

Lazaros Iliadis
Ilias Maglogiannis
Efthymoulos Kyriacou
Chrisina Jayne (Eds.)

Communications in Computer and Information Science

2582

Engineering Applications of Neural Networks

26th International Conference, EANN 2025
Limassol, Cyprus, June 26–29, 2025
Proceedings, Part II

Part 2

Communications in Computer and Information Science

2582

Series Editors

Gang Li , *School of Information Technology, Deakin University, Burwood, VIC, Australia*

Joaquim Filipe , *Polytechnic Institute of Setúbal, Setúbal, Portugal*

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.

Lazaros Iliadis · Ilias Maglogiannis ·
Efthymoulos Kyriacou · Chrisina Jayne
Editors

Engineering Applications of Neural Networks

26th International Conference, EANN 2025
Limassol, Cyprus, June 26–29, 2025
Proceedings, Part II

Editors

Lazaros Iliadis 
Democritus University of Thrace
Xanthi, Greece

Efthymoulos Kyriacou 
Cyprus University of Technology
Limassol, Cyprus

Ilias Maglogiannis 
University of Piraeus
Piraeus, Greece

Chrisina Jayne 
Teesside University
Middlesbrough, UK

ISSN 1865-0929

ISSN 1865-0937 (electronic)

Communications in Computer and Information Science

ISBN 978-3-031-96198-4

ISBN 978-3-031-96199-1 (eBook)

<https://doi.org/10.1007/978-3-031-96199-1>

© 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

EAAAI/EANN 2025 More than a Quarter of a Century Event!

The 26th *Engineering Applications and Advances of Artificial Intelligence* conference (EAAAI 2025) was the descendant of the *Engineering Applications of Neural Networks* (EANN) conference that started its journey in *Otaniemi Finland* in 1995. Since then, it has been organized in several European countries (e.g., UK, Bulgaria, Sweden, Italy, France, Cyprus, Poland, Spain, Greece). As Artificial Intelligence keeps advancing with the emergence of more efficient algorithms, coverage of a broader scientific spectrum becomes a necessity. The organizers are very proud of the continuous and perceptible presence of EAAAI in the literature for more than a quarter of a century.

On behalf of the EAAAI Organizing and Scientific Committees, it was a real honor and pleasure to welcome all Artificial Intelligence scholars who are looking for a well-established and prestigious forum to communicate, to publish and disseminate the innovative advances of their research. On the other hand, EAAAI 2025 provided the chance to get to know each other better, to develop strong bonds and to find paths of potential fruitful cooperation. Overall, the conference not only succeeded in offering a deep glimpse into the future, but it emphasized the Ethical principles that should govern Artificial Intelligence standards.

The conference accepted papers from the whole AI spectrum focusing on Engineering applications. However, it also accepted novel research in other scientific domains.

Through the years, the conference has evolved into a well-respected and mature event. The EAAAI community comprises of extended group of leading scientists, not only from Europe but from all over the globe.

The h-index of the EANN (now EAAAI) event is as high as 31. The Proceedings are always published in the Springer Communications in Computer and Information Science CCIS book series, which is Indexed in the Conference Proceedings Citation Index (CPCI) – part of Clarivate Analytics' Web of Science, EI Engineering Index (Compendex and Inspec databases), ACM Digital Library, DBLP, Google Scholar, IO-Port, MathSciNet, Scopus and Zentralblatt MATH. All the above clearly prove the novelty and quality of the state-of-the-art research presented and published and its important contribution to the promotion of scientific ideas, algorithms and methods, globally.

The fact that the Springer Proceedings of the conference have been downloaded more than 1,000,000 times so far supports the robust impact of the event among the scholars of AI globally.

EAAAI 2025 was successfully organized (both physically and remotely via Webex) from the 26th to the 29th of June 2025 once more in an interactive hybrid mode, aiming to provide the chance to scholars who cannot physically attend to submit, and present by interacting in real time. After 25 years (a quarter of a century) strong links have been developed with the members of our scientific community.

The venue of the 26th EAAAI 2025 was the Technical University of Cyprus, in the city of Limassol. Cyprus is one of the most iconic and gorgeous Mediterranean islands, with direct airplane connections to all over Europe. Moreover, the amazing atmosphere and the excellent academic environment of the Technical University of Cyprus offered a perfect venue for all AI scientists. The conference participants attending live had the chance to visit the beautiful, amazing and breath-taking island of Cyprus.

Nevertheless, the EAAAI team has extended experience in hybrid long-distance conference organization. We should mention that the 2020 and 2021 EAAAI (ex-EANN) conferences were broadcast 100% remotely due to Covid restrictions, whereas the 2022, 2023 and 2024 events were organized in a hybrid mode, all of them with great success.

These two Proceedings volumes comprise the papers that were accepted to be presented orally at the 26th EAAAI conference.

The diverse nature of the papers presented demonstrates the vitality of Artificial Intelligence algorithms and approaches. The conference provides a very wide forum for AI applications as well.

In total, 101 papers were initially submitted. All papers were peer reviewed by at least two independent academic referees. Where needed, a third referee was consulted to resolve any potential conflicts. A total of 40.5% of the submitted manuscripts (41 papers) were accepted to be published in the Springer proceedings as full papers.

The 5th Workshop on AI and Ethics (AIETH 2025) was organized and Coordinated by John MacIntyre under the auspices of the 26th EANN. It included short presentations from the panel members and an open Q&A session where the audience members were able to ask, and answer, important questions about the current and future development of Generative AI models. It aimed to emphasize the need for responsible and ethical global AI. The respective scientific community must be prepared to act preemptively and ensure that our societies will avoid negative effects of AI and of the 4th Industrial Revolution in general.

Organization

General Co-chairs

Lazaros Iliadis
Andreas Andreou
John MacIntyre

Democritus University of Thrace, Greece
Cyprus University of Technology, Cyprus
University of Abertay, UK

Program Co-chairs

Ilias Maglogiannis
Efthymoulos Kyriacou
Antonis Papaleonidas

University of Piraeus, Greece
Cyprus University of Technology, Cyprus
Democritus University of Thrace, Greece

Steering Committee

Lazaros Iliadis
Elias Pimenidis
Chrisina Jayne

Democritus University of Thrace, Greece
University of the West of England, UK
Teesside University, UK

Honorary Chairs

Plamen Angelov
Vera Kurkova
Leon Bobrowski

Lancaster University, UK
Czech Academy of Sciences, Czech Republic
Bialystok University of Technology, Poland

Organizing Co-chairs

Antonios Papaleonidas
Andreas Andreou
Efthymoulos Kyriacou
Harris Papadopoulos
Anastasios Panagiotis Psathas

Democritus University of Thrace, Greece
Cyprus University of Technology, Cyprus
Cyprus University of Technology, Cyprus
Frederick University, Cyprus
Democritus University of Thrace, Greece

Advisory Co-chairs

George Magoulas	Birkbeck, University of London, UK
Nikolaos Passalis	Aristotle University of Thessaloniki, Greece
Antonio Moran	University of León, Spain
Serafin Alonso	Universidad de León, Spain
Paulo Cortez	University of Minho, Portugal
Petia Koprinkova-Hristova	Bulgarian Academy of Sciences, Bulgaria

Doctoral Consortium Co-chairs

Anastasios Panagiotis Psathas	Democritus University of Thrace, Greece
-------------------------------	---

Publication and Publicity Co-chairs

Antonios Papaleonidas	Democritus University of Thrace, Greece
Anastasios Panagiotis Psathas	Democritus University of Thrace, Greece

Liaison Co-chairs

George A. Papakostas	Democritus University of Thrace, Greece
Ioannis Chocliouros	Hellenic Telecommunication Organization, Greece

Workshops Co-chairs

Peter Hajek	University of Pardubice, Czech Republic
Spyros Sioutas	University of Patras, Greece
Christos Makris	University of Patras, Greece
Ioannis Karydis	Ionian University, Greece

Special Sessions and Tutorials Co-chairs

Andreas S. Andreou	Cyprus University of Technology, Cyprus
Harris Papadopoulos	Frederick University, Cyprus
Efhtyvoulos Kyriacou	Cyprus University of Technology, Cyprus

Local Organizing/Hybrid Facilitation Committee

Aggeliki Tsouka	Democritus University of Thrace, Greece
Anastasios Panagiotis Psathas	Democritus University of Thrace, Greece
Anna Karagianni	Democritus University of Thrace, Greece
Athanasios Kalantzis	Democritus University of Thrace, Greece
Athanasios Saliakouras	Democritus University of Thrace, Greece
Christina Gkizioti	Democritus University of Thrace, Greece
Constantinos Stylianou	Cyprus University of Technology, Cyprus
Devis Beshello	Democritus University of Thrace, Greece
George Savva	Cyprus University of Technology, Cyprus
Iasonas Konstantinos Kazakos	University of Piraeus, Greece
Ioanna-Maria Erentzi	Democritus University of Thrace, Greece
Leonidas Skopelitis	Democritus University of Thrace, Greece
Michalis Gemenaris	Cyprus University of Technology, Cyprus
Nikos Zervis	Democritus University of Thrace, Greece
Panagiotis Restos	Democritus University of Thrace, Greece
Stelios Mappouras	Cyprus University of Technology, Cyprus
Tassos Giannakopoulos	Democritus University of Thrace, Greece
Theodoros Koutrakis	Democritus University of Thrace, Greece
Vasileios Kotsaris	Democritus University of Thrace, Greece

Program Committee

Abraham Yosipof	International Institute for Applied Systems Analysis, Austria
Aditya Gupta	Amazon, USA
Afef Mdhaftar	University of Sfax, Tunisia
Akrivi Krouska	University of West Attica, Greece
Alaa Marouf	Hosei University, Japan
Alaa Mohasseb	University of Portsmouth, UK
Alexander Ryjov	Lomonosov Moscow State University, Russia
Alexandros Kostopoulos	Hellenic Telecommunications Organization, Greece
Alexandros Vrochidis	Center for Research and Technology Hellas, Greece
Alexios Papaioannou	Democritus University of Thrace, Greece
Aliki Stefanopoulou	Centre for Research and Technology Hellas, Greece
Ameen Abdel Hai	Temple University, USA

