable Development Goals (SDGs) aim to reduce carbon emissions, provide clean energy and improve access to electricity [1]. Furthermore, countries signed a legally binding international treaty called the Paris Climate Agreement, in which they commit to reducing carbon emissions by 45% by the year 2030 and achieving net zero by 2050 [2]. The above macro-scale goals demonstrate a clear global drive towards cleaner energy sources. On a local scale, at present, the main source of base load power in South Africa is from nonrenewable (i.e. nuclear) or fossil fuels (i.e. coal) sources [37]. These unsustainable sources are fleeting and lead to environmental degradation [24]. The need to convert the base load to a more sustainable power source is becoming more imminent as the impacts of climate change are more readily seen.

Solar PV is the fastest-growing renewable energy and is forecast to be responsible for almost 60% of all renewable capacity expansion globally over 2018–2024 [3]. Theoretically, the amount of solar energy that touches the earth is 4200 times that the human population will consume in 2035 [14]. This excess means that focusing on developing this energy source can solve the world's energy demand. More so in South Africa, where energy is currently lacking. The amount of solar energy that energy providers can collect depends on the geographical location [14], which determines the amount of solar irradiance concentrated in that area.

Accurate solar resource data, namely, Direct Solar Irradiance (DNI) and Global Horizontal Irradiance (GHI), are needed for accurate planning, engineering and financing of a solar energy project [11]. Site maps acquired from Solar-Gis [4] and the Global Solar Atlas [5] show that South Africa has, on average, a DNI of 2687.1 kWh/m^2, GHI of 2156.9 kWh/m^2 and a specific PV power output of 1919.6 kWh/kWp. These statistics make South Africa a good place for concentrating solar power [13]. PV systems take solar energy and convert it into usable electrical energy. They do not add to the pollution of the environment, and they are relatively straightforward to install and use either commercially or in residential use cases [39].

The use of solar PV panels in converting solar energy into electrical energy is advantageous because they are recyclable, reliable and considered a clean source of energy [14]. The cost to operate and maintain the PV panels is lower relative to other renewable sources [16]. This makes solar PV more desirable for further exploration and consideration.

1.3 Current Challenges

It would be remiss to exclude the challenges of using solar PV for power. In [14], four weaknesses are highlighted: Firstly, the dependence on the availability of solar radiation, which is only available during the daytime. This calls for a UPS, which increases the cost of installation. Secondly, the panels are inefficient at converting solar energy into usable energy. According to [29], the highest conversion efficiency in research on crystalline silicon PV cells is 25%. The inefficiency

means more panels are required, thus more space to meet the demand. Thirdly, the overheating of panels further reduces the efficiency requiring more panels and more space to meet the energy demand. Fourthly, the initial investment cost is quite high. These are just some of the challenges that have been studied previously. This paper will expand upon the following challenges: intermittency, solar data availability and shadowing.

Intermittency. A well known challenge when it comes to solar PV is that of intermittency. An "Energy Quality Framework" was defined to categorize and measure variations in renewable power generation [38]. The main variations are power fluctuations and power intermittency. For solar energy, intermittency is considered far more challenging than power fluctuations. Intermittency refers to long term power variations on the timescale of hours to days to years. The effect translates to solar power, which shows intermittency in timescales of hours to months; this hurts the stability and reliability of power grids [38].

This challenge has often raised the importance of accurately forecasting [22] solar PV output to limit major fluctuations in the grid. The Duck Curve, which illustrates the power production over a day, indicates that a steep ramp rate is required on the conventional generation units to meet the fluctuations in demand due to intermittency of the system in circumstances where PV generation used in conjunction with conventional power supply or with other renewable alternatives [23]. As a result, it becomes critical to have accurate short and long-term PV generation forecasts for grid stabilisation and more efficient operation [18].

Through advances in computing over the years, using deep learning techniques to predict solar power output has been studied comprehensively. In [17], deep learning is defined as a sub-field of machine learning that deals with algorithms that closely resemble an oversimplified version of the neural network system of the human brain. It is characterized by machine learning the appropriate representations of data from raw data and can conclude without explicit feature engineering. This can also be called automatic feature engineering. This approach towards machine learning allows for faster computation of complex and large datasets. The section below presents studies implementing the deep learning approach towards forecasting solar PV power.

Study [12] proposes using the deep learning method TESDL for reliable forecasts for short-term forecasts of PV power output. [35] presents the use of Support Vector Machine (SVM), Random Forest (RF) and Gradient Boosting Machine (GBM) as machine learning algorithms to produce predictions of a 99% accuracy using the AUC performance metric. [39] indicates the most accurate and reliable model in that study to be the Matern 5/2 GPR algorithm in comparison to the SVM using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE) and R^2 as performance metrics which yielded results of 7.967, 5302 and 0.98 respectively.

In [15], the Long-Short Term Memory Neural Network (LSTM NN) is demonstrated to have superior forecasting performance for short-term forecasting (24 h); the performance was measured using Mean Absolute Percentage Error

(MAPE), RMSE and MAE, which yielded results of 22.31, 0.71 and 0.36 respectively. To shorten training time and achieve accuracy, models such as GBM have been researched, [10] shows that for studies where meteorological data is the input (with more than one feature), computational speed becomes important. The type of GBM referred to as Extreme Boosting Machine (XGboost) in this study, is demonstrated to produce high forecast performance with an accuracy of 93.56%, MAE 35.82, RMSE 81.70 and MSE 6689.60. The memory usage during computation is relatively low. Furthermore, [21] proposes using XGBoost for very short-term power prediction.

Solar Data Availability. Another well-established challenge is the availability of accurate, measured data for solar radiance for a specific region of interest. Solar radiance is the most widely used attribute for predicting PV output using machine learning algorithms. The study [28] uses solar irradiance and temperature to forecast PV output. There are various other studies, i.e. [27], which state irradiance as the most important factor in forecasting PV panel output. However, irradiation data is not easy to acquire for a specific location. Ground-based measurements require well-calibrated, accurate instruments that take consistent measurements over some time. These measurements are time-consuming and expensive to complete. In addition to consistency in measurements, quality control is paramount and must be closely monitored. Failure in the quality aspect will lead to inaccurate and erroneous data [36].

The widely used solar radiance data such as that obtained from SolarGis and National Solar Radiance Database (NSRDB) [30] is modelled - where measured data is extrapolated to cover smaller areas. Thus, with the use of this data, there are model inaccuracies that should be accounted for. In the case of the NSRDB dataset, inaccuracies may arise because the spatial resolution is 4 km. Where spatial resolution refers to the smallest area visible to the sensor on the satellite [19].

The above challenge (of solar irradiance) has led to research which uses other meteorological attributes to predict PV output with greater accuracy. The study [25] yielded an accurate PV output forecast with R^2 of 0.94 using only weather data (i.e. ambient temperature, humidity), time (i.e. Month) and geographical variables as dependent variables. Weather data, time and geographical data are more accessible. However, it still requires onsite measurements, which is not always available.

Shadowing. The last well-researched challenge for PV solar ouput is that of shadowing. In an urban setting, several buildings and other physical structures (i.e. trees) can shadow the PV module. This can lead to a reduction in efficiency of up to 30% just by a shadow of a single cell [33].

To conclude this section, none of the literature consulted uses a Digital Twin (DT) for Solar PV output prediction. The majority of research papers consulted for this study prioritise solar irradiance as the main input variable for prediction, only a few used meteorological data, time and geographical features for

prediction. Therefore, this study has identified a knowledge gap in using a DT for forecasting solar PV power output. In [34], a DT is defined as a virtual copy or model of any physical entity interconnected via data exchange. As a virtual, simulated model, the DT can yield a more comprehensive dataset with more attributes than actual site measurements. The DT also allows for the modelling of the surrounding structures, which is an opportunity to integrate the effect of shadowing into this study.

1.4 Significance

This study is significant because it provides a time- and cost-effective approach to generating a quality PV solar output dataset for forecasting. This is significant in South Africa, where there are limits to data availability whilst there is a drive towards using renewable energy. This study can also contribute towards the literature on renewable energy. Furthermore, the accurate forecasting of PV output data will contribute towards ensuring better grid stability, thus potentially assisting the operations.

1.5 Study Aim

Advancements in energy modelling software allow for modelling a solar PV plant in it's existing or desired geometry and orientation. Creating a DT makes it possible to generate a synthetic dataset that includes meteorological features, time, geographical features and the solar PV yield.

Ultimately, this study will replace actual onsite PV output data with a simulated IES VE dataset.

The physical PV plant chosen for this study on which the DT will be based is the PV plant located at the parking lot on the University of Johannesburg's(UJ), Kingsway(APK) campus.

1.6 Assumptions

The assumption made in this study is that the drawings provided by the facilities department accurately represent the actual PV plant; no site visitation was conducted to verify this. A further assumption is that the 3rd party PV yield data accurately represents the PV power generated by the plant.

2 Experimental Design

2.1 Data

Four (4) different datasets were collected and used to execute this study. All data collected or produced is described in the subsections below.

Meteorological Data. Historical meteorological data for the City of Johannesburg was obtained from Meteonorm 8. This data set consists of a historical time series of data obtained from 2000–2019 and is populated with meteorological parameters such as temperature, wind, and precipitation, amongst a few [6,32]. This dataset was exported from Meteonorm 8 as an EnergyPlus Weather File (EPW) and used as an input into the IES VE simulation model.

Building Physics Data. To build a digital twin model, obtaining the physical attributes of the building concerned is important. For this study, the UJ facilities management team were requested to provide the following documentation for the APK campus solar plant:

- Engineering drawings which detail the carport layout, dimensions, PV panel arrangement and sizes, site orientation and the surrounding buildings
- Data sheets showing the material properties of the PV panels

Furthermore, the UJ facilities management team provided the actual APK campus PV power yield for 2022. This data was provided in an Excel (CSV) file showing the monthly PV output power produced.

Digital Twin Synthetic Data. The data described in the sections above was used to create the 3D digital twin model on IES VE using the ModelIT application. ModelIT enables 3D model building and associated data [7]. Within IES VE, two separate simulations were run to obtain the desired dataset. The first simulation was a Suncast simulation. In its calculation, Suncast considers the site's position, the building's position, the date and time of day, and shading surfaces such as adjacent buildings [7]. It recognises the effect of solar gains and solar shading in its analysis.

The results of the Suncast simulation were incorporated into the second simulation called the ApacheSim. The ApacheSim is a dynamic thermal simulation tool capable of producing PV solar output data of the solar PV panels modelled [7].

The synthetic dataset produced by the above-mentioned simulations is a database containing hourly PV power yield predictions over a calendar year. The dataset also includes the hourly yield over a calendar year of additional attributes, which could influence the outcomes of this study. The additional attributes that were extracted from IES VE for further study were chosen based on previous literature, which suggested that these attributes may be important. The additional attributes are dry-bulb temperature, wet bulb temperature, wind speed, cloud cover, external relative humidity, direct radiation, global radiation, solar altitude, and diffuse radiation.

People On-Site Data. The final piece of data collected to contribute to this study is to establish a correlation between the PV power produced and the amount of people traffic on campus. The dataset used for this purpose was the

UJ 2022 academic calendar. This data was consolidated into a single spreadsheet with the independent variable Month and Number of days. The dependent variables are academic classes, study breaks, assessments, university holidays, research blocks, and weekends.

2.2 Tools and Libraries

CPU and GPU. The Python script for this study was run in the Google collaboratory (Colab). The Colab hosts a Jupyter notebook and grants free access to computing resources required to run deep learning scripts without having to download or install them onto the PC. The notebook itself is stored in Google Drive. Free access to disk space 107.72 GB and RAM 12.67 GB is provided on the Colab through Python 3 Google Compute Engine.

Forecasting. Building forecasting models can lead to very large models with complex API architectures and libraries. Nixtla is an open-source time-series ecosystem [8] providing pre-established forecasting models which can be implemented in Python without consuming too much GPU and RAM.

The Nixtla forecasting libraries are easily installed into a Python script using the package manager pip command. The Nixtla libraries implemented in this study are the StatsForecast and the NeuralForecast.

StatsForecast is a collective of commonly used univariate time series models such as ARIMA [8]. NeuralForecast is a collective of neural network-based forecasting models such as RNN [8,26]. Using the Nixtla libraries to run the forecasting ensures that the models can be concluded faster. Nixtla libraries also support the integration of exogenous variables. An exogenous variable is one whose variation is explained outside the model [20].

2.3 Method for Implementing Digital Twin

This section illustrates building a 3D model and running simulations on the model within the IES VE. The process began with collecting the actual solar PV plant data as detailed in Sect. 2.1. A digital replica of the actual solar PV was modelled in 3D using the IES VE modelling tool ModelIT. This tool allows for the model to be built to scale in the correct orientation, and it also allows for inputs of the material composition of the building elements modelled. The carport was modelled with six carports, each covered in solar PV panels. Each solar cell is made of Monocrystalline silicon, each panel has 144 cells and dimensions are $2115 \times 1052 \times 35$ mm. Each panel has an anodized aluminium alloy material. The module efficiency is 20%, and the operating module temperature of $-40°$ C to $+85°$C. The model built was subsequently simulated twice as detailed in Sect. 2.1 of this paper. The dataset generated through this process was exported directly into an Excel file.

2.4 Method for Implementing Forecasting Models

Implementing a forecasting model using deep learning requires a structured approach to ensure that the best practices are being employed in the process this is to yield the best outcome given the available data. In [31], a road map was developed by industry leaders for executing a data mining project. This road map is defined as the CRISP-DM Reference Model, which distinguishes the data mining process into six phases: business understanding, data preparation, modelling, evaluation and deployment. The CRISP-DM Reference Model was used as a guide in the methodology applied.

Data Collection. The methods used for data collection and types of data collected are outlined in Sect. 2.1 of this paper. To further describe the synthetic dataset generated in IES VE, the Table 2 below shows the attributes exported and their properties, namely, units of measurement and data type (Table 1).

Table 1. Synthetic dataset attributes and their properties

Attribute	SI unit	Datatype
Dry-bulb temperature	°C	float64
Wet-bulb temperature	° C	float64
Wind speed	m/s	float64
Cloud cover	oktas	float64
External relative humidity	%	float64
Direct radiation	W/m^2	float64
Global radiation	W/m^2	float64
Diffuse radiation	W/m^2	float64
Solar altitude	deg.	float64
PV generated electricity	kW	float64
Datetime		datetime64[ns]

Data Exploration. This study aims to replace the actual data with the synthetic data. Before replacing the one dataset for the other an intermediate data analysis was undertaken to ensure that the synthetic data is a good representation of the actual onsite data. The exploratory analysis was undertaken to explain the variability in PV power production between the synthetic and actual PV output values. The tool used in the analysis is statsmodels. Statsmodels is a Python file that provides built-in classes and functions for statistical data exploration, conducting statistical tests and estimating different statistical models [9].

Data Pre-processing. After exporting from IES VE, the raw synthetic dataset could not immediately be used within the Nixtla libraries. The dataset first needed to be cleaned. The techniques used to clean the data included dropping

and interpolating missing data, correcting the format of the datetime attribute and ensuring that each attribute is associated with the appropriate datatype within Python. Furthermore, the standard form for data in Nixtla libraries requires the following naming convention: the 'Datetime' column be renamed to 'ds', and the 'PV generated electricity (kW)' column be renamed to 'y'. Nixtla also calls for adding a 'unique_id' column, assigning a number to each time series within the data. In the case of this study, only one unique number, '1.0', was required since the data has one time series.

Train Test Split. The data was split into training and testing data. The model was trained on the data from January 1 at 00:00:00 up to November 30 at 23:00:00. The validity of the model was subsequently tested on the testing data, which includes data from December 1 at 00:00:00 up to December 31 at 23:00:00. In the split detailed above, the composition of the training to testing data equates to a 90% proportion used for training data.

3 Results and Discussion

3.1 Results from Digital Twin

The 3D digital model of the APK campus PV plant is shown in Fig. 1 below:



Fig. 1. Digital twin of the APK campus PV plant as modelled in IES VE with overshadowing buildings

The simulated data set produced a PV profile shown in Fig. 2 below. Where the maximum power generated is 779.6392 kW produced at noon in December. The minimum power produced by the digital twin is 0 kW, which occurs at hours without solar radiation (evenings and early mornings). The average power generated by the digital twin in an hour is 138.89 kW. During winter, the PV output is reduced when less solar radiation is experienced.

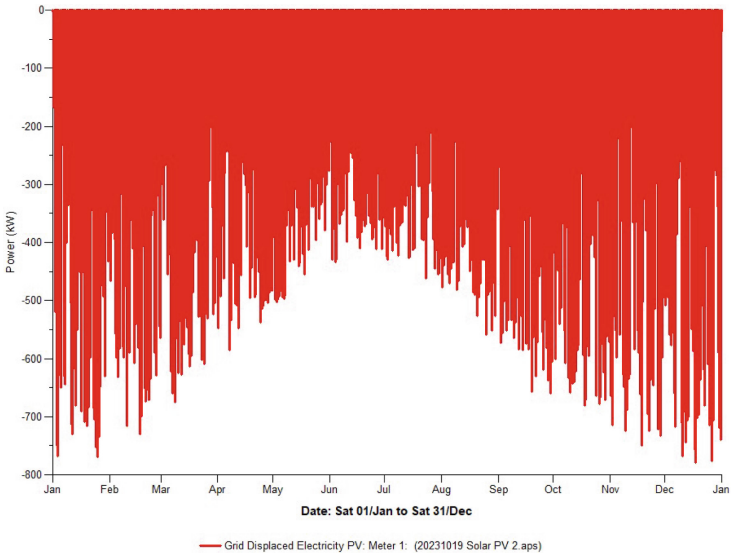


Fig. 2. PV output profile of synthetic data

3.2 Results from Statistical Data Analysis

Before the digital twin data could replace the actual onsite data, steps were taken to ensure that the synthetic data fairly represented the physical data that it replicates. The monthly PV power output for 2022, as seen in the synthetic dataset, follows the same trend on a month-to-month basis as the actual, except for some outliers. The correlation between the two datasets yielded a R^2 value of 0.061, meaning that only six (6) % of the variation in the PV generated (synthetic) is accounted for by the actual (actual) variation. This low value indicates that further investigation is required to understand the reasons.

In its simulation, the digital twin does not account for exogenous factors that can influence the amount of power generated in the plant. The most prominent factor is the academic calendar. The simulated PV values show the plant's full potential, assuming that non-meteorological factors remain constant. In reality, the foot traffic on campus determines the amount of PV power produced and stored onsite. With this in mind, the UJ academic calendar was included in the correlation calculations. The attributes added were the Research days, Academic classes and Assessment days. By including these factors, the R^2 value increased to 0.878, implying that 88% of the variability between the synthetic dataset and the actual dataset can be attributed to the amount of foot traffic on campus.

Given the findings above, this study proceeded with the experimental set-up for forecasting using the synthetic dataset whilst noting that the dataset represents the potential of the PV power plant.

3.3 Attributes Selected for Forecasting

It was mentioned in Sect. 2.4 of this paper that eleven (11) attributes were extracted from the simulation model, including the target variable. However, as suggested in phase three of [31], further data preparation was carried out as data integration. A correlation matrix was completed to assess which attributes have the most influential impact on the target variable "PV generated electricity (kW)". The results of the correlation matrix are shown in Fig. 3 below. The results show a strong correlation between the target variable and Direct Radiation (W/m2), Global Radiation (W/m2), Solar Altitude (deg) and a moderate correlation with the variables Diffuse radiation (W/m2), Dry-bulb Temperature(° C) and External relative humidity (%).

From the aforementioned analysis, only the variables with moderate and high correlation to the target variable were considered valuable towards obtaining the desired outcomes of the study. Therefore, these variables were carried forward as exogenous variables in the forecasting stages of the experimental setup.

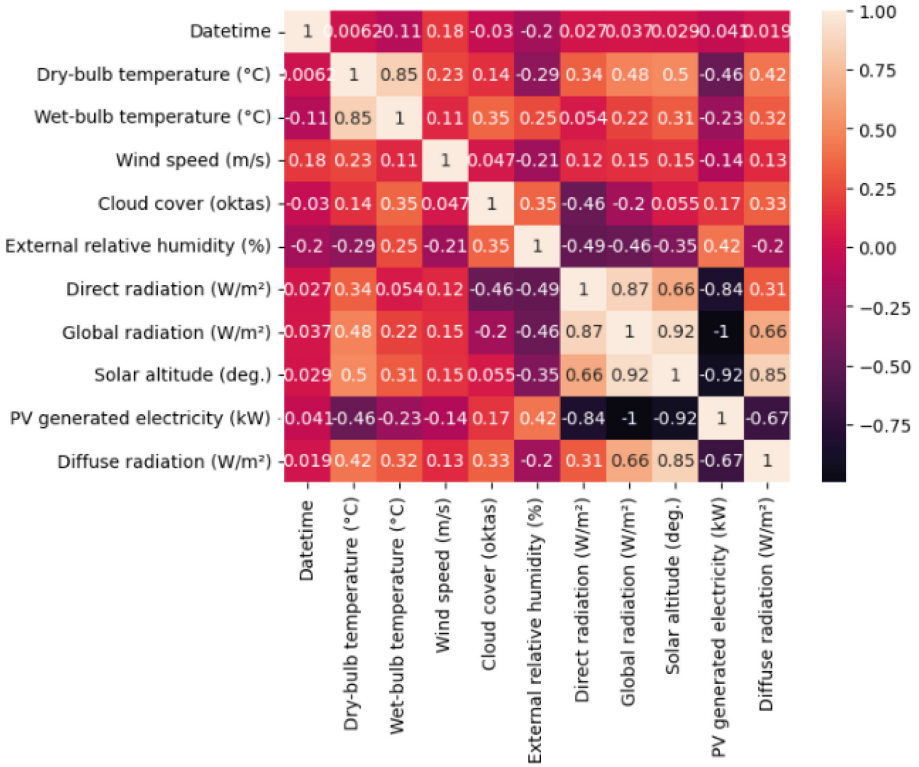


Fig. 3. Results of the correlation matrix

3.4 Forecasting Results Table

Table 2. Model performance results

Model	MSE	RMSE	MAE
LSTM	8347.58	91.36	54.49
RNN	9052.22	95.14	55.67
ARIMA	73410.34	270.0	219.16
Historical Average	53734.46	219.42	181.66
Naive	83803.57	273.51	177.80
RWD	83803.57	273.51	177.80
Seasonal Naive	12916.53	96.64	55.35

3.5 Results from Naive Forecast

The baseline models considered in this study are historical average, Naive, RWD and seasonal naive. The model that yielded the most favourable outcome is the seasonal naive. The dataset considered is highly seasonal. The seasonal naive captures the seasonality best, giving the lowest MSE. The MSE values were quite high, meaning the data points are widely dispersed around the mean. This may be attributed to the basic nature of the models and their failure to include outside factors and seasonality. On the rMSE, the Seasonal Naive produces the best outcome, meaning there are fewer deviations from the actual values.

3.6 Results from ARIMA Forecast

When the model was executed using a slightly more complex and widely applied model, the ARIMA. It was expected to outperform the baseline models since it considers exogenous variables. With a seasonal length set to one (1) for hourly data and the horizon set to 744 (meaning a prediction over one month for December).

The ARIMA outperforms some baseline models, namely, Naive and RWD, using the RMSE and MSE performance metrics. However, the Seasonal naive baseline model outperforms the ARIMA on RMSE, MAE and MSE. From this, we can infer that the seasonality of the data has a greater effect on the outcomes of the analysis than the meteorological exogenous variables considered. The most representative model is the one that can consider the seasonality in the input data.

3.7 Results from LSTM and RNN Forecast

Both the LSTM and RNN models were executed using the entire history for truncated train back-propagation, 200 epochs and including the exogenous variables and an hourly frequency. The LSTM model was the best when considering

the rMSE, MAE and MSE performance metrics, whilst the RNN produced a better MAPE.

Compared to the other methods used above, the LSTM and the RNN performed better than the baseline (naive, RWD, historical average) and statistical (ARIMA) models. Yielding the lowest rMSE values of 91.36 and 95.14, respectively. The MAE values yielded by both deep learning methods are 54.49 and 55.67, respectively; these values surpass the values obtained using the seasonal naive, which yielded 57.48. The deep learning models can account for both the seasonality in the data and the exogenous variables.

4 Conclusions and Recommendations

From the results above, the deep learning models provide an efficient and effective means for predicting PV power. Furthermore, digital twins should also be considered a powerful tool for generating comprehensive data for PV power yield, which can be used for feasibility studies, design purposes or on existing PV solar plants where measured data may not be available or hard to come by.

References

1. https://sdgs.un.org/goals . Accessed 19 June 2023
2. https://www.un.org/en/climatechange/paris-agreement . Accessed 19 June 2023
3. https://www.iea.org/reports/renewables-2019 . Accessed 09 June 2023
4. https://solargis.com/maps-and-gis-data/download/southafrica . Accessed 09 June 2023
5. https://globalsolaratlas.info/map?s=-28.816624,24.991639&m=site&c=-28.816624,24.991639,11 . Accessed 09 June 2023
6. https://meteonorm.com/en/ . Accessed 25 July 2024
7. https://www.iesve.com/software/virtual-environment . Accessed 09 June 2023
8. https://www.nixtla.io/open-source . Accessed 25 July 2024
9. https://www.statsmodels.org/stable/index.html . Accessed 26 July 2024
10. Aksoy, N., Genc, I.: Predictive models development using gradient boosting based methods for solar power plants. J. Comput. Sci. **67**, 101958 (2023)
11. Cebecauer, T., Suri, M.: Site-adaptation of satellite-based DNI and GHI time series: overview and SolarGIS approach. In: AIP Conference Proceedings, vol. 1734. 1. AIP Publishing LLC. 2016, p. 150002
12. Chang, R., Bai, L., Hsu, C.-H.: Solar power generation prediction based on deep learning. Sustain. Energy Technol. Assess. **47**, 101354 (2021)
13. Fluri, T.P.: The potential of concentrating solar power in South Africa. Energy Policy **37**(12), 5075–5080 (2009)
14. Guangul, F.M., Chala, G.T.: Solar energy as renewable energy source: SWOT analysis. In: 4th MEC international conference on big data and smart city (ICBDSC). IEEE 2019, pp. 1–5 (2019)
15. Hossain, M.S., Mahmood, H.: Short-term photovoltaic power forecasting using an LSTM neural network and synthetic weather forecast. IEEE Access **8**, 172524–172533 (2020)

16. Kabir, E., et al.: Solar energy: potential and future prospects. Renew. Sustain. Energy Rev. **82**, 894–900 (2018)
17. Ketkar, N., Moolayil, J.: Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch. Springer, Cham (2021). https://doi.org/10.1007/978-1-4842-5364-9
18. Kuo, W.-C., et al.: Assessment of different deep learning methods of power generation forecasting for solar PV system. Appl. Sci. **12**(15), 7529 (2022)
19. "Chapter 1 - A systematic view of remote sensing". In: Advanced Remote Sensing (Second Edition). Ed. by Shunlin Liang and Jindi Wang. Second Edition. Academic Press, 2020, pp. 1–57. isbn: 978-0-12-815826-5. https://doi.org/10.1016/B978-0-12-815826-5.00001-5. https://www.sciencedirect.com/science/article/pii/B9780128158265000015
20. Lleras, C.: Path analysis. Encycl. Soc. Meas. **3**(1), 25–30 (2005)
21. Ma, Z., et al.: Very short-term renewable energy power prediction using XGBoost optimized by TPE algorithm. In: 2020 4th International Conference on HVDC (HVDC). IEEE 2020, pp. 1236–1241 (2020)
22. Thabang Mathonsi and Terence L van Zyl. "Multivariate anomaly detection based on prediction intervals constructed using deep learning". In: Neural Computing and Applications (2022), pp. 1-15
23. Morrissey, K., Kahrobaee, S., Ioan, A.: Optimal energy storage schedules for load leveling and ramp rate control in distribution systems. In: 2020 IEEE Conference on Technologies for Sustainability(SusTech). IEEE 2020, pp. 1–4 (2020)
24. Omubo-Pepple, V.B., Israel-Cookey, C., Alaminokuma, G.I.: Effects of temperature, solar flux and relative humidity on the efficient conversion of solar energy to electricity. Eur. J. Sci. Res. **35**(2), 173–180 (2009)
25. Pasion, C., et al.: Machine learning modeling of horizontal photovoltaics using weather and location data. Energies **13**(10), 2570 (2020)
26. Perumal, R., Van Zyl, T.L.: Comparison of recurrent neural network architectures for wildfire spread modelling. In: International SAUPEC/RobMech/PRASA Conference, pp. 1–6. IEEE 2020 (2020)
27. Raj, V., et al.: Ensemble machine learning for predicting the power output from different solar photovoltaic systems. Energies **16**(2), 671 (2023)
28. Ramaswamy, S., Sadhu, P.K.: Forecasting PV power from solar irradiance and temperature using neural networks. In: International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS). IEEE 2017, pp. 244–248 (2017)
29. Saga, T.: Advances in crystalline silicon solar cell technology for industrial mass production. NPG Asia Mater. **2**(3), 96–102 (2010)
30. Sengupta, M., et al.: The national solar radiation data base (NSRDB). Renew. Sustain. Energy Rev. **89**, 51–60 (2018)
31. Shearer, C.: The CRISP-DM model: the new blueprint for data mining. J. Data Warehous. **5**(4), 13–22 (2000)
32. Sibolla, B.H., Coetzee, S., Van Zyl, T.L.: A framework for visual analytics of spatio-temporal sensor observations from data streams. ISPRS Int. J. Geo-Inf. **7**(12), 475 (2018)
33. Silvestre, S., Chouder, A.: Effects of shadowing on photovoltaic module performance. Progress Photovolt. Res. Appl. **16**(2), 141–149 (2008)
34. Singh, M., et al.: Digital twin: origin to future. Appl. Syst. Innov. **4**(2), 36 (2021)
35. Subramanian, E., et al.: Solar Power Prediction Using Machine Learning. arXiv preprint arXiv:2303.07875 (2023)

36. Suri, M., et al.: Accuracy-enhanced solar resource maps of South Africa. In: 3rd Southern African Solar Energy Conference, South Africa, 11–13 May 2015
37. Thopil, M.S., et al.: A review of grid connected distributed generation using renewable energy sources in South Africa. Energy Strategy Rev. **21**, 88–97 (2018)
38. Wu, C., Zhang, X.P., Sterling, M.: Solar power generation intermittency and aggregation. Sci. Rep. **12**(1), 1363 (2022)
39. Zazoum, B.: Solar photovoltaic power prediction using different machine learning methods. Energy Rep. **8**, 19–25 (2022)

Responsible and Ethical AI (Philosophy, Law and Humanities)



Requirements Engineering (RE) in Artificial Intelligence (AI) Systems Implementation: The Need to Emphasize Non-Functional Requirements (NFRs) for Ethical AI

Eddie Liywalii[1,2(✉)]

[1] Faculty of Humanities, Department of Philosophy, University of Pretoria (UP), Pretoria, South Africa
u22781197@tuks.co.za

[2] Center for Artificial Intelligence Research (CAIR), Ethics of AI Research Group, Pretoria, South Africa

Abstract. In traditional approaches to develop software systems that do not have an Artificial Intelligence (AI) or Machine Learning (ML) component, requirements determination activities – also called Requirements Engineering (RE) – are well-established and researched. When it comes to building software systems with one or more AI/ML components, the process is dependent heavily on data with limited or in some cases no insight into the non-functional requirements of the system's internal workings such as fairness, accountability and transparency, inter alia. In this paper, I review literature to probe how an emphasis on non-functional requirements during requirements determination can help address ethical requirements when implementing AI systems. The results show that currently technical experts often emphasize functional requirements more than non-functional requirements. Where they do, the emphasis is still on system-oriented non-functional requirements such as portability, maintainability, opacity, inter alia. However, AI being an interdisciplinary subject, encompassed in a bracket of sociotechnical systems, there is need to revisit this tendency by technical experts. I propose a framework for classifying ethical requirements under non-functional requirements in order to ensure equal emphasis on environment-oriented non-functional requirements such as privacy, fairness, eco-friendly among others. I also argue that the use of a value sensitive design (VSD) approach will help in the implementation of ethical AI systems in order to address ethical concerns from stakeholders. In addition, existing technical literature has focused more on using AI to manage RE activities (AI in RE), yet very limited research has been done on RE for AI systems (RE4AI). Requirements engineering is generally accepted as the most critical and complex process within the development of sociotechnical systems. The literature also confirms that during requirements determination for AI systems (RE4AI), an emphasis on non-functional requirements could actually help address some ethical issues that are raised by stakeholders.

Keywords: Ethical Requirements · Requirements Engineering (RE) · Non-functional requirements (NFRs) · ethical AI

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 423–439, 2025.
https://doi.org/10.1007/978-3-031-78255-8_25

1 Introduction

Requirements Engineering (RE) refer to the process of defining, documenting, and maintaining requirements in the engineering design process or software development lifecycle (SDLC). According to the IEEE[1], a requirement is a condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents (IEEE Std 610.12, 1990). In other words, a well-formed requirement is a statement of system functionality (a capability and I add property or attribute) that must be met or possessed by a system to satisfy a customer's need or to achieve a customer's objective, and that is qualified by measurable conditions and bounded by constraints (IEEE Std 1233, 1998) [1].

Koelsch also defines a requirement as "a need, desire, or want to be satisfied by a product or service" [2]. If this need or desire is not fulfilled, the product is not usable [3]. These requirements are outlined in a document called user requirements [4]. It is from these user requirements that systems requirements are developed. Hence, system requirements development, validation and verification activities include validating the system requirements against user requirements and verifying the correctness of system requirements documentation against these requirements [1].