Anastasia Spiliopoulou	Hellenic Telecommunications Organization S.A., Greece
Anastasios Panagiotis Psathas	Democritus University of Thrace, Greece
Andre de Carvalho	Universidade de São Paulo, Brazil
Andreas Andreou	Cyprus University of Technology, Cyprus
Andreas Kanavos	Ionian University, Greece
Andreas Konstantinidis	Frederick University, Cyprus
Andreas Menychtas	University of Piraeus, Greece
Andreea Udrea	University Politehnica of Bucharest, Romania
Andres Iglesias	University of Cantabria, Spain
Angel Lareo	Universidad Autónoma de Madrid, Spain
Antonio Morán	Universidad de León, Spain
António Pereira	Polytechnic University of Leiria, Portugal
Antonios Billis	Aristotle University of Thessaloniki, Greece
Antonios Kalampakas	American University of the Middle East, Kuwait
Antonios Papaleonidas	Democritus University of Thrace, Greece
Aristidis Likas	University of Ioannina, Greece
Arnaud Lewandowski	Université du Littoral Côte d'Opale, France
Asfak Ali	Jadavpur University, India
Asimina Dimara	Centre for Research and Technology Hellas, Greece
Asma Mekki	University of Sfax, Tunisia
Athanasios Kallipolitis	University of Piraeus, Greece
Athanasios Koutras	University of the Peloponnese, Greece
Athanasios Tsadiras	Aristotle University of Thessaloniki, Greece
Balakrishnan Natarajan	Sona College of Technology, India
Banafsheh Rekabdar	Portland State University, USA
Bharathi R.	People's Education Society University, India
Binyu Zhao	Heilongjiang University, China
Bo Mei	Texas Christian University, USA
Boudjelal Meftah	Mustapha Stambouli University of Mascara, Algeria
Bujar Raufi	Technological University Dublin, Ireland
Cen Wan, Birkbeck	University of London, UK
Charalambos Eliades	Frederick University, Cyprus
Charalampos Liapis	University of Patras, Computer Technology Institute and Press "Diophantus", Greece
Cheng Shang	Northwest A&F University, China
Chin Cheng Siong	Newcastle University, UK
Christina Plomariti	Aristotle University of Thessaloniki, Greece
Christoforos Papaioannou	Democritus University of Thrace, Greece
Christos Loizou	Cyprus University of Technology, Cyprus

Christos Makris	University of Patras, Greece
Christos Troussas	University of West Attica, Greece
Chuqing Zhao	Harvard University, USA
Claudio Giancaterino	Towards Innovation Lab, Italy
Cunjian Chen	Monash University, Australia
Cyril Drocourt	University of Picardy Jules Verne, France
Damien Lolive	University of Southern Brittany, France
Daniel Perez	Universidad de León, Spain
Daniel Stamate	Goldsmiths, University of London, UK
Daniel Vařata	Czech Technical University in Prague, Czechia
Dario Malchiodi	Università degli Studi di Milano, Italy
David Dembinsky	German Research Center for Artificial Intelligence, Germany
Denise Gorse	University College London, UK
Dennis Wong	Macao Polytechnic University, China
Despoina Mantziari	Aristotle University of Thessaloniki, Greece
Despoina Mouratidis	Ionian University, Greece
Diana-Laura Borza	Babeř-Bolyai University, Romania
Diego Garcıa	University of Oviedo, Spain
Dimitrios Michail	Harokopio University of Athens, Greece
Dionysios Koulouris	University of Piraeus, Greece
Doina Logofatu	Frankfurt University of Applied Sciences, Germany
Dominik Geissler	American Family Insurance, USA
DongNyeong Heo	Handong Global University, South Korea
Dulani Meedeniya	University of Moratuwa, Sri Lanka
Dumitru-Clementin Cercel	National University of Science and Technology Politehnica Bucharest, Romania
Dustin van der Haar	University of Johannesburg, South Africa
Edoardo Peretti	Politecnico di Milano, Italy
Efstathia Martinopoulou	Information Technologies Institute, Greece
Efstratios Georgopoulos	University of the Peloponnese, Greece
Efthymoulos Kyriacou	Cyprus University of Technology, Cyprus
Eirini Mathe	University of West Attica, Greece
Eirini Vasilaki	Hellenic Telecommunications Organisation S.A., Greece
Eleana Kafeza	Technology Innovation Institute, United Arab Emirates
Eleni Gerasimidi	Democritus University of Thrace, Greece
Eleni Vlachou	Ionian University, Greece
Elias Pimenidis	University of the West of England, UK
Ella Haig	University of Portsmouth, UK

Emanuele Lattanzi	University of Urbino, Italy
Emmanouil Rigas	Aristotle University of Thessaloniki, Greece
Evaggelia Pitoura	University of Ioannina, Greece
Fabrizio Marozzo	Università della Calabria, Italy
Faisal Alvi	Habib University, Pakistan
Farhin Farhad Riya	University of Tennessee, USA
Farzad Vazinram	University of Twente, Netherlands
Federico Corradi	Eindhoven University of Technology, Netherlands
Federico Vozzi	Consiglio Nazionale delle Ricerche, Italy
Fei Zuo	University of Central Oklahoma, USA
Flora Ferreira	University of Minho, Portugal
Florian Mirus	Autonomous Reply, Germany
Florin Leon	Gheorghe Asachi Technical University of Iași, Romania
Foteini Dervisi	University of Edinburgh, UK
Fotios Kounelis	Imperial College London, UK
Francesco Marcelloni	University of Pisa, Italy
Francesco Trovò	Politecnico di Milano, Italy
Franck Gechter	University of Johannesburg, South Africa
Friedhelm Schwenker	Ulm University, Germany
Gaetano Di Caterina	University of Strathclyde, UK
George Anastassopoulos	Democritus University of Thrace, Greece
George Magoulas	Birkbeck, University of London, UK
George Manias	University of Piraeus, Greece
George Papakostas	Democritus University of Thrace, Greece
Georgia Liapi	Cyprus University of Technology, Cyprus
Georgios Alexandridis	National & Kapodistrian University of Athens, Greece
Georgios Drakopoulos	University of West Attica, Greece
Georgios Vardakas	University of Ioannina, Greece
Gerasimos Vonitsanos	University of Patras, Greece
Giacomo Boracchi	Politecnico di Milano, Italy
Gregory Gasteratos	Ionian University, Greece
Guillem Rodríguez	Artificial Intelligence Research Institute, Spain
Hanae Belmajdoub	Mohammed V University in Rabat, Morocco
Hanae Belmajdoub	Mohammed V University in Rabat, Morocco
Haopeng Chen	Shanghai Jiao Tong University, China
Harris Papadopoulos	Frederick University, Cyprus
Hemn Abdalla	Wenzhou-Kean University, China
Honggang Zhang	University of Massachusetts Boston, USA
Hugo Camacho-Cruz	Universidad Autónoma de Tamaulipas, Mexico
Hussain Otudi	Temple University, USA

Ieswaria Sreedharan	Frederick Research Center, Cyprus
Ignacio Diaz-Blanco	Universidad de Oviedo, Spain
Ilias Maglogiannis	University of Piraeus, Greece
Ioannis Chamodrakas	National and Kapodistrian University of Athens, Greece
Ioannis Chochliouros	Hellenic Telecommunications Organisation S.A., Greece
Ioannis Hatzilygeroudis	University of Patras, Greece
Ioannis Karamitsos	Rochester Institute of Technology Dubai, United Arab Emirates
Ioannis Karampinis	Democritus University of Thrace, Greece
Ioannis Karydis	Ionian University, Greece
Ioannis Livieris	University of Piraeus, Greece
Ioannis Papaioannou	Centre for Research and Technology Hellas, Greece
Ioannis Pastaltzidis	Information Technologies Institute, Greece
Ioannis Tzitzios	Democritus University of Thrace, Greece
Ivo Bukovsky	University of South Bohemia in České Budějovice, Czechia
Jacky Mallett	Reykjavik University, Iceland
Jacopo Rossi	Sapienza University of Rome, Italy
Jan Treur	Vrije Universiteit Amsterdam, Netherlands
Javier Naranjo-Alcazar	Instituto Tecnológico de Informática, Spain
Jérémie Sublime	International Student Exchange Programs, France
Jingwen Fu	Xi'an Jiaotong University, China
Jinling Wang	Ulster University, UK
Jinpeng Dong	Xi'an Jiaotong University, China
Jinyuan Sun	University of Tennessee, USA
Jiri Sima	Institute of Computer Science of the Czech Academy of Sciences, Czechia
Jitesh Dundas	Independent Researcher, USA
João Ramos	Polytechnic University of Leiria, Portugal
Joost Vennekens	Vrije Universiteit Brussel, Belgium
Jordi Cosp-Vilella	Universitat Politècnica de Catalunya, Spain
Josua Spisak	University of Hamburg, Germany
Juan Fuertes	Universidad de León, Spain
Jumanah Alshehri	Imam Abdulrahman Bin Faisal University, Saudi Arabia
Kamil Dedecius	Czech Technical University in Prague, Czechia
Katia Kermanidis	Ionian University, Greece
Ken-ichi Fukui	Kansai University, Japan
Khalid Minaoui	Mohammed V University in Rabat, Morocco
Khoa Phung	University of the West of England, UK

Kiran Lekkala	University of Southern California, USA
Koloud Alkhamaiseh	Western Michigan University, USA
Konstantinos Delibasis	University of Thessaly, Greece
Konstantinos Giannakis	University of Bergen, Norway
Kostadin Cvejovski	Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme, Germany
Kostas Karatzas	Aristotle University of Thessaloniki, Greece
Kostas Katsios	Frederick University, Cyprus
Kostas Moutselos	University of Piraeus, Greece
Krati Saxena	Kyushu Institute of Technology, Japan
Kruthika K. R.	SandLogic Technologies, India
Kun Zhang	National Institute for Research in Digital Science and Technology, France
Kyriaki Fanaridou	Centre for Research and Technology Hellas, Greece
Lazaros Iliadis	Democritus University of Thrace, Greece
Leiping Jie	Guangdong Ocean University, China
Leo Thomas Ramos	Universitat Autònoma de Barcelona, Spain
Lyra Puspa	Vanaya NeuroLab, Indonesia
Madalina Erascu	West University of Timisoara, Romania
Magda Friedjungová	Czech Technical University in Prague, Czechia
Marcello Trovati	Edge Hill University, UK
Marcin Pietron	AGH University of Science and Technology, Poland
Maria Belesioti	Hellenic Telecommunications Organisation, Greece
Marian Cristian Mihaescu	University of Craiova, Romania
Markus Aho	University of Eastern Finland, Finland
Matthew Evanusa	University of Maryland College Park, USA
Mayukh Maitra	Walmart, USA
Md Saef Ullah Miah	American International University - Bangladesh, Bangladesh
Michail-Antisthenis Tsompanas	University of the West of England, UK
Michalis Gemenaris	Cyprus University of Technology, Cyprus
Michalis Rantopoulos	Hellenic Telecommunications Organisation, Greece
Michel Salomon	Université Bourgogne Franche-Comté, France
Miguel Prada	Universidad de León, Spain
Mirjana Ivanovic	University of Novi Sad, Serbia
Mohamed Ben Aouicha	University of Sfax, Tunisia
Mohamed Elleuch	University of Manouba, Tunisia
Mohammed Khennour	Kasdi Merbah University of Ouargla, Algeria

Mohib Ullah	Norwegian University of Science and Technology, Norway
Nafiz Fahad	Multimedia University, Malaysia
Nanning Zheng	Xi'an Jiaotong University, China
Naushin Nower	University of Dhaka, Bangladesh
Nermeen Abou Baker	Hochschule Ruhr West, Germany
Nichat Kiourt	Democritus University of Thrace, Greece
Nikolaos Kladovasilakis	Information Technologies Institute, Greece
Nikolaos Mitianoudis	Democritus University of Thrace, Greece
Nikos Kanakaris	University of Southern California, USA
Nikos Karacapilidis	University of Patras, Greece
Noémi Gaskó	Babeş-Bolyai University, Romania
Pabitra Mitra	Indian Institute of Technology Kharagpur, India
Panagiotis Gratsanis	Ionian University, Greece
Panagiotis Pintelas	University of Patras, Greece
Panagiotis Theocharopoulos	University of Thessaly, Greece
Panthadeep Bhattacharjee	National Institute of Technology Rourkela, India
Paul Spirakis	University of Liverpool, UK
Paul Stefan Popescu	University of Craiova, Romania
Paulo Cortez	University of Minho, Portugal
Pavel Petrović	Comenius University Bratislava, Slovakia
Pedro Zuccarello	Instituto Tecnológico de Informática, Spain
Perikos Isidoros	University of Patras, Greece
Petia Koprinkova-Hristova	Institute of Information and Communication Technologies, Bulgaria
Petr Hajek	University of Pardubice, Czechia
Petros Tzallas	Centre for Research and Technology Hellas, Greece
Phivos Mylonas	University of West Attica, Greece
Ram Sarkar	Jadavpur University, India
Rashmi Gupta	Kristiania University of Applied Sciences, Norway
Ricardo Marcacini	University of São Paulo, Brazil
Riccardo Renzulli	University of Turin, Italy
Rodrigo Clemente Thom de Souza	Federal University of Paraná, Brazil
Rodrique Kafando	Virtual University of Burkina Faso, Burkina Faso
Roman Mouček	University of West Bohemia, Czechia
Sahar Vahdati	Leibniz University Hannover and TIB, Germany
Salvatore Aiello	Politecnico di Torino, Italy
Sanchit Hira	Independent, USA
Sandrine Mouysset	Institut de Recherche en Informatique de Toulouse, France

Sanping Zhou	Xi'an Jiaotong University, China
Seiya Satoh	Tokyo Denki University, Japan
Selvine George Mathias	Technische Hochschule Ingolstadt, Germany
Serafin Alonso	Universidad de León, Spain
Sergey Dolenko	M.V. Lomonosov Moscow State University, Russia
Sheikh Sharfuddin Mim	Frankfurt University of Applied Sciences, Germany
Shelly Gupta	Temple University, USA
Shoumeng Qiu	Fudan University, China
Shylaja S. S.	People's Education Society University, India
Sie Long Kek	Universiti Tun Hussein Onn Malaysia, Malaysia
Song Guo	Xi'an University of Architecture and Technology, China
Sotiris Kotsiantis	University of Patras, Greece
Spyridon Tzimiris	Ionian University, Greece
Spyros Sioutas	University of Patras, Greece
Stefanos Kollias	University of Lincoln, UK
Stefanos Nikiforos	Ionian University, Greece
Stelios Krinidis	Democritus University of Thrace, Greece
Stelios Mappouras	Cyprus University of Technology, Cyprus
Takuya Kitamura	National Institute of Technology, Toyama College, Japan
Thanasis Moustakas	Centre for Research and Technology Hellas, Greece
Theodor Panagiotakopoulos	Hellenic Open University, Greece
Thomas Schuster	Pforzheim University, Germany
Thomas Sgourous	Centre for Research and Technology Hellas, Greece
Tiyu Fang	University of Jinan, China
Tomasz Kapuscinski	Rzeszow University of Technology, Poland
Uwe Handmann	Hochschule Ruhr West, Germany
V. Ramasubramanian	International Institute of Information Technology Bangalore, India
Vandana Ladwani	International Institute of Information Technology Bangalore, India
Vangelis Metsis	Texas State University, USA
Veljko Kovač	University of Amsterdam, Netherlands
Verner Ferreira	Universidade Federal do Rio Grande do Norte, Brazil
Vijeta Sharma	Norwegian University of Science and Technology, Norway
Vikramaditya Jakkula	Data Mining Tools Inc., USA

Vilson DalleMole	Universidade Tecnológica Federal do Paraná, Brazil
Weifeng Liu	China University of Petroleum, China
Wenxin Yu	Southwest University of Science and Technology, China
Xia Feng	Civil Aviation University of China, China
Xingpeng Zhang	Southwest Petroleum University, China
Xuewen Wang	China University of Geosciences, Wuhan, China
Yahong Lian	Nankai University, China
Yang Shao	Hitachi, Ltd., Japan
Yisong Chen	Georgia Institute of Technology, USA
Yuhao Huang	Xi'an Jiaotong University, China
Zhaoyun Ding	National University of Defense Technology, China
Zhichao Lian	Universidad de León, Spain
Ziad Doughan	Beirut Arab University, Lebanon
Zoran Obradovic	Temple University, USA

Keynote Lectures

Connecting AI to the Cyber-Physical World

Marios M. Polycarpou

Department of Electrical and Computer Engineering, University of Cyprus

Abstract. The development of cyber-physical systems with multiple sensor/actuator components and feedback loops has given rise to advanced automation applications, including energy and power, intelligent transportation, water systems, manufacturing, etc. Traditionally, feedback control has focused on enhancing the tracking and robustness performance of the closed-loop system; however, as cyber-physical systems become more complex and interconnected and more interdependent, there is a need to refocus our attention not only on performance but also on the resilience of cyber-physical systems. In situations of unexpected events and faults, computational intelligence can play a key role in improving the fault tolerance of cyber-physical systems and preventing serious degradation or a catastrophic system failure. The goal of this presentation is to provide insight into the design and analysis of intelligent monitoring methods for cyber-physical systems, which will ultimately lead to more resilient societies.

Engineering Information Systems and Smart Applications with LLMs

Massimo Mecella

School of Information Engineering, Sapienza Università di Roma, Italy

Abstract. The current evolution of AI, and of Generative AI in particular, namely Large Language Models (LLMs), makes it possible to adopt them as supporting tools for the engineering of information systems, in particular for their design and development based on service-based approaches and business-process orientation. In this keynote we will present, with an IS engineering attitude, recent approaches and applications for LLMs' usage during the design of information systems. Very recent works (in the last year) are emerging on how to use ChatGPT (one of the most widespread applications of LLMs) for the conceptual design, software development, and extraction of business process specifications from documents. But all of them are quite fragmented, and a unifying framework and pipeline is missing. We will also provide such a unifying principled view. Case studies based on the presenters' research activities will be presented, and a systematic analysis of the literature and practice will be presented as well.

Collectionless AI and Nature-Inspired Learning

Marco Gori

University of Siena, Italy

Abstract. AI is revolutionizing not only the entire field of Computer Science, but nearly all fields of Science. However, while application contexts explode and LLMs display embarrassing cognitive qualities, the entire AI research field seems headed toward saturation of the fundamental ideas that have enabled today’s spectacular results of large companies. Is the infamous “AI winter” perhaps creeping into research? In this talk I argue that the time is ripe for a fundamental rethinking of AI methodologies with the purpose to migrate intelligence from the cloud to the growing global population of devices with on-board CPUs.

From Language Models to Autonomous Agents: Evolutionary Trajectories and Industrial Transformations in AI-Driven Systems

John Soldatos

University of Glasgow, UK

Abstract. The rapid evolution of Large Language Models (LLMs) and AI agents is currently redefining the boundaries of AI: It enables unprecedented automation and decision-making capabilities in complex domains. This talk examines the architectural and algorithmic milestones in LLM development, from transformer-based foundations to today's multi-agent frameworks which are capable of collaborative reasoning.

Drawing on case studies in finance (e.g., real-time risk assessment, stock selection) and industrial automation (e.g., asset maintenance, industrial training, and human-robot collaboration), the talk highlights how LLMs transition from passive tools to proactive, context-aware agents. The practical contributions of the talk include novel architectures for solving complex problems with LLMs and AI agents, while considering the need for ethical governance frameworks to address bias, transparency, and accountability.