Due to the nature of requirements, there are two types of requirements. These are functional requirements and non-functional requirements. Functional requirements represent features of the system and the business rules to what the system should include. Whereas non-functional requirements include system qualities and constraints [2, 3]. In other words, functional requirements specify what the system must do while non-functional requirements specify how the system must do the job or the attributes the system must possess.

By the definition of a requirement one gets a view that the emphasis is more on functional requirements than on non-functional requirements. This is because the focus is more on 'what' the product or service achieves than on 'how' it achieves it. Ahmad, Bano [3] even notes that non-functional requirements (NFR) are more difficult to present, despite several modeling languages and tools available that can be used to display their properties [3]. This situation is more pronounced when we consider RE4AI. The bias might be exacerbated with the imminent automation of RE (AI in RE) [5]. However, in this study I direct my interest to the need to emphasize non-functional requirements for ethical AI systems during RE. Below, I make a comparative analysis of traditional software lifecycle against AI/ML development lifecycle:

There is limited well-established and researched RE activities, especially ethical requirements, when it comes to building software systems with one or more AI/ML components as compared to traditional approaches in building software systems. As Table 1 shows, the process of implementing AI/ML components in software systems is inclined heavily towards data. There is limited or no insight into the non-functional requirements of the AI/ML system's internal workings or how the foundation models comply with ethical or regulatory requirements such as fairness, accountability and transparency, safety, trustworthiness inter alia. While such governance frameworks exist,

[1] The Institute of Electrical and Electronics Engineers (IEEE), is the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity.

Table 1. Comparison of Traditional SDLC and AI/ML Model development lifecycle

Typical stages for traditional software development lifecycle [6, 7]	Comparative analysis of the traditional SDLC against AI/ML lifecycle	Typical stages for AI/ML model development lifecycle [8, 9]
Project identification and initiation	In traditional approach, the emphasis is on project planning while in AI/ML it is on data	Data collection
Feasibility study		Data cleansing
Requirements gathering	In traditional approach requirements gathering is pronounced while in AI/ML it is dataset attributes	Feature extraction and selection
Analysis	Developer appreciates the nature of the system while a foundation model will remain a black box	Model training
Design	Developer designs the system while model evaluation is based on data used to train the model before implementation	Model evaluation
Implementation		Model deployment and integration
Operation and Maintenance		

in many jurisdictions there is still no guidelines to ensure that software systems with AI/ML components comply with ethical or regulatory provisions particularly in Africa.

Traditional software engineering involves collecting requirements, analysis, and detailed designs to implement an executable program which mainly includes writing code [10]. On the other hand, engineering software with an AI component involves additional configuration aspects, such as data collection, selecting an appropriate algorithm (for instance, a machine learning or natural language processing algorithm), and training the model based on the desired input/output with relatively less emphasis on the source code writing. This is because the source code is in the foundation model and was coded by developer of the base model [3]. In Table 1, I give a comparative analysis of the difference between the two development methodologies (traditional SDLC and AI/ML model development lifecycle.

RE for AI systems (RE4AI) is not as established as the traditional RE approaches for non-AI software, and AI systems usually lack precise requirements and proper RE techniques [11, 12]. The configuration process, in other words requirements analysis in AI-based software, that is, selecting and validating data and other activities, need to be specified in the requirements phase [3]. This owes to the differences between the RE practices for traditional software which are mostly deterministic compared to AI-based software customized from the foundation models, which are mostly uncertainty-prone and thus a black-box. This necessitates adaptation of the existing tools and RE methods in order to change the status quo [3].

In this paper, I review literature to establish the importance of non-functional requirements in implementing ethical AI also referred to as responsible AI or human-centred AI (HCAI). I use search strings such as non-functional requirements in AI systems, or requirements engineering for AI systems to retrieve as much articles on the subject matter as possible. The repositories used are Google and Google scholar. Using a framework for the classification of requirements, I attempt to make a case for a sub-classification of ethical requirements under the non-functional requirements. I look at approaches to ethical AI and show how a Value Sensitive Design (VSD) approach can be used to embed the non-functional requirements while building ethical AI systems.

1.1 What are Functional and Non-Functional Requirements?

On the one hand a functional requirement specifies an action that a system must be able to perform [13], without considering physical constraints; a requirement that specifies input/output behavior of a system. In other words, at issue is what the system (hardware, software, data) should do. On the other hand a non-functional requirement specifies system properties [14], such as environmental and implementation constraints, performance, platform dependencies, maintainability, extensibility, and reliability. In other words, at issue are characteristics the system should have.

According to Mate and Silva [1], non-functional requirements are often classified into the following categories:

Performance requirements: pertaining to requirements that specify performance characteristics that a system or system component must possess, for example, maximum CPU usage, maximum memory footprint.

External interface requirements: describe a requirement that specifies hardware, software, or database elements with which a system or system component must interface or that sets forth constraints on formats, timing, or other factors caused by such an interface.

Design constraints: refer to a requirement that affects or constrains the design of a system or system component, for example, language requirements, physical hardware requirements, software development standards, and software quality assurance standards.

Quality attributes: A requirement that specifies the degree to which a system possesses attributes that affect quality, for example, correctness, reliability, maintainability, portability.

The problem with the above categorization is that it lacks a clear emphasis on ethical requirements imposed on the system due to consideration of the societal values or ethical concerns from stakeholders. In addition, quality attributes could have been a superset of ethical requirements. However, quality attributes also contains other requirements such as reliability, maintainability, portability which still speak to the technical aspects of the system (system oriented). This is in comparison with ethical requirements that speak to users or environment aspects (non-system or environment oriented).

Subsequently, to the above categorization, I add ethical requirements so that non-functional requirements are now classified as ethical requirements, which are requirements with respect to its environment[2] which include individual (user/developer), business, ecological/environmental, social/cultural/religious or political while other requirements include those with respect to the system such as (opacity, portability, maintainability, durability, inter alia) shown below:

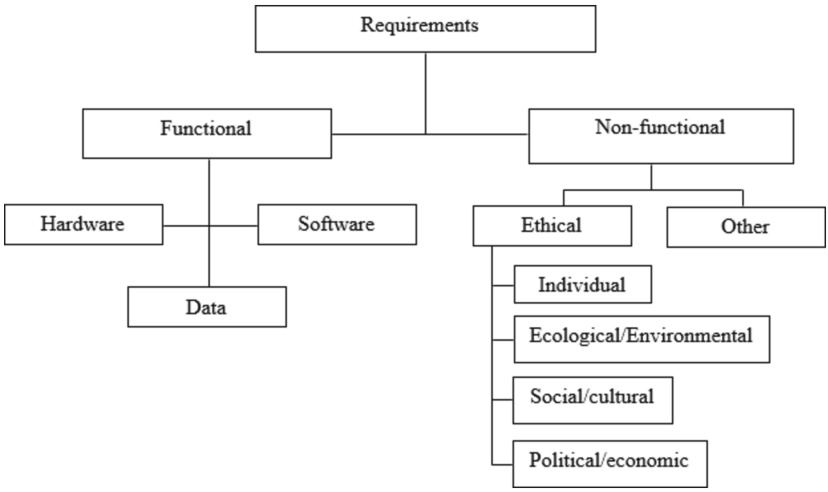


Fig. 1. Classification of requirements for ethical AI

The ethical requirements could be classified into individual (which includes user or developer), ecological/ environmental, social/cultural, and political/economic. Ethical issues are first considered at individual level by each person as a moral agent or patient within their ecological environment. Some ethical concerns would require a community to come together to solve, hence becoming a social problem. When smaller communities fail to deal with the ethical concern, these reach the attention of political leaders and thus will be dealt with by government who might respond by putting legal and regulatory frameworks in place to address the ethical concerns.

2 Requirements Engineering (RE) in Ethical AI Systems Design

I argue that there is need to emphasize non-functional requirements of AI systems because of for instance the nuances of concepts such as moral status, autonomy and agency introduced in AI systems. These new concepts have not only questioned what it means to be human, but they have also opened reflections on how humans relate to technology, other humans, and the world through technology. I discuss these concepts briefly in order to put in context how these ethical concerns among other many concerns necessitate the need to emphasize ethical requirements in AI systems.

[2] By environment I mean all external entities and factors that the system interacts with apart from presumed environment prone to pollution.

To add to these efforts, I propose a framework for classifying ethical requirements into categories that will make it easier to address them. This classification will prove advantageous before architecture and design phase. This is because architects need preferential classification of software requirements, so that they can easily implement different types of requirements into different architectural components [5].

Society has come up with efforts to establish universal standards or guiding principles for embedding human values and norms into what IEEE calls autonomous and intelligent systems (A/IS) today. But as these systems are instilled with increasing autonomy in making decisions and manipulating their environment, it is essential that they are embedded with or designed to adopt, learn, and follow the norms and values of the community they serve. Moreover, their actions should be transparent in signaling their norm compliance and, if need be, they must be able to explain their actions. This is essential if humans are to develop appropriate levels of trust in A/IS in the specific contexts and roles in which A/IS function[3].

According to Bostrom and Yudkowsky, the possibility of creating thinking machines raises a host of ethical issues that relate both to ensuring that such machines do not harm humans and other morally relevant beings, but also to the moral status of the machines themselves [15]. They argue that two criteria are commonly proposed as being importantly linked to moral status, either separately or in combination: sentience[4] and sapience (or personhood).

Sullins on the other hand, argues that it is not necessary for a robot to have personhood in order to be a moral agent. He details three requirements for a robot to be seen as a moral agent. The first is achieved when the robot is significantly autonomous from any programmers or operators of the machine. The second is when one can analyze or explain the robot's behavior only by ascribing to it some predisposition or 'intention' to do good or harm. And finally, robot moral agency requires the robot to behave in a way that shows and understanding of responsibility to some other moral agent [16].

Thus, robots with all of these criteria as described above could assume moral rights as well as responsibilities, if not already, regardless of their status as persons [17–19].

3 Existing Literature on Non-functional Requirements in AI Systems Requirements Engineering

According to a mapping study by Martínez-Fernández et al., non-functional requirements (NFRs) has become one of the hottest topic in the RE4AI-related literature [20]. This presumably, could be due to the realization of what I have discussed so far.

[3] Embedding Values into Autonomous and Intelligent Systems, The IEEE Global Initiative on Ethics of Autonomous and Independent systems.

[4] Sentience: the capacity for phenomenal experience or qualia, such as the capacity to feel pain and suffer; while Sapience: a set of capacities associated with higher intelligence, such as self- awareness and being a reason-responsive agent 15. Bostrom, N. and E. Yudkowsky, *The ethics of artificial intelligence*, in *Artificial intelligence safety and security*. 2018, Chapman and Hall/CRC. p. 57–69.

Several works outline a number of challenges related to RE particularly in AI systems [3, 21]. For instance, the importance of context, consideration of data-related requirements. Of great importance to this current work is the realization of the need to define new types of non-functional requirements, an aspect also mentioned by Horkoff's seminal paper on the topic [22].

Several authors have explored which NFRs apply to AI-based systems. A good number of papers explore a designated type of NFR in detail, for example, safety, and performance [20, 23]. Other authors adopt a holistic perspective and investigate which NFRs apply to AI-based systems. For instance, Habibullah and Horkoff conducted an interview-based survey with ten practitioners to elicit NFR types, their priorities, and most relevant NFR-related challenges [24].

In summary, they state: (1) NFR types can be grouped into thematically-relevant clusters; (2) there are a number of new NFR types specifically related to AI-based systems, or whose relevance excels in this context, such as trust, ethics and explainability [24]; (3) other traditional NFR types, such as usability, are not considered so prioritary (although as usual, there are conflicting views on the importance of this and other NFR types in AI-based systems) [22] in [20].

In this paper, I recommend a new category of non-functional requirements I call ethical requirements to carter for non-technical or non- system oriented requirements such as trust, safety among others.

Recent research efforts strive to aid in designing AI systems with explainability as a requirement [25]. Nevertheless, a systematic and overarching approach to ensure explainability by design is still missing. In fact, often it is not even clear what precisely is meant when demanding explainability. In trying to address this challenge, A. Kohl et al. investigate the elicitation, specification, and verification of explainability as a non-functional requirement (NFR). They do this with a long-term vision of establishing a standardized certification process for the explainability of software-driven systems in tandem with appropriate development techniques [25].

In this paper, I recommend that a standardized certification process caters for all ethical requirements, including non-system oriented, and not just explainability for example. Non-system oriented requirements would be done through system validation by stakeholders.

Agbese, Mohanani [26] and others, recommend the involvement of management in implementing AI ethics by making ethical requirements part of their management practices. However, research indicates that most work on implementing ethical requirements in software engineering management primarily focuses on technical development, with scarce findings for middle and higher-level management [26].

4 The Need to Emphasize Non-Functional Requirements for Ethical AI

For the AI technologies to be acceptable and thus successfully deployed for business in both private and public sectors they have to meet user requirements. These requirements are both functional and non-functional. The challenge my study brings to the fore is that apart from concentrating on business and technical requirements, ethical requirements

in AI systems implementation must receive equal attention from the developers of these technologies. I argue that ethical issues or concerns will arise when values held by stakeholders seemingly appear to be negatively impacted by the introduction of AI technologies in society.

In this paper, I use the phrase 'ethical AI' to refer to AI technology that adheres to well defined ethical guidelines regarding fundamental values, including individual rights, privacy, non-discrimination, and non-manipulation. I argue that ethical AI is not limited to what is permissible by law (what I would call 'legal AI'). Legal limits pertaining to the use of AI set a minimum threshold of acceptability, while ethical AI adheres to policies that go beyond legal requirements to ensure respect for fundamental human values [27].

However, Eitel-Porter [27] also uses the phrase 'responsible AI' to mean the same thing as ethical AI. Virginia Dignum uses 'responsible AI to mean AI systems that are developed "along fundamental human principles and values, to ensure human flourishing and well-being in a sustainable world" [28]. I use 'ethical AI' to include 'responsible AI'. In addition, I also use 'ethical AI' to include human-centred AI (HCAI).

In the book tilted Human-centered AI (HCAI), Ben Shneiderman considers HCAI from three central ideas, the first idea is a framework for thinking about technology design, which shows how creative designers can imagine highly automated systems that preserve human control, the second idea is to show how the two central goals of AI research – emulating human behavior and developing useful applications – are both valuable, and the third idea bridges the gap between widely discussed ethical principles of HCAI and the realistic steps needed to realize them [29].

In this third idea in the book, Shneiderman describes how to adapt proven technical practices, management strategies, and independent oversight methods in the development of AI systems. He also guides software team leaders, business managers, and organization leaders in developing: 1) Reliable systems based on sound software engineering practices, such as a VSD approach in AI systems design which I recommend (see 6.1), 2) Safety culture through business management strategies, and 3) Trustworthy certification by independent oversight [29].

I thus view all the three phrases as overlapping in the following sense. In this paper I use the phrase 'ethical AI' taking into account the above definitions of the other phrases. The two underlining themes in all the three phrases is how their specific definitions strive to bring out and emphasize the human significance aspect and the need to adapt AI systems development processes in order to foster this human significance. For instance, in his third idea Shneiderman talks about bridging the gap between widely discussed ethical principles of HCAI, and I add responsible or ethical AI, and the realistic steps needed to realize them.

Therefore, based on this human significance and the need for reconsideration of software development processes to carter for ethical AI, I argue that an emphasis on non-functional requirements in requirements engineering, taking into account the above classification, will work towards the realization of ethical AI systems in order to promote human values such as privacy, human autonomy, control, wellbeing, and flourishing [27–29].

5 Categorization of Non-Functional Requirements

According to Cysneiros and do Prado Leite, non-functional requirements will play a critical role in delivering trustworthy and socially responsible systems by adopting principles associated with corporate social responsibility [30]. In their paper, Cysneiros and do Prado Leite argue that the more critical the domain is (such as safety, privacy and ethics), the less consumers seem to trust software to make decisions on their behalf or even to be used. They recommend corporate social responsibility (CSR) may play an essential role in creating a trust paradigm between customers and businesses.

However, their interest is to promote loyalty and customer retention in order to enhance customer trust and thus increasing corporate profit as the ultimate motive. However, from an ethical point of view, the motive is not economic profit but holistic human wellbeing and flourishing. Thus, from an ethical point of view, for AI systems to be acceptable by stakeholders, I recommend that non-functional requirements should include ethical requirements that the systems must possess in order to ensure that they align to societal values where they are deployed. I classify these ethical requirements into four levels. These are individual (user or developer), social/cultural/religious, economic/political and ecological/environmental requirements.

These ethical requirements are layered in the sense that when a new problem or phenomenon of moral nature, that is an ethical issue, emerges in a society, it is first perceived and experienced at a personal or moral level within the moral and natural ecological environment where the individual human beings (users and developers inclusive) live. From this level, a community comes together to react to the matter by devising social norms or ethical frameworks through established institutions to govern relations among members or tackle common issues affecting members of the community as a result of the issue at hand.

Where a smaller community cannot handle the challenges posed by the issue at hand, governments come in to make policies, laws and other governance framework providing for rewards and sanctions to protect society at large. This can be depicted as ripples from the point of disturbance moving outward [31]. This is depicted in Fig. 2 below (adopted from Laudon and Laudon) to show how the ethical issues progressively propagates through a community like ripples in a still pond of water.

I propose a classification of ethical requirements based on the corresponding classification of values affected by the implementation of the AI systems based on the ethical issues and concerns raised by stakeholders [32].

This classification of values shown in Table 2 above would enable categorization of corresponding requirements for building ethical AI (value-aligned AI systems) implemented through an approach such as VSD approach as shown in Fig. 1. Personal or moral values raise ethical issues at individual level; then social/cultural ethical issues which include religious values, while economic values are considered at political/economic level of ethical issues.

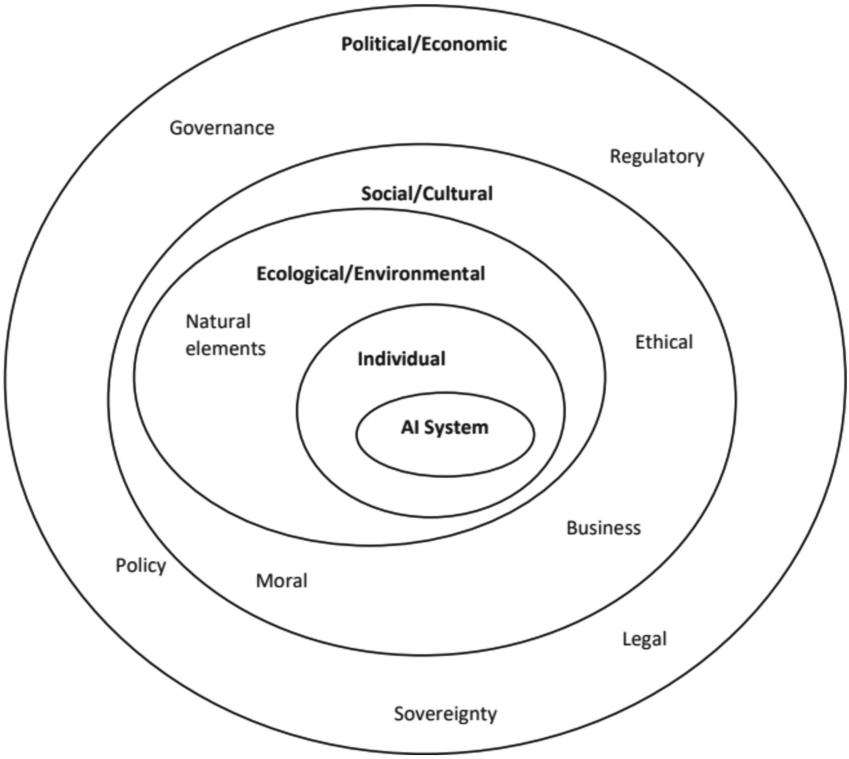


Fig. 2. Ripple effects of an AI system in a society

Table 2. Classification of values

Classification of values	Ethical Concern	Ethical Issue
Personal/moral	I value my privacy	Human right (dignity)
Social	Discrimination/bias	Is AI system fair
Economic	Fear of financial loss/job loss AI replacing humans as tool of production	Equal opportunity/digital divide/human agency
Religious/cultural	Against my religious/cultural beliefs	Human right (autonomy)
Ecological/safety	Effect on climate/Possible accidents	Is AI system safe
Political	Sovereignty/Democracy	Does AI system uphold sovereignty/democracy

6 Approaches to Ethical AI

Despite a rapid growth of ethics documents meant to guide artificial intelligence (AI) development, M. Findlay and J. Seah note that the promotion of AI ethics has nonetheless proceeded with little input from AI practitioners themselves [33]. I argue that AI ethics is one of the cornerstones in the implementation of ethical AI in society. It is the bedrock upon which guiding principles for governance and regulatory frameworks for ethical AI could be anchored. AI ethics can be used to define why, how, and when AI technologies such as generative AI can be used for the good of humanity.

I am therefore, inclined to agree with M. Findlay and J. Seah that this is an emerging gap that needs to be addressed in order to develop meaningful ethical approaches to AI use and development [33]. I argue that it is AI ethics that instigates the call for not only the generation of motivation, among all AI actors, to adopt and habituate practices that influence technology (AI) development (developers) but also the use in a positive manner for the benefit of all (users), of ethical AI. AI ethics also as an applied ethics brings together the necessary focus on "technical discourses" as well as the "genuinely social and personality-related aspects" of adhering to AI ethics guidelines [34] such as the need to relook at RE for ethical AI.

Apart from the need to put governance frameworks in place and a call for AI actors to adhere to principles of some kind in their line of duty, there is also need to look at AI systems development methodologies followed to produce value-laden or ethical AI to address ethical issues due to the potential autonomous, agency, and moral status attributes of some AI systems. There is as a result, growing interest among academics and policymakers in how to ensure that AI systems' actions and decisions do not conflict with human values, laws, or reasonable expectations about how systems should behave [35].

These considerations fall in the scholarly discipline of ethics or moral philosophy. In this paper, I shy away from a philosophical debate on ethical theories and the evaluation of decisions, actions and outcomes pertaining to AI systems against given ethical theories. My interest is to recommend how to build machines that exhibit design attributes as well as make decisions and effect actions that are aligned with human values. Such studies have become a centre of investigation in AI research and in the philosophy of technology [35].

For instance, in my work elsewhere I discussed how a value-sensitive design (VSD) approach for value aligned AI based on relevant normative ethics could be used to protect vulnerable groups of society, such as children growing in Africa, from the adverse impact of AI systems [36].

I therefore, do not only look at ethical AI from a point of machines, or system biased (technical) non-functional requirements such as portability, maintainability, and durability among others. However, I look at ethical AI by also looking at environment oriented (non-technical) non-functional requirements such as privacy, fairness, safety, and so on, based on definitions given earlier in this paper or as Prem puts it, take the notion of ethical AI simply as an abbreviation for an AI system that performs actions that when taken by a human would be considered ethical [35].

The point of reference still remains humans. Hence, it still remains the responsibility of humans to implement (design, develop, deploy and operate, maintain and refine)

AI systems that reflect this ethical requirement, and I argue below that this is best done through a VSD approach. I also argue that a VSD approach is supported by the development of a framework used to identify values espoused by stakeholders, thus ethical issues that arise and how these can be addressed through a VSD approach.

6.1 A VSD Approach for Non-Functional Requirements

Ingredients to the VSD approach to developing ethical AI are values espoused by society and principles practiced by the developers during the development process. The first assumption made is that the humans implementing AI will be ethical and adhere to provided principles and guidelines [27, 37]. The second assumption made is that societal values that maybe affected by AI technologies will be known during requirements elicitation when interacting with stakeholders [38].

VSD is an approach to the design of computer systems and software that aims to take care of and incorporate human values in a comprehensive manner throughout the design process and embed these values into the design [39]. VSD is so far the most reviewed approach for the purpose of ethics and technology since the increased attention to the integration of moral values into the conception, design, and development of emerging information technology (IT) [40]. VSD approach provides an opportunity for adopting a design philosophy that embeds values into the design of artificial agents at the early stages of AI development and possible adjustments throughout the lifecycle of artifacts [41].

The first assumption ignores the fact that unlike other fields of practice such as medicine, AI development lacks (1) common aims and fiduciary duties, (2) professional history and norms, (3) proven methods to translate principles into practice, and (4) robust legal and professional accountability mechanisms. These differences suggest we should not yet celebrate consensus around high-level principles that hide deep political and normative disagreement [42].

The second assumption is that there is only epistemic uncertainty about which values will be impacted upon by a technology, however, digital platforms exhibit a novel form of uncertainty, namely, ontological uncertainty [38]. This means that even with full information and overview, it cannot be foreseen what users or developers will do with digital platforms and how they will be impacted by the technological artifacts [38].

Other than the above two assumptions many assumptions could be construed[5]. Therefore, principles and the assumed knowledge about societal values at a given time alone are not sufficient for ensuring ethical AI development and use in enterprises and public sector. However, I suggest that businesses and governments also require strong, mandated governance controls including tools for managing development processes and creating associated audit trails to enforce their principles. In addition, understanding the ethics of AI will help to establish a framework for building ethical AI [27, 37].

[5] I add that the third assumption made is that humans are not homogenous, thus will not espouse the same principles. In addition, humans are irrational beings, which means that their actions will always be affected by various external facts which in turn will affect decisions made in making sensitive decisions such as embedding values in AI systems.

I make a recommendation for the use of a VSD approach in order to factor in and emphasize non-functional requirements based on the assumption that, 1) the human developers will adhere to provided principles and guidelines, and 2) business and governance frameworks and processes put in place will adequately foster all essential aspects of ethical AI development process.

The table below is an example of an ethical AI framework used to illustrate conceptualization of the fairness principle and how it could be factored in the implementation of ethical AI using a VSD approach for instance.

Table 3. Example for the main components of an ethical AI framework for the principle of "fairness" [Prem [35]].

Concept	Bias
Concern	Not treating people fairly, e.g., taking decisions that are influenced by a person's gender or social background
Principle	Fairness
Remedy	Testing systems for bias and using unbiased data sets, etc

In the above framework, fairness is considered as a principle. However, fairness could also be considered as a value or indeed an ethical dilemma. For instance, a society that considers fairness as an obligation, it is thus valued as such and poses a dilemma in a scenario where non-discriminatory action has to be taken such as in employment recruitment, education opportunity, justice dispensation among others.

I recommend a framework below as a mechanism for clarifying further a concept of a given value or principle, and the accompanying ethical issue, concern, dilemma or indeed implication. This is done by way of what I call a statement of clarification in this case made by the concerned subject (user or developer). The framework also includes consideration of the perspective from which the concept is being analyzed from, that is user or developer perspective in order to know where the remedy must be applied and thus solve the uncertainty in the VSD approach while prescribing an appropriate remedy for the societal value impacted as shown in Table 2 [38]. Therefore, ethical dilemma, ethical issue, ethical concern and ethical implication columns in Table 3 are suggested expansion of Table 2 after the principle row but before the remedy row (Table 4).

In the framework above, the concept of privacy can be viewed as a value claimed by a user. This can be formulated in a statement of an ethical principle to ensure the privacy of users in AI system implementation. An ethical dilemma would arise when a user is faced with a situation in which they are required to make decision whether or not to use the AI system despite being unsure if their privacy is assured. This would then be a concern each time a user contemplates on whether to use the AI system or not. An ethical implication is a result of the occurrence of the risk, in this case the loss of privacy of the user while using the AI system.

Table 4. Ethical concepts uncertainties clarifications

Word/phrase	*Value (claim)*	*Ethical principle*	*Ethical dilemma*	*Ethical issue*	*Ethical concern*	*Ethical implication*
Concept	Privacy					
Statement for clarification	I value my privacy	Ensure privacy unless, justice[6] requires otherwise	Should I use the system even if my privacy is not assured?/What is the best way to assure privacy?	Is privacy assured/at stake?	I'm worried about my privacy. The system does not seem to guarantee privacy	What loss will people, society incur as a result of the risk of loss of privacy?
Perspective [User (U)/Developer (D)]	*U*	*D*	*U, D*	*U, D*	*U, D*	*D*

7 Conclusion

In conclusion, existing traditional approaches for building software systems have well established requirements engineering activities. However, this is not the case for software systems with AI/ML components. This is because AI/ML systems concentrate on data availability more than anything else. This is exacerbated further by a lack of emphasis on non-technical (environmental oriented) non-functional requirements such as fairness, accountability, transparency, safety, privacy among others, during requirements engineering in all software engineering activities and AI systems in particular.

I reviewed literature to probe how an emphasis on non-functional requirements during requirements determination could help address the status quo. Non-functional requirements is becoming an area of interest from a number of researchers. I attempt to contribute to the body of knowledge by firstly giving a comparative analysis of the traditional software development lifecycle against software with an AI/ML component development lifecycle.

Secondly, I recommend the introduction of a categorization of ethical requirements under non-functional requirements when implementing software systems particularly AI systems in order to emphasize ethical requirements that arise in AI technologies. I also argue that currently, the emphasis is more on system oriented (technical) non-functional requirements than environment oriented (non-technical) non-functional requirements.

[6] The implication is that one may lose their privacy in an event of any legal proceedings such as investigations for instance; according to The Zambian Data Protection Act of 2018 https://www.parliament.gov.zm/sites/default/files/documents/acts/Act%20No.%203%20The%20Data%20Protection%20Act%202021_0.pdf.

Thirdly, in order to conceptualize ethical issues and minimize uncertainties in what values will be affected and how they will be affected, I recommend grouping values or ethical issues into individual, ecological/environmental, social/cultural, and political/economic based on the ripple effect of the introduction of AI systems in society from individual to political level. I argue that AI ethics is the foundation upon which guiding principles for governance and regulatory frameworks for ethical AI could be built upon.

I argue that there is need for approaches to ethical AI that can be used to conceptualize values and principles. I recommend the use of a framework that lay out the concept, concern, principle and remedy for easy understanding of ethical requirements. In addition, I recommend a framework that includes clarification of the value and principle. The framework also includes consideration of the perspective from which the concept is being analyzed from, that is, user or developer.

I recommend the use of a VSD approach in determining and embedding of non-functional requirements classified as ethical requirements based on the assumption that, 1) the human developers will adhere to provided principle guidelines, and 2) business and governance frameworks and processes put in place will adequately foster all essential aspects of ethical AI development process.