Contents – Part II

Machine Learning/Pattern Recognition

A Human-Centric Decision Support System for Zero-Defect Manufacturing Enabled by Human-in-the-Loop Learning	3
<i>Andreana Mitsiaki, Christina Tsita, Nikolaos Dimitriou, and Dimitrios Tzovaras</i>	
A Norm-Chatbot: Local LLM Vision with Vision-Based RAG for Complex Production Documents and Task-Specific Responses	17
<i>Michael Sahl Lystbæk and Ulrik Sahl Lystbæk</i>	
Artificial Intelligence Versus Food Wastage in Bakeries	31
<i>Elias Pimenidis, Yahia Abdelrahman, and Anastasios Panagiotis Psathas</i>	
Assistive X for Amyotrophic Lateral Sclerosis (ALS)	43
<i>Radhika Khatri, Samin Rizwan, Faryal Khan, Sudais Yasin, Faisal Alvi, and Qasim Pasta</i>	
Autonomous Navigation in Swarm of UAVs Using Spatio Temporal Data and Constrained-Reinforcement Learning	54
<i>Abhudaya Shrivastava, Christos Petridis, Marijana Vacic, and Zoran Obradovic</i>	
Data-Driven BLEVE Overpressure Prediction Using Explainable Machine Learning	68
<i>Zifan Shi, Qilin Li, Ling Li, Yanda Shao, and Hong Hao</i>	
ESN Architectures for Industrial Process Modelling to Develop Digital Twins	81
<i>Raúl González-Herbón, Serafín Alonso, Miguel A. Prada, Ignacio Díaz, Manuel Domínguez, and Juan J. Fuertes</i>	
Harnessing Machine Learning for Rain Induced Landslide Detection and Analysis	94
<i>Shelly Gupta, Hussain Otudi, Ameen Abdel Hai, Rafaa Aljurbua, Jovan Andjelkovic, Abdulrahman Alharbi, and Zoran Obradovic</i>	

Leveraging Computational Geometry for Data Augmentation in Medical Flow Fields Classification	109
<i>Riccardo Margheritti, Andrea Schillaci, Carlotta Pipolo, Maurizio Quadrio, and Giacomo Boracchi</i>	
Machine Learning Analysis of Dissolved Oxygen at the “G. Giannouli” Hydrological Station Greece	123
<i>Nichat Kiourt, Lazaros Iliadis, Chsristos Akratos, and Antonis Papaleonidas</i>	
Privacy-Preserving and Personalized AI Modules for E-Commerce Platforms	137
<i>Arantzazu Florez Tapia, Izar Azpiroz, Ane M. Florez-Tapia, Amaia Gil, Elena Zotova, Roger Solsona, Igor García Olaizola, and Marco Quartulli</i>	
Quench Detection and Localization via Interpretable Machine Learning	151
<i>Alessandro Biagiotti, Dario Malchiodi, Samuele Mariotto, and Lucio Rossi</i>	
Spatiotemporal Multiplex Network Model for Predicting Forced Outage Severity in Distribution Grids	165
<i>Rafaa Aljurbua, Rashid Baembitov, Christos Petridis, Daniel Saranovic, Mladen Kezunovic, and Zoran Obradovic</i>	
Machine Learning/Time Series	
AI-Based Water Quality Monitoring Module for the Segura Hydrographic Confederation Platform	181
<i>Izar Azpiroz, David Velásquez, Breno da Costa Paulo, Iker Landa del Barrio, Juan Odriozola, and Mikel Maiza</i>	
An Integrated Approach for Short-Term Forecasting of Highway Vehicle Flows Based on Singular Spectrum Analysis and Artificial Neural Networks ...	195
<i>Nikiforos Botzoris, Anastasios Panagiotis Psathas, Antonios Papaleonidas, and Lazaros Iliadis</i>	
Assessing Mobility Policies by Traffic Simulation and Change Detection	210
<i>Felipe Bagni, Edoardo Peretti, Danilo Macció, Cristiano Cervellera, and Giacomo Boracchi</i>	
Automatic Long-Term Forecasting of Mortality Rates with Generalized Regression Neural Networks	224
<i>Giovanna Apicella, Michele La Rocca, Cira Perna, and Marilena Sibillo</i>	

Contextualized Segmentation of Milling Processes Using Discrete Rule-Based Pattern Recognition	238
<i>Lukas Klehr, Bastian Engelmann, Frank-Michael Schleif, and Daniel Regulin</i>	
Estimation of River Ebro Streamflow Using Fuzzy Inference Systems and FCM	255
<i>Anastasios Panagiotis Psathas, Stamatios Petkopoulos, Antonios Papaleonidas, Luis Garrote, Mike Spiliotis, and Lazaros Iliadis</i>	
Forecasting vehicle crossing volumes by using Nonlinear Autoregressive Neural Networks sets	269
<i>Ioannis Skopelitis, Antonios Papaleonidas, Anastasios Panagiotis Psathas, Lazaros Iliadis, and George N. Botzoris</i>	
Retrospective Clustering of COVID-19 Mortality Time Series Using Dynamic Time Warping	285
<i>Murat Razi and Manuel Graña</i>	
Author Index	299

Contents – Part I

Deep Learning/Classification

A Real-Time Human Action Recognition Model for Assisted Living	3
<i>Yixuan Wang, Cristina Muntean, Pramod Pathak, and Paul Stynes</i>	
A Survey of Federated Learning-Based Intrusion Detection Methods in Medical IoT	17
<i>Ahlem Harhad, David Durand, Cyril Drocourt, and Gil Utard</i>	
AI-Based Automatic Counting and Classification of <i>aedes</i> Mosquito Eggs In Field Traps	31
<i>Jordi Grau-Haro, Javier Naranjo-Alcazar, Jesus Lopez-Ballester, David Almenar, and Pedro Zuccarello</i>	
An Empirical Review of Uncertainty Estimation for Quality Control in CAD Model Segmentation	45
<i>Gerico Vidanes, David Toal, Andy Keane, Daniel Xu Zhang, Marco Nunez, and Jon Gregory</i>	
Brain Inspired Learning for Neural Networks	59
<i>Jayani Hewavitharana, Amida Anand, Peter Giese, Carolina Moretti Ierardi, and Kathleen Steinhofel</i>	
Comparative Analysis of Machine Learning Techniques for Chronic Kidney Disease Prediction Efficiency	73
<i>Md. Rabiul Islam and Doina Logofătu</i>	
Detecting Anomalous Self-citations Using Citation Network Analysis and LLMs	87
<i>Farhaan Ebadulla, B. V. Gaurav, H. Manoj, Arti Arya, Nazmin Begum, and K. Kruthik</i>	
Early Detection of Voice Pathology from Cry Analysis Using Non-interpretable Features and Parallel 1D CNN	100
<i>Mouad El Omari, Younes El Belghiti, Hanae Belmajdoub, Khalid Minaoui, and Samir Saoudi</i>	
Easy, Fast and Reliable Modulo and Linear Congruential Generator Approximation with Artificial Neural Networks	112
<i>Sara Boancă</i>	

EEG-Based Hybrid Emotion Recognition Model with Statistical-Wavelet Features and Modality-Agnostic Loss	128
<i>Asfak Ali, Jotiraditya Banerjee, Debam Saha, Akash Dutta, Friedhelm Schwenker, and Ram Sarkar</i>	
ESN with Delayed Inputs to Model Industrial Processes	141
<i>José Ramón Rodríguez-Ossorio, Antonio Morán, Juan J. Fuertes, Claudio Gallicchio, Lidia Roca, and Manuel Domínguez</i>	
Exploring Knowledge Distillation for Model Compression in Edge Environments	154
<i>Telma Garção, David Belo, Joana Sousa, and João Ferreira</i>	
Exploring Various Sequential Learning Methods for Deformation History Modeling	168
<i>Muhammed Adil Yarkin, Mihkel Korgesaar, Jani Romanoff, Joshua Stuckner, Ümit Işlak, and Hasan Kurban</i>	
FusionNet: Leveraging Dual Speech Separation Networks for Enhanced Multi-speaker Isolation	181
<i>Sumedh Ravindran, Shreyas Malleash, Raghavendra S. Bhatagunaki, Srikrishna R. Chitnis, and Shylaja S. Sharath</i>	
Hybrid Deep Learning and Gradient Boosting for Superior Sentiment Analysis: A Comparative Study	195
<i>Saddam Hossain and Doina Logofătu</i>	
Implementing Hybrid Tsetlin Machine and Q-Learning for Solving the Job Shop Scheduling	209
<i>Reshma Maharjan, Per-Arne Andersen, and Lei Jiao</i>	
Maximum Interstory Drift Ratio (MIDR) Equations for R/C Buildings Using Machine Learning Procedures	225
<i>Ioannis Karampinis, Konstantinos Morfidis, Konstantinos Kostinakis, and Lazaros Iliadis</i>	
MRI-Based Brain Tumor Classification Using Ensemble CNN, VGG16, and ResNet50 Model	240
<i>Anshuli Chikhale and Divyam Kakani</i>	
Needle-in-the-Haystack Testing LLMs with a Complex Reasoning Task	254
<i>Thomas Schuster, Marian Lambert, Nico Döring, and Julius Trögele</i>	

Utilizing Multiple Data Sources to Improve Prediction of Severe Weather
Events Through Spatio-Temporal Analysis 267
 *Hussain Otudi, Shelly Gupta, Ameen Abdel Hai, Rafaa Aljurbua,
 Abdulrahman Alharbi, and Zoran Obradovic*

Author Index 281

Machine Learning/Pattern Recognition



A Human-Centric Decision Support System for Zero-Defect Manufacturing Enabled by Human-in-the-Loop Learning

Andreana Mitsiaki^(✉) , Christina Tsita , Nikolaos Dimitriou ,
and Dimitrios Tzovaras 

Information Technologies Institute, Centre for Research and Technology Hellas,
57001 Thessaloniki, Greece
amitsiaki@iti.gr

Abstract. This paper presents a Decision Support System (DSS) for Zero Defect Manufacturing (ZDM), deployed in three factories. The DSS comprises modular components that enhance decision-making through advanced visualization for monitoring, prediction, analysis, and defect detection. Its design prioritizes usability, offering comprehensive visual analytics, easy management of connected devices, human-Artificial Intelligence (AI) interaction, and interface expandability. The DSS captures product images using RGB cameras, which are analyzed by AI algorithms to detect defects. Operators can review the detection results through a user-friendly web interface and provide feedback. The significant novelty of the system lies in its application of human-in-the-loop (HITL) Learning, where human feedback is used to retrain the AI models, progressively improving the accuracy of the defect detection, as demonstrated in real-world experiments. At the same time, operators learn from their interactions with the AI, while their empirical knowledge is complemented by real-time informed data. The system fosters a synergistic collaboration between human operators and AI, reducing defective products and inspection time. In one factory, the DSS led to 90% reduction in product inspection time, enabling operators to complete inspections faster and focus on other production tasks. End-user evaluations across the three factories showed a 93% perceived ease of use and a 90% intention to use. These results highlight the system's effectiveness in advancing digital transformation in quality control, aligning with Industry 4.0/5.0 objectives.

Keywords: defect detection · decision support · artificial intelligence · human-in-the-loop Learning

1 Introduction

“Industry 4.0” refers to the fourth industrial revolution, which integrates intelligent digital technologies into production lines and has become the main standard for the manufacturing sector’s digital transformation [1]. The term “Industry 5.0” emerged later to emphasize the human-centric perspective in digital transformation [2, 3]. According to

Platform Industrie 4.0, “Industry 4.0” already incorporates sustainability goals, including workforce well-being, through human-centric approaches. Like previous industrial revolutions, the fourth revolution will require considerable time to fully mature before transitioning to the next phase [3]. Although technology is advancing rapidly, it represents only one aspect of the broader digital transformation, which also involves people and processes. Meanwhile, the term “Industry 5.0” is frequently used in the literature. Regardless of which terminology is considered most appropriate, a strategic goal of the manufacturing sector is the creation of ecosystems that are human-centric, eco-friendly, and resilient, with the ability to adapt to production disruptions [1, 2]. A key element in this endeavor is communication among humans, machines, and products throughout the manufacturing process [4].

AI and Internet of Things (IoT) are core technologies in industrial automation and the smart factory concept [5], playing a critical role in realizing this vision. Central to this is the integration of ZDM into quality control processes, with a focus on early defect detection and corrective action [6]. ZDM consists of four main strategies: detect, predict, prevent, and repair, used in combination to ensure quality from the outset by leveraging data to predict and prevent defects [7]. It supports “Right-First-Time” (RFT) manufacturing by minimizing resource waste and shifting from traditional off-line sample measurements to in-line data collection during production. This transition improves manufacturing efficiency, accuracy and quality, while reducing defects [8]. IoT, supported by AI and operator feedback, further enhances production yield and efficiency. DSS combines these elements, creating an integrated framework to optimize manufacturing operations [9].

The OPTIMAI DSS [10] was developed to meet the requirements of ZDM within the Industry 4.0/5.0 framework, aiming to reduce defects, enhance product quality, and minimize resource waste, while supporting human operators. The system leverages AI to predict and prevent defects before they occur, and to enable early detection of faulty products. This prevents defect propagation along the production line, reducing the waste of raw materials and energy. When a defect is detected, the system can notify the operator for manual intervention or automatically execute the necessary corrective actions. Additionally, if a defective product cannot be repaired, it can be redirected to the digital marketplace, allowing part of its value to be salvaged.

A key innovation of the DSS is the integration of the HITL Learning approach into the quality control process. In this approach, human operators provide feedback to the AI models, allowing them to retrain and improve over time. Unlike traditional active learning, where models select uncertain or informative instances for labeling, this method involves human experts actively reviewing AI predictions, validating results, and offering corrective feedback. This ensures continuous system improvement and adaption to new data and production conditions. This synergy between human expertise and data-driven insights enhances decision-making in manufacturing. The system has been deployed and tested in three pilot production sites, tailored to the specific needs of each manufacturing line. Collected data is analyzed and visualized through interactive DSS interfaces, enabling users to engage with the system and refine its performance. DSS implementation has significantly reduced quality control time while achieving high operator acceptance, demonstrating its practical value in smart manufacturing. The

next section presents related work, followed by the proposed DSS interfaces, deployment methodology per pilot, the HITL Learning framework, and user evaluations and experiments. The final section provides the conclusions.

2 Related Work

In Industry 4.0, the Industrial Internet of Things (IIoT) enables efficient interaction among heterogeneous systems through event-driven frameworks, utilizing autonomous tools for data-driven decisions. According to [11], DSSs are classified based on their focus—data, communication, documents, and knowledge. Among these, knowledge-driven DSSs utilizing technologies such as data mining and AI to support autonomous decision-making. The same study highlights a hybrid DSS developed to automate the identification and rectification of problems in industrial production. To simulate production and evaluate its effectiveness, the system integrates both knowledge-based and data-driven approaches within a dynamic scheduling framework. In a real-world semiconductor application, the DSS outperformed previous decision-making strategies, significantly improving Key Performance Indicators (KPIs).

By improving IoT data processing, AI-based DSS streamline operations, facilitate predictive maintenance (PdM), and improve quality control. The work of [12] examines AI applications in production planning, energy management, and supply chain management, showing how DSS boost efficiency, reduce costs, and enhance dependability. It also highlights AI's potential for autonomous decision-making in industry and discusses implementation techniques and the challenges that may arise during deployment.

The IoT, Big Data, and Machine Learning (ML) are essential tools for intelligent production, PdM, and product quality improvement. A relevant publication [13] presents a DSS that addresses the challenge of annotating component wear data in a real industrial environment. The DSS is based on a feature extraction strategy and a ML model, integrated into a cloud architecture for data collection, analysis, and storage. The work of [14] explores an integrated DSS for achieving ZDM, combining production techniques and product design, with a focus on risk identification, prioritization, and management, reducing production time and cost, and improving decision-making efficiency.

Another research [15] proposes a dynamic scheduling tool with a DSS that utilizes ZDM strategies (detection, repair, prediction, and prevention) to minimize defects in industrial settings. The paper presented in [16] develops an extensible hybrid DSS to assist operators in decision-making regarding defective components, aiming to reduce defects at later stages without disrupting production. The system can expand to incorporate growing industrial data. The importance of the operator's role in the process is emphasized in [17], particularly in ultrasound testing, a common non-destructive method for flaw detection that relies on operator expertise. To support this, DSS have been developed to automate the process and provide valuable insights, assisting the operator in making more accurate and efficient decisions.

According to the above, DSS leverage modern technologies such as IoT, Big Data, and ML to automate processes and improve production, while incorporating ZDM strategies for detecting, predicting, and preventing defects. In the context of OPTIMAI, this approach was expanded by developing a hybrid DSS that integrates AI sensors into the

production line to collect real-time data and support operators in making decisions about defective products. Additionally, HITL Learning is employed, allowing operators to participate in training the AI model, fostering collaboration between humans and machines for optimal performance.

3 Decision Support System Interfaces

The OPTIMAI DSS consists of six modules, each with its respective interface: Monitoring, Prediction, Analysis, Defect detection, Event-Notifications, and Administration.

The “Monitoring” module provides real-time visualizations of production status, including outputs from the prediction and defect detection from AI models. These insights are displayed through interactive dashboards, enabling operators to make timely decisions and take preventive actions. Data from sensors, cameras, and edge devices are continuously aggregated and processed, offering up-to-date information that enhances responsiveness and reduces downtime.

The “Prediction” module has been developed for lift manufacturing to improve the calibration process of the lift’s power unit, and specifically the valve block that controls the vertical movement of the lift (Fig. 1). It uses a neural action-prediction model to simulate the weights’ motion based on the desired velocity, allowing qualitative evaluation. A virtual environment with generative AI converts 1D velocity sequences into 2D synthetic frames, enabling operators to visualize the anticipated motion of the weights before implementing adjustments, thus reducing trial-and-error iterations [18].

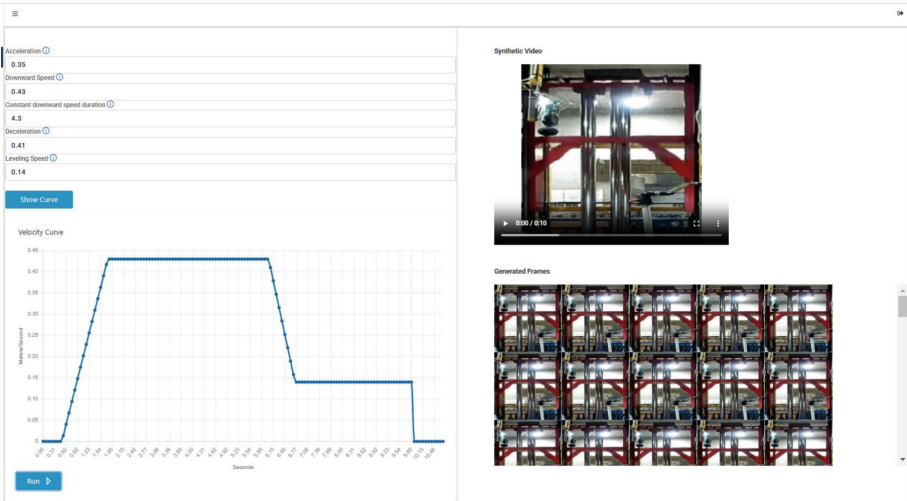


Fig. 1. Prediction simulates lift movement with synthetic frames and video.

The “Analysis” module refers to historical data to support long-term decision-making. It allows users to review the outcomes of the AI modules and the data collected during the selected timeframe to identify patterns and hypothesize potential causes of

defects or suboptimal operations. Depending on the pilot, this module includes graphs for defect detection, yield and the learning curve of AI defect detection models (explained in Sect. 5).

The “Defect Detection” module refers to the quality control interface that displays the output of the AI-based defect detection solutions in real time (Fig. 2). Each inspected product is analyzed by the underlying computer vision model, with results immediately visualized on the DSS interface. The system highlights detected anomalies, allowing operators to validate findings and, if necessary, initiate corrective actions or provide feedback to refine the detection models (details in Sects. 4, 5).

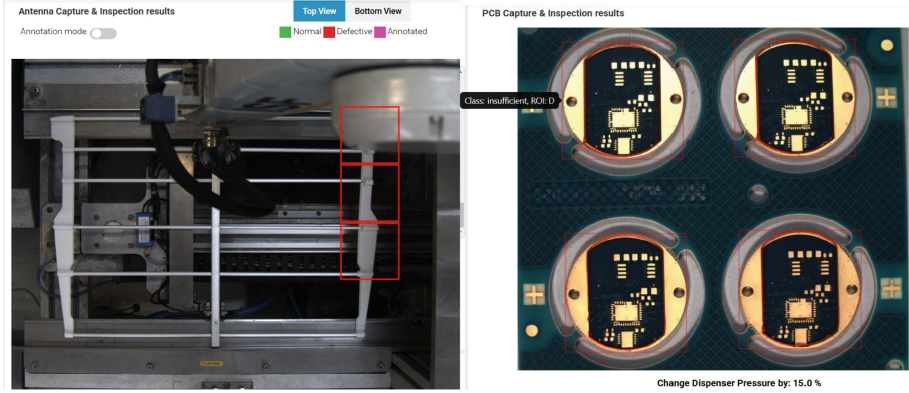


Fig. 2. Defect detection (a) in antenna and (b) PCB production lines

The purpose of the “Events-Notifications” module is to notify users of real-time events, such as system faults or defect discoveries, enabling swift responses and problem resolution. All notifications are stored, providing a historical record of events generated by external modules (e.g., defect detection, digital marketplace). This event history supports better decision-making for long-term process optimization.