References

1. Mate, J.L., Silva, A.: Requirements Engineering for Sociotechnical Systems. IGI Global (2005)
2. Koelsch, G.: Requirements Writing for System Engineering, vol. 2016. Springer, Heidelberg (2016)
3. Ahmad, K., et al.: What's up with requirements engineering for artificial intelligence systems? In: 2021 IEEE 29th International Requirements Engineering Conference (RE). IEEE (2021)
4. Dennis, A., Wixom, B., Tegarden, D.: Systems Analysis and Design: An Object-Oriented Approach with UML. Wiley (2015)
5. Kaur, K., Kaur, P.: The application of AI techniques in requirements classification: a systematic mapping. Artif. Intell. Rev. **57**(3), 57 (2024)
6. Kyeremeh, K.: Overview of system development life cycle models. Available at SSRN 3448536 (2019)
7. Dennis, A., Wixom, B.H., Roth, R.M.: Systems Analysis and Design. Wiley (2008) & sons
8. Lorenzoni, G., et al.: Machine learning model development from a software engineering perspective: a systematic literature review. arXiv preprint arXiv:2102.07574 (2021)
9. De Silva, D., Alahakoon, D.: An artificial intelligence life cycle: from conception to production. Patterns **3**(6) (2022)
10. Van Vliet, H., Van Vliet, H., Van Vliet, J.: Software Engineering: Principles and Practice, vol. 13. Wiley, Hoboken (2008)
11. Agarwal, M., Goel, S.: Expert system and it's requirement engineering process. In: International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014). IEEE (2014)
12. Khomh, F., et al.: Software engineering for machine-learning applications: the road ahead. IEEE Softw. **35**(5), 81–84 (2018)
13. Desfray, P., Raymond, G.: Modeling Enterprise Architecture with TOGAF: A Practical Guide Using UML and BPMN. Morgan Kaufmann (2014)
14. Maxim, B.R., Kessentini, M.: An introduction to modern software quality assurance. In: Software Quality Assurance, pp. 19–46. Elsevier (2016)

15. Bostrom, N., Yudkowsky, E.: The ethics of artificial intelligence. In: Artificial Intelligence Safety and Security, pp. 57–69. Chapman and Hall/CRC (2018)
16. Sullins, J.P.: When is a robot a moral agent. Mach. Ethics **2011**(6), 151–161 (2001)
17. Sullins, J.P.: When is a robot a moral agent. Mach. Ethics **2006**(6), 23–30 (2006)
18. Asaro, P.M.: What should we want from a robot ethic? In: Machine Ethics and Robot Ethics, pp. 87–94. Routledge (2020)
19. Floridi, L.: Artificial agents and their moral nature. In: Kroes, P., Verbeek, PP. (eds.) The Moral Status of Technical Artefacts. Philosophy of Engineering and Technology, vol. 17, pp. 185–212. Springer, Dordrecht (2013). https://doi.org/10.1007/978-94-007-7914-3_11
20. Ferrari, A., Penzenstadler, B.: Requirements Engineering: Foundation for Software Quality: 29th International Working Conference, REFSQ 2023, Barcelona, Spain, April 17–20, 2023, Proceedings, vol. 13975. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-29786-1
21. Heyn, H.-M., et al.: Requirement engineering challenges for ai-intense systems development. In: 2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN). IEEE (2021)
22. Franch, X., Jedlitschka, A., Martínez-Fernández, S.: A requirements engineering perspective to AI-based systems development: a vision paper. In: Ferrari, A., Penzenstadler, B. (eds.) REFSQ 2023. LNCS, vol. 13975, pp. 223–232. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-29786-1_15
23. Sadiq, M.: Modeling the non-functional requirements in the context of usability, performance, safety and security. In: Computer Science and Software Engineering. Blekinge Institute of Technology: SE – 372 25 Ronneby, Sweden (2007)
24. Habibullah, K.M., Horkoff, J.: Non-functional requirements for machine learning: understanding current use and challenges in industry. In: 2021 IEEE 29th International Requirements Engineering Conference (RE). IEEE (2021)
25. Köhl, M.A., et al.: Explainability as a non-functional requirement. In: 2019 IEEE 27th International Requirements Engineering Conference (RE). IEEE (2019)
26. Agbese, M., et al.: Implementing ai ethics: making sense of the ethical requirements. In: Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering (2023)
27. Eitel-Porter, R.: Beyond the promise: implementing ethical AI. AI Ethics **1**(1), 73–80 (2021)
28. Dignum, V.: Responsible Artificial Intelligence: How to Develop and Use AI in a Responsible Way. Springer, Heidelberg (2019). https://doi.org/10.1007/978-3-030-30371-6
29. Shneiderman, B.: Human-centered AI. Issues Sci. Technol. **37**(2), 56–61 (2021)
30. Cysneiros, L.M., do Prado Leite, J.C.S.: Non-functional requirements orienting the development of socially responsible software. In: Nurcan, S., Reinhartz-Berger, I., Soffer, P., Zdravkovic, J. (eds) BPMDS EMMSAD 2020. LNBIP, vol. 387, pp. 335–342. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-49418-6_23
31. Laudon, K., Laudon, J.: Management Information Systems: International Edition, 11/E. KC Laudon, vol. **11** (2009)
32. Díaz-Rodríguez, N., et al.: Connecting the dots in trustworthy artificial intelligence: from AI principles, ethics, and key requirements to responsible AI systems and regulation. Inf. Fusion **99**, 101896 (2023)
33. Findlay, M., Seah, J.: An ecosystem approach to ethical AI and data use: experimental reflections. In: 2020 IEEE/ITU International Conference on Artificial Intelligence for Good (AI4G). IEEE (2020)
34. Ruttkamp-Bloem, E.: The quest for actionable AI ethics. In: Gerber, A. (ed.) SACAIR 2021. CCIS, vol. 1342, pp. 34–50. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-66151-9_3
35. Prem, E.: Approaches to ethical AI. In: Werthner, H., Ghezzi, C., Kramer, J., Nida-Rümelin, J., Nuseibeh, B., Prem, E., Stanger, A., p. 225 (2024)

36. Liywalii, E.: Artificial intelligence (AI) and children in africa: a sandboxed childhood and a normative ethics point of view. EasyChair (2023)
37. Siau, K., Wang, W.: Artificial intelligence (AI) ethics: ethics of AI and ethical AI. J. Database Manage. (JDM) **31**(2), 74–87 (2020)
38. de Reuver, M., van Wynsberghe, A., Janssen, M., van de Poel, I.: Digital platforms and responsible innovation: expanding value sensitive design to overcome ontological uncertainty. Ethics Inf. Technol. **22**(3), 257–267 (2020). https://doi.org/10.1007/s10676-020-09537-z
39. Friedman, B., et al.: Value sensitive design and information systems. In: Doorn, N., Schuurbiers, D., van de Poel, I., Gorman, M. (eds) Early engagement and new technologies: Opening up the laboratory. Philosophy of Engineering and Technology, vol. 16, pp. 55–95. Springer, Dordrecht (2013). https://doi.org/10.1007/978-94-007-7844-3_4
40. Manders-Huits, N.: What values in design? The challenge of incorporating moral values into design. Sci. Eng. Ethics **17**(2), 271–287 (2011)
41. Umbrello, S., De Bellis, A.F.: A value-sensitive design approach to intelligent agents. In: Artificial Intelligence Safety and Security. CRC Press, Roman Yampolskiy (2018)
42. Mittelstadt, B.: Principles alone cannot guarantee ethical AI. Nat. Mach. Intell. **1**(11), 501–507 (2019)

Democratic AI: Justification for a Broad View of Public Reason

Paige Benton[1,2(✉)]

[1] The African Centre for Epistemology and Philosophy of Science, University of Johannesburg, Johannesburg, South Africa
pbenton@uj.ac.za
[2] Centre for AI Research (CAIR), CSIR, Pretoria, South Africa

Abstract. Literature on how and why AI technology undermines democracy is increasing. In this paper, I argue that a narrow view of public reason is insufficient for public justification of AI technology. Instead, I claim that a broad view of public reason is a more appropriate method of justification for the development and implementation of democratic AI. Public reason requires one to justify their decisions in terms of values that all can accept, i.e., values embedded in the public political culture of democracy. This is an attractive method of justification for democratic societies since it enables persons to provide justifications that their fellow citizens can accept even if they hold conflicting moral truths. The narrow view of public reason could account for constitutional amendments to strengthen AI regulation. Yet, it does not require CEOs of companies to develop publicly justifiable decisions for creating or implementing an AI algorithm since the CEO's decisions do not form part of the basic structure. For AI regulation to be democratic, the sphere of influence of public reason needs to be extended into the background culture, considering that the use of AI technology is not confined to the public sphere (i.e., basic structure) but instead is integrated into the non-public sphere of society. Therefore, I claim a broad view of public reason is essential.

Keywords: Public Reason · Background Culture · Democratic AI · The Politics of AI · Political Liberalism

1 Introduction

There is an increase in literature discussing how and why AI technology undermines democracy [6, 14, 15, 19, 45]. Within this literature is one stream of research investigating the risks of opaque decision-making AI[1] [13, 21, 54]. If opaque decision-making algorithms are not transparent and cannot be explained, how should their development and use be justified in a democratic society? Questions like this are essential when discussing how AI technology can support or undermine democratic ideals. To address

[1] AI is defined as "a machine-based system that can, for a given set of human-defined objectives, make predictions, recommendations, or decisions influencing real or virtual environments" See: OECD.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 440–455, 2025.
https://doi.org/10.1007/978-3-031-78255-8_26

opacity in AI, thinkers have invoked discussions of public reason. Public reason refers to the practice of providing justifications for decisions and policies that rely on shared political values that reflect democratic norms, such as liberty, equality, fairness and mutual respect [43].

Thinkers like Binns [7] and Maclure [29] have argued for using public reason as the demarcator for algorithmic accountability or the theoretical justification for a "strong explainability requirement". In contrast, Da Silva [16] argues that Maclure's strong explainability requirement is unnecessary to meet the conditions of public reason. Public reason is an attractive method of justification for democratic societies since it enables persons to develop public reasons that their fellow citizens can accept even if they disagree about moral truth [36, 43]. This is precisely why thinkers have argued for its use both in political liberal theory and in AI ethics [7, 16, 29].

This paper does not propose a technical solution to the problem of AI opacity, nor does it aim to show how public reason can resolve issues of opacity. Instead, I aim to provide a justification for why a broad view of public reason should be the method of justification for distinguishing the just vs. unjust development and use of AI technology in general. The outcome I claim is that a broad view of public reason is a better political, philosophical framework of justification to align decision-making (within the realm of AI technology) with democratic ideals. Thus, this paper adds to the current debate on AI and public reason by demonstrating the limits of a narrow view of public reason in developing democratic AI. My choice to engage in the debate on public reason is precisely motivated by contributions such as Binns and Maclure. Scholars in the field of AI ethics have increasingly turned to public reason as a framework to address opaque decision-making and ensure alignment with democratic values. I aim to demonstrate how a broad view of public reason can extend beyond explainability requirements and opacity concerns.

I claim that the broad view of public reason can help ensure that the development and implementation of AI technology can align with democratic practices and values beyond the basic structure. If confined to the narrow view of public reason, I suggest that AI regulation may cause developers and users of AI technologies to undermine democratic societal goals and values. This is because the narrow view of public reason only regulates constitutional essentials in the basic structure. Thus, the narrow view could ensure constitutional amendments that help strengthen the democratic use of AI. However, it cannot ensure that design decisions developed in the background culture of democratic societies align with democratic ideals; for example, the narrow view does not require that the CEOs of companies develop publicly justifiable decisions for creating or implementing an AI algorithm. However, many impactful decisions in the AI realm are made in the background culture of society, which includes private corporations and industries. CEOs and companies often wield significant power over AI development and implementation, but the narrow view of public reason doesn't require them to publicly justify their decisions. The broad view of public reason can account for the latter as its scope extends beyond constitutional essentials, including any situation where citizens wield coercive power over each other [36].

In Sect. 2, I briefly discuss the problem of opacity to demonstrate why it poses a political issue for democratic AI. This discussion establishes the foundation for a more

detailed examination in Sect. 4, where I explore how opacity contributes to the erosion of institutional trust. In Sect. 3, I explain the two views of public reason (narrow vs broad view). With this background in place, I address how AI regulation would benefit from a broad view of public reason due to the institutional trust it could help strengthen. In demonstrating this, I use the CEO as an example since it effectively illustrates the need for a broad view of public reason, as it highlights how powerful, yet unregulated, decision-makers in the private sector can influence AI development without public scrutiny or alignment with democratic ideals under the narrow view. In Sect. 4, I demonstrate how the broad view is more attractive than the narrow one in safeguarding institutional trust. After that, I present a critique of public reason that applies to both the narrow and broad views. I discuss Robert Talisse's [50] critique of public reason as an exclusionary epistemology in relation to implications for Big Tech. The implication of this argument is that public reason, in general, may be too demanding, particularly in cases where corporate leaders, like CEOs, must align their decisions with publicly acceptable reasons precisely because it means excluding personal moral convictions. Let us now turn to the problem of opacity as a political problem.

2 The Problem of AI Opacity

"How do we guarantee that authorities tasked with making decisions that impact our daily lives ensure those decisions are transparent and equitable?" [30]. Questions such as these plague political philosophy and have arisen in discussions on AI opacity.

Burrell's seminal paper, "How the machine 'thinks': Understanding opacity in machine learning algorithms", [9] highlights three forms of opacity. Firstly, intentional opacity. 'Intentional opacity' results from developers and corporations purposefully omitting or obscuring information to protect their intellectual property or trade secrets [13]. This kind of opacity is intentional, as trade secrets are deliberately kept from the public domain in order for corporations to capitalise on their proprietary technology and, as such, ensure the greatest market value of their product.

The second kind of opacity is "technical illiteracy" [9]. 'Technical illiteracy' refers to how AI operating systems trained by machine learning algorithms can be opaque to the general public. Here, the ability to write, read and interpret the code is a specialised skill set that the majority of the general public does not have access to [9]. This leads to many individuals being unable to critically engage with or challenge the algorithms that affect their lives, leading to the unequal power imbalance between the small group of experts who create algorithms and the large group of citizens who are subject to their outcomes. To remove this level of opacity from machine learning and deep learning technology, it would require a transformational digital literacy policy, civic initiative and governmental resources, see Diakopoulos [18] suggestion for digital literacy for journalists as a safeguard for the general public.

The third opacity is the opacity of scale. This form of opacity arises from the complexity of machine learning algorithms and the massive scale of their datasets. Before machine learning, the decision-making of symbolic AI could be explained by developers as the conditions and environment of symbolic AI were contained and extracting decision trees enabled the operating system to be transparent, explainable and interpretable. In contrast, with machine learning and deep learning, the process is vastly

more complex due to massive data sets, numerous data points, and weighing different informational nodes [29]. This results in no systematic path between the input data and output functionality. As a result, all stakeholders in a "machine learning ecosystem" cannot understand, explain or interpret the behaviour of the machine, leading to what the industry terms the Black Box Problem [32]. Burrell [9] refers to this problem as the "curse of dimensionality". The 'curse of dimensionality' arises due to the scale of data and the practical complications of the difficulty of human oversight in interpreting the data and analysing the input and output relations. This leads to even experts battling to follow the internal logic underlying the operating system's behaviour, which causes this opacity [33].

Facchini and Termine [20] develop a taxonomy to emphasize the complexity of opacity and the context-dependent nature of opaque systems. This taxonomy demonstrates that the degree of opacity of an algorithm or model varies depending on a select set of circumstances, such as where it is used, the purposes for which it is used, and the characteristics of the user employing the system. The three spheres of opacity, Facchini and Termine, are identified as access, link, and semantic opacity [20].

Briefly, access opacity refers to one's ability to understand the AI system. The degree to which a system may be more or less opaque can depend on the user's background knowledge, the complexity of the system, and the transparency of the policies. In contrast, link opacity arises when the relationship between the model and the phenomenon it represents is unclear. An example of this could be that a user may not know if the underlying process of the system is causal or mechanistic as such, the pragmatic understanding of the system is opaque. Lastly, semantic opacity results when there is confusion about the meaning and interpretation of the system. This opacity can arise when a user has difficulty understanding inferences made by the system. What is important to note is the dynamic nature of opacity. Opacity can increase or decrease for a variety of factors at a given time, and this increase or decrease is not uniform for all users, experts, etc. Thus, the search for technical solutions to the problem of opacity needs to consider nuanced approaches tailored to context-specific situations. One of the technical solutions has led to the rise of Explainable Artificial Intelligence (i.e., XAI). Within this literature, some thinkers argue that only interpretable models should be implemented [46]. In contrast, others aim to explain decisions after the operating machine has made them [56].

Non-technical solutions to address technical and intentional opacity are articles 13–15 and 22 of the European Union General Data Protection Regulation (GDPR), which insists on the right to explanation [55].The United Nations Educational, Scientific and Cultural Organization (UNESCO) Recommendation on the Ethics of AI[2] focuses on the values of transparency and explainability, trustworthiness, reasonability and accountability as essential in all aspects of the "AI life cycle" to highlight the need to ensure that decision-making by AI operating systems aligns with, as opposed to undermining citizens human rights.

Technical, intentional, large-scale or context-dependent opacity have contributed to discriminatory outcomes in automated decision-making across many sectors of society, raising ethical concerns. Automated decision-making is when an AI operating system undertakes certain decision-making tasks that were previously performed by

[2] See: [53].

human beings. Automated decision-making is employed in education, judicial, health, law enforcement, and economic sectors to automate decisions relating to university admission[3], parole eligibility[4], diagnosing patients[5], and identifying criminal hotspots[6] and top candidates for job positions[7]. Within these sectors, issues of bias have arisen. For example, the Correctional Offender Management Profiling for Alternative Sanctions (COMPAS) risk-assessment algorithm unfairly identified black offenders to be at a higher risk than white offenders for reoffering [22]. This leads to individuals like Eric Loomis receiving longer sentences due to their COMPAS score [13]. In Eric Loomis's case, the biased decision-making resulted from technical and intentional opacity. For the former, pre-existing bias was built into the data sets on which the automated decision-making was trained. There was intentional opacity as the decision-making process of the software was considered a trade secret, and Loomis and his lawyer were prevented from assessing it (ibid.).

The problem of AI opacity, no matter its origin, is a political problem in liberal constitutional democratic societies. In the case above, COMPAS was used to exercise coercive power over individuals such as Loomis. Yet, the justification for this coercive power could not be provided to Loomis and his lawyer. This lack of public justification Maclure [29] terms "AI's public reason deficit". Persons are entitled to know how and why a decision that impacts their life was made. AI opacity amplifies the inability to achieve this. I agree with Maclure [29] when she states: "AI needs to be brought under the authority of public reason so that automated decisions can be scrutinised and assessed". I claim that without justification of public reason to hold AI developers accountable, the lack of explainability and transparency will undermine citizens' trust in social, political and economic institutions. This breakdown of trust is addressed in Sect. 4. For now, I turn to the scope of public reason to clarify the distinction in terminology before demonstrating why the broad view of public reason is best suited to managing AI technology.

3 Public Reason in Political Philosophy

Liberal theorists such as John Rawls, Thomas Nagel, and Charles Larmore use public reason as a method of justification. This method requires that political decisions, principles, and institutions be legitimate if they result from reasonable agreement [44]. Scholars consider public reason a neutral method of justification since the search for reasonable agreement cannot rest on promoting a singular conception of the good [39, 44]. Political philosophers such as Rawls [44] use the term 'a conception of the good' to encompass different moral, philosophical, religious, metaphysical, or spiritual doctrines that persons follow, such as Christianity. Public reason is an attractive method of justification for liberal theorists because of its neutrality. This is important as liberal theorists who rely on public reason presuppose that reasonable moral pluralism is a

[3] See: [34].
[4] See: [26].
[5] See: [40].
[6] See: [31].
[7] See: [25].

foundational aspect of a just society [11, 44, 52]. In other words, a just society respects people's right to choose their conception of the good (i.e., respects persons' liberty of conscience). Given this, public reason enables democratic legitimacy, allowing agreement to be reached based on public reasons. These reasons are 'public' because their justification is that which all persons can accept regardless of their notion of the good.

Suppose the foundation of public reason cannot be a conception of the good for the reasons explained above. In that case, the foundation has to rely on the political and moral values embedded in the public political culture of a liberal democratic society. Public political culture is a Rawlsian term that refers to the background values that form the foundation of liberalism [41]. These values include "the virtue of reasonableness and a sense of fairness, a spirit of compromise and a readiness to meet others halfway" [44]. Thus, public reason relies on citizens to internalise these values and act from and in line with the values. The question is when should citizens exercise their use of public reason? The narrow view of public reason requires citizens to exercise public reason in only matters of constitutional essentials [36, 43]. In contrast, the broad view requires that all political issues – i.e., any time a decision that can lead to citizens wielding coercive power over one another – must meet public justification [36]. I explain the nuances of this distinction in terms of the scope of the views below.

Quong regards John Rawls's account of public reason as a narrow view [36]. To reiterate, the narrow view only applies to discussions of constitutional essentials. For Rawls, constitutional essentials are the fundamental rights and liberties that are safeguarded by the first principle of justice and features of the second principle, such as equality of opportunity and a social minimum [44].

Proponents of the narrow view of public reason argue that its limited scope is sufficient for a just society, as it ensures that public reason is applied to the basic structure. Rawls considers the basic structure of society as the main subject of justice [44]. The basic structure denotes the essential institutions that mould the lives of citizens, such as the political constitution, the organisation of the economy, and the judicial system [44]. All these institutions exist within the basic structure as they are non-voluntary institutions, core to citizens' participation within society. The basic structure is the foundational sphere of justice for Rawls since the institutions here regulate what Rawls refers to as "background justice" [43]. Citizens' aspirations, social circumstances, careers, and personal opportunities can be realised to the degree to which the basic structure of society is just [44]. Rawls claims that just institutions in the basic structure would permeate just decisions throughout society, hence its importance [42]. Suppose public reason regulates the constitutional essentials which regulate the basic structure. In that case, the essential institutions within a society will be structured in line with public reason, as defended by Rawls [41–44].

To illustrate, neither the narrow nor the broad view would apply directly to family members, whereby members themselves have to invoke the method of public reason to determine how to treat one another around the dinner table. Instead, the narrow view of public reason applies indirectly to the structure of the family. This means that family members will not have to invoke the method themselves but instead turn to the constitutional essentials, such as the constitution's content created in line with the requirements of public reason for guidance. A criticism of the narrow view of public

reason is that it assumes all political domains of society will be just if public reason is to apply to the basic structure since just actions filter down from just principles. Proponents of this view do not justify why public reason should not or cannot be applied to all domains [36].

In contrast to the narrow view, the broad view proposed by Quong [36] claims that public reason should be applied in every situation that can be regarded as political. A problem is viewed as political when "citizens exercise coercive power over one another" [36]. For Quong [36], public reason is an ideal method of reasoning that should " regulate the behaviour of legislators, judges, and ordinary citizens". Thus, the broad view expands the scope of public reason from government, public officials and courts to ordinary citizens. Quong does not suggest that citizens implement public reason in every aspect of their lives, such as in the family scenario discussed above. Citizens should appeal to non-public reasons (i.e., reasons based on a conception of the good) in their private lives. However, the broad view requires public reasons to be invoked by ordinary citizens when reasoning about decisions in the public sphere. In other words, public reasoning should be implemented by citizens to discuss how, why, and when the use of coercive power is legitimate [36].

Allow a toy example to highlight this distinction in scope between the two forms of public reason. Say the CEO of a social media platform decides to develop a business model for her company, and in doing so, she decides to commodify the attention of platform users. The narrow scope of public reason would not require the CEO to make this decision in line with the requirements of public reason since this decision is neither a constitutional essential nor a fundamental justice issue. The business model based on the commodification of users' attention does not lie within the basic structure of society, as businesses are part of the background culture. Nor does the business model impact one's access to primary goods or equal access to rights and liberties since citizens will still have their basic rights and access to equal opportunities (two of the five primary goods).[8] The outcome is that the CEO (in line with the narrow view) could make the decision by appealing to non-public reason, i.e., the board deciding on the company's business model in line with her own conception of the good and moral values.

Would it be legitimate for the CEO to promote her conception of the good by choosing a business model that tracks user data to sell targeted advertisements? Regarding the narrow view, one could argue that it is justified, as CEOs are not confined to public reasons when making business decisions. Additionally, tracking users' data to sell targeted advertisements does not impede their rights and liberties, considering that users still choose not to use the social media platform or change their privacy settings. Suppose one could show that the CEO's choice of business model does prevent users from exercising their fundamental rights and liberties. In that case, an argument can be made

[8] The five primary goods are: "(a) First, the basic liberties as given by a list, for example, freedom of thought and liberty of conscience; freedom of association; and the freedom defined by the liberty and integrity of the person, as well as by the rule of law; and finally the political liberties; (b) Second, freedom of movement and choice of occupation against a background of diverse opportunities; (c) Third, powers and prerogatives of offices and positions of responsibility, particularly those in the main political and economic institutions;(d) Fourth, income and wealth; and (e) Finally, the social bases of self-respect" [42].

within the framework of the narrow view of public reason to determine this business model as unjust in a democratic society.

In comparison to the narrow view, the broad view of public reason requires that the CEO uses public reasons when deciding on the kind of business model for her social media company and decisions made to promote this model, such as the use of targeted advertisements. People must justify these business decisions via public reasons or reasons that could be eventually translated into public reasons since the broad view requires public reason to be used in any instance where coercive power is enforced on citizens.

One could argue that deciding on the kind of business model of a social media company is the exercise of coercive power since the platform can impact citizens' freedom of expression depending on what content the recommendation algorithms push vs how the users' own content is amplified or suppressed.[9] Thus, content moderation policies should be subject to public justification. The broad view is attractive as it does not allow for non-public reasons in situations where coercive power is displayed in society, even if it falls part of the background culture. The advantage of the broad view in this instance is that it removes the possibility of a person's or group's (in this example, the CEOs') conception of the good infiltrating public spaces.

Recognising the danger of a CEO imposing their conception of the good onto a public forum does not require imagination. The genuine risk that Elon Musk poses to democratic public discourse on X (formally known as Twitter) raises this issue. Elon Musk's numerous decisions as CEO of X have led to journalists and academics questioning the role of X in undermining democratic political life [1, 5, 27, 51]. This is seen in Musk's decisions to use the new 'blue tick' system, whereby individuals pay for the use of this blue tick, disbanding the trust and safety council and letting go of numerous employees tasked with ethical oversight roles [8]. Many users suspect that far-right content is being amplified as recommended content [28, 35]. According to an analysis by the Center for Countering Digital Hate [10], "Musk has posted false or misleading claims about the Democrats importing voters on 42 occasions, amassing 747 million views…Musk has posted false or misleading claims about voting integrity seven times, amassing 288 million views, often criticising mail-in ballots and electronic voting machines". Most recently, Musk posted an AI-generated video of the Vice President of America, Kamala Harris, in which the video attributed her to say: "I was selected because I am the ultimate diversity hire…so if you criticise anything I say, you're both sexist and racist." His post was viewed 133 million times and breached X's community guidelines. X requires that all AI-generated content carry a warning label, 'misleading media'. Thus, Musk's very own post did not meet the criteria of his platform.

These incidents highlight how Musk, on his acquisition of *Twitter to X*, uses the platform to promote his political values by amplifying content he agrees with. In theory, a person who holds a far-right political view that undermines democratic values such as (fairness, equality, freedom, liberty of conscience, tolerance, etc.), should have the same opportunity as those holding moderate or far-left views in the public sphere to express their views freely. However, when a CEO amplifies content on a social media platform to promote one subset of political values or amplify content from the right

[9] For literature on recommendation algorithms see: [3, 4, 24].

instead of the centrists or left, then issues of coercive power arise. When people are influenced by what they see on social media, it influences their thoughts and beliefs, impacting their behaviour within society. The riots that recently happened in the United Kingdom [17] and the events in Myanmar in 2019 [2] are two examples that highlight how social media wields coercive power. The narrow view of public reason cannot act as a justificatory framework since social media is not a constitutional essential. Non-constitutional essentials impact power dynamics and institutional trust as illustrated by this example of Musk. X users who do not hold far-right values may question the legitimacy of the content they encounter, leading them to question the institution of X and distrust the platform's views on the neutrality of the information they receive. A strength of the broad view is that it would require Elon Musk to make his decisions at X publicly justifiable, aiding the creation of just power dynamics and institutional trust.

4 AI Regulation, Public Reason and Institutional Trust

Before I address how and why citizens' trust in social, political, and economic institutions will be diminished if AI developers are not held accountable to a broad public reason framework, allow me to explain the relationship between public reason and institutional trust.

Citizens develop institutional trust as a form of trust when the basic structure of society is just [43]. Institutional trust is essential, as citizens will only be willing to adhere to the demands of justice if they see that the basic structure is set up for mutual benefit. Considering that a liberal constitutional democracy relies on citizens' mutual commitment to upholding justice, the basic structure must be set up to the advantage of all citizens. The idea that democratic society depends on the mutual cooperation of persons to ensure that all people are recognised as free and equal requires the development of reciprocity between citizens.

For democratic citizens to acquire reciprocity, they need to trust in their fellow citizens. Specifically, they need trust in citizens who are not like-minded, i.e., citizens with whom they share no other bonds other than their civic commitment to justice [42]. Citizens achieve a minimum level of reciprocity when they develop civic friendship with one another, as they see other citizens as willing to internalise the demands of justice (for example, by acting in line with the laws of a country).

There are many instances in current democratic societies around the globe where scholars question the breakdown of reciprocity, such as the rise of the Black Lives Matter Movement, the inauguration of Donald Trump in 2016, and the economic disparity between citizens [48]. Thus, the breakdown of institutional trust and reciprocity in a democratic society is not a result of AI development, such as opaque operating systems underlying social media. However, AI development adds to the breakdown of already weakened institutional trust, thus amplifying the instability of democratic institutions. It is not only opaque AI but also AI development in the background culture that impacts the rights and liberties of citizens and undermines institutional trust. Let us look at the former.

Opaque AI undermines institutional trust as it amplifies inequalities between citizens. Returning to the COMPAS example, Loomis and other black offenders were unfairly

targeted by the AI operating system. As a result of this algorithmic bias and opaque decision-making, this group of citizens may lose trust in the judicial system as they do not see themselves as equal beneficiaries of the system of justice in comparison to their fellow white citizens. The same could be said for those who are unfairly denied university admission, access to home loans, job positions, or suffer misdiagnosis due to how opaque AI may discriminate against persons. If citizens find evidence that makes them question the fairness and equality of the system, to their detriment, they are more likely to undermine the system than support it. Maclure [29] suggests that the narrow view of public reason can help strengthen ties of institutional trust regarding high-risk algorithms. However, I disagree, given the limited scope of the narrow view.

When AI technologies are governed by the requirements of the narrow view, any decision that impacts the basic structure of society or the constitutional essentials would be required to make the decision-making process behind these technologies open to the public. In theory, the COMPAS risk management algorithm would have to be publicly justifiable since it is used in the judicial system, which forms part of the Rawlsian basic structure. Although the narrow view could help require public justification and make high-risk AI operating systems align with democratic and liberal ideals this narrow view presents significant limitations to the requirements that can be in place to protect the background culture.

AI technologies permeate various aspects of society beyond the basic structure and constitutional essentials, as discussed in the previous section. These technologies infiltrate the private spheres of citizens' lives; as such, the role and usage of technologies infiltrate both the background culture and private spheres of citizens' lives. Banks use AI algorithms in the background culture for loan approvals, healthcare providers for diagnostic testing, businesses for recruitment, and companies for self-driving cars, among other areas of AI infiltration.

In these examples, AI algorithms and their implications move beyond the basic structure. Research shows that AI-driven hiring algorithms exhibit biases that can unfairly disadvantage certain groups of people, such as women and minorities [12, 47]. This algorithm would undermine public trust in institutions, as women and minorities could start questioning the legitimacy of the hiring system. Thus, I claim that the narrow view is insufficient in scope as it does not require public justification for hiring algorithms since hiring practices are not traditionally seen as a constitutional essential; in instances like this, as long as the constitution itself aligns with public reason, then hiring practices (part of the background culture) can promote non-public reasons. Thus, the narrow view accommodates injustices in areas in the background culture. The narrow view, due to its insufficient scope, can encourage inequalities that would undermine public and institutional trust and civic friendship.

One could argue that the narrow view's limitation of scope is not a failure of the view since the background culture would be just in an ideal theory in the basic structure itself, to begin with. That could, in theory, be the case; however, in nonideal theory and in thinking about justice for a nonideal democratic world, one needs to aim for the ideals of justice on both a top-down approach (basic structure to background culture) as well as a bottom-up approach (restricting background culture to public reason to align it with the basic structure).

In contrast to the above, under the broad view, hiring algorithms, university admissions algorithms, social media business models, and content moderation policies (to name a few examples) would need public justification. The broad view, in theory, ensures that any opaque AI (whether technically, intentionally, due to scale or context-dependent opacity) must conform to public justification.[10] Thus, in theory, this helps ensure that the decision-making processes within an AI technology's life cycle are transparent insofar as citizens would be provided with reasons for why and how the technology is applied to them.

These 'reasons' will be public, hence aligning with the democratic ideals in public political culture, meaning that they are reasons all reasonable[11] citizens can support. The broad view could help prevent biases that undermine institutional trust and civic friendship since discriminatory algorithms would be easily identifiable since the algorithms themselves could not remain intentionally opaque (i.e. due to propriety reasons) as the reason to withhold information would not align with public reason ideals. If the algorithm itself is technically opaque or opaque due to scale and context, then other aspects of the decision-making process within the life cycle of the AI technology can still be justified by appealing to public reason, even if the interpretability and explainability of the algorithm itself cannot be. Thus, the broad view by extending the scope of public reason beyond constitutional essentials could help promote fairness and equality in the background culture, which would help strengthen institutional trust.

Let us return to the example of Elon Musk and his promotion of content on X that aligns with his conception of the good. When citizens see that AI technologies can be used for the benefit of a select person (Musk) or group of persons (people that also agree with far-right content) at the expense of other citizens' interests (receiving equal recommendations of all political views), the disenfranchised users of X may lose trust in the business, or CEO. This trust may diminish as these users do not receive the mutual benefit of the business model or technology etc. that their fellow citizen(s) do, those that have their views amplified on X.

By holding Musk accountable to a broad view of public reason, citizens' trust in X may increase, as a broad view ensures that the justification of business models, the use of algorithmic recommendations, and advertisements, etc., serve and promote the democratic interests of citizens in general as opposed to a select number. This is crucial in a liberal democracy, where citizens' mutual commitment to justice is foundational for a stable society. Public and institutional trust is imperative for effectively creating a background culture supporting democratic aims. It is precisely this trust that a narrow view in a nonideal society cannot safeguard, whereas a broad view can. Given that so far, I have argued for the strengthens of public reason in general and a broad view specifically, it is appropriate to now turn to a challenge to using public reason as the method of justification in democratic societies.

[10] See: Da Silva's [16] method for making AI appeal to public reason without the constraints of strong explainability requirements.

[11] The term 'reasonable person' refers to persons who are willing to abide by the constraints of reasonability, meaning that they are open to constraining their moral views, and acknowledging all persons are free and equal and that society should be fair and open in terms of opportunities, see: [23, 37, 38, 43] for a more detailed discussion.

5 A Challenge to Public Reason

Robert Talisse [50] criticises public reason as 'epistemically exclusionary', which applies to both the broad and narrow views.[12] Talisse argues that public reason, in theory, could lead to political instability instead of political stability [50]. As summarised, public reason defines the kinds of arguments that should be considered epistemically legitimate. For example, suppose one is to argue that women should not have a legal right to abortion. In that case, the justification needs to be political instead of moral (i.e., appeal to values in the public political culture instead of, say, comprehensive values of a religious doctrine). Thus, public reason favours or amplifies liberal arguments that conform to the boundaries of public reason (i.e., arguments that appeal to the political values embedded in the public political culture). According to Talisse [50], the implication of this is that public reason leads to an "epistemically closed system: what counts as a reason, and what qualifies as an acceptable conclusion, are set in advance by the concept of the reasonable. This means that no antiliberal position could be reasonable, no matter how tightly argued or well-supported". Thus, a political system upholding public reason cannot acknowledge the epistemic force of non-liberal arguments. This leads to public reason excluding epistemic arguments that cannot be given public justification. Persons with excluded epistemic views will not attempt to propose their views in public forums as they know they cannot be publicly supported. Instead, these same persons will find like-minded citizens who also hold illiberal views and develop associational groups around these views. According to Cass Sunstein, individuals in like-minded groups have a higher chance of developing extremist views since persons in discussions with other like-minded individuals tend to hold a more extremist version of their initial beliefs [49]. Thus, since public reason epistemically excludes non-liberal people's views, non-liberal persons hold more extreme positions of their own views, fuelling extremism within democratic life, therefore undermining the stability of democratic institutions.