The “Administration” interface manages DSS entities, including sensors, actuators, inspected products, monitored processes, and defect detection activities. The Blockchain module stores administrative actions executed by the users to ensure the immutability of critical data. These essential datasets are stored within Smart Contracts on distinct Private Ethereum networks, developed specifically for each pilot [19, 20].

4 Methodology

4.1 Pilots Brief Description

In antenna manufacturing, the applied solution enables automated defect detection in robotic cells, synchronized with the robots’ operation in mass production. This advancement allows real-time inspection and classification of the produced antennas, enabling operators to immediately identify defects, rather than relying on later manual checks that may miss the cause of the defect [19].

In PCB manufacturing, the system significantly improves quality control through AI. At the quality control workstation, the operator initiates the process, replacing the traditional microscope with the DSS interfaces. This reduces inspection time by approximately 20 min (depending on the PCB model), bringing it down to about 2 min (a 90% reduction), including the time needed for the operator to evaluate the AI results [21].

The lift manufacturing pilot focuses on quality control of the lift power unit and its hydraulic valve block, which controls the upward and downward movement of the produced lift. The operator monitors the valve block's performance in real-time through the DSS interface, enabling quick identification of suboptimal performance and on-site adjustments using a tablet. This reduces calibration time, as the operator no longer needs to repeatedly move between the site and the side office PC to observe the measurement graphs and calibrate the valves [18].

4.2 Methodology Implemented for Each of the Pilots

As presented in Fig. 3, various sensors and actuators were installed along the three production lines to collect real-time data. The DSS visualizes this data either by executing Python-based HTTP requests to a Middleware layer or through MQTT protocols. In the vision-based measurement systems, image outputs are analysed by AI models to detect defects. The results are sent to the Middleware and/or trigger automated actions in the production line. The DSS receives the data from Middleware and visualizes it across the various aforementioned modules (Sect. 3) according to each pilot's requirements. It supports the visualization of multiple defect detection processes results, which may derive from the analysis of the same captured image by different AI models. The DSS incorporates an HITL Learning approach (Sect. 5), allowing users to provide their feedback and to retrain the AI models. This feedback is transmitted back to the Middleware, into a separate entity.

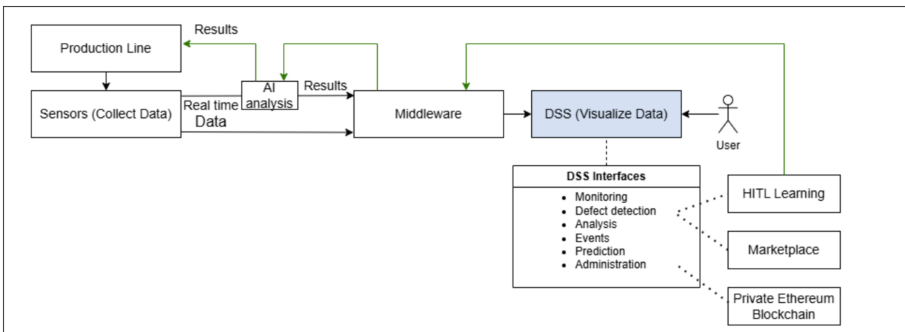


Fig. 3. The overall OPTIMAI Methodology for instrumentation and visualization of the data collected and analyzed by AI in various factories

In antenna manufacturing, the defect detection process is fully automated. A PLC device controls the robotic arm, which positions an antenna onto the hydraulic press for inspection every few seconds [19]. Upon grabbing an antenna, the PLC sends a 5V

external trigger to the RGB camera, signaling it to capture an image. This image is then processed by an AI-driven defect detection system utilizing three AI models, each employing different algorithms. Each model's output is sent to a distinct Middleware entity. Finally, the PLC of the next robotic arm receives a signal indicating a defective antenna, prompting the robotic arm to place it in a separate rack. This process has significantly optimised the production line, compared to the previous practice, where antennas were stacked together and quality inspection occurred asynchronously.

In the second pilot, PCB manufacturing, the operator initiates PCB inspection by placing the board under the camera and pressing the appropriate button in the DSS interface. The camera captures a high-resolution image of the PCB, which is then analyzed by the defect detection AI models. As in the previous factory, the results are depicted in the DSS interfaces, mainly in the “Defect Detection” interface, including the captured image with overlaid defect indicators. Various industrial processes that generate defects are examined, with different AI models employed for the inspection. This paper focuses on glue/epoxy dispensing and glue dispensing calibration processes.

In the glue/epoxy dispensing processes, the AI model calculates the volume of the dispensed glue on the PCB substrate and categorizes it as normal, insufficient or excessive, using colour-coded indicators [21]. The operator inspects the results and provides feedback (details in Sect. 5). Defective PCBs may also be sent to the marketplace to recover residual scrap value. Moreover, depending on the PCB type, the DSS may recommend adjustments to the dispenser pressure, which the operator reviews and decides whether to implement.

In the “Monitoring” tab for lift manufacturing, workers test the hydraulic valve block, which controls elevator movement. During testing, they monitor its performance in real-time, checking the valve block's velocity, sound, pressure, and vibration during both directions of lift movements. If the measurements fall outside the specified range, workers make necessary adjustments to comply with quality control standards [18].

5 Human-in-the-Loop Learning

The DSS displays quality inspection results in the defect detection interface by retrieving the image and visualising inspection results, stored in JSON format, including ground truth annotations as coloured polygons overlaid on the captured image (Fig. 4). If users inspect the results and notice a false detection, missed defect, or misclassified defect type, they can provide feedback to the AI model by switching to annotation mode (Fig. 5). In this mode, users can refine detection results by selecting a specific area and provide textual feedback. Upon saving, the newly annotated area is marked in pink.

The interface also includes retraining settings, allowing manual initiation or automatic triggering once a defined annotation threshold is reached, requiring no further user input.

Every time an annotation is created, a new JSON file is generated for the HITL Learning module. During a retraining session, these JSON files are batched and uploaded to the appropriate Middleware entity. The AI model then uses this data as input for retraining. Once retraining is complete, the updated model can be deployed for defect detection, allowing accuracy improvement as more annotated data is collected. Details of retraining sessions, such as the number of annotations, timestamps, and model performance,

COCO [23] weights to reduce training time. The dataset was split into a 90% training set and a 10% validation set, retraining the model from the initial stage of the process. Evaluation was based on accuracy, precision, recall, and F1-score. Training ran for up to 300 epochs, with early stopping triggered if no improvement was observed on the validation set over 50 consecutive epochs, to avoid overfitting and reduce computation time. In the PCB pilot, deep learning-based methods were employed for detecting conductive glue defects. Specifically, instance segmentation (Mask R-CNN) was used to locate glue deposits, followed by a residual neural network (R2esNet) for volume regression from RGB images [21]. To evaluate the regression models, Mean Squared Error (MSE) was employed as the evaluation metric, with a 10-fold cross-validation retraining process. This iterative approach leverages real-time data and user annotations, leading to significant MSE reduction across phases.

HITL Learning modules integrated into defect detection systems enhance the accuracy of inspection models by incorporating user feedback. Experimental results indicate that model accuracy increases with the integration of annotated defective samples into the training set. The most substantial improvements are observed in the early stages of HITL, where data scarcity significantly impacts performance. As the training set grows, the model continues to improve, suggesting that initial limitations stemmed from insufficient data rather than algorithmic shortcomings.

6 User Testing

6.1 Methodology

Designers and developers of the DSS maintained continuous communication with end-users through questionnaires and workshops, utilizing wireframes and mockups to gather feedback for iterative design, following an agile process to integrate new features based on this feedback. Combined with intermediate user testing, this process helped moderators to better understand operators' workflows and refine the system. The final user assessment, conducted across the pilot sites, evaluated an end-to-end HITL production scenario, focusing on DSS interfaces. Participants were introduced to the system via a demonstration of the production line instrumentation, camera setup, data flow, and results visualization. Each participant tested the DSS and completed a background information and user experience questionnaire (BIQ, UXQ). The BIQ gathered demographics, job-related details, and prior experience with computer applications.

In PCB and antenna manufacturing, the first UXQ section assessed the usefulness of the different DSS modules [24]. The second section, based on the Technology Acceptance Model (TAM) – TAM2 [25], evaluated job relevance and the perceived usefulness for daily work. The third section used the System Usability Scale (SUS) [26]. The fourth section focused on the defect detection module, with participants rating how easy it was to (a) understand detection results, (b) use inspection tools, and (c) provide feedback via annotation mode. Participants also assessed their trust in the system's results (with and without their feedback) [27] and their intention to use the system [25]. All questions used 5-point Likert scale from 1–5 (1: strongly disagree, 5: strongly agree). Overall satisfaction was measured with the Net Promoter Score (NPS) [28,29]. Open-ended questions at the end gathered comments based on likes and dislikes. For lift manufacturing, the

UXQ included TAM- derived items on perceived usefulness, intention to use, quality and ease of understanding the results. An additional item rated UI aesthetics, along with SUS, NPS and open-ended questions.

6.2 Results

Ten operators participated in each pilot study. In the PCB and lift pilots, two participants were women, while in the antenna, one participant was a woman. In the lift pilot, most participants (7) were aged 35–44, with 90% having more than 5 years of experience. In the PCB and antenna pilots, participants' ages ranged from 18–55 and 18–64, respectively. In the PCB pilot, 40% of users were aged 25–34, while in the antenna pilot, 40% were aged 35–44. PCB participants were evenly distributed across experience ranges (<5 years: 4, 5–10 years: 3, > 10 years: 3), while 70% of antenna users had over 10 years of experience. Participants' expertise was related to quality control operations and optimization, either in practical roles or at the management level. Regarding job relevance, in the antenna pilot, 30% of users were neutral, and the rest were positive (30% agree, 40% strongly agree) (Table 1). In the PCB pilot, 40% of participants found relevant and 60% very relevant.

In the lift pilot, 80% of users were positive about system's usefulness in their work, while 20% were neutral. In the antenna line, positivity was higher, with 10% neutral. In the PCB line, all participants were positive (30% agree, 70% strongly agree). Regarding performance improvement, nearly all lift line users were positive, with 10% disagreeing. In the antenna line, 20% were neutral, 50% agreed, and 30% strongly agreed. In the PCB line, 20% were neutral, 30% agreed, and 50% strongly agreed.

The intention to use the system was higher in antenna and PCB pilots. In the lift pilot, one user disagreed, another was neutral, and the rest were evenly split between agreeing and strongly agreeing (40% each). In the antenna line, all users intended to use the system (60% agree, 40% strongly agree), while in the PCB line 10% were neutral, 30% agreed, and 60% strongly agreed. For ease of use, both the lift and PCB lines had one neutral response, with 30% agreeing and 60% strongly agreeing. In the antenna line, 40% agreed, and 60% strongly agreed. Regarding learnability, 10% of lift line users were neutral, 40% agreed and 50% strongly agreed. In the other two lines, all users were positive: 40% agreed and 60% strongly agreed in the antenna line, while PCB, responses were evenly split.

In the lift line, both the quality of the output and the ease of understanding were rated positively (quality: 10% neutral, 40% agree, 50% strongly agree, understanding: 10% neutral, 30% agree, 60% strongly agree). In the antenna line, users found the results easy to understand (40% agree, 60% strongly agree), and the inspection tools easy to use (30% agree, 70% strongly agree), but found the annotation mode slightly more difficult to use (20% neutral, 10% agree, 70% strongly agree). In the PCB line, the results were also easy to understand (10% neutral, 30% agree, 60% strongly agree). Users rated the inspection tools as easy to use (10% neutral, 20% agree, 70% strongly agree), and found it easy to correct false results (30% agree, 70% strongly agree). In both the antenna and PB lines, users expressed greater trust in the system's results when considering their ability to provide their feedback, with positive ratings ranging from 80% to 100% in the antenna line and from 70% to 100% in the PCB line. The NPS score was marginally

favorable for the lift line at 20% (promoters 40% - detractors 20%), marginally excellent for the antenna line at 50% (promoters 60% - detractors 10%), and excellent for the PCB pilot at 60% (promoters 60% - detractors 0%).

Table 1. Indicative UXQ results

Parameter	Item	Positive rates (%)		
		<i>Lift</i>	<i>Antenna</i>	<i>PCB</i>
Relevance (TAM)	In my work, usage of the system is relevant	-	70	100
System’s Usefulness (TAM)	I think that the system would be useful at my work	80	90	100
	I think that using the system would improve my performance in my work	90	80	80
Intention to use (TAM)	Assuming I have access to the system, I intend to use it	80	100	90
Ease of use (SUS)	I thought the system was easy to use	90	100	90
Learnability (SUS)	I would imagine that most people would learn to use this system very quickly	90	100	100
Output of the system (TAM)	The overall output of the system is of high quality	90	-	-
	It was easy to understand the output of the system	90	-	-
Defect detection with AL related				
Ease of defect detection (task-related)	It was easy to understand the defect detection results	-	100	90
	It was easy to use the inspection tools (checkboxes, opacity slider, zoom in/out and pan)	-	100	90
	It was easy to correct any false results	-	80	100
Trust (inspired by 27)	I feel I can trust the results of the system	-	80	70
	I feel I can trust the results of the system as long as I provide my feedback	-	100	100
“How likely is it that you would recommend OPTIMAI system to a college?” - NPS score		20	50	60

In the antenna line, the system's strengths identified in the open-ended question were in defect detection, analysis and usability. Indicatively, users appreciated its visual appeal, the way defects are displayed, and its ease of use. They also noted that the system "included the feedback from the previous evaluation everything that we asked is integrated." Weaknesses were related to the defect detection, as well as general comments and suggestions for possible extensions. Indicatively, users recommended making the annotation mode switch more visible, enhancing the prominence of overall results, and adding options in the administration and monitoring modules. In the PCB line, users highlighted the system's accurate defect detection compared to the current method, its colour-coded visualizations, inspection options, and the clear interface. Suggestions included improving navigation and increasing the size of the captured image area. In the lift line, positive comments focused on the depicted graphs, especially the normal ranges and the reference curves, and on having "all the data in a timely manner". Negative comments included the small graphs size and a suggestion to allow users to modify the reference curves independently.

7 Conclusions

The proposed DSS includes modules for Monitoring, Prediction, Analysis, Event-Notifications, Administration, and Defect detection for ZDM. Its interfaces assist operators by providing real-time monitoring of production and defect occurrences. Integrated HITL Learning methods enable operators to provide annotations and actively participate in AI model retraining, improving AI results accuracy. User testing results highlighted the ease of use, higher acceptance when operators provide feedback, and strong intentions to use and recommend the system. Users' appreciation for the system's responsiveness to their feedback during the design phase, where their suggestions were incorporated after testing, validated the project's commitment to agile development. Operators' acceptance and reduced inspection times demonstrate the system's alignment with Industry 4.0/5.0 and ZDM goals, effectively leveraging AI to improve production efficiency and workforce satisfaction. However, some limitations were noted. Sample sizes were small, with no more than 10 participants per pilot site, as it was not feasible to involve more operators due to the limited number of workers performing relevant roles in the factories. Regarding the interface, graph size issues must be addressed to better support lift operators using tablet devices, while the low NPS indicates a need to further explore and resolve any user concerns in the lift pilot.

Acknowledgments. This work was supported by the European Commission through the H2020 programme under grant OPTIMAI "Optimizing Manufacturing Processes through Artificial Intelligence and Virtualization" (ID: 958264), and through the Horizon Europe programme under grant ENCIRCLE "Enabling Circular Value Chains via Production Digitization and Human Empowerment" (ID:101178230).

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

References

1. Kagermann, H., Wahlster, W.: Ten years of Industrie 4.0. *Science* **4**(3), 26 (2022)
2. Ghobakhloo, M., Iranmanesh, M., Tseng, M. L., Grybauskas, A., Stefanini, A., Amran, A.: Behind the definition of Industry 5.0: a systematic review of technologies, principles, components, and values. *J. Ind. Prod. Eng.* **40**(6), 432–447 (2023)
3. Platform Industrie 4.0 (Ed.) The fourth industrial revolution and industry 5.0 - a critical perspective - Joint statement by the Research Council Industrie 4.0 and the Platform Industrie 4.0 on the use of the term “Industry 5.0”, Berlin, Germany (2024). <https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Industry50.html>. Accessed 15 Nov 2024
4. Albers, A., et al.: Procedure for defining the system of objectives in the initial phase of an industry 4.0 project focusing on intelligent quality control systems. *Procedia Cirp* **52**, 262–267 (2016)
5. Radanliev, P., et al.: Artificial intelligence and the internet of things in industry 4.0. *CCF Trans. Pervasive Comput. Interact.* **3**, 329–338 (2021)
6. Papageorgiou, E.I., et al.: Short survey of artificial intelligent technologies for defect detection in manufacturing. In: 2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA). IEEE (2021)
7. Psarommatis, F.: A generic methodology and a digital twin for zero defect manufacturing (ZDM) performance mapping towards design for ZDM. *J. Manuf. Syst.* **59**, 507–521 (2021)
8. Kiraci, E., et al.: Non-reproducible alignment and fitting algorithm effects on Laser Radar measurement. In: 2017 IEEE International Instrumentation and Measurement Technology Conference (I2MTC). IEEE (2017)
9. Vafeiadis, T., et al.: A framework for inspection of dies attachment on PCB utilizing machine learning techniques. *J. Manage. Anal.* **5**(2), 81–94 (2018)
10. Margetis, G., et al.: Aligning emerging technologies onto I4. 0 principles: towards a novel architecture for zero-defect manufacturing. In: 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE (2022)
11. Psarommatis, F., Kiritsis, D.: A hybrid decision support system for automating decision making in the event of defects in the era of Zero Defect Manufacturing. *J. Ind. Inf. Integr.* **26**, 100263 (2022)
12. Soori, M., et al.: AI-based decision support systems in industry 4.0, a review. *J. Econ. Technol.* (2024)
13. Rosati, R., et al.: From knowledge-based to big data analytic model: a novel IoT and machine learning based decision support system for predictive maintenance in Industry 4.0. *J. Intell. Manuf.* **34**(1), 107–121 (2023)
14. Akbar, M.A., et al.: Integrated-decision support system (DSS) for risk identification and mitigation in manufacturing industry for zero-defect-manufacturing (ZDM): a state-of-the-art review. *Int. J. Adv. Manuf. Technol.*, 1–39 (2024)
15. Psarommatis, F., Kiritsis, D.: A scheduling tool for achieving zero defect manufacturing (ZDM): a conceptual framework. In: *Advances in Production Management Systems. Smart Manufacturing for Industry 4.0: IFIP WG 5.7 International Conference, APMS 2018, Seoul, Korea, August 26-30, 2018, Proceedings, Part II*. Springer International Publishing (2018)
16. Zörrer, H., et al.: Using business analytics for decision support in zero defect manufacturing of composite parts in the aerospace industry. *IFAC-PapersOnLine* **52**(13), 1461–1466 (2019)
17. Cruz, F.C., et al.: Efficient feature extraction for an automatic ultrasound testing decision support system. In: 2014 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings. IEEE (2014)
18. Evangelidis, A., et al.: Efficient deep Q-learning for industrial equipment calibration in elevator manufacturing. *IEEE Trans. Ind. Inform.* (2024)