How could this epistemic exclusion apply to the context of AI justification? Let us return to the example hypothetical example of the CEO of a social media company. Suppose this CEO is presenting her justification for the content that should be moderated on her platform. In terms of the broad view, since the CEO can wield coercive power over citizens, the CEO's justification must meet the conditions of public reason. Suppose this CEO wants to use content filters to block content promoting transgender rights. The CEO must justify her stance via reasons the public can accept. Thus, if the CEO wants to justify her antitransgender stance in the name of promoting 'traditional gender values' associated with her moral comprehensive doctrine, then the CEO would be epistemically excluded from doing so in line with Talisses' reasoning. The question then is: Is public reason too epistemically demanding or unfair if it omits all epistemic justification that is illiberal? I agree that public reason is demanding since it would require Elon Musk (for example) to bracket his moral convictions and hold his political views to public scrutiny under conditions all could accept.

No social system can accommodate all notions of the good or promote all ways of life. Some values will necessarily be excluded at the expense of others, some rules will prevent others and that is the nature of a political system. Unavoidably, we need a

[12] For other criticisms see: [36].

method to justify actions from being classified as just or unjust. The moral assumption of a democratic society is that a just society safeguards freedom, equality, tolerance, and liberty of conscience, which must be the constraint of our reasoning when deciding what is right or wrong. Thus, although public reason is epistemically exclusionary, it is the constraints of democratic politics which delineate what should be excluded. Why should a democratic political system not be exclusionary to illiberal political views? Views that undermine precisely the foundation of justice. Views that undermine freedom, equality and fairness for all citizens cannot be supported with public reasons.

6 Conclusion

In conclusion, I have highlighted the need for a broad view of public reason to apply to justify AI technology. The debate of public reason in AI literature arises from the questions of justifications for opacity. Public reason, in general, has the potential to provide a method of justification for AI technology that all reasonable persons can agree with, as the constraints of justification are values embedded in the public political culture of democratic societies. The narrow view of public reason is undesirable due to its limited scope. There are many areas where citizens exercise coercive power over one another, extending beyond constitutional essentials and the basic structure. As was illustrated with the examples of the CEO in the background culture. Given this, the narrow view falls short of addressing the broader implications of AI technologies in the background culture. By demanding public justification in all political matters, the broad view offers a more comprehensive safeguard against injustices amplified by AI technologies, thereby fostering more robust freedom, equality, institutional trust, and a stable society.

References

1. Al Jazeera: The right-wing lurch of X under Elon Musk (2024). Al Jazeera. https://www.aljazeera.com/economy/2024/8/13/the-right-wing-lurch-of-x-under-elon-musk. Accessed 26 July 2024
2. Amnesty International: Myanmar: Facebook's systems promoted violence against Rohingya; Meta owes reparations – new report (2022). Amnesty International. https://www.amnesty.org/en/latest/news/2022/09/myanmar-facebooks-systems-promoted-violence-against-rohingya-meta-owes-reparations-new-report/. Accessed 26 August 2023
3. Adomavicius, G., Zhang, J.: Stability of recommendation algorithms. ACM Trans. Inf. Syst. **30**(4), 1–31 (2012)
4. Banker, S., Khetani, S.: Algorithm overdependence: how the use of algorithmic recommendation systems can increase risks to consumer well-being. J. Public Policy Mark. **38**(4), 500–515 (2019)
5. Benton, B., Choi, J.A., Luo, Y., Green, K.: Hate speech spikes on Twitter after Elon Musk acquires the platform. School of Communication and Media, Montclair State University (2022). https://digitalcommons.montclair.edu/scom-facpubs/33/. Accessed 6 Aug 2024
6. Bernholz, L., Landemore, H., Reich, R.: Digital Technology and Democratic Theory. University of Chicago Press. (2021). https://www.bibliovault.org/BV.landing.epl?ISBN=9780226748603. Accessed 6 Aug 2024
7. Binns, R.: Algorithmic accountability and public reason. Philos. Technol. **31**(4), 543–556 (2018)

8. Bland, A.: Monday briefing: how Elon Musk is shaping a dangerous political moment. The Guardian (Online) (2024). https://www.proquest.com/docview/3091494415/citation/1DC559E6E81A47B8PQ/1. Accessed 10 July 2024
9. Burrell, J.: How the machine 'thinks': Understanding opacity in machine learning algorithms. Big Data Soc. **3**(1), 1–12 (2016). https://doi.org/10.1177/2053951715622512
10. Center for Countering Digital Hate: Musk misleading election claims viewed 1.2bn times on X – with no fact checks. Center for Countering Digital Hate (2024). https://counterhate.com/research/musk-misleading-election-claims-viewed-1-2bn-times-on-x-with-no-fact-checks/. Accessed 5 Aug 2024
11. Charney, E.: Political liberalism, deliberative democracy, and the public sphere. Am. Polit. Sci. Rev. **92**(1), 97–110 (1998)
12. Chen, Z.: Ethics and discrimination in artificial intelligence-enabled recruitment practices. Humanit. Soc. Sci. Commun. **10**(1), 1–12 (2023)
13. Chesterman, S.: Through a glass, darkly: artificial intelligence and the problem of opacity. Am. J. Comp. Law **69**(2), 271–294 (2021)
14. Coeckelbergh, M.: Democracy, epistemic agency, and AI: political epistemology in times of artificial intelligence. AI Ethics **3**(4), 1341–1350 (2023)
15. Coeckelbergh, M.: Why AI Undermines Democracy and What to Do About It. Wiley (2024)
16. Da Silva, M.: Explainability, public reason, and medical artificial intelligence. Ethic Theory Moral Prac. **26**(5), 743–762 (2023)
17. Davies, C.: Elon Musk should face arrest if he incited UK rioters, says ex-Twitter chief. The Guardian (2024). https://www.theguardian.com/technology/article/2024/aug/12/elon-musk-should-face-arrest-if-he-incited-uk-rioters-says-ex-twitter-chief. Accessed 4 June 2023
18. Diakopoulos N.: Algorithmic accountability reporting: on the investigation of black boxes. Report, Tow Center for Digital Journalism, Columbia University (2013)
19. El-Bermawy, M.M.: Your filter bubble is destroying democracy. Wired (2016). https://www.wired.com/2016/11/filter-bubble-destroying-democracy/. Accessed 7 July 2019
20. Facchini, A., Termine, A.: Towards a taxonomy for the opacity of AI systems. In: Müller, V.C. (ed.) PTAI 2021. Studies in Applied Philosophy, Epistemology and Rational Ethics, vol. 63, pp. 73–89. Springer, Cham (2021). https://doi.org/10.1007/978-3-031-09153-7_7
21. Fainman, A.A.: Technology and the Law: The problem with opaque AI. Thinker 82 (2019)
22. Flores, A.W., et al.: False positives, false negatives, and false analyses: a rejoinder to machine bias: there's software used across the country to predict future criminals and it's biased against blacks. Fed. Probation **80**, 38 (2016)
23. Friedman, M.: Autonomy, Gender, Politics, vol. 248. Oxford University Press (2003)
24. Gupta, S.: A literature review on recommendation systems. Int. Res. J. Eng. Technol. **7**, 3600–3605 (2020)
25. Kim, J-Y., Heo, W.: Artificial intelligence video interviewing for employment: perspectives from applicants, companies, developer and academicians. Inf. Technol. People (2021). https://doi.org/10.1108/ITP-04-2019-0173
26. Kleinberg, J., Lakkaraju, H., Leskovec, J., Ludwig, J., Mullainathan, S.: Human decisions and machine predictions. Q. J. Econ. **133**(1), 237–293 (2017). https://doi.org/10.1093/qje/qjx032
27. Kolhatkar, S.: Elon Musk plans to profit from Twitter, not create a town square for global democracy CounterPunch. (2022). https://rozenbergquarterly.com/page/54/?e=25. Accessed 10 May 2023
28. Le Monde: How Elon Musk uses X to support the far right and its financial interests. Le Monde (2024). https://www.lemonde.fr/en/international/article/2024/08/13/how-elon-musk-uses-x-to-support-the-far-right-and-its-financial-interests_6714673_4.html. Accessed 5 Aug 2024

29. Maclure, J.: AI, explainability and public reason: the argument from the limitations of the human mind. Mind. Mach. **31**(3), 421–438 (2021)
30. McKinlay, S.: Trust and Algorithmic Opacity. In: Big Data and Democracy, pp. 153–166. Edinburgh University Press (2020)
31. Miller, S.: Machine learning, ethics and law. Aust. J. Inf. Syst. (2019). https://doi.org/10.3127/ajis.v23i0.1893
32. Mittelstadt, B., et al.: Explaining explanations in AI. In: Proceedings of the Conference on Fairness, Accountability, and Transparency, pp. 279–288 (2019)
33. Müller, V.C.: Deep opacity undermines data protection and explainable artificial intelligence. Overcoming Opacity Mach. Learn. **18** (2021)
34. Newton, D.: Artificial Intelligence grading your 'neuroticism'? Welcome to college's new frontier. USA Today (2021) https://www.usatoday.com/story/news/education/2021/04/26/ai-infiltrating-collegeadmissions-teaching-grading/7348128002/
35. PBS News: How Elon Musk uses his X social media platform to amplify right-wing views. PBS News (2024). https://www.pbs.org/newshour/politics/how-elon-musk-uses-his-x-social-media-platform-to-amplify-right-wing-views. Accessed 27 July 2024
36. Quong, J.: The scope of public reason. Polit. Stud. **52**(2), 233–250 (2004)
37. Quong, J.: Liberalism without Perfection, 1st edn. Oxford University Press (2011)
38. Quong, J.: Liberalism without perfection: Replies to Lister, Kulenovic, Zoffoli, Zelic, and Baccarini. Filoz. drus. **25**(1), 96–122 (2014)
39. Quong, J.: Public reason (2013). https://seop.illc.uva.nl/entries/public-reason/. Accessed 10 May 2020
40. Raghu, M., Blumer, K., Corrado, G., Kleinberg, J., Obermeyer, Z., Mullainathan, S.: The algorithmic automation problem: prediction, triage, and human effort (2019). https://arxiv.org/abs/1903.12220
41. Rawls, J.: The idea of public reason revisited. Univ. Chicago Law Rev. **64**(3), 765 (1997)
42. Rawls, J.: A Theory of Justice. Revised edn., p. 538. Belknap Press of Harvard University Press, Cambridge (1999)
43. Rawls, J.: Political Liberalism: Expanded Edition. Columbia Classics in Philosophy Edition. Columbia University Press (2005)
44. Rawls, J., Kelly, E.: Justice as Fairness: a Restatement, p. 214. Harvard University Press, Cambridge (2001)
45. Rini, R.: Social media disinformation and the security threat to democratic legitimacy. NATO Association of Canada: Disinformation and Digital Democracies in the 21st Century, pp. 10–14 (2019)
46. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nat. Mach. Intell. **1**(5), 206–215 (2019)
47. Scatiggio, V.: Tackling the issue of bias in artificial intelligence to design AI-driven fair and inclusive service systems (2020). https://www.politesi.polimi.it/handle/10589/186118. Accessed 10 June 2024
48. Scheffler, S.: The rawlsian diagnosis of Donald Trump. Boston Review (2019). https://www.bostonreview.net/articles/samuel-scheffler-rawlsian-diagnosis-donald-trump/. Accessed 10 Apr 2022
49. Sunstein, C.R. (ed.) The law of group polarization. In: Debating Deliberative Democracy. Blackwell, London (2003)
50. Talisse, R.B.: Social epistemology and the politics of omission. Episteme **2**(2), 107–118 (2006)
51. The Washington Post: How X owner Elon Musk uses his social platform to amplify his right-wing views. The Washington Post (2024). https://www.washingtonpost.com/politics/2024/08/13/elon-musk-x-influence-uk-us-politics/c80d5a3a-5928-11ef-93a9-023ab69f91f5_story.html. Accessed 15 July 2024

52. Tomasi, J.: Liberalism Beyond Justice: Citizens, Society, and the Boundaries of Political Theory, p. 163. Princeton University Press, Princeton (2001)
53. UNESCO: Recommendation on the Ethics of Artificial Intelligence (2021). https://unesdoc.unesco.org/ark:/48223/pf0000374266. Accessed 26 Aug 2023
54. Von Eschenbach, W.J.: Transparency and the black box problem: why we do not trust AI. Philos. Technol. **34**(4), 1607–1622 (2021
55. Wachter, S., Mittelstadt, B., Floridi, L.: Why a right to explanation of automated decision-making does not exist in the general data protection regulation. Int. Data Priv. Law **7**(2), 76–99 (2017)
56. Zednik, C.: Solving the black box problem: a normative framework for explainable artificial intelligence. Philos. Technol. **34**(2), 265–288 (2021)

Symbolic AI and Knowledge Representation and Reasoning

Towards Propositional KLM-Style Defeasible Standpoint Logics

Nicholas Leisegang[1(✉)], Thomas Meyer[1], and Sebastian Rudolph[2,3]

[1] University of Cape Town and CAIR, Cape Town, South Africa
lsgnic001@myuct.ac.za, tommie.meyer@uct.ac.za
[2] Technische Universität Dresden, Dresden, Germany
sebastian.rudolph@tu-dresden.de
[3] Center for Scalable Data Analytics and Artificial Intelligence, Dresden/Leipzig, Germany

Abstract. The KLM approach to defeasible reasoning introduces a weakened form of implication into classical logic. This allows one to incorporate exceptions to general rules into a logical system, and for old conclusions to be withdrawn upon learning new contradictory information. Standpoint logics are a group of logics, introduced to the field of Knowledge Representation in the last 5 years, which allow for multiple viewpoints to be integrated into the same ontology, even when certain viewpoints may hold contradicting beliefs. In this paper, we aim to integrate standpoints into KLM propositional logic in a restricted setting. We introduce the logical system of Defeasible Restricted Standpoint Logic (DRSL) and define its syntax and semantics. Specifically, we integrate ranked interpretations and standpoint structures, which provide the semantics for propositional KLM and propositional standpoint logic respectively, in order to introduce ranked standpoint structures for DRSL. Moreover, we extend the non-monotonic entailment relation of rational closure from the propositional KLM case to the DRSL case. The main contribution of this paper is to characterize rational closure for DRSL both algorithmically and semantically, showing that rational closure can be characterized through a single representative ranked standpoint structure. Finally, we conclude that the semantic and algorithmic characterizations of rational closure are equivalent, and that entailment-checking for DRSL under rational closure is in the same complexity class as entailment-checking for propositional KLM.

Keywords: Knowledge representation · Non-monotonic reasoning · Defeasible reasoning · Standpoint logic

1 Introduction

Within the field of symbolic AI, much work has been dedicated to reasoning with information which is incomplete, or is seemingly contradictory. One avenue in which contrasting beliefs is explored is through non-monotonic reasoning, which considers cases where conclusions can be made based on a given set of beliefs, and then withdrawn if new information arises that contradicts previous conclusions. A specific form of such non-monotonic reasoning was established by Kraus et al. [19], who introduced a defeasible implication $\mid\!\sim$ into classical propositional logic, where the term $\alpha \mid\!\sim \beta$ reads as "α *typically implies* β". More specifically, this reads as saying that all the most

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 459–475, 2025.
https://doi.org/10.1007/978-3-031-78255-8_27

typical occurrences of α should also satisfy β. Thus, we are able to incorporate rules with exceptions into our logic, since we can have instances where α is true but β is not, and merely interpret this as some exceptional instance of α. This approach to defeasible reasoning is referred to as the KLM framework.

Another instance where contradictory information occurs in knowledge representation, is where we wish to integrate the views of two agents into a single ontology or knowledge base, where the views of these agents may contradict each other. One way to do this is to weaken or exclude the contradicting beliefs between agents. However, this method may sacrifice the accuracy of the beliefs represented. In order to remedy this, Gómez Álvarez and Rudolph [15] introduce *standpoint* modal operators, $\Box_s$ and $\Diamond_s$, to the propositional case where $\Box_s\phi$ reads that "*it is unequivocal to s that* ϕ" and $\Diamond_s\phi$ reads that "*it is possible to s that* ϕ". Standpoint modal operators have also been introduced by Gómez Álvarez et al. [16] in the case of first-order and description logics. However, the above cases of standpoint logics continue to be monotonic in nature.

In this paper, we consider the case where we combine both approaches in order to represent, in a single logical framework, multiple agents who hold (possibly contradicting) beliefs, where these beliefs may be defeasible in their nature. We will also build a non-monotonic notion of entailment, which extends the well-known system of rational closure in propositional KLM [21]. A motivating example for this case is given below.

Example 1. In the 19th century, crops imported into the USA were divided into fruits and vegetables, where fruits were exempt from import tax. This led to a court case on whether a tomato should be legally classified as a fruit or a vegetable. From a botanical standpoint, tomatoes are fruits and furthermore all fruits are also vegetables, since vegetables are considered all commonly eaten plants. This can be expressed using standpoint logics with the set:

$$\{\Box_B(tomato \rightarrow fruit), \Box_B(fruit \rightarrow vegetable)\},$$

where B represents the botanical standpoint. However, the court considered a different standpoint based on the culinary use of tomatoes. This states that vegetables are those crops which are savoury and fruits are those which are sweet. That is,

$$\{\Box_C(savoury \leftrightarrow vegetable), \Box_C(sweet \leftrightarrow fruit), \Box_C(tomato \mid\sim savoury),$$

$$\Box_C(fruit \mid\sim \neg vegetable) \wedge \Box_C(vegetable \mid\sim \neg fruit)\},$$

where C represents the "culinary-use" standpoint on crops. The first two statements above tell us the specified definitions for fruits and vegetables. The third statement tells us that tomatoes are usually used in savoury dishes, and the last proposition states that in a culinary perspective, fruits and vegetables are usually distinct from each other. The court concluded that from the "culinary-use" standpoint, tomatoes were usually considered vegetables, and therefore should be taxed. We can represent this by $\{L \preceq C, \Box_L(vegetable \rightarrow \neg fruit)\}$, where $L \preceq C$ tells us that the legal standpoint L holds true each conclusion of C's standpoint, and the second proposition states that fruits and vegetables are strictly distinct, legally speaking. Importantly, this system allows for both internal exceptions, and strict disagreements between standpoints. For example,

B holds that every fruit is a vegetable, while L holds that no fruit is a vegetable. We also see that it is possible that a certain kind of exceptional tomato is not savoury from a culinary standpoint. In fact, from a culinary perspective it seems possible to consider a tomato a fruit and a vegetable (i.e. $\Diamond_C(tomato \rightarrow (fruit \wedge vegetable))$), although this is not possible from a legal perspective.

In order to formally analyse examples as the one above, we formally introduce *Defeasible Restricted Standpoint Logic* (DRSL) and describe how this can be used to reason non-monotonically with multi-perspective knowledge bases. In Section 2, we provide a background on propositional KLM defeasibility, the non-monotonic entailment relation of rational closure and propositional standpoint logic in the classical case. In Section 3, we describe the syntax and semantics for DRSL. In Section 4, we describe how rational closure can be lifted from the propositional case to the case of standpoint logic, and show a correspondence between an algorithmic and a semantic definition of rational closure. Furthermore, we show that the algorithm for entailment-checking in rational closure is in the same complexity class as entailment-checking in the propositional case. An extended version of this paper with proofs is available in an arXiv submission with the same title as this paper.

2 Background

2.1 Propositional KLM and Rational Closure

In this section we define the syntax and semantics for KLM in the propositional case, as well as recall certain results which we rely upon later on. In particular, we are considering the case where $\mid\!\sim$ forms a rational consequence relation as is defined by Lehmann and Magidor [21]. We use a slightly more expressive syntax than the traditional KLM which allows for conjunctions between propositions.

That is, we define the language of KLM propositional logic $\mathcal{L}^{\mid\sim}$, over a set of propositional atoms $\mathcal{P}$, as defined by

$$\phi ::= \alpha \mid \alpha \mid\!\sim \beta \mid \phi \wedge \phi,$$

where α and β are Boolean formulas with atoms in $\mathcal{P}$.

The semantics for $\mathcal{L}^{\mid\sim}$ is referred to as ranked interpretations. These were defined originally as a Kripke-style semantics [19]. However, in this paper we use an equivalent notion of ranking functions to define such an interpretation [8]. For the definition below $\mathcal{U}$ refers to the set of classical interpretations for the atoms in $\mathcal{P}$. That is, each $u \in \mathcal{U}$ is a map that assigns each $p \in \mathcal{P}$ a truth value.

A **ranked interpretation** is therefore defined as a function $R : \mathcal{U} \rightarrow \mathbb{N} \cup \{\infty\}$, such that the following convexity property is satisfied: if $R(u) < \infty$, then for every $0 \leq j < R(u)$ there exists $v \in \mathcal{U}$ such that $R(v) = j$.

A ranked interpretation intuitively tells us how typical a state of the world ought to be, with those valuations with higher ranks being less typical than those with lower ranks. Those valuations with rank ∞ then refer to states of the world that are "impossible". We define $[\![\alpha]\!] = \{u \in \mathcal{U} \mid u \Vdash \alpha\}$ for any Boolean formula α, and let $\mathcal{U}^R = \{u \in \mathcal{U} \mid$

$R(u) \neq \infty\}$. Then we say $R \Vdash \alpha$ iff $[\![\alpha]\!] \supseteq \mathcal{U}^R$. We also say that $R \Vdash \alpha \mathrel{|\!\sim} \beta$ if and only if $\{u \in [\![\alpha]\!] \mid R(v) \not< R(u) \text{ for all } v \in [\![\alpha]\!]\} \subseteq [\![\beta]\!]$. That is, if all the valuations satisfying α with a minimal rank also satisfy β, or "the most typical α-instances are instances of β". We further note that a Boolean formula α can be expressed equivalently by a defeasible implication $\neg\alpha \mathrel{|\!\sim} \bot$ [19].

For a set $\mathcal{K} \in \mathcal{L}^{|\sim}$ we also say that $R \Vdash \mathcal{K}$ when $R \Vdash \phi$ for all $\phi \in \mathcal{K}$. We therefore define that $R \Vdash \phi_1 \wedge \phi_2$ if and only if $R \Vdash \{\phi_1, \phi_2\}$. The reason we include conjunctions in the language $\mathcal{L}^{|\sim}$ in this paper is to allow for increased expressivity when we bound formulas by standpoint modal operators in Section 3.

Example 2. We consider a knowledge base $\mathcal{K} = \{p \rightarrow b, b \mathrel{|\!\sim} f, p \mathrel{|\!\sim} \neg f\}$. Then a ranked interpretation which is a model of $\mathcal{K}$ is represented in the table below, where each valuation is placed in the row corresponding to its rank. Moreover, each valuation is represented by a sequence of atomic propositions in $\mathcal{K}$, where a bar is placed over the top of each atom which is false in the valuation. For example, the valuation $pb\bar{f}$ is the valuation where p and b are true and f is false, and has rank 1 in the table below:

∞	$p\bar{b}f, p\bar{b}\bar{f}$
2	pbf
1	$pb\bar{f}, \bar{p}b\bar{f}$
0	$\bar{p}bf, \bar{p}\bar{b}f, \bar{p}\bar{b}\bar{f}$

It is also well-known in the KLM literature that there are multiple non-equivalent notions of defeasible entailment defined to characterize what a KLM knowledge base $\mathcal{K} \subseteq \mathcal{L}^{|\sim}$ entails. The first of these, defined by Lehmann and Magidor [21], is *rational closure*. This is defined using a single ranked interpretation $R^{\mathcal{K}}_{RC}$, or equivalently, via an algorithmic approach. The algorithmic definitions are given by the `RCProp` algorithm for entailment-checking in Algorithm 2 which in turn calls on the `BaseRank` algorithm described in Algorithm 1. These are originally proposed by Freund [12] but resemble closer those occurring in Casini et al. [8]. In these algorithms, we assume that each knowledge base $\mathcal{K}$ only contains elements of the form $\alpha \mathrel{|\!\sim} \beta$, and that entailment checking is only done for elements of this form. This can be done by changing each Boolean formula α into $\neg\alpha \mathrel{|\!\sim} \bot$ and by splitting any conjunction of formulas into a set of its conjuncts.

In order to define the semantic notion of rational closure, one uses an order on models of $K \subseteq \mathcal{L}^{|\sim}$ denoted $\preceq_{\mathcal{K}}$, where $R_1 \preceq_{\mathcal{K}} R_2$ iff $R_1(u) \leq R_2(u)$ for all $u \in \mathcal{U}$.[1] It is observed by Giordano et al. [14] that the set of ranked models of $\mathcal{K}$, denoted $\mathcal{R}^{\mathcal{K}}$, has a minimum with respect to the order $\preceq_{\mathcal{K}}$. This minimal element is denoted $R^{\mathcal{K}}_{RC}$. This assumption states that we ought to reason as though things are as typical as they can be unless we have reason to believe otherwise. This corresponds to moving valuations in $\mathcal{U}$ to their lowest rank possible without contradicting anything occurring in the knowledge base $\mathcal{K}$. From this, we obtain the following representation result for

[1] It is worth noting this holds with respect to valuations $\mathcal{U}$ on $\mathcal{P}$, although not all atomic propositions in $\mathcal{P}$ may occur in $\mathcal{K}$. In the literature these two sets of atoms are often considered to be the same, but this is not the case when we introduce standpoints, since there may be atoms that do not occur in the beliefs of one standpoint that occur in the beliefs of another.

Algorithm 1. BaseRank

Input: A propositional KLM knowledge base $\mathcal{K}$
Output: An ordered tuple $(R_0, ..., R_{n-1}, R_\infty, n)$

$i := 0$;
$E_0 := \{\alpha \rightarrow \beta \mid \alpha \mathrel{|\!\sim} \beta \in \mathcal{K}\}$;
repeat
  $E_{i+1} := \{\alpha \rightarrow \beta \in E_i \mid E_i \vDash \neg\alpha\}$;
  $R_i := E_i - E_{i+1}$;
  $i := i + 1$;
until $E_{i-1} = E_i$;
$R_\infty := E_{i-1}$;
if $E_{i-1} = \emptyset$ **then**
  $n := i - 1$;
else
  $n := i$;
end if
return $(R_0, ..., R_{n-1}, R_\infty, n)$

Algorithm 2. RCProp

Input: A propositional KLM knowledge base $\mathcal{K}$ and a defeasible implication $\alpha \mathrel{|\!\sim} \beta$
Output: True if $\mathcal{K} \mathrel{|\!\approx}_{RC} \alpha \mathrel{|\!\sim} \beta$ and False otherwise

$(R_0, ..., R_{n-1}, R_\infty, n) :=$ BaseRank($\mathcal{K}$);
$i := 0$;
$R := \bigcup_{j=0}^{j<n} R_j$;
while $R_\infty \cup R \vDash \neg\alpha$ and $R \neq \emptyset$ **do**
  $R := R - R_i$;
  $i := i + 1$;
end while
return $R_\infty \cup R \vDash \alpha \rightarrow \beta$

propositional rational closure. For a knowledge base $\mathcal{K}$ and a defeasible implication $\alpha \mathrel{|\!\sim} \beta$, RCProp($\mathcal{K}, \alpha \mathrel{|\!\sim} \beta$)=True if and only if $R^{\mathcal{K}}_{RC} \Vdash \alpha \mathrel{|\!\sim} \beta$ [21,24]. This result can be extended to conjunctions of defeasible implications by once again treating them as sets of their conjuncts.

Definition 1. *Consider $\phi \in \mathcal{L}^{\mathrel{|\!\sim}}$. If $\phi = \alpha \mathrel{|\!\sim} \beta$, we say ϕ is in the rational closure of a knowledge base $\mathcal{K}$, or write $\mathcal{K} \mathrel{|\!\approx}_{RC} \phi$, when $R^{\mathcal{K}}_{RC} \Vdash \alpha \mathrel{|\!\sim} \beta$. Equivalently, where RCProp($\mathcal{K}, \alpha \mathrel{|\!\sim} \beta$)=True. If $\phi = \phi_1 \wedge \phi_2$, we say $\mathcal{K} \mathrel{|\!\approx}_{RC} \phi$ when $\mathcal{K} \mathrel{|\!\approx}_{RC} \phi_1$ and $\mathcal{K} \mathrel{|\!\approx}_{RC} \phi_2$.*

We also note the original complexity result for rational closure in the propositional case, which states that `RCProp` is computable in P^{NP} [21].

2.2 Standpoint Logics

Standpoint logic is introduced by Gómez Álvarez and Rudolph [15] in the propositional case using the following syntax and semantics.

Definition 2. *Consider a vocabulary $\mathcal{V} = (\mathcal{P}, \mathcal{S})$, where $\mathcal{P}$ is a finite set of propositional atoms and $\mathcal{S}$ is a finite set of standpoint symbols containing the universal standpoint $*$. The language $\mathcal{L}_\$$ over $\mathcal{V}$ is defined by $\phi ::= s_1 \preceq s_2 \mid p \mid \neg\phi \mid \phi \wedge \phi \mid \Box_s \phi$, where $s_1, s_2, s \in \mathcal{S}$ and $p \in \mathcal{P}$.*

In this syntax, $s_1 \preceq s_2$ are called *standpoint sharpening statements*. The Boolean connectives $\vee$, $\rightarrow$, $\leftrightarrow$ are defined via $\neg$ and $\wedge$ in their usual manner, and for each standpoint $s \in \mathcal{S}$ we define $\Diamond_s := \neg\Box_s\neg$. The semantics for $\mathcal{L}_\$$ are defined as follows.

Definition 3. *A **standpoint structure** is a triple $M = (\Pi, \sigma, \gamma)$ where Π is a non-empty set of precisifications, $\sigma : \mathcal{S} \to P(\Pi)$ is a function which assigns each standpoint a non-empty set of precisifications and $\sigma(*) = \Pi$, and $\gamma : \Pi \to P(\mathcal{P})$ is a function which assigns each precisification a set of atoms.*

Intuitively, the map σ allows one to allocate to a standpoint s the set of all "reasonable ways to make s's beliefs correct", and γ assigns a set of basic propositions which are "true" in that precisification. Equivalently, it associates to each π a classical valuation on $\mathcal{P}$, describing the contents of what the precisification π holds true. This notion is emphasised by the definition of the satisfaction relation $\Vdash$.

Definition 4. *For a standpoint structure M and a precisification $\pi \in \Pi$, we define the satisfaction relation $\Vdash$ inductively as follows:*

- $M, \pi \Vdash p$ *iff* $p \in \gamma(\pi)$.
- $M, \pi \Vdash \neg\phi$ *iff* $M, \pi \nVdash \phi$.
- $M, \pi \Vdash \phi_1 \wedge \phi_2$ *iff* $M, \pi \Vdash \phi_1$ *and* $M, \pi \Vdash \phi_2$.
- $M, \pi \Vdash \Box_s \phi$ *iff* $M, \pi' \Vdash \phi$ *for all* $\pi' \in \sigma(s)$.
- $M, \pi \Vdash s_1 \preceq s_2$ *iff* $\sigma(s_1) \subseteq \sigma(s_2)$
- $M \Vdash \phi$ *iff* $M, \pi \Vdash \phi$ *for all* $\pi \in \Pi$.

In the above $\phi, \phi_1\phi_2 \in \mathcal{L}_{\mathbb{S}}$, $s, s_1, s_2 \in \mathcal{S}$ and $p \in \mathcal{P}$.

Based on these semantics, and those for KLM, we propose an integration of these two logics in the following section.

3 Defeasible Restricted Standpoint Logic

3.1 Syntax

In the language of DRSL, we consider the case where standpoint modal operators are applied to propositional KLM formulas, with some restrictions of how standpoint bound KLM formulas can be combined. Formally, this is defined below.

Definition 5. *A **vocabulary** is a pair $\mathcal{V} = (\mathcal{P}, \mathcal{S})$ where $\mathcal{P}$ is a finite set of propositional atoms and $\mathcal{S}$ is a finite set of standpoints containing the universal standpoint $*$. The language $\mathcal{L}_{\mathbb{S}}^{\mid\sim}$ is defined as follows,*

$$\psi ::= \phi \mid \#_s \psi \mid \psi \wedge \psi \quad \text{or} \quad e ::= s_1 \preceq s_2,$$

where $\# = \Box$ or $\# = \Diamond$, s_1 and s_2 are standpoints and ϕ is a formula in $\mathcal{L}^{\mid\sim}$ with atoms in $\mathcal{P}$.

Note here that we define the syntax in such a way that we can only apply standpoint modal operators to the outside of existing formulas in the language of KLM logic. For example, $\Box_s(p \mid\sim (q \vee r))$ would be a valid formula in $\mathcal{L}_{\mathbb{S}}^{\mid\sim}$ while $\Box_s p \mid\sim \Box_{s'}(q \vee r)$ is not valid. We also define the dual operator $\Diamond_s$ explicitly since we do not allow for negation of modal operator bound formulas. We also do not allow for disjunctions of

formulas in our syntax. Due to this, we lose the ability to express the related operator $\mathcal{D}_s$ defined by $\mathcal{D}_s\phi = \Box_s\neg\phi \vee \Box_s\phi$, which is read to mean that ϕ's truth is determined in s's standpoint [15]. However, we are able to express in the case where ϕ is Boolean the additional standpoint operator $\mathcal{I}_s\phi := \Diamond_s\neg\phi \wedge \Diamond_s\phi$, which states that ϕ's truth is not determined in s's standpoints. If ϕ is not Boolean, and contains some defeasible implication, we are not able to express this using our given syntax.

3.2 Semantics

The semantics we introduce to this standpoint logic follows the intuition given by Gómez Álvarez et al. [16] such that we define a set of "precisifications" of formulas, which correspond to "worlds" in a possible-worlds style semantics. Moreover, they intuitively refer to making precise those views held within a standpoint, and different ways in which such a standpoint can be made precise. Every precisificaton is then mapped to an underlying semantic structure for the "base logic". In first order, or description logics, this corresponds to a first-order interpretation structure [16]. Using the standard semantics for KLM defeasible logic, we will construct a ranked interpretation for each given precisification.

Definition 6. *Given a vocabulary $\mathcal{V} = (\mathcal{P}, \mathcal{S})$, a **ranked standpoint structure** is a triple $M = (\Pi, \sigma, \gamma)$ where:*

- *Π is a non-empty set of precisifications.*
- *$\sigma : \mathcal{S} \rightarrow P(\Pi)$ is a function that maps each standpoint symbol to a non-empty set of precisifications and in particular, $\sigma(*) = \Pi$.*
- *$\gamma : \Pi \rightarrow \mathcal{R}$ is a function, where $\mathcal{R}$ is the set of ranked interpretations over $\mathcal{P}$. That is, for each precisification π, $\gamma(\pi)$ is a ranked interpretation.*

We say that M is *valid* if and only if for each $s \in \mathcal{S}$, for every $\pi \in \sigma(s)$, there exists some valuation u such that $\gamma(\pi)(u) \neq \infty$. We then say that a ranked standpoint structure is a *model* for a formula $\xi \in \mathcal{L}_{\mathrm{S}}^{\mid\sim}$ if and only if $M \Vdash \xi$.[2] The relation $\Vdash$ is defined as follows.

Definition 7. *Given a ranked interpretation structure M and a precisification $\pi \in \Pi$, the satisfaction relation $\Vdash$ is defined inductively as follows:*

- *$M, \pi \Vdash \phi$ iff $\gamma(\pi) \Vdash \phi$ for the ranked interpretation $\gamma(\pi)$.*
- *$M, \pi \Vdash \Box_s\psi$ iff $M, \pi' \Vdash \psi$ for all $\pi' \in \sigma(s)$.*
- *$M, \pi \Vdash \Diamond_s\psi$ iff $M, \pi' \Vdash \psi$ for some $\pi' \in \sigma(s)$.*
- *$M, \pi \Vdash \psi_1 \wedge \psi_2$ iff $M, \pi \Vdash \psi_1$ and $M, \pi \Vdash \psi_2$.*
- *$M, \pi \Vdash s_1 \preceq s_2$ iff $\sigma(s_1) \subseteq \sigma(s_2)$.*
- *$M \Vdash \psi$ iff $M, \pi \Vdash \psi$ for all $\pi \in \Pi$.*

where $\phi \in \mathcal{L}^{\mid\sim}$, $\psi \in \mathcal{L}_{\mathrm{S}}^{\mid\sim}$ and $s, s_1, s_2 \in \mathcal{S}$.

[2] This definition of a model extends to sets of formulas in the usual way.

Something to note on this semantics is that the application of multiple modal operators over a single formula reduces to just one formula, as is in the case for other standpoint logics [15]. That is, $M \Vdash \#^1_{s_1}\#^2_{s_2}...\#^n_{s_n}\phi$ if and only if $M \Vdash \#^n_{s_n}\phi$, for any ranked standpoint structure M, any $s_1, ..., s_n \in \mathcal{S}$ and any $\phi \in \mathcal{L}^{\sim}$. Therefore, in the rest of this paper we only consider those formulas bound by one or no standpoint modal operators.

Another result which is evident from these semantics is that for any formula $\phi \in \mathcal{L}^{\sim}$ and any ranked standpoint structure M, we have $M \Vdash \phi$ if and only if $M \Vdash \Box_*\phi$. This is also the case in other versions of standpoint logics [15]. These two facts, along with the following Lemmas allow us to define a *normal form* for a formula in DRSL.

Lemma 1. *For any ranked standpoint structure M and any $\psi_1, \psi_2 \in \mathcal{L}^{\sim}_{\mathbb{S}}$, we have that $M \Vdash \Box_s(\psi_1 \wedge \psi_2)$ if and only if $M \Vdash \Box_s\psi_2 \wedge \Box_s\psi_2$.*

Lemma 2. *Let M be a ranked standpoint structure, $\phi_1, ..., \phi_k \in \mathcal{L}^{\sim}$. Then $M \Vdash \Diamond_s(\phi_1 \wedge \#^2_{s_2}\phi_2 \wedge ... \wedge \#^k_{s_k}\phi_k)$ if and only if $M \Vdash \Diamond_s\phi_1 \wedge \#^2_{s_2}\phi_2 \wedge ... \wedge \#^k_{s_k}\phi_k$.*

From these two results we obtain the following.

Corollary 1. *Any DRSL formula ψ can be represented equivalently by a formula in the form $\#^1_{s_1}\phi_1 \wedge ... \wedge \#^k_{s_k}\phi_k$, where $\phi_1, ..., \phi_n \in \mathcal{L}^{\sim}$.*

For example, the formula

$$\psi = \Diamond_s((p_1 \mathrel{|\!\sim} p_2) \wedge (p_3 \mathrel{|\!\sim} p_4) \wedge \Box_t(p_5 \mathrel{|\!\sim} p_6) \wedge (p_7 \mathrel{|\!\sim} p_8))) \wedge p_9$$

can be equivalently represented by

$$\widetilde{\psi} = \Diamond_s((p_1 \mathrel{|\!\sim} p_2) \wedge (p_3 \mathrel{|\!\sim} p_4)) \wedge \Box_t(p_5 \mathrel{|\!\sim} p_6) \wedge \Box_t(p_7 \mathrel{|\!\sim} p_8) \wedge \Box_* p_9$$

Definition 8. *Any formula of the form specified in Corollary 1 is said to be in **normal form**. The process of rewriting a formula ψ in normal form is called **normalization** of ψ.*

This definition for the semantics for DRSL gives rise to a natural Tarskian notion of entailment, which is analogous to *ranked entailment* defined in the propositional case of KLM [21].

Definition 9. *The **ranked entailment** relation $\models_R$ is defined as follows. Consider a knowledge base $\mathcal{K}$ of propositions in $\mathcal{L}^{\sim}_{\mathbb{S}}$ and a proposition $\xi \in \mathcal{L}^{\sim}_{\mathbb{S}}$. Then $\mathcal{K} \models_R \xi$ if and only if every ranked standpoint structure M which is a model of $\mathcal{K}$ is also a model of ξ.*

Although ranked entailment is a notion worth investigating further, it is monotonic, and therefore does not give us the desired system of non-monotonic reason we wish to construct in the DRSL case. This is shown in the lemma below.

Lemma 3. *Let $\mathcal{K}$ be a DRSL knowledge base. Then for any $\psi \in \mathcal{L}^{\sim}_{\mathbb{S}}$, we have that $\mathcal{K} \models_R \xi$ implies $\mathcal{K} \cup \{\psi\} \models_R \xi$.*

Therefore, in order to construct some kind of non-monotonic entailment for DSRL, we propose an extension of rational closure, as defined by Lehmann and Magidor [21] in the propositional case, for standpoint logics.

4 Rational Closure for DRSL

In this section we define an algorithm for rational closure in DRSL. Furthermore, we show that this algorithm can also be described semantically via a single ranked standpoint structure. This is analogous to the semantic and algorithmic representations of rational closure in the propositional case as is described by Lehman and Magidor [21].

4.1 Rational Closure Algorithm

In this section we introduce the rational closure algorithm for DRSL, and describe its complexity. This begins by constructing a "representative set" of KLM knowledge bases for each standpoint $s \in \mathcal{S}$. We split a knowledge base with formulas in $\mathcal{L}_{\mathrm{S}}^{\mid\sim}$ into several knowledge bases in $\mathcal{L}^{\mid\sim}$ which represent the KLM-knowledge bases that each standpoint s takes into account. In the base case, we determines a knowledge based only on those formulas that are unequivocal for s, called K_s, which include formulas that are bound by $\Box_t$ where $t = s$ or t is some standpoint for which "subsumes" s in the sense that s is a sharpening of t. Then we construct a separate knowledge base for the standpoint s for each formula of the form $\Diamond_s \phi$ for $\phi \in \mathcal{L}^{\mid\sim}$. That is, we create a new knowledge base representing the precisification of s where ϕ is true. We then associate each standpoint with a set of representative KLM knowledge base, denoted Know_s, which includes K_t and K_t^{ϕ} where $t = s$ or t is a sharpening of s. This is described in Algorithm 3.

Example 3. Using the same knowledge base in Example 1 we can split our DRSL knowledge base into three KLM knowledge bases:

$$\begin{aligned} K_B =& \{tomato \rightarrow fruit, fruit \rightarrow vegetable\} \\ K_C =& \{savoury \leftrightarrow vegetable, sweet \leftrightarrow fruit, tomato \mid\sim savoury, \\ & fruit \mid\sim \neg vegetable, vegetable \mid\sim \neg fruit\} \\ K_L =& \{savoury \leftrightarrow vegetable, sweet \leftrightarrow fruit, tomato \mid\sim savoury, \\ & fruit \rightarrow \neg vegetable\} \end{aligned}$$

In K_B and K_C we simply include those propositions bound by $\Box_B$ and $\Box_C$, respectively. However, in the case of K_L, since $L \preceq C$ is in our knowledge base, any statement unequivocal to C must also be unequivocal to L. Hence, we include statements bound by $\Box_L$ and $\Box_C$ in K_L. Then, we associate with each standpoint the sets of knowledge bases "relevant" to those standpoints, giving us: $\mathrm{Know}_B = \{K_B\}$, $\mathrm{Know}_C = \{K_C, K_L\}$ and $\mathrm{Know}_L = \{K_L\}$.Here note that since $L \preceq C$, we have to include K_L in Know_C, since L is a sharpening of C's standpoint and therefore is seen as a more specific version of C's view. If we were to add a formula of the form $\Diamond_L \phi$ to our knowledge base, this would create another set in the splitting defined by $K_L^{\phi} = K_L \cup \{\phi\}$. Intuitively, this means if our knowledge base asserts that it is possible for L to believe that ϕ is true, then we must consider the possibility of L's view where this is true, but that it is a divergent possible precisification to the set of statements K_L which are those unequivocal to L. In this case we would also obtain that $\mathrm{Know}_C = \{K_C, K_L, K_L^{\phi}\}$ and $\mathrm{Know}_L = \{K_L, K_L^{\phi}\}$.

However, before we implement our splitting algorithm on a knowledge base $\mathcal{K} \subseteq \mathcal{L}_{\mathrm{S}}^{|\sim}$, we reduce $\mathcal{K}$ into a canonical form. We first *normalize* each formula in $\mathcal{K}$ so that it only contains formulas in normal form. We then say that a DRSL knowledge base $\mathcal{K}$ is *conjunction-free* if there are no formulas of the form $\psi_1 \wedge \psi_2$ in $\mathcal{K}$. If $\mathcal{K}$ is not conjunction free, it is easy to observe that we can construct an equivalent knowledge base $\mathcal{K}'$ by splitting any conjunction in $\mathcal{K}$ into a set of its conjuncts. If $\mathcal{K}$ contains only normalized formulas and is conjunction free, we say that $\mathcal{K}$ is in *normal form*. That is, it only contains formulas of the form $\#_s\phi$, where ϕ is in $\mathcal{L}^{|\sim}$.

We are then able to define Algorithm 3, which splits a DRSL knowledge base into a set of propositional KLM knowledge bases. We then apply propositional rational closure point-wise on these sets for entailment-checking over a DRSL knowledge base. A last definition used in the algorithm is as follows.

Definition 10. *Let $\mathcal{K}$ be a DRSL knowledge base. For standpoint symbols s and t, we say that $s \preceq^{+} t$ if $t = s$, $t = *$ or there exists a finite sequence of standpoint symbols $t_1, ..., t_n$ such that $s \preceq t_1, t_1 \preceq t_2, ..., t_{n-1} \preceq t_n, t_n \preceq t \in \mathcal{K}$.*

After splitting a DRSL knowledge base into several propositional KLM knowledge bases, we apply another algorithm for checking whether a (non-standpoint sharpening) formula $\xi \in \mathcal{L}_{\mathrm{S}}^{|\sim}$ is in the rational closure of a DRSL knowledge base $\mathcal{K}$, and therefore whether $\mathcal{K}$ defeasibly entails ξ. In order to do this we presume the ability to call on Algorithm 2, for rational closure in the propositional case. We define this in Algorithm 4.

Intuitively, Algorithm 4 examines the modal operator occurring at the front of a DRSL formula ξ and based on this decides which KLM knowledge bases to consider when entailment-checking. If $\xi = \Box_s\psi$, then we check if ψ is in the rational closure of every KLM-propositional knowledge base which coincides with the standpoint s. That is, we check it is entailed by rational closure in every knowledge base in Know_s. If $\xi = \Diamond_s\psi$ then we check if ψ is entailed in *some* rational closure of a KLM knowledge base in Know_s. In the case where ξ has no modal operator we equivalently can check for $\Box_*\xi$. Using this, we are able to propose a reasonable definition of rational closure for DRSL, which can be seen as a natural extension of rational closure in the propositional case.

Definition 11. *We say that ϕ is in the **rational closure** of $\mathcal{K}$, or write $\mathcal{K} \mathrel{|\!\approx_{RC}} \phi$, if and only if* RCStandpoint$(\mathcal{K}, \phi) =$ True.

Note 1. Algorithm 4 is easily adaptable to other instances of algorithmically defined consequence operations used for propositional KLM. For example, lexicographic closure is another non-monotonic entailment operator introduced in the literature for the propositional case [20]. A more general algorithmic approach for defeasible entailments in the propositional KLM setting is introduced by Casini et al. [8], including the case for lexicographic closure. In such a case, the above algorithm for rational closure in standpoints could be extended to use other similar algorithms for defeasible entailment, by calling on the lexicographic closure algorithm instead of `RCProp`, for example. Hence, the use case for the above algorithm is not necessarily restricted to the case of rational closure.

Algorithm 3. StandpointSplit

Input: A defeasible standpoint knowledge base $\mathcal{K}$ in normal form.
Output: A set of propositional KLM knowledge bases associated to each standpoint in $\mathcal{K}$, denoted $\bigcup_{s\in S\cup\{*\}}\{\text{Know}_s\}$.

Define S as the set of standpoints occurring in the formulas of $\mathcal{K}$.
for $s \in S$ **do**
  $K_s := \emptyset$;
  while $t \in S$ and $s \preceq^+ t$ **do**
    for $\Box_t\phi \in \mathcal{K}$ **do**
      $K_s := K_s \cup \{\phi\}$;
    end for
  end while
end for
for $s \in S$ **do**
  for $\Diamond_s\phi \in \mathcal{K}$ **do**
    $K_s^{\phi} := K_s \cup \{\phi\}$;
  end for
end for
for $s \in S$ **do**
  $S_{\preceq s} := \{t \in S \mid t \preceq^+ s\}$;
  $\text{Know}_s := \cup_{t\in S_{\preceq s}}(\{K_t\} \cup \{K_t^{\phi} \mid \Diamond_t\phi \in \mathcal{K}\})$;
end for
$\text{Know}_* = \bigcup_{s\in S}\text{Know}_s - K_*$;
return $\bigcup_{s\in S\cup\{*\}}\{\text{Know}_s\}$;

Algorithm 4. RCStandpoint

Input: A defeasible standpoint knowledge base $\mathcal{K}$ in normal form and a (non-standpoint sharpening) defeasible standpoint formula ψ in normal form.
Output: True if ψ is in the rational closure of $\mathcal{K}$, otherwise False.

if $\psi = \phi_1 \wedge \phi_2$ **then**
  if RCStandpoint(ϕ_1) = True and RCStandpoint(ϕ_2) = True **then**
    return True;
  else
    return False;
  end if
else if $\psi = \Box_s\phi$ **then**
  for $K \in \text{Know}_s$ **do**
    if RCProp(K, ϕ)=False **then**
      return False;
    end if
  end for
  return True;
else if $\psi = \Diamond_s\phi$ **then**
  for $K \in \text{Know}_s$ **do**
    if RCProp(K, ϕ) = True **then**
      return True;
    end if
  end for
  return False;
end if

We can also show $\mid\!\approx_{RC}$ as defined above is indeed non-monotonic, as opposed to ranked entailment. Consider the example below.

Example 4. Consider the knowledge base, $\mathcal{K} = \{p \rightarrow b, b \mid\sim f, s \preceq *\}$. If we compute StandpointSplit($\mathcal{K}$), since all the non-subsumption items in our knowledge base are universal, we obtain just one set in our splitting $K_s = \mathcal{K} - \{s \preceq *\}$. Then, since there are no exceptional antecedents we are able to use classical reasoning to conclude $\mathcal{K} \mid\!\approx_{RC} \{p \mid\sim f, \Box_s(p \mid\sim f)\}$. However, if we consider the amended knowledge base, $\mathcal{K}' = \mathcal{K} \cup \{\Box_s(p \mid\sim \neg f), t \preceq *\}$ we are able to show that $\mathcal{K}' \not\mid\!\approx_{RC} \Box_s(p \mid\sim f)$ and $\mathcal{K}' \not\mid\!\approx_{RC} p \mid\sim f$. Hence, $\mid\!\approx_{RC}$ is non-monotonic. It is also interesting to note that since $p \mid\sim \neg f$ is only asserted for s and not for other standpoints, we still have $\mathcal{K}' \mid\!\approx_{RC} \Box_t(p \mid\sim f)$.

Note 2. In Algorithm 3, it can be seen that we intentionally exclude K_* as a knowledge base when we consider our rational closure. This is due to the fact that we do not

want to construct a "basic universal" standpoint as an individual precisification when we are querying over our logic. If we allow for K_* to be constructed we may have $\mathcal{K} \not\approx_{RC} \Box_* \phi$ simply because ϕ is not in the rational closure of K_*. On the other hand, we may conclude $\mathcal{K} \not\approx_{RC} \Diamond_* \phi$ simply because ϕ is in the rational closure of K_*. In both cases, this seems undesirable. In the first case, we may lose conclusions when we are reasoning about what finite agents agree upon, when these agreements are not the consequence of some universal specified rule. In the worst case, if $\mathcal{K}_* = \emptyset$, there is nothing non-trivial in the rational closure of K_* and we are unable to entail anything non-trivial of the form $\Box_* \phi$, even if every known standpoint $s \in S - \{*\}$ agrees upon ϕ. This would be the case in Example 4. In the second case, it feels erroneous to conclude $\Diamond_* \phi$ based on the fact that ϕ is in the rational closure of K_*, since intuitively $\Diamond_* \phi$ reads "there is a standpoint which can be interpreted as entailing ϕ". However, if the only such standpoint is $*$, then this standpoint does not correspond to any considered agents viewpoint, and so $\Diamond_* \phi$ does not seem a reasonable conclusion. One downside to the exclusion of K_* is that it means that `RCStandpoint` is not usable in the case where $\mathcal{K}$ only consists of formulas in the form $\Box_* \psi$. However, in this case $\mathcal{K}$ is equivalent to a propositional KLM knowledge base and so propositional rational closure can be used for entailment-checking. It is also possible to enforce the inclusion of a knowledge base equal to K_* by adding $\Diamond_* \top$ to $\mathcal{K}$. Since $\top$ adds only a trivial element to K_*, $K_*^{\top}$ is equivalent to K_*.

Having defined and motivated the previous algorithm as a natural extension to rational closure, we now consider the complexity of using `StandpointSplit` and `RCStandpoint` to check for entailment in DRSL.

Lemma 4. *`StandpointSplit` is computable in polynomial time.*

Theorem 1. *`RCStandpoint` is computable in P^{NP}, and if the materialization of each defeasible entailment in the knowledge base and queried formula is a Horn clause then `RCStandpoint` is computable in polynomial time.*

This theorem follows a general pattern in the standpoint logic literature, which shows in various cases that standpoint modal operators can be added, sometimes with additional restrictions, to a certain system of logic without increasing the complexity of the logic [16,18]. The result here comes from the fact that `RCProp` is in P^{NP}. Then, if our reduced DRSL knowledge base has length k, the `RCStandpoint` algorithm consists of computing `RCProp` at most k times over a knowledge base of at most k propositional KLM formulas.

4.2 Semantic Characterisation for Rational Closure

Besides the previous algorithm, we can equivalently characterize the rational closure for a knowledge base $\mathcal{K}$ using our previously defined semantics for DRSL. In particular, we can define a single ranked interpretation which is a model for a (non-standpoint sharpening) formula ξ if and only if $\mathcal{K} \not\approx_{RC} \xi$.

Definition 12. *Let $\mathcal{K}$ be a knowledge base in normal form, then define the interpretation $M_{RC}^{\mathcal{K}} = (\Pi, \sigma, \gamma)$ as follows:*

- $\Pi = \{\pi_s \mid s \in S - \{*\}\} \cup (\bigcup_{s \in S \cup \{*\}} \{\pi_s^\phi \mid \Diamond_s \phi \in \mathcal{K}$ *for some* $s \in S\})$.
- $\sigma(s) = \{\pi_t \mid t \preceq^+ s\} \cup \{\pi_t^\phi \mid t \preceq^+ s\}$ *for each* $s \in S$.
- *For each* π_s, *we have* $\gamma(\pi_s) = R_{RC}^{K_s}$ *and for each* π_s^ϕ, *we have* $\gamma(\pi_s^\phi) = R_{RC}^{K_s^\phi}$.

where K_s *and* K_s^ψ *are the sets as described in the standpoint splitting algorithm, and* R_{RC}^K *is the ranking function representing the rational closure of the (propositional) KLM knowledge base* K.

This is constructed with a similar motivation to `RCStandpoint`. Any standpoint $s \in \mathcal{S}$ must have some "base" precisification which is characterized by the KLM propositions hold in every precisification for s. This corresponds to the rational closure ranked interpretation for this set of propositions. Then, any proposition $\Diamond_s \phi$ characterizes a new precisification of s where ϕ is taken into account. It is worth noting that, as expected, $M_{RC}^{\mathcal{K}}$ is a well-defined model of $\mathcal{K}$.

Lemma 5. $M_{RC}^{\mathcal{K}} = (\Pi, \sigma, \gamma)$ *is a model of* $\mathcal{K}$.

Through construction, it is also clear that the algorithmic and semantic characterization of rational closure follow the same principles. That is, we split the standpoint knowledge base into various KLM-propositional knowledge bases, and connect these to a certain standpoint. Then, for each knowledge base, we perform a "pointwise" rational closure. In the algorithmic case, we call on the `RCProp` algorithm, and use the ranked interpretations for rational closure in the semantic case. From this, we are able to see that entailment checking under the rational closure model $M_{RC}^{\mathcal{K}}$ gives the same result as the algorithm `RCStandpoint`. This is formalized with the following theorem.

Theorem 2. *For all* $\psi \in \mathcal{L}_{\$}^{\sim}$ *and all* $\mathcal{K} \subseteq \mathcal{L}_{\$}^{\sim}$, *we have that* $\mathcal{K} \mid\!\approx_{RC} \psi$ *if and only if* $M_{RC}^{\mathcal{K}} \Vdash \psi$.

This gives us the main result of our paper. As in the propositional case for KLM, we have a non-monotonic entailment operation which can be equivalently characterized through an algorithm, and a single representative semantic model. We now consider applying the above to our original motivating example.

Example 5. Using the same knowledge base as Example 1 we use the splitting defined in Example 3 to construct our ranked interpretations at each precisification, where $\gamma(\pi_B)$, $\gamma(\pi_C)$ and $\gamma(\pi_L)$ are the ranked interpretations defined by the rational closure of K_B, K_C and K_L, respectively. Since K_B is a Boolean knowledge base, the rational closure contains every valuation satisfying K_B in row 0 and every other valuation in row ∞. For the other two knowledge bases we have the following ranked interpretations corresponding to rational closure. In the representation of these interpretations, we change each atomic proposition to the first letter of its name (e.g. $t = tomato$) and the propositions $sweet$ and $savory$ are represented by s_w and s_a respectively:

∞	all other valuations	all other valuations
1	$ts_a\overline{v}s_w f, t\overline{s_a v}s_w f, t\overline{s_a v s_w f}, \overline{t}s_a v s_w f$	$t\overline{s_a v}s_w f, t\overline{s_a v s_w f}$
0	$ts_a\overline{v s_w f}, \overline{t}s_a\overline{v s_w f}, \overline{ts_a v}s_w f, \overline{ts_a v s_w f}$	$ts_a\overline{v s_w f}, \overline{t}s_a\overline{v s_w f}, \overline{ts_a v}s_w f, \overline{ts_a v s_w f}$

Here, the middle column represents the ranking function of $\gamma(\pi_C)$ and the rightmost represents $\gamma(\pi_L)$. We are then able to use these models to check if DRSL statements are in the rational closure for $\mathcal{K}$. First, note that $\gamma(\pi_C) \Vdash tomato \mathrel{|\!\sim} vegetable$ and $\gamma(\pi_L) \Vdash tomato \mathrel{|\!\sim} vegetable$. Moreover, $\gamma(\pi_B) \Vdash tomato \rightarrow vegetable$, which is a stronger condition than $tomato \mathrel{|\!\sim} vegetable$. Thus, $M^{\mathcal{K}}_{RC}, \pi \Vdash tomato \mathrel{|\!\sim} vegetable$ for each $\pi \in \Pi$ and $\mathcal{K} \mathrel{|\!\approx}_{RC} tomato \mathrel{|\!\sim} vegetable$. We are also able to check for non-entailment. For example, $\mathcal{K} \not\mathrel{|\!\approx}_{RC} \Box_L(tomato \rightarrow \neg fruit)$ since the valuation $ts_a^{-}vs_w f$ has rank 1 under $\gamma(\pi_L)$. This seems reasonable intuitively, since it allows for the legal position to consider a tomato a fruit in an exceptional case, such as the possibility of a sweet "dessert tomato".

Note 3. It is worth noting here that we have been operating under the assumption that $S \cup \{*\}$, the set of named standpoints in the knowledge base, is equal to $\mathcal{S}$, the set of all standpoints expressible in our vocabulary. This is akin to a closed-world assumption with regards to standpoints in the given knowledge base. This may seem restrictive since we are unable to query for a standpoint which is not named in the knowledge base. However, the addition of standpoints which have no formulas in the knowledge face a similar problem to the addition of K_* as in Note 2. For example, if we added another standpoint X to Example 5 then we are required in our model to consider π_X. However, since there are no statements pertaining to X, $\gamma(\pi_X)(u) = 0$ for any $u \in \mathcal{U}$. Then we are no longer able to conclude $tomato \mathrel{|\!\sim} vegetable$[3] in our model simply due to the fact that it does not hold in $\gamma(\pi_X)$. Therefore, we keep this assumption, and note that in the case where we wish to consider an additional viewpoint X that does not already occur in the knowledge base, we can equivalently add the statement $X \preceq *$ into the knowledge base as in Example 4.

5 Related Work

An early notion of a logical semantics incorporating standpoints was introduced by Bennet [1], using a fairly different approach to those used in this paper. The syntax and semantics for standpoint modalities used in this paper was developed in the propositional case by Gómez Álvarez and Rudolph [15]. Similar notions of standpoints have since been introduced to first order logic [16], various description logics [16–18] and linear temporal logic [13]. The defeasible logic of KLM was originally introduced by Kraus et al. [19] with rational closure defined by Lehmann and Magidor [21]. Although defeasible implication has not been studied previously in the case for standpoints, similar notions have been previously considered. Britz et al. [3,4] consider introducing $\mathrel{|\!\sim}$ as a non-monotonic consequence relation between propositions in modal logics. Another aspect of defeasibility in modal logic are *defeasible modal operators*, which have been considered by Britz and Varzinczak [5] in the general case and Chafik et al. [10] in the case of linear temporal logic. Other non-modal extensions of KLM-style defeasibility and derived non-monotonic entailments have been considered in the description logics case [2,6,9], as well as for first order logic and some of its decidable fragments [7,11,22].

[3] In fact, we lose entailment for any non-trivial statements of the form $\Box_* \phi$.

6 Conclusion

The main focus of this paper was to integrate standpoint modal operators into KLM-style defeasible propositional logic. In order to do this we defined Defeasible Restricted Standpoint Logic (DRSL), by introducing a syntax which allows KLM propositional symbols to be bound by standpoint modal operators. We introduced the semantics of DRSL through ranked standpoint structures, combining the notion of ranked interpretations for propositional KLM [21], and the semantic approaches used to define other standpoint logics [15,16]. We then introduced a non-monotonic entailment relation for DRSL, which we proposed as a natural extension to rational closure in the propositional case [21], denoted $\mid\!\!\approx_{RC}$. This was first defined via a rational closure algorithm, which we showed provided a method for entailment-checking in DRSL that was of the same complexity class as the rational closure algorithm in the propositional case. Lastly, we showed that $\mid\!\!\approx_{RC}$ could be equivalently defined semantically, specifically constructing a single ranked standpoint structure $M^{\mathcal{K}}_{RC}$ such that $\mathcal{K} \mid\!\!\approx_{RC} \psi$ if and only if $M^{\mathcal{K}}_{RC} \Vdash \psi$.

For future work, the most obvious question would be to enquire whether a more expressive version of DRSL can be defined (for example including negation and disjunction of standpoint bound formulas), and what effects allowing for a more expressive logic would have on the complexity and semantic constructions linked to a non-monotonic entailment, such as rational closure. It seems initially that this case would be significantly more difficult to consider, since we lose the Single Model property when we introduce negation and disjunction into KLM in the propositional case [23]. Another notion to explore, although mentioned briefly in the paper, is the possibility of extending other non-monotonic consequence relations defined in the propositional case to the standpoint case, such as lexicographic closure [20]. It would also be desirable to investigate the proof-theoretic properties of $\mid\!\!\approx_{RC}$ in the standpoint logic case and compare those to the original KLM postulates given by Kraus et al. [19]. Another interesting problem to investigate would be the integration of standpoints into more expressive KLM-style defeasible logics, such as defeasible description logics (DDLs) [2]. Since standpoint have seen to be integratable in other description logics [16,18], it seems plausible that this would work in the case of DDLs.

Acknowledgments. Nicholas Leisegang wishes to thank the School of Embedded Composite Artificial Intelligence (SECAI) – project 57616814 funded by BMBF (the Bundesministerium für Bildung und Forschung) and DAAD (German Academic Exchange Service) – and the University of Cape Town's Artificial Intelligence Research Unit (AIRU), who facilitated the research visit to TU Dresden which made this collaboration possible. Nicholas Leisegang is funded by the UCT Science Faculty Fellowship.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bennett, B.: Standpoint semantics: a framework for formalising the variable meaning of vague terms. Understanding Vagueness–Logical, Philosophical and Linguistic Perspectives. College Publications pp. 261–278 (2011)

2. Britz, K., Casini, G., Meyer, T., Moodley, K., Sattler, U., Varzinczak, I.: Principles of KLM-style defeasible description logics. ACM Transactions on Computational Logic (TOCL) **22**(1), 1–46 (2020)
3. Britz, K., Meyer, T., Varzinczak, I.: Preferential reasoning for modal logics. Electronic Notes in Theoretical Computer Science **278**, 55–69 (2011), proceedings of the 7th Workshop on Methods for Modalities (M4M'2011) and the 4th Workshop on Logical Aspects of Multi-Agent Systems (LAMAS'2011)
4. Britz, K., Meyer, T., Varzinczak, I.: Normal modal preferential consequence. In: Proceedings of the 25th Australasian Joint Conference in Artificial Intelligence. pp. 505–516. Springer, Heidelberg, Germany (2012)
5. Britz, K., Varzinczak, I.: From KLM-style conditionals to defeasible modalities, and back. Journal of Applied Non-Classical Logics **28**(1), 92–121 (2018)
6. Casini, G., Meyer, T., Moodley, K., Nortjé, R.: Relevant closure: A new form of defeasible reasoning for description logics. In: Logics in Artificial Intelligence: 14th European Conference, JELIA 2014, Funchal, Madeira, Portugal, September 24-26, 2014. Proceedings 14. pp. 92–106. Springer (2014)
7. Casini, G., Meyer, T., Paterson-Jones, G., Varzinczak, I.: KLM-style defeasibility for restricted first-order logic. In: Proceedings of the 6th International Joint Conference on Rules and Reasoning. pp. 81–94. Springer, Cham, Switzerland (September 2022)
8. Casini, G., Meyer, T., Varzinczak, I.: Taking defeasible entailment beyond rational closure. In: Calimeri, F., Leone, N., Manna, M. (eds.) Logics in Artificial Intelligence, pp. 182–197. Springer International Publishing, Cham (2019)
9. Casini, G., Straccia, U.: Lexicographic closure for defeasible description logics. In: Proceeding of the Australasian Ontology Workshop. vol. 969, pp. 28–39 (2012)
10. Chafik, A., Cheikh-Alili, F., Condottaa, J.F., Varzinczak, I.: Defeasible linear temporal logic. Journal of Applied Non-Classical Logics **33**(1), 1–51 (2023)
11. Delgrande, J., Rantsoudis, C.: A preference-based approach for representing defaults in first-order logic. In: Proceedings of the 18th International Workshop on Non-Monotonic Reasoning, NMR (2020)
12. Freund, M.: Preferential reasoning in the perspective of Poole default logic. Artificial Intelligence **98**(1), 209–235 (1998)
13. Gigante, N., Gómez Álvarez, L., Lyon, T.S.: Standpoint linear temporal logic. In: Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning, KR 2023, Rhodes, Greece, September 2-8, 2023. pp. 311–321 (2023)
14. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.: Semantic characterization of rational closure: From propositional logic to description logics. Artificial Intelligence **226**, 1–33 (2015)
15. Gómez Álvarez, L., Rudolph, S.: Standpoint logic: Multi-perspective knowledge representation. In: Proceedings of the 12th International Conference (FOIS 2021) Frontiers in Artificial Intelligence and Applications. Frontiers in Artificial Intelligence and Applications, vol. 3344, pp. 3 – 17. IOS Press (2021)
16. Gómez Álvarez, L., Rudolph, S., Strass, H.: How to agree to disagree: Managing ontological perspectives using standpoint logic. In: Proceedings of the 21st International Semantic Web Conference (ISWC 22). Lecture Notes in Computer Science, vol. 13489. Springer (October 2022)
17. Gómez Álvarez, L., Rudolph, S., Strass, H.: Pushing the boundaries of tractable multiperspective reasoning: A deduction calculus for standpoint EL+. In: Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning, KR 2023, Rhodes, Greece, September 2-8, 2023. pp. 333–343 (2023)

18. Gómez Álvarez, L., Rudolph, S., Strass, H.: Tractable diversity: Scalable multiperspective ontology management via standpoint EL. In: Elkind, E. (ed.) Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23. pp. 3258–3267. International Joint Conferences on Artificial Intelligence Organization (2023)
19. Kraus, S., Lehmann, D., Magidor, M.: Nonmonotonic reasoning, preferential models and cumulative logics. Artificial Intelligence **44**(1–2), 167–207 (1990)
20. Lehmann, D.: Another perspective on default reasoning. Annals of Mathematics and Artificial Intelligence **15**, 61–82 (1995)
21. Lehmann, D., Magidor, M.: What does a conditional knowledge base entail? Artificial Intelligence **55**(1), 1–60 (1992)
22. Morris, M., Ross, T., Meyer, T.: Defeasible disjunctive datalog. In: Proc. of the South African Forum for Artificial Intelligence Research FAIR. pp. 208–219. Cape Town, South Africa (December 2019)
23. Paterson-Jones, G., Casini, G., Meyer, T.: A Boolean extension of KLM-style conditional reasoning. In: Gerber, A. (ed.) Artificial Intelligence Research, pp. 236–252. Springer International Publishing, Cham (2020)
24. Pearl, J.: System Z: a natural ordering of defaults with tractable applications to nonmonotonic reasoning. In: Proceedings of the 3rd Conference on Theoretical Aspects of Reasoning about Knowledge. p. 121-135. TARK '90, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1990)



Non-monotonic Extensions to Formal Concept Analysis via Object Preferences

Lucas Carr[1(✉)], Nicholas Leisegang[1], Thomas Meyer[1], and Sebastian Rudolph[2,3]

[1] University of Cape Town and CAIR, Cape Town, South Africa
{crrluc003,lsgnic001}@myuct.ac.za, tommie.meyer@uct.ac.za
[2] Technische Universität Dresden, Dresden, Germany
sebastian.rudolph@tu-dresden.de
[3] Center for Scalable Data Analytics and Artificial Intelligence Dresden/Leipzig, Dresden/Leipzig, Germany

Abstract. Formal Concept Analysis (FCA) is an approach to creating a conceptual hierarchy in which a *concept lattice* is generated from a *formal context*. That is, a triple consisting of a set of objects, G, a set of attributes, M, and an incidence relation I on $G \times M$. A *concept* is then modelled as a pair consisting of a set of objects (the *extent*), and a set of shared attributes (the *intent*). Implications in FCA describe how one set of attributes follows from another. The semantics of these implications closely resemble that of logical consequence in classical logic. In that sense, it describes a monotonic conditional. The contributions of this paper are two-fold. First, we introduce a non-monotonic conditional between sets of attributes, which assumes a preference over the set of objects. We show that this conditional gives rise to a consequence relation that is consistent with the postulates for non-monotonicty proposed by Kraus, Lehmann, and Magidor (commonly referred to as the KLM postulates). We argue that our contribution establishes a strong characterisation of non-monotonicity in FCA. To our knowledge, this is a novel view of FCA as a formalism which supports non-monotonic reasoning. We then extend the influence of KLM in FCA by introducing the notion of *typical* concepts through a restriction placed on what constitutes an acceptable preference over the objects. Typical concepts represent concepts where the intent aligns with expectations from the extent, allowing for an exception-tolerant view of concepts. To this end, we show that the set of all typical concepts is a meet semi-lattice of the original concept lattice. This notion of typical concepts is a further introduction of KLM-style typicality into FCA, and is foundational towards developing an algebraic structure representing a concept lattice of prototypical concepts.

Keywords: Formal Concept Analysis · Non-monotonic reasoning · Rational consequence relations · Defeasible reasoning

1 Introduction

Formal Concept Analysis (FCA) is a lattice-theoretic approach to representing and reasoning about concepts and hierarchies in data. The view of concepts adopted in FCA has clear philosophical underpinnings, describing a concept as a pair representing the

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 476–492, 2025.
https://doi.org/10.1007/978-3-031-78255-8_28

dualism between extension, *what a concept refers to*, and intension, *what a concept means* [8,9,19]. As such, the setting of FCA works with data in the form of a formal context, describing a relationship between objects and attributes.

An important component of FCA involves discovering implications in the data which describe a complete correspondence between sets of attributes [19]. A concern might be that complete correspondence is too strict a requirement, and implications may fail to capture relationships that, while useful, are only partial correspondences between attribute sets. To illustrate this point, we introduce a variation of the customary example of penguins, where we look at species of animals and some of their attributes (see Example 1 for a more detailed description) [10,13]. In our example we have three objects which have the attribute `bird`: duck, robin, and penguin. Of these, duck and robin also have the attribute `flies`, and obviously penguin does not. Consequently, the implication $\texttt{bird} \rightarrow \texttt{flies}$ is not valid in our context. One may argue that this behaviour is completely correct. Clearly, not all birds fly. A compelling response is that birds that do not fly are exceptions to the rule, and that the ability to express that birds usually do fly is useful.

Association rules offer an existing approach to address this concern. These rules introduce a notion of confidence, which quantifies the proportion of data that conforms to a given rule, if the confidence is above a user-defined threshold are then accepted [8]. While association rules are capable of capturing partial relationships, this approach may be considered somewhat blunt, relying on something analogous to *majority rules*. The reliance on a threshold for rule acceptance limits the expressivity of these rules.

We propose an alternate approach which adopts the view that we can define a preference relation over the objects in our data. While we are – at present – agnostic about the origin of this preference relation, from an applied perspective it might represent an external sentiment that one object is more *typical* than another. Partial implications between attribute sets can then be captured through the idea that the correspondence holds for *preferred* objects [10,13].

Once we have a preference relation over the set of objects, and if we regard the relation as representative of the typicality of objects, we can introduce a notion of a *typical concept*. The motivation for typical concepts, and what they mean, rests on the notion that, given some set of attributes, we want a coherent way to derive a concept which has the attributes we *typically* expect. Continuing with the example from before, we might expect that, while the intent of the "bird concept" derived from the attribute `bird` does not include the attribute `flies`, it may be useful to have a notion of a typical concept which includes the attributes of prototypical birds.

The remainder of the paper is structured as follows: Section 2 provides an account of some basic notions in FCA. Then, Section 3 provides an introduction to classical notions of consequence, non-monotonic reasoning, and an intuition for how these translate to FCA. We introduce the announced extensions to FCA in Section 4 and discuss the semantics and properties of the consequence relation they allow. In Section 5 we develop the notion of typical concepts, derived from the preferential view of objects. In Section 6 we discuss some attempts to bring KLM-style defeasibility into other logical systems. We also discuss existing work within FCA which proposes to introduce more expressive notions of concepts.

2 Formal Concept Analysis

Two fundamental notions in FCA are *formal context* and *formal concept*. A formal context has a set-theoretic definition, where three sets of objects, attributes, and a binary relation are defined. However, in reasonably sized instances it can be represented as a cross-table, where rows represent the objects, and columns are the attributes. Naturally, the presence of an '×' at a position in the table indicates that the respective object has the attribute from the respective column.

Definition 1. *A* formal context *is a triple* (G, M, I) *where* G *refers to a finite, non-empty set of objects,* M *to a finite, non-empty set of attributes, and* $I \subseteq G \times M$ *is an incidence relation. For* $(g, m) \in I$*, we say object* g *has attribute* m*, otherwise* g *does not have attribute* m*.*

Example 1. Figure 1 shows a cross-table representation of a formal context, describing different animal species and some of their attributes.

	northern	southern	flies	antarctic	bird
orca	×	×		×	
duck	×	×	×		×
robin	×	×	×		×
penguin		×		×	×

Fig. 1. A context of animal species and some of their attributes. For example, Orcas are found in the northern hemisphere, southern hemisphere, and the antarctic.

In the build-up to defining formal concepts, one introduces two operators which define an order-reversing Galois connection between the power sets $\mathcal{P}(G)$ and $\mathcal{P}(M)$. These two operators, which share the same notation, describe the derivation from a set of objects to the shared attributes, and vice versa [8,9].

Definition 2. *In a formal context,* $\mathbb{K} = (G, M, I)$*,* the derivation operator $(\cdot)'$ *is defined for sets* $A \subseteq G$ *and* $B \subseteq M$ *as:*

$$A' := \{m \in M \mid \forall g \in A, (g, m) \in I\}$$
$$B' := \{g \in G \mid \forall m \in B, (g, m) \in I\}$$

For a set $A \subseteq G$ *of objects (resp. attributes),* A' *is just a set of attributes (resp. objects), and so* A'' *would be a set of objects (resp. attributes). The double application of derivation operators is in fact a closure operator, meaning it is* extensive, idempotent, and monotonic.

Concepts are represented as a pairs of sets of objects and attributes; as one might expect, these sets are not arbitrary. In fact, they completely determine one another. Conventionally, we might denote a concept as (A, B) with $A \subseteq G, B \subseteq M$. This may be a useful notation to express the demarcation between the extent and intent, but it is not strictly necessary, for, (A, A') and (B', B) would refer to the same concept.

Definition 3. *Let $\mathbb{K} = (G, M, I)$ be a formal context. A pair (A, B), where $A \subseteq G$ and $B \subseteq M$, is a* formal concept *iff $A' = B$ and $B' = A$.*

In fact, given an arbitrary set of objects, $A \subseteq G$, one can entirely describe a concept by (A'', A') – the same principle holds for sets of attributes [9]. If we refer back to Example 1, starting from two sets of objects and attributes, respectively, $\{\texttt{duck, penguin}\}$ and $\{\texttt{antarctic}\}$, we find the following concepts:

$$(\{\texttt{duck,penguin}\}'', \{\texttt{duck,penguin}\}') = (\{\texttt{duck,robin,penguin}\}, \{\texttt{southern,bird}\})$$
$$(\{\texttt{antarctic}\}', \{\texttt{antarctic}\}'') = (\{\texttt{penguin,orca}\}, \{\texttt{southern,antarctic}\})$$

These two concepts suggest that one can introduce a partial ordering $\leq$ to concepts (induced by the subset relationship on their extents), establishing the notion of sub- and super-concepts. Equipped with this ordering, the set of all concepts of a formal context $\mathbb{K}$, denoted by $\mathfrak{B}(\mathbb{K})$, forms a complete lattice $\underline{\mathfrak{B}}(\mathbb{K}) := (\mathfrak{B}(\mathbb{K}), \leq)$, referred to as a concept lattice [8,9]. Figure 4 provides an example of a concept lattice. Attributes connected to a concept from above (inclusive) are part of that concept's intent, while objects connected from below (inclusive) are in the extent. For a set of concepts, $\{(A_t, B_t) \mid t \in T\} \subseteq \mathfrak{B}(\mathbb{K})$, the supremum (*smallest super-concept*), and infimum (*greatest sub-concept*) are given by

$$\bigvee_{t\in T}(A_t, B_t) := \big((\bigcup_{t\in T} A_t)'', \bigcap_{t\in T} B_t\big) \text{ as well as}$$
$$\bigwedge_{t\in T}(A_t, B_t) := \big(\bigcap_{t\in T} A_t, (\bigcup_{t\in T} B_t)''\big).$$

2.1 Implications

A significant aspect of FCA involves implications between sets of attributes. These express the notion that certain attributes indicate the presence of others, in all objects of a formal context [8,19].

Definition 4. *Let $\mathbb{K} = (G, M, I)$ be a formal context. An* implication over M *is an expression of the form $A \to B$ with $A, B \subseteq M$. We say that the implication is respected by another set $C \subseteq M$, iff $A \not\subseteq C$ or $B \subseteq C$. $\mathbb{K}$ respects the implication, written $\mathbb{K} \models A \to B$, iff for every $g \in G$, g' respects $A \to B$.*

It is useful to note that a formal context $\mathbb{K} = (G, M, I)$ respects an implication $A \to B$ over M exactly if $B \subseteq A''$, which, in turn, is equivalent to $A' \subseteq B'$.

3 Logical Consequence and Nonmonotonic Reasoning

3.1 Logical Consequence

A classical notion of logical consequence describes the circumstances under which one sentence is said to follow (logically) from another [6]. In a model-theoretic view, it is given as follows.

Definition 5. *For two sentences α, β in the language $\mathcal{L}$, we say that β is a* logical consequence *of α, expressed as $\alpha \models \beta$, iff for every valuation $u \in \mathcal{U}$ where $u \Vdash \alpha$ then also $u \Vdash \beta$.*

This definition can easily be generalised to the notion of *entailment*: given a set of facts it would be useful to know *what else* we can know, or, what is entailed [10].

Definition 6. *Given a set of sentences $\mathcal{KB}$, another sentence α is* entailed *by $\mathcal{KB}$ iff for every valuation $u \in \mathcal{U}$ with $u \Vdash \mathcal{KB}$ also $u \Vdash \alpha$ holds. This is expressed as $\mathcal{KB} \models \alpha$.*

A consequence operator, $\mathcal{C}n$, provides a general way to derive all sentences that should follow from a set of sentences $\mathcal{KB}$, under *some* notion of logical consequence [10]. Using classical entailment from Definition 6, we could have $\mathcal{C}n(\mathcal{KB}) := \{\alpha \mid \mathcal{KB} \models \alpha\}$ [5,10]. A Tarskian consequence operator satisfies the following properties (which describe a closure operator).

$$\text{Monotonocity:} \quad \textit{if } \Gamma \subseteq \Gamma' \text{ then } \mathcal{C}n(\Gamma) \subseteq \mathcal{C}n(\Gamma') \tag{1}$$

$$\text{Idempotence:} \quad \mathcal{C}n(\Gamma) = \mathcal{C}n(\mathcal{C}n(\Gamma)) \tag{2}$$

$$\text{Inclusion:} \quad \Gamma \subseteq \mathcal{C}n(\Gamma) \tag{3}$$

3.2 Consequence Relations

A more abstract notion of consequence is a *consequence relation*. This is a set of pairs, $\{(\Gamma_1, \gamma_1), \ldots, (\Gamma_n, \gamma_n), \ldots\}$, where it is typical to allow Γ_i to represent a set of formulae, and γ_i to represent a single formula in the language. The inclusion of a pair (Γ_i, γ_i) in the consequence relation-denoted as $\Gamma_i \vdash \gamma_i$-means that γ_i can be inferred from Γ_i [7,10].

Consequence relations may be characterised by the properties they satisfy and, as such, correspond to a certain kind of reasoning—an example of this is found in Definition 7 [7]. Conversely, an algorithmic description of a kind of reasoning may in turn give rise to a consequence relation, the properties of which provide strong intuition for the implicit pattern of reasoning.

3.3 Non-monotonic Reasoning

Non-monotonic reasoning is concerned with developing formal reasoning process in which a conclusion drawn under a premise can be withdrawn under the addition of another premise [13]. The justifications for why we may want to reason non-monotonically are easy to accept – it is quite obvious that when humans reason we do so under the implicit assumption that, upon receiving new information we can change our mind. Without this assumption, navigating life would be very difficult; we would hesitate to come to any conclusion out of fear it may be the wrong one. Moreover, we frequently make statements for which we know exist exceptions, under the assumption that upon encountering an exception it would be treated as such [16].

In the setting of propositional logic, the problem is often introduced in the following way [10,13,21]: we want to accept that "penguins are birds", "birds usually fly" and "penguins do not fly". However, when we translate this into classical logic with the

propositions $\texttt{penguin} \rightarrow \texttt{bird}$, $\texttt{bird} \rightarrow \texttt{fly}$ and $\texttt{penguin} \rightarrow \neg\texttt{fly}$, we are forced to conclude $\neg\texttt{penguin}$. With respect to the attribute logic of FCA, we find a similar issue. Consider the context in Example 1. We do not have $\{\texttt{bird}\} \rightarrow \{\texttt{flies}\}$ as a valid implication, since penguins are an object with `bird` but not `flies`. This illustrates the final point made in the last paragraph – that monotonicity prevents us from making statements with known exceptions.

In the realm of concepts, we encounter another instance of this problem: the concept determined by `bird` is not a sub-concept of the concept determined by `flies`. Again, we lack the expressivity for the idea that *typical* birds are flying animals. In Section 4 and Section 5 we propose a solution to this lack of expressivity drawing inspiration from [12,13].

3.4 Rational Consequence Relations

The style of non-monotonic reasoning we aim to develop in FCA is that of rational consequence relations. A rational consequence relation, $\mid\!\sim_R$, is based on preferential logics, where an order (or ranking) over valuations conveys the notion that certain valuations are preferred to others [12,14,20]. The semantics describe a notion of consequence where, if α, β are formulae in the language then β is a rational consequence of α, $\alpha \mid\!\sim_R \beta$, iff β is true in all the *most preferred* models of α.

The KLM postulates, introduced in [12,14], with the inclusion of *Rational Monotonicity* are a set of inference rules that characterise a rational consequence relation.

Definition 7. *A consequence relation, $\mid\!\sim$, constitutes a rational consequence relation iff it satisfies Reflexivity, Left Logical Equivalence (LLE), Right Weakening (RW), Cut, And, Or, Cautious Monotonicity and Rational Monotonicity.*

Reflexivity $A \mid\!\sim A$ Reflexivity is a somewhat basic notion of any notion of consequence; it essentially prevents a self-defeating pattern of reasoning where given something, you conclude as a consequence not that thing.

Left Logical Equivalence (LLE) $\frac{\models A \equiv B, A \mid\!\sim C}{B \mid\!\sim C}$ LLE enforces the notion that two things that are equivalent (under a coherent notion of equivalence) should have the exact same consequences.

Right Weakening (RW) $\frac{\models A \rightarrow B, C \mid\!\sim A}{C \mid\!\sim B}$ RW states that the consequence of a classical implication, which has a defeasible consequence as a premise, can itself be derived from the original defeasible implication.

Cut $\frac{A \wedge B \mid\!\sim C,\ A \mid\!\sim B}{A \mid\!\sim C}$ Cut allows us to use existing defeasible consequences in the premise of a new defeasible implication. However, the new implication is subject to failure should the defeasible conclusion in its premise be retracted.

Or $\frac{A \mid\!\sim C, B \mid\!\sim C}{A \vee B \mid\!\sim C}$ Or is that given distinct premises with a common defeasible consequence, we should be able to draw this conclusion from the disjunction of the premises – that is, we need not explicitly know which one is true.

And $\frac{A|\sim B, A|\sim C}{A|\sim B \wedge C}$ And tells us that two consequences can be concluded at the same time.

Cautious Monotonicity (CM) $\frac{A|\sim B, A|\sim C}{A \wedge B|\sim C}$ CM ensures that adding a defeasible consequence that could already have been derived from our premises should never invalidate another defeasible consequence that could be derived from our original premises.

Rational Monotonicity (RM) $\frac{A|\sim B,\ A|\not\sim \neg C}{A \wedge C|\sim B}$ RM expresses the notion, similar to conflicts in Default Logic, that only when a premise that was expected to be false is added, should we retract a rational consequence. In another sense, we can assume that when new information that does not explicitly contradict existing knowledge, we can retain existing conclusions [10,14].

4 Introducing Nonmonotonicity in FCA

4.1 Extended Formal Context

As a precursor to the non-monotonic conditionals in Subsection 4.3, we introduce the notion of an extended formal context.

Definition 8. *Let* $\mathbb{K} = (G, M, I)$ *be a formal context. An* extended formal context *simply adds a partial order over* G*. We denote this as a quadruple,* $\mathbb{K}^{\preceq} = (G, M, I, \preceq)$.

The partial ordering is intended to convey a preference relation between objects. That is, for two objects, $g, h \in G$, if $g \preceq h$, then we say that g is more preferred, or more typical, in comparison to h. For example, `robin` $\preceq$ `penguin` tells us a robin is "more typical" than a penguin in our context.

4.2 Minimisation

The preference relation between objects provided by an extended formal context enables us to define how the derivation operators might behave when restricted to only care about minimal objects. We formalise this notion as a *minimised derivation*.

Definition 9. *Let* $\mathbb{K}^{\preceq} = (G, M, I, \preceq)$ *be an extended formal context, and let* $B \subseteq M$*. Then the* minimised derivation*,* $\underline{B}'$*, of* B*, is the set*

$$\underline{B}' := \{g \in B' \mid \nexists h \in B' \text{ such that } h \prec g\}$$

As a reminder, for a set of attributes, $B \subseteq M$, B' provides us with all the objects which *have* all attributes of B. Following this, B'' extends B by including all other attributes which are shared by those objects which share attributes B.

If $\underline{B}'$ describes the process of going from a set of attributes to the minimal objects which share B, a reasonable question might concern the procedure for returning to a "closed" set of attributes: presumably, the set of attributes common to those objects described by $\underline{B}'$. This is, in fact, just the composition of Definition 9 and Definition 2, taking the form $(\underline{B}')'$ - henceforth referred to as a *minimised-return operation*.

Theorem 1. *Let $\mathbb{K}^{\preceq} = (G, M, I, \preceq)$ be an extended formal context, the minimised-return operator, $(\underline{\cdot}')'$, applied to a set of attributes is nonmonotonic, extensive, and idempotent.*

Proof. If the minimised-return operator were monotonic, for the sets $A, B \subseteq M$, $A \subseteq B$ would imply $(\underline{A}')' \subseteq (\underline{B}')'$. Refer back to the formal context in Example 1 extended with the partial order `robin` $\preceq$ `penguin`. Let $A := \{$`bird`$\}$ and $B := \{$`bird, antarctic`$\}$. Clearly, $A \subseteq B$, however, application of the minimised-return operator gives $(\underline{A}')' = \{$`bird, flies, southern, northern`$\}$ and $(\underline{B}')' = \{$`bird, antarctic, southern`$\}$, and so $(\underline{A}')' \not\subseteq (\underline{B}')'$.

To show that the minimised-return operator is extensive, it should hold that for a set $A \subseteq M$, $A \subseteq (\underline{A}')'$. We already know that in the classical case, $A \subseteq A''$. So, for all objects $g \in A'$, it is the case that $A \subseteq g'$. It is also clear that the minimised derivation of a set of attributes is a subset of the classical derivation – that is, $\underline{A}' \subseteq A'$. This means that for all objects $h \in \underline{A}'$, $A \subseteq h'$. It should then be straightforward to see that A is a subset of the set given by $\bigcap\{g' \mid g \in \underline{A}'\}$, which is equivalent to $A \subseteq (\underline{A}')'$.

The final task is to show that the minimised-return operator is idempotent. Given a set $A \subseteq M$, let $C := (\underline{A}')'$. It is enough to show that $\underline{A}' = \underline{C}'$. C is the set of attributes such that all objects, $g \in \underline{A}'$, have C in their intent. It is also clear, since all objects in $\underline{A}'$ have A in their intent, that $A \subseteq C$. Then, for $h \in \underline{C}'$ it follows that h has attributes C, and thus A, in its intent. Since h is minimal w.r.t. C, and $g \in C'$, it cannot be the case that $g \preceq h$. Conversely, since g is minimal w.r.t. A, and $h \in A'$, it cannot be that $h \preceq g$. Then, since both g and h are elements of A' and C', and are arbitrary, the sets $\underline{A}'$ and $\underline{C}'$ refer to the same objects. Then, applying a derivation to the same sets would of course yield the same result, and so the minimised-return operator is idempotent. □

4.3 Non-monotonic Conditional Through Minimisation

From the discussion in Subsect. 2.1 it is clear that the semantics of classical implications in FCA can be defined through applications of closure operators on sets of attributes. With the introduction of the minimised-return operator, one might expect to find a different notion of consequence.

The minimised-return operator still refers to the same kind of process as the closure-operator, going from a set of attributes to a set of objects, and then back to a set of attributes. The similarity enable us to define a semantics which is only a minor departure from the original implication. For now we will use *non-monotonic conditional* to refer to the minimised-derivation based implications, and denote this by $\rightsquigarrow$.

Definition 10. *Let $\mathbb{K}^{\preceq} = (G, M, I, \preceq)$ be an extended formal context, with $A, B \subseteq M$. $\mathbb{K}^{\preceq}$ respects a* non-monotonic conditional *$\mathbb{K}^{\preceq} \models A \rightsquigarrow B$ iff $B \subseteq (\underline{A}')'$ which is equivalent to $\underline{A}' \subseteq B'$.*

Although we have yet to prove the characterisation, the non-monotonic conditional from above describes a notion of consequence with strong semblance to the preferential consequence relations discussed in Subsect. 3.4.

	warrior	demigod	hero
Jason	×		×
Achilles	×	×	×
Minos		×	

Fig. 2. Extended formal context of figures in Greek mythology

Example 2. Let the cross table below represent an extended formal context, where the partial order over objects is: $\texttt{Achilles} \preceq \texttt{Jason, Minos}$.

In Greek mythology there is a strong correspondence between figures who are heroes, and demigods. So much so that there is debate as to whether *hero* and *demigod* actually have distinct meanings. Thus, with some creative licence, we may wish to express the notion that `hero` *usually implies* `demigod`. The classical implication, of course, fails to express this - as `Jason` is a counter-example. Instead, we express this notion as the non-monotonic conditional $\texttt{hero} \rightsquigarrow \texttt{demigod}$.

In our definition of classical implications, see Definition 4, we first introduced $\rightarrow$ as an object-level operator, saying that a set $C \subseteq M$ respects an implication $A \rightarrow B$ iff $A \not\subseteq C$ or $B \subseteq C$ (we can think of C as an object intent). However, most of the time when we talk about implications in FCA, we really refer to the meta level notion analogous to entailment that $K \models A \rightarrow B$.

For non-monotonic conditionals it makes even less sense to speak about the object level, we are interested in what our ordering allows us to conclude – and the ordering is explicitly on the meta-level. Consequently, whenever we speak about a non-monotonic conditional, we are speaking about the implication over the entire formal context.

4.4 Non-monotonic Conditionals and the KLM Postulates

We now present an argument that the consequence relation given by the non-monotonic conditional described in the previous sub-section satisfies all the KLM postulates required to be a characterisation of rational consequence relations. We should remind ourselves that the postulates are usually described in a truth-theoretic language, such as propositional logic. A consequence being that some initial work needs to be translated into the attribute-logic formalism considered here.

For *Reflexivity* and *RW*, the translation is obvious and doesn't require additional intuition. All that is required for *LLE* is to describe what a notion of equivalence between sets of attributes means. To this end, we say that in a formal context two attribute sets are equivalent, $A \equiv B$, iff $A' = B'$. *Cut* says that if $\alpha \wedge \beta \mid\!\sim \gamma$ and $\alpha \mid\!\sim \beta$ then $\alpha \mid\!\sim \gamma$. The conjunction of two formulae in a truth-theoretic logic requires satisfaction of each formula. In attribute logic, the equivalent notion is given by the union of two attribute sets, satisfied by objects with *all* attributes from both sets. The conjunction in *CM* and *And* is re-phrased in the same way.

It is not immediately clear what *RM* might mean, as it is uncommon to talk about implications with a negation. We use the following definitions to make this notion explicit.

Definition 11. *Let $\mathbb{K} = (G, M, I)$ be a formal context and $A \rightarrow \neg B$ an implication over M. We say that a set $C \subseteq M$ respects the implication $A \rightarrow \neg B$ iff $A \not\subseteq C$ or $B \not\subseteq C$. $\mathbb{K}$ respects the implication, $\mathbb{K} \models A \rightarrow \neg B$, iff for every object g, g' respects $A \rightarrow \neg B$. Equivalently, $A' \cap B' = \emptyset$.*

Definition 12. *An extended formal context, $\mathbb{K}^{\preceq}$, respects the non-monotonic conditional $A \rightsquigarrow \neg B$ iff, for every object g in $\underline{A}'$, $B \not\subseteq g'$. Equivalently, $\mathbb{K}^{\preceq} \models A \rightsquigarrow \neg B$ iff $\underline{A}' \cap B' = \emptyset$.*

The intuition is that one set of attributes corresponds to the absence of another if there are no objects which have both sets in their intent. On the object level of a formal context, for an object $g \in G$, if $g \not\models A \rightarrow B$ then $g \models A \rightarrow \neg B$. We do not, however, have that if $\mathbb{K} \not\models A \rightarrow B$ then $\mathbb{K} \models A \rightarrow \neg B$. Translating the intuition of $A \rightarrow \neg B$ to $A \rightsquigarrow \neg B$ is straightforward. The non-monotonic conditional, $A \rightsquigarrow \neg B$, concerns only objects $g \in \underline{A}'$, which, by definition of the minimised derivation operator, must have A in their intent. Consequently, we need only show that g does not have B in its intent.

We have yet to address the *Or* postulate. The difficulty here is that attribute logic does not have (nor is it intuitive to introduce) a notion of disjunction [15]. As such, we recognise a departure from the KLM framework, and omit the *Or* postulate entirely. Consequence relations that satisfy only *Reflexivity, LLE, RW, Cut*, and *CM* are considered *cumulative consequence relations* [13]. However, while we do not have*Or*, we show that we still have *Rational Monotonicity*, and continue to regard our relation as a characterisation of *rational consequence**. [1]

Theorem 2. *The consequence relation derived from $\rightsquigarrow$ satisfies Reflexivity, Left Logical Equivalence, Right Weakening, Cut, And, Cautious Monotonicity, and Rational Monotonicity, and thus, constitutes a rational consequence relation*.* [2]

This gives us a notion of non-monotonic consequence that corresponds to what we may think *logical* non-monotonic reasoning should look like, as described by rational consequence. We are able to use intermediary results as the basis for more conclusions through the use of *Cut* and *CM* [10,21]. *RM* is a useful property in the sense that it reduces the work one has to do - unless we have some consequence that is contradictory to what we expect, we can retain existing conclusions. The notion of consequence also preserves classical implications in FCA, and so it is strictly more expressive than regular implications in FCA [21].

Now is a good place to remind ourselves of what we gain from introducing non-monotonic conditionals to FCA. If we accept rational consequence relations as a good account of non-monotonic reasoning, and that we can impose a preference relation on objects in a formal context, non-monotonic conditionals provide FCA with the expressivity to reason about what may be the *typical* case.

[1] The absence of the *Or* postulate is indicated by an asterisk.

[2] The proofs for the non-monotonic conditional satisfying the KLM-postulates* can be found in the Arxiv submission with a matching title.

5 Typical Concepts

The partial ordering over objects was introduced as a means to developing a non-monotonic consequence relation. We now show how this ordering, under some constraint, gives rise to a notion of typical concepts.

5.1 Naive Typical Concepts

Definition 13. *Let $\mathbb{K}^{\preceq} = (G, M, I, \preceq)$ be an extended formal context. Then, for a set $A \subseteq M$, we define a* typical concept *as a concept of the form*

$$\left(\left(\underline{A}'\right)'', \left(\underline{A}'\right)' \right).$$

The set of all typical concepts of an extended formal context is denoted $\mathfrak{T}(\mathbb{K}^{\preceq})$.

In this case, the intent of a typical concept is the set of attributes common to the minimal objects which have A in their intent. The extent is then all objects which have this extended set of attributes. Referring back to Example 1, we add the partial order $\texttt{duck} \preceq \texttt{penguin}$ and $\texttt{robin} \preceq \texttt{penguin}$, expressing the sentiment that ducks and robins are more *typical* than penguins, but incomparable to orcas. Given only the attribute $\texttt{bird}$, $(\{\texttt{duck}, \texttt{robin}\}, \{\texttt{northern}, \texttt{southern}, \texttt{flies}, \texttt{bird}\})$ is the derived typical concept. Without this notion of a typical concept, the derived concept would be $(\{\texttt{duck}, \texttt{robin}, \texttt{penguin}\}, \{\texttt{southern}, \texttt{bird}\})$. Of course, both of these concepts exist in the classical concept lattice; however, typical concepts provide an instrument to, given only the condition of $\texttt{bird}$, to arrive at a concept which we might consider a more natural characterisation of things that are birds. We should remind ourselves that, while this example is picked to match real-world expectations, we should think of "expectations" as being an expression of the preference relation on objects.

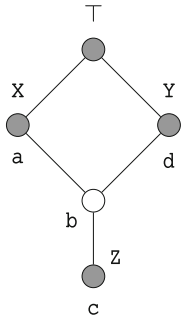
Perhaps un-intuitively, this formulation allows for a typical concept to contain non-typical objects in its extent, as long as they share all the attributes of their typical counterparts. We should recognise, however, that they do not contribute to describing the intent, they just happen to be consistent with it. Although we ascribe specific meaning to $\underline{A}'$, it can be considered to be an arbitrary set of objects. Then, per Definition 3, an arbitrary typical concept, $((\underline{X}')'', (\underline{X}')')$ where $X \subseteq M$, is always equivalent to some formal concept. As such, we can define a map from the set of concepts to the set of typical concepts.

Definition 14. *Let $\mathbb{K}^{\preceq} = (G, M, I, \preceq)$ be an extended formal context,then $\mathfrak{B}(\mathbb{K}^{\preceq})$ denote the set of concepts of $\mathbb{K}^{\preceq}$, and $\mathfrak{T}(\mathbb{K}^{\preceq})$ denote the set of typical concepts of $\mathbb{K}^{\preceq}$. We define a map:*

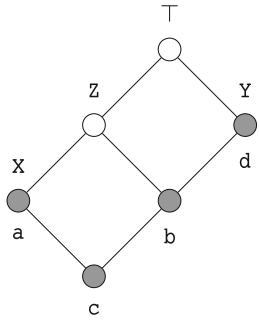
$$\varphi : \mathfrak{B}(\mathbb{K}) \mapsto \mathfrak{T}(\mathbb{K}^{\preceq})$$

For a concept $(A, B) \in \mathfrak{B}(\mathbb{K}^{\preceq})$, φ is defined as:

$$\varphi(A, B) := ((\underline{B}')'', (\underline{B}')')$$



(a) c ≺ b and a,d ≺ c



(b) a ≺ b,c,d

Fig. 3. Classical concept lattices, with the typical concepts derived from the respective ordering marked as grey

The mapping function inherits idempotency from the minimised-return operator. For a typical concept $(A, B) \in \mathfrak{T}(\mathbb{K}^{\preceq})$, there must exist some set $C \subseteq M$ such that $(\underline{C}')' = B$. It follows that $(\underline{(\underline{C}')'})' = (\underline{C}')' = B$. Consequently, $\varphi(A, B) = (A, B)$, or, typical concepts map to themselves. Differently put, $\varphi(A, B) = \varphi(\varphi(A, B))$. We also note that by the extensive property of minimized-derivation, for any concept (A, B), we have $(\underline{B}')' \supseteq B$ and therefore $\varphi(A, B) = ((\underline{B}')'', (\underline{B}')') \leq (A, B)$. That is, φ is *anti-extensive*.

Theorem 3. *In an extended formal context,* $\mathbb{K}^{\preceq} = (G, M, I, \preceq)$*, let* $(A, B) \in \mathfrak{B}(\mathbb{K}^{\preceq})$*, then* $(A, B) \in \mathfrak{T}(\mathbb{K}^{\preceq})$ *iff* $(\underline{B}')'' = A$*.*

Proof. Let $\mathbb{K}^{\preceq} = (G, M, I, \preceq)$ be an extended formal context with a concept, $(A, B) \in \mathfrak{B}(\mathbb{K}^{\preceq})$. Suppose (A, B) were a typical concept, then there exists a set $C \subseteq M$ such that $A = (\underline{C}')''$ and $B = (\underline{C}')'$, per the definition of a typical concept. Through the properties of the minimised-return operator in Theorem 1 we have that $\underline{C}' = \underline{(\underline{C}')'}$, equivalently, $\underline{C}' = \underline{B}'$. Then, obviously, $(\underline{C}')'' = (\underline{B}')''$. □

A next point of departure might be to investigate the structure the set of typical concepts. Certainly, we have that $\mathfrak{T}(\mathbb{K}^{\preceq}) \subseteq \mathfrak{B}(\mathbb{K}^{\preceq})$. Then in particular, we want to investigate whether the subset of "typical" objects forms a sub-lattice, if we can form a sub-lattice of only those "typical" concepts, we are able to use all the same algebraic tools to analyze the data in our context "prototypically", as opposed to the classical analysis which relies on strict implications and concepts. Using the same approach to ordering typical concepts as we did for formal concepts (see Sect. 2a), two examples make it clear that the current formulation unfortunately does not guarantee that the pair $(\mathfrak{T}(\mathbb{K}^{\preceq}), \leq)$, henceforth denoted $\underline{\mathfrak{T}}(\mathbb{K}^{\preceq})$, will form any type of lattice. Figure 3a is the concept lattice of some formal context, $\mathbb{K}$. If one were to extend $\mathbb{K}$ with the partial order c ≺ b and a,d ≺ c, the resultant set of typical concepts would be the four concepts

marked with grey nodes. We pay special attention to the concepts

$$C_1 := \Big((\underline{X}')'', (\underline{X}')'\Big) = \Big((\mathtt{a})'', (\mathtt{a})'\Big) = \Big(\{\mathtt{a},\mathtt{b},\mathtt{c}\}, \{\mathtt{X}\}\Big)$$
$$C_2 := \Big((\underline{Y}')'', (\underline{Y}')'\Big) = \Big((\mathtt{d})'', (\mathtt{d})'\Big) = \Big(\{\mathtt{b},\mathtt{c},\mathtt{d}\}, \{\mathtt{Y}\}\Big)$$

The greatest common sub-concept of these two concepts is given by

$$C_1 \wedge C_2 = \big(\{\mathtt{a},\mathtt{b},\mathtt{c}\} \cap \{\mathtt{b},\mathtt{c},\mathtt{d}\}, (\{\mathtt{X}\} \cup \{\mathtt{Y}\})''\}\big) = \big(\{\mathtt{b},\mathtt{c}\}, \{\mathtt{X},\mathtt{Y}\}\big)$$

However, from Theorem 3, it is clear that this is not a typical concept, as $(\underline{\{\mathtt{X},\mathtt{Y}\}}')'' = \{\mathtt{c}\} \neq \{\mathtt{b},\mathtt{c}\}$. This result tells us that the subset of typical concepts is not closed under meets. If we shift attention to Fig. 3b, it becomes clear that it is not closed under joins either, nor does it guarantee a lattice with respect to the concept-lattice ordering. For the typical concepts

$$C_1 := \Big((\underline{X}')'', (\underline{X}')'\Big) = \Big((\mathtt{a})'', (\mathtt{a})'\Big) = \Big(\{\mathtt{a},\mathtt{b}\}, \{\mathtt{X}\}\Big)$$
$$C_2 := \Big((\underline{\{Y,Z\}}')'', (\underline{\{Y,Z\}}')'\Big) = \Big((\mathtt{b},\mathtt{c})'', (\mathtt{b},\mathtt{c})'\Big) = \Big(\{\mathtt{b},\mathtt{c}\}, \{\mathtt{Y},\mathtt{Z}\}\Big)$$

the least common super-concept is given by

$$C_1 \vee C_2 = \big((\{\mathtt{a},\mathtt{c}\} \cup \{\mathtt{b},\mathtt{c}\})'', \{\mathtt{X},\mathtt{Z}\} \cap \{\mathtt{Y},\mathtt{Z}\}\}\big) = \big(\{\mathtt{b},\mathtt{c}\}, \{\mathtt{Z}\}\big).$$

For $C_1 \vee C_2$, the closure of the minimised-derivation of the intent, $(\underline{\mathtt{Z}}')'' = \{\mathtt{a},\mathtt{c}\}$, does not equal the concept extent. So, we do not have that typical concepts are closed under joins. To make matters worse, the top element of the lattice is the concept $(\{\mathtt{a},\mathtt{b},\mathtt{c},\mathtt{d}\}, \emptyset)$, but not a typical concept, as $(\underline{\emptyset}')'' = (\mathtt{a})'' = \{\mathtt{a},\mathtt{c}\}$, and so we do not have a lattice structure of typical concepts. In fact, we do not even have an upper bound to the set of typical concepts in Fig. 3b.

5.2 Restriction on the Partial Order

One cause of the aforementioned problem lies in the fact that we allow non-minimal objects to be included in typical concepts. This allows for rankings where non-minimal objects, which should be in the meet or join of two concepts, are *lost*. As a solution to this, we may restrict the partial orders to prevent non-minimal objects from being included in typical concepts.

Definition 15. *Let* $\mathbb{K}^{\preceq} = (G, M, I, \preceq)$ *be an extended formal context. A* valid partial order *over objects is one which for all* $A \subseteq M$, $\underline{A}' = (\underline{A}')''$.

Consider a set of attributes $X \subseteq M$ and then take $Y \subseteq M$ to be the set of all attributes common to the minimal objects which satisfy X – of course, Y would be the set X with additional attributes. The restriction to orders $\preceq$ that are valid ensures that any object that has all the attributes in Y must itself be minimal w.r.t. X. Put differently, Y serves as a total characterisation of the minimal objects satisfying attributes X, if an object matches this characterisation, then it should have been one of these minimal objects.

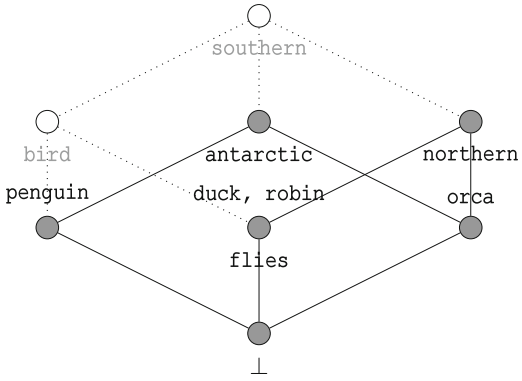


Fig. 4. The concept lattice for Example 1. Given the preference order `duck` $\preceq$ `penguin` and `robin` $\preceq$ `penguin`, the dark grey concepts show the structure of typical concepts

If we accept this restriction on the partial order, the set of typical concepts preserves meets, that is, the greatest common sub-concept of two typical concepts is itself a typical concept. Before this is shown, Definition 13, Definition 14, and Theorem 3 might be amended to reflect that the closure of a set $\underline{A}'$, where $A \subseteq M$, is superfluous.

Theorem 4. *For an extended formal context, $\mathbb{K}^{\preceq} = (G, M, I, \preceq)$, with a restriction on the partial-order such that for any $A \subseteq M$, $\underline{A}' = (\underline{A}')''$, the greatest common sub-concept of two typical concepts is itself a typical concept. Hence, the subset of typical concepts form a $\wedge$-subsemilattice of the concept lattice.*

Proof. We note that it is a sufficient condition to show that $\underline{A}' \cap \underline{B}' = \underline{C}'$ for some $C \subseteq M$. We claim that we can choose $C = (\underline{A}')' \cup (\underline{B}')'$. Note

$$\underline{A}' \cap \underline{B}' = (\underline{A}')'' \cap (\underline{B}')'' = ((\underline{A}')' \cup (\underline{B}')')',$$

where the first equality follows from our assumed condition, and the second from our preliminary lemma for Galois connections. Then $\underline{\underline{A}' \cap \underline{B}'} = \underline{((\underline{A}')' \cup (\underline{B}')')'}$. Note that

$$\underline{\underline{A}' \cap \underline{B}'} = \{g \in \underline{A}' \cap \underline{B}' \mid \nexists h \in \underline{A}' \cap \underline{B}' \text{ such that } h < g\}.$$

Therefore, if $g \in \underline{A}' \cap \underline{B}'$, since g is a minimal member of A' and of B' there is no h in A' or in B' with $h < g$ by definition and so $g \in \underline{A}' \cap \underline{B}'$. Therefore $\underline{A}' \cap \underline{B}' \subseteq \underline{\underline{A}' \cap \underline{B}'}$, and clearly $\underline{\underline{A}' \cap \underline{B}'} \subseteq \underline{A}' \cap \underline{B}'$. If the intersection has no elements $\underline{A}' \cap \underline{B}' = \emptyset = \underline{\emptyset}$. Hence $\underline{A}' \cap \underline{B}' = \underline{\underline{A}' \cap \underline{B}'} = \underline{((\underline{A}')' \cup (\underline{B}')')'}$ and we are done. □

It is also worth noting here that the bottom element of the concept lattice is always a typical concept since $\varphi(\bot) \leq \bot$, by anti-extensivity, and so $\varphi(\bot) = \bot$. This result allows us to begin characterizing the algebraic structure of those concepts considered "typical" in an extended formal context. However, there is still much structure that this restriction on the order does not account for. In particular, we do not have that $\underline{\mathfrak{T}}(\mathbb{K}^{\preceq})$ preserves least common super-concepts – Fig. 3b is an example of a concept lattice

whose order satisfies the condition in Definition 15, but does not have a $\top$-concept. That being said, we are still able to preserve some structure using a somewhat intuitive restriction on the order in our extended formal context.

6 Related Work

There is a large corpus of work on developing KLM-style notions of non-monotonic reasoning outside of the standard of propositional logic. [4], and more extensively [18], introduce KLM-style defeasible reasoning to datalog, which can be considered a fragment of first-order logic used for database queries. [1,2] are recent efforts to introduce a notion of defeasible subsumption to description logics (DL) that follows the KLM properties. Remaining in the realm of DL, [17] is a general characterisation of to KLM in DL $\mathcal{ALC}$. [3] investigates the KLM framework in the context of defeasible modalities, introducing new modal operators for *defeasible necessity* and *distinct possibility*.

With respect to work that extends our view of concepts in FCA, [11] introduces *Rough Concept Analysis*, a merging of rough set theory and FCA that uses equivalence classes on objects to define upper and lower-bound approximations of concepts. [22] investigates an expansion which enables "rough concepts" to be defined not only by objects.

7 Conclusions and Future Work

By extending a formal context with a preference relation on the objects we have introduced a non-monotonic variant of implication between attribute sets which characterise a rational consequence relation*. This strictly increases the expressivity of the attribute logic of FCA by creating a notion of non-monotonicity that corresponds to the KLM view of how a *logical* non-monotonic system should behave. In terms of FCA, we have introduced a way of discovering and representing relationships between attribute sets that tolerates exceptions, and is capable of representing what our data (formal context) shows in the *typical* case.

With a slight restriction on what constitutes a *valid* preference relation over objects, we presented a formalisation of typical concepts which has its foundations in a KLM-style typicality. We were able to show that with this notion we could create a structure of typical concepts that at least preserved the sub-concept relation, and as such is a *meet-subsemilattice* of the original concept lattice.

To our knowledge, the introduction of KLM-style typicality, and preferences over a formal context as a whole, presents a novel view on, and non-trivial extension to FCA.

This work represents the initial investigation of introducing KLM-style typicality into FCA. As such, we believe there is a considerable amount that remains to be looked at. Addressing concerns of existing work, we aim to find an approach to defining typical concepts which is closed under joins (super-concepts), essentially meaning that the structure of the set of typical concepts would be a sub-lattice of the original concept lattice. To this end, we could neatly "reduce" a concept lattice to its typical counterpart. In another branch of this work, we would like to investigate the relationship between the set of all non-monotonic conditionals and the typical concept lattice; determining if something analogous to a *canonical basis* can be found.

Acknowledgments. This work is a culmination of effort that took place when students at the University of Cape Town visited the Technical University of Dresden. We would like to express immense appreciation and gratitude to the School of Embedded Composite Artificial Intelligence (SECAI) – project 57616814 funded by BMBF (the Bundesministerium für Bildung und Forschung) and DAAD (German Academic Exchange Service) – for funding the visit, and to its members in Dresden who were so welcoming and engaging.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Britz, K., Casini, G., Meyer, T., Moodley, K., Sattler, U., Varzinczak, I.: Principles of klm-style defeasible description logics. ACM Transactions on Computational Logic (TOCL) **22**(1), 1–46 (2020)
2. Britz, K., Casini, G., Meyer, T., Varzinczak, I.: A klm perspective on defeasible reasoning for description logics. In: Description Logic, Theory Combination, and All That: Essays Dedicated to Franz Baader on the Occasion of His 60th Birthday, pp. 147–173. Springer (2019)
3. Britz, K., Varzinczak, I.: From klm-style conditionals to defeasible modalities, and back. J. Appl. Non-Classical Logics **28**(1), 92–121 (2018). https://doi.org/10.1080/11663081.2017.1397325
4. Casini, G., Meyer, T., Paterson-Jones, G., Varzinczak, I.: Klm-style defeasibility for restricted first-order logic. In: International Joint Conference on Rules and Reasoning. pp. 81–94. Springer (2022)
5. Citkin, A., Muravitsky, A.: Consequence relations an introduction to the tarski-lindenbaum method (2021), https://arxiv.org/abs/2106.10966
6. Etchemendy, J.: Tarski on truth and logical consequence. The Journal of Symbolic Logic **53**(1), 51–79 (1988), http://www.jstor.org/stable/2274427
7. Gabbay, D.M.: A general theory of structured consequence relations. Theoria: An International Journal for Theory, History and Foundations of Science pp. 49–78 (1995)
8. Ganter, B., Obiedkov, S., Rudolph, S., Stumme, G.: Conceptual exploration. Springer (2016)
9. Ganter, B., Wille, R.: Formal concept analysis: mathematical foundations. Springer Science & Business Media (2012)
10. Kaliski, A.: An overview of KLM-style defeasible entailment. Master's thesis, University of Cape Town (2020)
11. Kent, R.E.: Rough concept analysis: a synthesis of rough sets and formal concept analysis. Fund. Inform. **27**(2–3), 169–181 (1996)
12. Kraus, S., Lehmann, D., Magidor, M.: Nonmonotonic reasoning, preferential models and cumulative logics. Artif. Intell. **44**(1–2), 167–207 (1990)
13. Kraus, S., Lehmann, D., Magidor, M.: Nonmonotonic Reasoning, Preferential Models and Cumulative Logics (Feb 2002), http://arxiv.org/abs/cs/0202021, arXiv:cs/0202021
14. Lehmann, D., Magidor, M.: What does a conditional knowledge base entail? Artificial Intelligence **68**(2) (1994)
15. Lei, Y., Sui, Y., Cao, C.: The logical operations on relations, scales and concept lattices. In: 2009 Fifth International Conference on Semantics, Knowledge and Grid. pp. 160–167 (2009https://doi.org/10.1109/SKG.2009.37
16. Makinson, D.: How to go nonmonotonic. In: Handbook of Philosophical Logic, pp. 175–278. Springer (2005)

17. Moodley, K., Meyer, T., Sattler, U.: Practical defeasible reasoning for description logics. In: STAIRS 2014, pp. 191–200. IOS Press (2014)
18. Paterson-Jones, G.: KLM-Style Defeasible Reasoning for Datalog. Master's thesis, University of Cape Town (2022)
19. Rudolph, S.: Relational Exploration - Combining Description Logics and Formal Concept Analysis for Knowledge Specification. Universitätsverlag Karlsruhe (December 2006)
20. Shoham, Y.: A semantical approach to nonmonotonic logics, p. 227–250. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1987)
21. Strasser, C., Antonelli, G.A.: Non-monotonic Logic. In: Zalta, E.N., Nodelman, U. (eds.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, Summer 2024 edn. (2024)
22. Yao, Y.: Rough-set concept analysis: interpreting rs-definable concepts based on ideas from formal concept analysis. Inf. Sci. **346**, 442–462 (2016)

Knowledge Compilation for KLM-Style Defeasible Reasoning

Luke Slater[1(✉)], Thomas Meyer[1], and Jesse Heyninck[1,2]

[1] University of Cape Town and CAIR, Cape Town, South Africa
sltluk001@myuct.ac.za, tommie.meyer@uct.ac.za
[2] Open University, Heerlen, The Netherlands
jesse.heyninck@ou.nl

Abstract. In artificial intelligence, knowledge compilation involves transforming knowledge into a form that facilitates efficient reasoning. We introduce a method for knowledge compilation of ranked interpretations within the KLM framework for defeasible reasoning. Leveraging binary decision diagrams (BDDs), a well-established tool in knowledge compilation, we demonstrate that ranked interpretations can be effectively represented using a BDD with multiple terminal nodes, termed a ranked binary decision diagram (RBDD). Additionally, we show that an existing procedure for checking entailment against a BDD can be adapted with minimal modification to check defeasible entailment against an RBDD.

Keywords: knowledge representation and reasoning · defeasible reasoning · rational closure · knowledge compilation · binary decision diagrams

1 Introduction

In artificial intelligence, the field of knowledge representation and reasoning (KRR) aims to model human understanding and decision-making processes. One important aspect of knowledge representation is defeasible reasoning, where conclusions can be retracted when new information becomes available. The Kraus, Lehmann, and Magidor (KLM) framework is a well-known approach for formalizing defeasible reasoning [10,12,13]. Implementations of KLM-style defeasible reasoning typically divide knowledge into distinct levels of typicality and apply a classical SAT solver. This method comes with certain limitations, as it requires multiple calls to the SAT solver for a single query, and becomes computationally prohibitive when handling large knowledge bases.

Knowledge compilation, a subset of KRR, focuses on transforming knowledge into a form that allows efficient reasoning [6]. In this work, we introduce a method for compiling defeasible knowledge bases within the KLM framework. Our approach conducts extensive SAT solving offline on the knowledge base, requiring minimal SAT solving on query-specific information at runtime.

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 493–508, 2025.
https://doi.org/10.1007/978-3-031-78255-8_29

We leverage binary decision diagrams (BDDs), an idea first introduced by Lee [11], and a well-established tool in knowledge compilation, to represent ranked interpretations effectively. Specifically, we propose the concept of a ranked binary decision diagram (RBDD), a BDD with multiple terminal nodes that capture the different ranks of a ranked interpretation. Additionally, we demonstrate how existing procedures for checking entailment against a BDD can be adapted to check defeasible entailment against an RBDD with minimal modifications. This facilitates the integration of defeasible reasoning into existing knowledge compilation frameworks by leveraging the well-established algorithms and techniques for BDD manipulation.

We begin by introducing the necessary preliminaries on propositional logic (Sect. 2) and KLM-style defeasible reasoning, including ranked interpretations, defeasible entailment, and rational closure (Sect. 3.3). We then provide an overview of knowledge compilation using BDDs in the classical context (Sect. 4), explaining how they are employed to verify classical entailment. In Sect. 5, we introduce ranked binary decision diagrams and demonstrate how they can be used to efficiently verify defeasible entailment.

2 Propositional Logic

Propositional logic is composed of a set of $N = n + 1$ boolean variables $X = \{x_0, \ldots, x_n\}$. Elements in $\mathbb{B}^N = \{0, 1\}^N$ are referred to as valuations and will be represented as binary vectors denoted with boldface, e.g., $\mathbf{x}$. Semantically, a formula (or function) $f(x_0, ..., x_n)$ is defined as a mapping $f : \mathbb{B}^N \mapsto \mathbb{B}$, and it is assumed that all formulas have the same N arguments. Intuitively, $f(\mathbf{x})$ indicates whether the formula f is true or false in $\mathbf{x}$. To enhance readability, we will alternate between propositional logic notation and Boolean algebra notation. Propositional logic notation will be used when discussing defeasible reasoning, whereas Boolean algebra notation will be applied to equations related to the compilation process. Formulas are thus recursively defined as:

$$f := 1 \mid 0 \mid x_i \mid \overline{f} \mid f \cdot g \mid f + g \mid f \rightarrow g$$
$$f := \top \mid \bot \mid x_i \mid \neg f \mid f \wedge g \mid f \vee g \mid f \rightarrow g$$

When a formula f evaluates to 1 under some valuation $\mathbf{x}$ it is said to be a *model* of f, denoted $\mathbf{x} \Vdash f$. The set of such valuations is denoted S_f. A knowledge base $\mathcal{K}$ is a set of formulas. The set of models that satisfy all formulas in a knowledge base is denoted as $S_\mathcal{K}$. When $S_\mathcal{K} \subseteq S_f$, $\mathcal{K}$ is said to *entail* f, denoted $\mathcal{K} \models f$ [2]. Generating all models, a task known as #SAT ("sharp SAT"), to verify entailment is highly inefficient. A more practical approach is to add the negation of the query to the knowledge base and use a standard SAT solver to check for a contradiction. A formula is a contradiction if it has no models.

3 KLM-Style Defeasible Reasoning

The KLM framework provides a nonmonotonic extension of propositional logic that centers around the addition of defeasible implication, denoted $f \mathrel{|\!\sim} g$, read

"*defeasibly*, f implies g." Intuitively, this means that typically, if f is true, g is also expected to be true. The central goal of the framework is to study different approaches for determining under which conditions such statements should be retracted, and the different patterns in reasoning that result from doing so.

Example 1. Consider fruit, which is generally considered sweet ($f \rightarrow s$). Now consider a lemon, which is a type of fruit ($l \rightarrow f$), but also isn't sweet ($l \rightarrow \neg s$). Classical reasoning would render a lemon impossible, because its existence would result in a contradiction. However, the idea that fruit is sweet is defeasible, as is the idea that a lemon is not sweet. This information is thus captured by the defeasible knowledge base $\mathcal{K} = \{l \rightarrow f, f \mathrel{|\!\sim} s, l \mathrel{|\!\sim} \neg s\}$.

3.1 Ranked Interpretations

Defeasible implication is defined semantically through structures called *ranked interpretations* [5,13]. Ranked interpretations essentially replace the single notion of truth in classical logic with multiple degrees of typicality, where lower ranks represent "stronger" typicality. This approach assigns a specific rank to each model of the classical formulas in a defeasible knowledge base, reflecting its typicality. Models that are deemed classically impossible are assigned a rank of ∞, indicating their exclusion from consideration.

Definition 1. *A ranked interpretation is a mapping* $R : \mathbb{B}^N \mapsto \mathbb{N} \cup \{\infty\}$ *where for any* $\mathbf{x} \in \mathbb{B}^N$ *such that* $R(\mathbf{x}) = i$*, there must exist a* $\mathbf{x} \in \mathbb{B}^N$ *such that* $R(\mathbf{x}) = j$ *for every* $j < i$ *[5], with the exception that rank 0 may be empty.*

Note that rank ∞ is allowed to be empty.

Ranked interpretations partition $\mathbb{B}^N$ into $n+2$ sets $S_0, \ldots, S_n, S_\infty$ such that $S_i = \{\mathbf{x} \mid R(\mathbf{x}) = i\}$, for $i < \infty$. The characteristic formula of S_i, denoted as f^i, is defined by $f^i(\mathbf{x}) = 1$ if $\mathbf{x} \in S_i$ and 0 otherwise. These formulas are collectively represented by the vector $\boldsymbol{f} = \begin{bmatrix} f^0 \ldots f^n \end{bmatrix}^T$, and referred to as the characteristic vector of a ranked interpretation. The corresponding model vector for characteristic vector $\boldsymbol{f}$ will be denoted as $\boldsymbol{S_f} = \begin{bmatrix} S_0 \ldots S_n \end{bmatrix}^T$. Rank ∞ is represented implicitly in both cases.

Note 1. Here, the superscript T indicates the transpose of a vector, which converts a row vector into a column vector or vice versa.

Definition 2. *A ranked interpretation satisfies a defeasible implication* $f \mathrel{|\!\sim} g$*, denoted as* $R \Vdash f \mathrel{|\!\sim} g$*, iff all the minimally ranked models of* f *(excluding* ∞*) are models of* g *[5].*

Definition 3. *A ranked interpretation is a model of a defeasible knowledge base iff* $\forall f \mathrel{|\!\sim} g \in \mathcal{K}, R \Vdash f \mathrel{|\!\sim} g$ *[5].*

3.2 Defeasible Entailment

Ranked interpretations are also used to define several forms of defeasible entailment, much like models are used to define classical entailment in traditional knowledge bases. A crucial distinction, however, is that defeasible entailment is not a universally agreed upon concept; multiple definitions of defeasible entailment exist. In their study of defeasible entailment KLM introduced a set of rationality axioms, known as the KLM properties [13]. Any form of defeasible entailment that satisfies these properties is considered *rational*. Lehmann and Magidor further demonstrated that any form of defeasible entailment representable by a ranked interpretation is guaranteed to be rational. Two well-known patterns of defeasible reasoning that are representable by ranked interpretations are rational closure [13] and lexicographic closure [12].

Definition 4. *A defeasible formula is defeasibly entailed by a defeasible knowledge base $\mathcal{K}$, denoted $\mathcal{K} \mid\!\!\sim_R f \mid\!\sim g$, according to some ranked interpretation R that is a model of $\mathcal{K}$, iff $R \Vdash f \mid\!\sim g$ [5].*

3.3 Rational Closure

To provide additional context, we will briefly outline the most widely recognized definition of defeasible entailment: rational closure [13]. For further exploration of alternative forms of defeasible entailment, readers are referred to the relevant literature [5,12].

Semantically, rational closure can be understood through an ordering $\preceq_{\mathcal{K}}$ over the set of all ranked interpretations that are models of a defeasible knowledge base $\mathcal{K}$. Specifically, for two ranked interpretations R_1 and R_2, $R_1 \preceq_{\mathcal{K}} R_2$ holds if for every valuation $\mathbf{x} \in \mathbb{B}^N$, $R_1(\mathbf{x}) \leq R_2(\mathbf{x})$ [5]. It has been shown that there always exists a single most preferred ranked interpretation in this ordering, which uniquely defines the rational closure of the knowledge base [9], denoted as $R^{\mathcal{K}}_{RC}$.

Definition 5. *A defeasible implication $f \mid\!\sim g$ is defeasibly entailed according to the rational closure of $\mathcal{K}$, denoted $\mathcal{K} \mid\!\!\sim^{\mathcal{K}}_{RC} f \mid\!\sim g$, iff $R^{\mathcal{K}}_{RC} \Vdash f \mid\!\sim g$.*

A key procedure for understanding rational closure syntactically, and its most common implementation, is the `BaseRank` algorithm. This procedure transforms the set of defeasible implications from the knowledge base, where each defeasible implication $f \mid\!\sim g$ is replaced by its classical counterpart $f \rightarrow g$, into a totally ordered partition of these classical formulas, denoted as $\mathcal{K}_0, \ldots, \mathcal{K}_n$, with the set of classical formulas from the original knowledge base represented as $\mathcal{K}_\infty$ [13]. Given this partition of formulas, entailment in rational closure can be verified using the `RationalClosure` algorithm.

Alternatively, one can construct the characteristic vector $\boldsymbol{f} = \left[f^0 \ \ldots \ f^n\right]^T$ representing the ranked interpretation that defines the rational closure of this knowledge base. This vector, and similar ones for other forms of defeasible entailment, will prove particularly useful later for the compilation of ranked interpretations through BDDs. Let $\overline{\mathcal{K}}_i = \{\neg f \mid f \rightarrow g \in \mathcal{K}_i\}$.

Algorithm 1: `BaseRank`

Input: A defeasible knowledge base $\mathcal{K}$
Output: An ordered tuple $(\mathcal{K}_0, ..., \mathcal{K}_n, \mathcal{K}_\infty)$

$i := 0$;
$\mathcal{K}_\infty := \mathcal{K} \backslash \{f \mathrel{|\!\sim} g \in \mathcal{K}\}$;
$\mathcal{K}_i := \{f \rightarrow g \mid f \mathrel{|\!\sim} g \in \mathcal{K}\}$;
repeat
    $\mathcal{K}_{i+1} := \{f \rightarrow g \in \mathcal{K}_i \mid \mathcal{K}_\infty \cup \mathcal{K}_i \models \neg f\}$;
    $\mathcal{K}_i := \mathcal{K}_i \backslash \mathcal{K}_{i+1}$;
    $i := i + 1$;
until $\mathcal{K}_i = \emptyset$ ***or*** $\mathcal{K}_{i-1} = \emptyset$;
if $\mathcal{K}_{i-1} = \emptyset$ **then**
    $\mathcal{K}_\infty := \mathcal{K}_\infty \cup \mathcal{K}_i$;
    $i := i - 1$;
return $(\mathcal{K}_0, \ldots, \mathcal{K}_{i-1}, \mathcal{K}_\infty)$;

Algorithm 2: `RationalClosure`

Input: An ordered tuple $(\mathcal{K}_0, ..., \mathcal{K}_n, \mathcal{K}_\infty)$ and query $f \mathrel{|\!\sim} g$
Output: **true**, if $\mathcal{K} \mathrel{|\!\approx} f \mathrel{|\!\sim} g$, and **false**, otherwise

$i := 0$;
$\mathcal{K} := \bigcup_{i=0}^{j<n} \mathcal{K}_j$;
while $\mathcal{K}_\infty \cup \mathcal{K} \models \neg f$ ***and*** $\mathcal{K} \neq \emptyset$ **do**
    $\mathcal{K} := \mathcal{K} \backslash \mathcal{K}_i$;
    $i := i + 1$;
return $\mathcal{K}_\infty \cup \mathcal{K} \models f \rightarrow g$;

Theorem 1. *A characteristic vector for the ranked interpretation defining the rational closure of a knowledge base $\mathcal{K}$ can be constructed from the partitions $\mathcal{K}_0, ..., \mathcal{K}_n, \mathcal{K}_\infty$ produced by the* `BaseRank` *algorithm using the following equation:*

$$f^i = \bigwedge \mathcal{K}_\infty \wedge \bigwedge \mathcal{K}_i \wedge \bigwedge_{i<j} \overline{\mathcal{K}}_j \wedge \bigwedge_{j<i} \neg f^j$$

The equation works as follows: All classical formulas within the knowledge base must be satisfied by every finitely ranked model in the ranked interpretation ($\bigwedge \mathcal{K}_\infty$). Additionally, models in rank i must satisfy all defeasible formulas associated with ranks j such that $i < j$ ($\bigwedge_{i<j} \mathcal{K}_j$), in accordance with Definition 3. The antecedent of defeasible statements assigned to higher ranks should be impossible in this rank ($\bigwedge_{i<j} \overline{\mathcal{K}}_j$). Models already assigned a rank in earlier levels ($j < i$) should not be reconsidered for the current rank ($\bigwedge_{j<i} \neg f^j$).

Example 2. Lets add the statement that fruit is generally juicy ($f \mathrel{|\!\sim} j$) to our knowledge base from Example 1, which becomes $\mathcal{K} = \{l \rightarrow f, f \mathrel{|\!\sim} s, l \mathrel{|\!\sim} \neg s, f \mathrel{|\!\sim} j\}$. Rational closure separates lemons from typical fruit by creating the partition $\mathcal{K}_0 = \{f \rightarrow s, f \rightarrow j\}$, $\mathcal{K}_1 = \{l \rightarrow \neg s\}$, and $\mathcal{K}_\infty = \{l \rightarrow f\}$. The rational closure of this knowledge is then represented by the characteristic vector $\boldsymbol{f}$:

$$\boldsymbol{f} = \begin{bmatrix} f^0 = (l \rightarrow f) \wedge (f \rightarrow s) \wedge (f \rightarrow j) \wedge \neg l \\ f^1 = (l \rightarrow f) \wedge (l \rightarrow \neg s) \wedge \neg f^0 \\ f^2 = (l \rightarrow f) \wedge \neg f^0 \wedge \neg f^1 \end{bmatrix}$$

$$\boldsymbol{S_f} = \begin{bmatrix} S_{f^0} = \{\overline{l}\overline{f}j\overline{s}, \overline{l}\overline{f}js, \overline{l}f\overline{j}\overline{s}, \overline{l}f\overline{j}s, \overline{l}fjs\} \\ S_{f^1} = \{\overline{l}f\overline{j}\overline{s}, \overline{l}f\overline{j}s, \overline{l}fj\overline{s}, lfj\overline{s}, lf\overline{j}\overline{s}\} \\ S_{f^2} = \{lfjs, lf\overline{j}s\} \end{bmatrix}$$

Given this information, we may wish to deduce new insights, such as whether lemons are as juicy as typical fruits, i.e., $\mathcal{K} \not\approx l \not\sim j$. Examining the vector, we observe that the lowest-ranked models of lemon are $lfj\overline{s}$ and $lf\overline{j}\overline{s}$. Since one model satisfies juicy while the other does not, the entailment does not hold.

3.4 The Complexity of Defeasible Entailment

Inspection of Algorithms 1 and 2 also allow to explain the high computational complexity of defeasible entailment according to rational closure when implemented through a classical SAT solver: both algorithms involve a polynomial number of calls to a SAT-solver. For a defeasible knowledge base with n defeasible statements, the worst-case scenario for Algorithm 1 arises when each defeasible statement is assigned a distinct cell in the partition. In this case, the first iteration of the loop makes n calls to the SAT solver, the next iteration makes $n-1$ calls, and so on, resulting in $\frac{n(n+1)}{2}$ calls to the SAT solver. Consequently, the complexity of Algorithm 1 is $O(n^2 \cdot T_{SAT}(n))$, where $T_{SAT}(n)$ denotes the time complexity of the SAT solver with n defeasible statements (converted to classical statements). Assuming the worst-case complexity for SAT solving, this gives a worst case complexity of $O(n^2 \cdot 2^n)$. The worst-case scenario for Algorithm 2 occurs when line 3 is executed until the final partition is removed from $\mathcal{K}$, resulting in a linear number of calls to a SAT solver, giving a complexity of $O(n \cdot 2^n)$. It is the complexity of Algorithm 2 that we aim to improve upon with this paper.

4 Knowledge Compilation with Binary Decision Diagrams

Knowledge compilation addresses the computational intractability of general propositional reasoning [6]. The approach involves an initial, computationally expensive transformation into a target language during an offline phase, enabling efficient, polynomial-time query processing on the compiled representation during the online phase. Binary Decision Diagrams (BDDs) achieve this by providing a compact representation of the semantic information for a given formula.

The basic ideas behind BDDs were first introduced by Lee in 1959 [11], and Akers in 1978 [1], based on the concept of Shannon expansion - the expansion of a formula into positive and negative cofactors [16]. This decomposition is crucial for making the construction of a BDD computationally feasible, as a naive approach would require evaluating the formula for all valuations in $\mathbb{B}^N$.

Definition 6. *The cofactors of a formula f are sub-formulas that result from partially evaluating f. The cofactor of $f(x_0, ..., x_n)$ with respect to x_i is denoted f_{x_i} [16]:*

$$f_{x_i} = f(x_0, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) \qquad f_{\overline{x}_i} = f(x_0, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n)$$

Theorem 2. *A BDD can be produced by decomposing a formula f by recursively applying* Boole's expansion theorem *for any Boolean formula $f(x_0, ..., x_n)$ and ordering of variables $x_0, \ldots, x_n$ [4]:*

$$f = x_i \cdot f_{x_i} + \overline{x}_i \cdot f_{\overline{x}_i}$$

Each node in the binary decision diagram represents a variable x_i, and each subgraph f_{x_i} and $f_{\overline{x}_i}$ corresponds to a subformula, with the left and right children representing the cofactors with respect to x_i and $\overline{x}_i$, respectively. Additionally, the diagram contains two terminal nodes, representing the Boolean constants 0 and 1, which indicate the evaluation of the formula as false or true based on the input variables. By partially evaluating the cofactors at each recursive step, we can determine exactly when an edge should be created to a terminal node.

BDDs were later refined and popularized by Randal Bryant in 1986 [4], who significantly enhanced their utility by introducing several key improvements. Bryant established a canonical form by imposing a fixed variable ordering and applying reduction rules. These rules involve removing redundant nodes and merging isomorphic subgraphs (i.e., identical subformulas) [8].

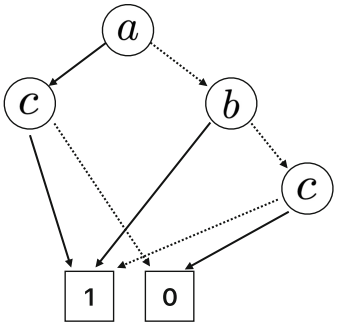
Theorem 3. *Two identical subformulas (subgraphs) can be merged by recursively merging their equivalent cofactors:*

$$f \equiv g \iff f_{x_i} \equiv g_{x_i} \text{ and } f_{\overline{x}_i} \equiv g_{\overline{x}_i}$$

Example 3. Figure 1 shows a BDD and its corresponding cofactor expansion for the formula $f(a, b, c)$, with variable ordering $x_0 = a$, $x_1 = b$, and $x_2 = c$. Since BDDs represent formulas in a canonical form, any formula with these models will produce the same diagram. A dotted arrow indicates false, while a non-dotted arrow represents true. First f is expanded into its cofactors for variable a to get $f = a \cdot f_a + \overline{a} \cdot f_{\overline{a}}$. Next f_a and $f_{\overline{a}}$ are expanded into their cofactors for variable b to get $f = a \cdot (b \cdot f_{ab} + \overline{b} \cdot f_{a\overline{b}}) + \overline{a} \cdot (b \cdot f_{\overline{a}b} + \overline{b} \cdot f_{\overline{a}\overline{b}})$. A tautology is detected in $f_{\overline{a}b}$ so the diagram contains an edge going directly to the 1 terminal node. The sub formulas f_{ab} and $f_{a\overline{b}}$ are equivalent and are thus merged.

BDDs offer a compact representation of all models of a formula, making them particularly suited for knowledge compilation. A significant advantage of constructing a BDD through recursive cofactor expansion is that it avoids the need to compute all models, an expensive and impractical task known as #SAT. Instead, a standard SAT solver can check each cofactor for a tautology or a contradiction, effectively allowing the search to end early down a particular branch in the diagram.

The theoretical worst-case complexity for constructing a BDD using recursive cofactor expansion occurs when no decision path terminates early, leading to an



$$S_f = \{\overline{a}bc, abc, \overline{a}b\overline{c}, \overline{a}\overline{b}\overline{c}, a\overline{b}c\}$$
$$S_f = a \cdot \{bc, \overline{b}c\} + \overline{a} \cdot \{bc, b\overline{c}, \overline{b}\overline{c}\}$$
$$S_f = a \cdot (b \cdot \{c\} + \overline{b} \cdot \{c\}) + \overline{a} \cdot (b \cdot \{c, \overline{c}\} + \overline{b} \cdot \{\overline{c}\})$$

Fig. 1. Recursive cofactor expansion

exponential number of SAT solver calls. This results in a worst-case complexity of $O(2^n \cdot 2^n)$. However, this complexity is highly dependent on various factors, and worst-case scenarios are exceptionally rare. In practice, the complexity typically ranges between $O(n \cdot 2^n)$ and $O(2^n \cdot 2^n)$. Bryant's reduction procedure has a complexity of $O(|G| \cdot \log |G|)$, where $|G|$ denotes the vector count of the BDD graph represented by G [4]. Consequently, this complexity can range from $O(n \cdot \log n)$ to $O(2^n \cdot \log 2^n)$.

4.1 Variable Ordering

The size of the BDD generated by the procedure in the previous section heavily depends on the chosen variable ordering. For some formulas, the difference can be dramatic-one ordering might yield a compact, linear-sized BDD, while another could produce an exponentially large diagram [4]. Finding the optimal ordering is a well-known challenge and has been proven NP-hard [3]. While both the best and worst orderings can significantly affect BDD size, they are difficult to identify in practice. Just as finding the optimal ordering is computationally demanding, so too is locating the worst one. Fortunately, many "good enough" orderings exist that result in manageable BDDs. Furthermore, may well known strategies exist for finding such variable orderings. Notable examples include the sifting algorithm [15], genetic algorithms [7], and simulated annealing [3].

4.2 Verifying Entailment

To verify entailment, we construct the conjunction of two binary decision diagrams: one representing the knowledge base and the other representing the negation of the query. If the resulting BDD simplifies to 0, it confirms that the query is entailed. Bryant provides an efficient procedure for creating the conjunction

of two BDDs with a complexity of $O(|G_1| \cdot |G_2|)$, where $|G_1|$ and $|G_2|$ denote the vertex counts of the respective graphs. The complexity of verifying entailment can thus range anywhere between $O(1)$ and $O(2^n \cdot 2^n)$. By investing additional resources to determine an optimal variable ordering for the knowledge base, the complexity of verifying entailment can approach often approach near-linear performance. We will now provide an outline of Bryant's algorithm with a focus on conjunction, the operation used for verifying entailment.

Theorem 4. *Boole's expansion theorem can be extended to decompose the conjunction of two formulas f and g into the conjunction of their cofactors [4]:*

$$f \cdot g = x_i \cdot (f_{x_i} \cdot g_{x_i}) + \overline{x}_i \cdot (f_{\overline{x}_i} \cdot g_{\overline{x}_i})$$

The above equation essentially illustrates that the conjunction of two formulas can be decomposed into the conjunction of its cofactors. A recursive procedure can thus be used to combine two BDDs by combining their corresponding subgraphs, starting with their respective root nodes. The following rules apply when combining formula graphs with roots v_1, v_2 and follow a priority order. $index(v_1)$ denotes the index of the variable represented by node v_1 in the variable ordering, while $left(v_1)$ and $right(v_1)$ represent the left and right children of node v_1 respectively.

1. If either v_1 or v_2 has value 0: Create a new terminal with value 0.
2. If both are terminals: Create a new terminal with value $v_1 \cdot v_2$.
3. If either is non-terminal and $index(v_1) = index(v_2)$: Create vertex u with index i, then recursively combine $left(v_1)$ with $left(v_2)$ to form $left(u)$, and $right(v_1)$ with $right(v_2)$ to form $right(u)$.
4. If either is non-terminal, $index(v_1) = i$ and $index(v_2) > i$ or v_2 is terminal: Create vertex u with index i, then recursively combine $left(v_1)$ with v_2 to form $left(u)$, and $right(v_1)$ with v_2 to form $right(u)$.

Rule 4 is similarly applied when the roles of each node are switched. After completing the process, any remaining redundant nodes are removed, and any additional equivalent sub-formulas are merged. The complexity of this procedure is further reduced using table storing entries (v_1, v_2, u) to avoid redundant evaluations of the same nodes as a result of merging identical sub functions [4].

5 Ranked Binary Decision Diagrams

We now show that a ranked interpretation represented by characteristic vector $\boldsymbol{f} = \begin{bmatrix} f^0 & \dots & f^n \end{bmatrix}^T$ can be compactly represented by a BDD with multiple terminal nodes, each corresponding to truth for a specific f^i. Such diagrams are known in the literature as multi-terminal BDDs (MTBDDs), and can used for any function $f : \mathbb{B}^N \mapsto \tilde{R}$, with $\tilde{R}$ being any finite set [8]. More generally, they can be used for any function $f : \tilde{D} \mapsto \tilde{R}$ by encoding the members of $\tilde{D}$ in binary [8]. Here, we will refer to a BDD representing a ranking function as a Ranked BDD (RBDD).

Definition 7. *The cofactors of a ranking represented by characteristic vector* $\boldsymbol{f}$ *with respect to* x_i *and* $\overline{x}_i$ *are respectively denoted as* $\boldsymbol{f}_{x_i}$ *and* $\boldsymbol{f}_{\overline{x}_i}$*, and are simply the cofactors of the formulas in* $\boldsymbol{f}$*:*

$$\boldsymbol{f}_{x_i} = \left[f^0_{x_i} \ldots f^n_{x_i}\right]^T \quad \boldsymbol{f}_{\overline{x}_i} = \left[f^0_{\overline{x}_i} \ldots f^n_{\overline{x}_i}\right]^T$$

Theorem 5. *A characteristic vector* f *can be decomposed to produce an RBDD by recursively applying booles expansion theorem to each formula in the vector:*

$$\boldsymbol{f} = x_i \cdot \boldsymbol{f}_{x_i} + \overline{x}_i \cdot \boldsymbol{f}_{\overline{x}_i}$$

Note 2. For $\boldsymbol{f} = \left[f^0 \ldots f^n\right]^T$, $x_i \cdot \boldsymbol{f}$ denotes $\left[x_i \cdot f^0 \ldots x_i \cdot f^n\right]^T$. Similarly, for any two characteristic vectors $\boldsymbol{f} = \left[f^0 \ldots f^n\right]^T$ and $\boldsymbol{g} = \left[g^0 \ldots g^n\right]^T$, $\boldsymbol{f} + \boldsymbol{g}$ denotes $\left[f^0 + g^0 \ldots f^n + g^n\right]^T$.

A characteristic vector can thus be decomposed to produce a RBDD in a similar manner to a regular BDD using a standard SAT solver. Each formula in the vector should be recursively expanded at the same time. If a tautology is detected in one of them then one should create an edge to its corresponding truth terminal node. Once a contradiction is detected in one of the formulas then one no longer needs to expand it along with the other formulas. An overall contradiction will only occur if it occurs in every formula in the vector, in which case an edge to the ∞ terminal node should be created. Finally, identical sub ranked interpretations should be merged using a similar rule to the original.

Theorem 6. *Identical sub ranked interpretations in an RBDD can be merged by recursively merging equivalent cofactors:*

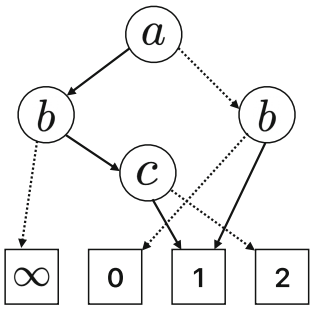
$$\boldsymbol{f} \equiv \boldsymbol{g} \iff \boldsymbol{f}_{x_i} \equiv \boldsymbol{g}_{x_i} \quad \textit{and} \quad \boldsymbol{f}_{\overline{x}_i} \equiv \boldsymbol{g}_{\overline{x}_i}$$

Definition 8. *Two rankings represented by characteristic vectors* $\boldsymbol{f} = \left[f^0 \ldots f^n\right]^T$ *and* $\boldsymbol{g} = \left[g^0 \ldots g^n\right]^T$ *are equivalent iff* $f^0 \equiv g^0 \ldots f^n \equiv g^n$.

The worst-case complexity of constructing an RBDD from a characteristic vector through recursive cofactor expansion is only marginally increased by the number of ranks. If there are n ranks, then instead of making a single SAT solver call as in the construction of a regular BDD, n SAT solver calls are required for constructing an RBDD. Despite this increase, the overall complexity of constructing an RBDD remains fundamentally similar to that of constructing a regular BDD.

Example 4. Figure 2 illustrates the binary decision diagram for some characteristic vector $\boldsymbol{f}(a, b, c, d)$. Below the diagram is the corresponding cofactor expansion, represented by the model vector $\boldsymbol{S_f}$. As with regular BDDs, any characteristic vector $\boldsymbol{f}$ that has the model vector $\boldsymbol{S_f}$ will result in the same diagram. First, vector $\boldsymbol{f}$ is expanded into its cofactors for variable a to get $\boldsymbol{f} = a \cdot \boldsymbol{f}_a + \overline{a} \cdot \boldsymbol{f}_{\overline{a}}$. A contradiction is detected in f^0_a and in $f^2_{\overline{a}}$, and they do not need to be expanded any further. Next $\boldsymbol{f}_a$ and $\boldsymbol{f}_{\overline{a}}$ are expanded into their cofactors for variable b

to get $\boldsymbol{f} = a \cdot (b \cdot \boldsymbol{f}_{ab} + \overline{b} \cdot \boldsymbol{f}_{a\overline{b}}) + \overline{a} \cdot (b \cdot \boldsymbol{f}_{\overline{a}b} + \overline{b} \cdot \boldsymbol{f}_{\overline{a}\overline{b}})$. In the case of $\boldsymbol{f}_{a\overline{b}}$, a contradiction is detected in every cofactor in the vector, and an edge is created directly to the ∞ node. Tautologies are detected in $f^1_{\overline{a}b}$ and $f^0_{\overline{a}\overline{b}}$, and edges are created directly to their respective terminal nodes.



$$\boldsymbol{S_f} = \begin{bmatrix} \{\overline{a}\overline{b}\overline{c}, \overline{a}\overline{b}c\} \\ \{\overline{a}bc, abc, \overline{a}b\overline{c}\} \\ \{ab\overline{c}\} \end{bmatrix}$$

$$\boldsymbol{S_f} = a \cdot \begin{bmatrix} \emptyset \\ \{bc\} \\ \{b\overline{c}\} \end{bmatrix} + \overline{a} \cdot \begin{bmatrix} \{\overline{b}\overline{c}, \overline{b}c\} \\ \{bc, b\overline{c}\} \\ \emptyset \end{bmatrix}$$

$$\boldsymbol{S_f} = a \cdot (b \cdot \begin{bmatrix} \emptyset \\ \{c\} \\ \{\overline{c}\} \end{bmatrix} + \overline{b} \cdot \begin{bmatrix} \emptyset \\ \emptyset \\ \emptyset \end{bmatrix}) + \overline{a} \cdot (b \cdot \begin{bmatrix} \emptyset \\ \{c, \overline{c}\} \\ \emptyset \end{bmatrix} + \overline{b} \cdot \begin{bmatrix} \{\overline{c}, c\} \\ \emptyset \\ \emptyset \end{bmatrix})$$

Fig. 2. Recursive cofactor expansion of a ranked interpretation.

5.1 Conjoining Ranked BDDs

Conjoining Generic Rankings. For generic rankings with no underlying ranking procedure - where their conjunction is simply the conjunction of their corresponding ranks - the following rules apply for conjunction on BDDs for rankings represented by characteristic vectors $\boldsymbol{f} = \begin{bmatrix} f^0 \dots f^n \end{bmatrix}^T$ and $\boldsymbol{g} = \begin{bmatrix} g^0 \dots g^n \end{bmatrix}^T$. It is easy to see that no modification is required to the procedure outlined above, as it does not discriminate between terminal nodes.

Theorem 7. *The RBDDs representing two generic rankings* $\boldsymbol{f}$ *and* $\boldsymbol{g}$ *can be conjoined by recursively conjoining their corresponding cofactors.*

$$\boldsymbol{f} \cdot \boldsymbol{g} = x_i \cdot (\boldsymbol{f}_{x_i} \cdot \boldsymbol{g}_{x_i}) + \overline{x}_i \cdot (\boldsymbol{f}_{\overline{x}_i} \cdot \boldsymbol{g}_{\overline{x}_i})$$

Querying Each Rank Simultaneously. RBDDs enable the compact representation of all models of each rank within a ranked interpretation into a single shared structure. The next step involves devising a method to query each rank in a unified procedure. By constructing a BDD for the query, allowing its terminal node to represent all natural numbers, $\mathbb{N}$, and letting the terminal nodes in a RBDD represent the singleton set containing their value, we can employ set-wise operations instead of boolean operations. We will informally refer to this query BDD as a threshold BDD (TBDD), as it distinguishes what is possible and what is impossible with regard to the query prior to the conjunction with the RBDD.

Note 3. A TBDD is constructed in exactly the same manner as a standard BDD.

Theorem 8. *Each rank in a ranked interpretation, as represented by an RBDD, can be queried classically by conjoining it with a TBDD that encodes the query.*

The result is a modified RBDD, where finitely ranked models that do not satisfy the query are redirected to the ∞ terminal node. This process is important for several reasons. First, as we will see in the next section, it enables the verification of defeasible entailment. Second, it provides a method for integrating classical and defeasible knowledge. For example, if an RBDD represents a defeasible knowledge base and one wishes to add a classical statement, the RBDD can be easily updated by conjoining it with a TBDD that encodes the classical statement.

The procedure for conjoining an RBDD with a TBDD closely mirrors the original method introduced by Bryant. In fact, all it requires is very minor modification to rules 1 and 2 of the conjunction procedure for nodes v_1 and v_2, which now represent the sets V_1 and V_2 respectively.

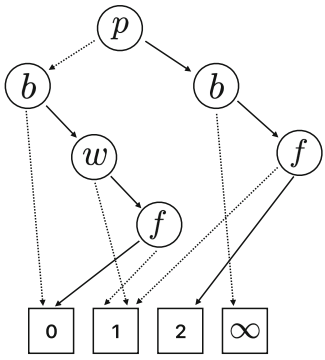
1. If either v_1 or v_2 has value ∞: Create a new terminal with value ∞.
2. If both are terminals: Create a new terminal with value $V_1 \cap V_2$.

Verifying Defeasible Entailment. Recall that verifying defeasible entailment for a query $f \mid\!\sim g$ on a ranked interpretation R involves finding the lowest rank i such that $R(\mathbf{x}) = i$ and $\mathbf{x} \Vdash f$, and then checking whether all such valuations satisfy $\mathbf{x} \Vdash g$. This is equivalent to finding the lowest-ranked valuation $\mathbf{x}_1$ such that $\mathbf{x}_1 \Vdash f \wedge g$, the lowest-ranked valuation $\mathbf{x}_2$ such that $\mathbf{x}_2 \Vdash f \wedge \neg g$, and verifying that $R(\mathbf{x}_1) < R(\mathbf{x}_2)$.

Theorem 9. *Defeasible entailment of a statement $f \mid\!\sim g$ can be verified on an RBDD by separately conjoining it with the TBDDs representing $f \wedge g$ and $f \wedge \neg g$, and comparing the terminal nodes representing the lowest finite rank in each diagram.*

Example 5. Consider a typical bird, which we generally agree can fly and thus has wings. Now, consider a penguin, which, although a bird, cannot fly. This information is represented by the knowledge base $\mathcal{K} = \{b \mid\!\sim f, b \mid\!\sim w, p \rightarrow b, p \mid\!\sim \neg f\}$. Rational closure differentiates reasoning about typical birds from reasoning

about typical penguins by creating the following partitions: $\mathcal{K}_0 = \{b \rightarrow f, b \rightarrow w\}$, $\mathcal{K}_1 = \{p \rightarrow \neg f\}$, and $\mathcal{K}_\infty = \{p \rightarrow b\}$. Figure 3 illustrates the ranked interpretation corresponding to the rational closure of this knowledge.



$$\boldsymbol{S_f} = \begin{bmatrix} S_{f^0} = \{\overline{p}\overline{b}w\overline{f}, \overline{p}\overline{b}wf, \overline{p}\overline{b}\overline{w}\overline{f}, \overline{p}\overline{b}\overline{w}f, \overline{p}bwf\} \\ S_{f^1} = \{\overline{p}b\overline{w}\overline{f}, \overline{p}b\overline{w}f, \overline{p}bw\overline{f}, pbw\overline{f}, pb\overline{w}\overline{f}\} \\ S_{f^2} = \{pbwf, pb\overline{w}f\} \end{bmatrix}$$

$$\boldsymbol{f} = \begin{bmatrix} f^0 = (p \rightarrow b) \wedge (b \rightarrow f) \wedge (b \rightarrow w) \wedge \neg p \\ f^1 = (p \rightarrow b) \wedge (p \rightarrow \neg f) \wedge \neg f^0 \\ f^2 = (p \rightarrow b) \wedge \neg f^0 \wedge \neg f^1 \end{bmatrix}$$

Fig. 3. RBDD representing characteristic vector $\boldsymbol{f}$

We currently lack explicit information about penguins, except that they cannot fly. Given this, we might want to deduce additional information about penguins, such as whether they have wings, i.e., $R \models p \mathrel{|\!\sim} w$. To verify entailment, we must identify the lowest ranked models that satisfy p and check if they all satisfy w. This process involves finding the lowest rank that contains a model satisfying $p \wedge w$ and the lowest rank that contains a model satisfying $p \wedge \neg w$, and then comparing these ranks. First, we create two TBDDs: one for $p \wedge w$ and another for $p \wedge \neg w$. Each TBDD is then conjoined separately with the RBDD representing the knowledge base, following the previously described procedure. Finally, we examine the incoming edges of the terminal nodes in the resulting RBDDs and compare them to determine if the defeasible entailment holds. Figure 4 illustrates the relevant diagrams, with the left representing $p \wedge w$ and the right representing $p \wedge \neg w$.

In both cases, we observe that rank 0, representing typical birds, has been retracted and is no longer part of either diagram. To verify entailment, we must identify the lowest remaining rank in each RBDD. As they are the same in both cases, the entailment does not hold. The entailment would hold only if the lowest remaining rank in the RBDD resulting from the conjunction with $p \wedge \neg w$

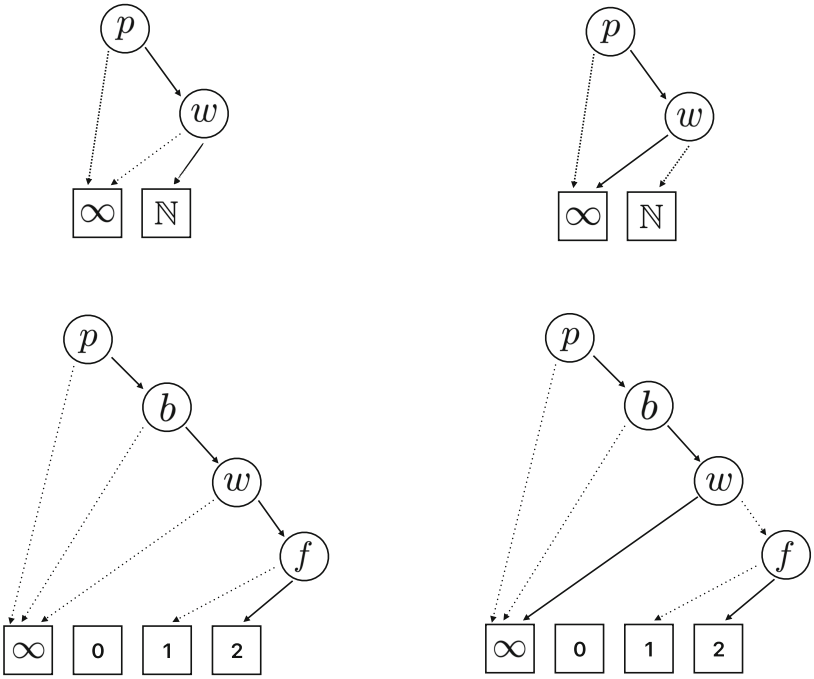


Fig. 4. TBDDs and resulting RBDDs

was greater than the lowest remaining rank in the RBDD resulting from the conjunction with $p \wedge w$.

6 Related Work

This paper builds on work done by Slater et al. in [17] in which regular BDDs are created seperately for each rank in a ranked interpretation. BDDs with multiple terminal nodes have been studied previously in providing efficient representations of large matrixes, along with methods for performing procedures such as matrix multiplication and Walsh transformation in [8].

7 Conclusion

This paper provides a preliminary introduction to the concept of compiling defeasible knowledge bases using binary decision diagrams (BDDs), laying the groundwork for further exploration in this area. While we have initiated a discussion on the complexity of the proposed procedures, much work remains to fully develop and validate this approach. A future paper will include correctness proofs for theorems 5 to 9, a comprehensive complexity analysis, along with implementation and experimental evaluation to formally compare this method with traditional techniques. Additionally, we aim to investigate more direct algorithms for

verifying defeasible entailment against an RBDD, as well as any other queries that could be of use to the field.

Future research directions include exploring the use of RBDDs in explainable AI and developing shared RBDDs for compactly representing multiple rankings [14]. It is plausible that more efficient methods for verifying defeasible entailment may exist. Additionally, there may be opportunities to combine two compiled defeasible knowledge bases by leveraging their respective RBDDs. Variable ordering is another promising research area; investigating various established strategies [3,7] for determining effective variable ordering in RBDDs is crucial for making this approach practical. Specialized algorithms designed to optimize variable ordering based on the characteristics of a given defeasible knowledge base could prove particularly valuable.

References

1. Akers: Binary decision diagrams. IEEE Trans. Comput. **C-27**(6), 509–516 (1978)
2. Ben-Ari, M.: Propositional logic: formulas, models, tableaux. In: Mathematical Logic for Computer Science, pp. 1, 7–47. Springer, London (2012). https://doi.org/10.1007/978-1-4471-4129-7_2
3. Bollig, B., Wegener, I.: Improving the variable ordering of OBDDs is NP-complete. IEEE Trans. Comput. **45**(9), 993–1002 (1996)
4. Bryant: Graph-based algorithms for Boolean function manipulation. IEEE Trans. Comput. **C-35**(8), 677–691 (1986)
5. Casini, G., Meyer, T., Varzinczak, I.: Taking defeasible entailment beyond rational closure. In: Calimeri, F., Leone, N., Manna, M. (eds.) JELIA 2019. LNCS (LNAI), vol. 11468, pp. 182–197. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-19570-0_12
6. Darwiche, A., Marquis, P.: A knowledge compilation map. CoRR **abs/1106.1819** (2011), http://arxiv.org/abs/1106.1819
7. Drechsler, R., Becker, B., Göckel, N.: Genetic algorithm for variable ordering of OBDDs. Comput. Digit. Tech. IEE Proceedings - **143**, 364 – 368 (12 1996)
8. Fujita, M., McGeer, P.C., Yang, J.C.Y.: Multi-terminal binary decision diagrams: an efficient data structure for matrix representation. Formal Methods Syst. Des. **10**(2), 149–169 (1997)
9. Giordano, L., Gliozzi, V., Olivetti, N., Pozzato, G.: Semantic characterization of rational closure: from propositional logic to description logics. Artif. Intell. **226**, 1–33 (2015)
10. Kraus, S., Lehmann, D., Magidor, M.: Nonmonotonic reasoning, preferential models and cumulative logics. Artif. Intell. **44**(1), 167–207 (1990)
11. Lee, C.Y.: Representation of switching circuits by binary-decision programs. Bell Syst. Tech. J. **38**(4), 985–999 (1959)
12. Lehmann, D.: Another perspective on default reasoning. Ann. Math. Artif. Intell. **15**, 61–82 (1995)
13. Lehmann, D., Magidor, M.: What does a conditional knowledge base entail? Artif. Intell. **55**(1), 1–60 (1992)
14. Minato, S.I., Ishiura, N., Yajima, S.: Shared binary decision diagram with attributed edges for efficient boolean function manipulation. In: Proceedings of the 27th ACM/IEEE Design Automation Conference, pp. 52–57 (1991)

15. Rudell, R.: Dynamic variable ordering for ordered binary decision diagrams. In: Proceedings of 1993 International Conference on Computer Aided Design (ICCAD), pp. 42–47 (1993)
16. Shannon, C.E.: The synthesis of two-terminal switching circuits. Bell Syst. Tech. J. **28**(1), 59–98 (1949)
17. Slater, L., Meyer, T.: Extending defeasible reasoning beyond rational closure. In: Pillay, A., Jembere, E.J., Gerber, A. (eds.) SACAIR 2023. CCIS, vol. 1976, pp. 151–171. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-49002-6_11

Author Index

A
Adigun, Matthew O. 282
Agyemang, Alex 107
Ajagbe, Sunday Adeola 282
Amir, Hamza 144

B
Badenhorst, Jaco 74, 91
Benton, Paige 440
Brooks, William 253
Brown, Dane 300, 334

C
Carr, Lucas 476
Chieza, Kenneth 300, 334
Connan, James 334

D
Davel, Marelie H. 192, 235, 253, 319
de Waal, Alta 21
Derks, Iena Petronella 21

G
Gaibie, Adeeb 144
Govender, Avashna 91

H
Helberg, Albert S. J. 192
Heyninck, Jesse 493

I
Isafiade, Omowunmi E. 53

K
Kotzé, Eduan 161
Kunene, Dumisani 176

L
Langa, Kelly 3
Leisegang, Nicholas 459, 476
Liywalii, Eddie 423
Loots, Theodor 21

M
Malan, Katherine Mary 38
Malatjie, Patrick 176
Martin, Andreas 372
Matlala, Boitumelo 270
Matthee, Machdel 355
Meyer, Thomas 459, 476, 493
Modipa, Thipe I. 319
Moodley, Deshendran 144
Moongela, Harry 355
Mouton, Coenraad 235, 253
Mudali, Pragasen 282
Mutembei, Leonard L. 207

N
Nana, Muhammad Ahmed 176
Nandutu, Irene 144
Ngorima, Simbarashe Aldrin 192

O
Okuboyejo, Olaperi 3

P
Potgieter, Hermanus L. 235

R
Ramalepe, Simon Phetole 319
Roos, Darren Craig 38
Rudolph, Sebastian 459, 476

S
Salie, Daanyaal 300, 334
Schlippe, Tim 107

© The Editor(s) (if applicable) and The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
A. Gerber et al. (Eds.): SACAIR 2024, CCIS 2326, pp. 509–510, 2025.
https://doi.org/10.1007/978-3-031-78255-8

Sekopa, Teboho 176
Seletani, Rofhiwa 176
Senekal, Burgert A. 161
Senekane, Makhamisa C. 207
Siphuma, Elekanyani 124
Slater, Luke 493
Smith, Jarod 21
Sobantu, Sive 53
Spahic-Bogdanovic, Maja 372
Ssemugabi, Samuel 390
Stander, Jean-Pierre 21

T
Turpin, Marita 355

V
van der Haar, Dustin 270
van der Merwe, Alta 355
van Zyl, Terence 124, 207, 405
Vandapalli, Hima 270
Vilakazi, Nomfundo 405

W
Wang, Hairong 3
Wenger, Seline 372

Z
Zandamela, Frank 176