19. Leontaris, L., et al.: A blockchain-enabled deep residual architecture for accountable, in-situ quality control in industry 4.0 with minimal latency. *Comput. Ind.* **149**, 103919 (2023)
20. Mitsiaki, A., Dimitriou, N., Margetis, G., Votis, K., Tzovaras, D.: Enhancing defect traceability and data integrity in industry 4.0 using blockcahin technology. 1173–1184 (2023). <https://doi.org/10.7712/150123.9866.443273>
21. Evangelidis, A., et al.: A deep regression framework toward laboratory accuracy in the shop floor of microelectronics. *IEEE Trans. Ind. Inform.* **19**(3), 2652–2661 (2022)
22. Glenn, J.: YOLOv8. <https://github.com/ultralytics/ultralytics/tree/main> (2023)
23. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: *Computer vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, part v 13*. Springer International Publishing (2014)
24. Davis, F. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly*
25. Venkatesh, V., Davis, F.D.: A theoretical extension of the technology acceptance model: four longitudinal field studies. *Management Science* (2000)
26. System Usability Scale (SUS): <https://digital.gov/topics/usability/>
27. Ntoa, S., Margetis, G., Antona, M., Stephanidis, C.: User experience evaluation in intelligent environments: a comprehensive framework. *Technologies* **9**(2), 41 (2021)
28. Baehre, S., O'Dwyer, M., O'Malley, L., Lee, N.: The use of Net Promoter Score (NPS) to predict sales growth: insights from an empirical investigation. *J. Acad. Mark. Sci.* **50**(1), 67–84 (2022)
29. What is NPS? Your ultimate guide to Net Promoter Score. <https://www.qualtrics.com/uk/experience-management/customer/net-promoter-score/?rid=ip&prevsite=en&newsite=uk&geo=GR&geomatch=uk>. Accessed 28 Nov 2023



A Norm-Chatbot: Local LLM Vision with Vision-Based RAG for Complex Production Documents and Task-Specific Responses

Michael Sahl Lystbæk¹(✉)  and Ulrik Sahl Lystbæk²

¹ Aarhus University, Birk Centerpark 15, Herning, Denmark
msl@btech.au.dk

² KYOCERA UNIMERCO Tooling A/S, Drejervej 2, Sunds, Denmark

Abstract. Large language models (LLMs) have a significant impact on the development of applied solutions for industrial purposes, with their general knowledge capabilities supporting more autonomous work processes. The weights learned from these extensively trained models provide the backbone for fine-tuning and retrieval-augmented generation (RAG) of LLMs, allowing them to address specific work-task operations in unique enterprise contexts. In this study, a locally running vision-based LLM framework is proposed, including custom vision-based RAG and a business-oriented evaluation method for handling sensitive core business documents. This approach enhances data privacy and prevents leakage of sensitive data. The LLM adapts core enterprise knowledge from in-house data spaces using affordable computational resources. Our proposed norm-chatbot for enterprise LLM data processing runs on a commercial Nvidia RTX 4070Ti GPU processor in an offline environment. A RAG vision technique is applied to handle complex document layouts by transforming them into image content, addressing the challenges that traditional text-based indexing and RAG methods face when documents consist of mixed data types such as text, tables, images illustrations, graphs, etc. The proposed framework uses RAG data processing with a PDF-to-image converter on top of an LLM vision model that embeds the images with textual information. The framework combines ColPali RAG tested with the LLAMA 3.2 11b Vision and LLAVA 1.6 models for highly semantic document processing, specifically for both descriptive and question answering (Q&A) tasks. From the study we found a clear trade-off between fast content generation and precision. However, the proposed framework shows promising results in generating contextual meaningful content in two different knowledge generation tasks for processing custom datasets of enterprise norms and a standard database to support tool development and operations in a Danish tool manufacturing company with a global customer segment.

Keywords: Large Language Model · Retrieval-Augmented Generation · Machine Learning · Applied AI

1 Introduction

Large language models (LLMs) have revolutionized the development of applied solutions across various industrial domains, in particular by supporting autonomous and efficient work processes. Their extensive general knowledge, derived from learned weights, forms the basis for fine-tuning and retrieval-augmented generation (RAG), allowing these models to address specific operational needs within unique enterprise contexts.

Vision language models (VLMs) are multimodal models that enable cross-modal understanding and reasoning, capable of encoding both text and visual data and responding with both image captioning and visual question answering [20]. LLMs are statistical models that have learned from their input data. Their responses are based on probabilistic logic, which is why they sometimes guess right and sometimes guess wrong [17, 19, 31]. Hallucination is one of the drawbacks of LLMs, caused by bias in training data and outdated information [35]. To mitigate hallucination in foundation models (FMs), external knowledge can be introduced into the models [5, 9, 29]. External knowledge should be injected into FMs with caution due to cost and the risk of losing information, which places requirements on the positioning of relevant information in the input context [23].

Retrieving data is challenged by the variety of layouts in which it appears - plain text, tables, or images, or a combination of all three on a single page. Although text encoder models can handle pure text input sequences and, to a limited extent, tables, text embedded in image formats requires a model capable of image classification and optical character recognition (OCR) [7].

In this study, we propose a locally running vision-based LLM framework for processing sensitive core business data. The framework uses a evaluation metric to assess the precision of operation-specific question answering by comparing generated feedback with ground truth (GT) answers found in the business norms and standards. It integrates custom vision-based RAG to encourage contextual precision, with a focus on a local approach to enhance data privacy by ensuring that sensitive information is protected and not leaked. The model adapts core enterprise knowledge from in-house data spaces, leveraging affordable computational resources to maintain privacy while enabling efficient processing. This approach is tailored to meet the needs of a Danish tool manufacturing company with a global customer base, providing an efficient method for processing a custom dataset of enterprise norms and standard databases that supports operational tool development tasks.

Combining multimodal models with RAG enhances the handling of data complexity, reduces hallucinations, and minimizes knowledge inconsistencies in question-and-answer (Q&A) predictions, resulting in more accurate and meaningful responses in domain-specific knowledge fields [7]. In this study, we propose a chatbot for answering specific norm-related questions within the domain of tool manufacturing in an industrial context, where industrial norms, standards, and procedures can comprise thousands of highly descriptive documents containing a mix of text, tables, graphs, and complex illustrations. These documents contain sensitive data that are critical to the competitiveness of the company. This

study makes the following theoretical contributions by extending the capabilities of RAG-based Q&A tasks:

- Integrating multimodal retrieval to handle visually complex, structured enterprise documents for norm-based reasoning.
- Ensuring data privacy by providing locally running generative AI models for sensitive business data.
- Reducing hallucination through vision-based retrieval indexing.

By combining vision-based retrieval, LLM text reasoning, and evaluation of a multimodal framework, this study shows promising results for providing an early-stage norm-based chatbot for industrial applications.

1.1 Large Language Models

LLMs are mainly transformer-based models [33], with self-attention being the building block of transformers [31]. Transformers [33] have played a pivotal role in the development of generative AI [31]. A causal language model (LM) predicts the next word (or token) in a sequence based on preceding words. Nucleus sampling (top-p sampling) is a text generation technique used in causal LMs to produce more natural and coherent text compared to traditional, non-deterministic sampling methods [14, 31]. For text-based LMs, there are three different generation techniques: greedy search, beam search decoding, and sampling (*top-k* or *top-p*) [31]. FMs basically face three challenges [13]: outdated information, lack of domain knowledge, and hallucination.

Hybrid and multimodal models designed to handle both parametric and non-parametric memory, such as seq2seq NLP models, are primarily employed in the text domain, using techniques to enhance contextual understanding for domain-specific tasks [7, 19]. Previous research has used Google BART as a baseline seq2seq pre-trained transformer-based language model [19].

1.2 Retrieval-Augmented Generation

Previous studies have focused on RAG based on text or OCR retrieval methods, individually or as hybrid methods for data retrieval [3, 6, 8–10]. Although recent research has shifted towards vision-based retrieval and reasoning methods to handle highly complex document structures, vision LLMs have been proposed as zero-shot solutions to deal with highly visual data [7]. A major challenge when working with norms and standards in enterprise tasks is that such data often consists of descriptive images (illustrations), tables, and text.

2 Related Work

From the literature, we found that most studies focus on traditional index search combined with text and OCR retrieval, both of which have limited performance when dealing with visually complex PDF datasets consisting of a mixture of text,

tables, and figures [30]. Dual-level retrieval, introduced by [10], which incorporates graph-based indexing, has solved some challenges by improving the precise generation of text content. However, it still struggles with inconsistencies when dealing with complex visual input. Recent literature has addressed these issues by using multimodal models with image reasoning capabilities, combining transformer-based LLMs and video transformers (ViT) [18] for image classification and text recognition and processing [7, 30]. LLAMA models from Meta [28] provide both locally running LMs and the multimodal LLAMA 3.2 Vision model with 11b parameters [27] and the lighter LLAVA 1.6 7b parameter model [11], which both have the ability to process text and visual data in mixed datasets. Local models are one of the crucial parameters to ensure reliable and secure generative chatbots for core business data processing. Nevertheless, we have not been able to locate previously literature addressing the need for norm-based chatbots for handling sensitive corporate data.

3 Methodology

The pipeline for developing and improving the norm-based chatbot follows a structured and iterative process, as illustrated in Fig. 1. Each module contributes to the enhancement of the chatbot by integrating RAG vision methods.

3.1 Proposed Model for Norm-Based Question Answering

This study introduces a RAG-based pipeline designed for norm-based Q&A in industrial tool manufacturing. The proposed model integrates multimodal retrieval and reasoning to handle these highly visually complex documents. The system ensures reliable and secure generative AI interaction by leveraging local models that process structured and unstructured data, while minimizing hallucinations and knowledge inconsistencies.

The model consists of four main modules:

1. **RAG** - Multimodal data retrieval and data indexing.
2. **Inference** - Query processing and retrieval-based answer generation.
3. **Reasoning** - Answer generation using a state-of-the-art VLM.
4. **Evaluation** - Evaluation of generated answers against GT.

The RAG module is responsible for extracting relevant information from a structured norm database containing a complex combination of text, tables, and image-based illustrations. It consists of a multimodal model with the following elements [7]:

- PDF-to-image conversion: Preprocesses PDFs into image format.
- Vision encoder: Extracts features from images, tables, and graphical content, enabling document structure analysis.
- Large language model: Handles text-based retrieval and document indexing.

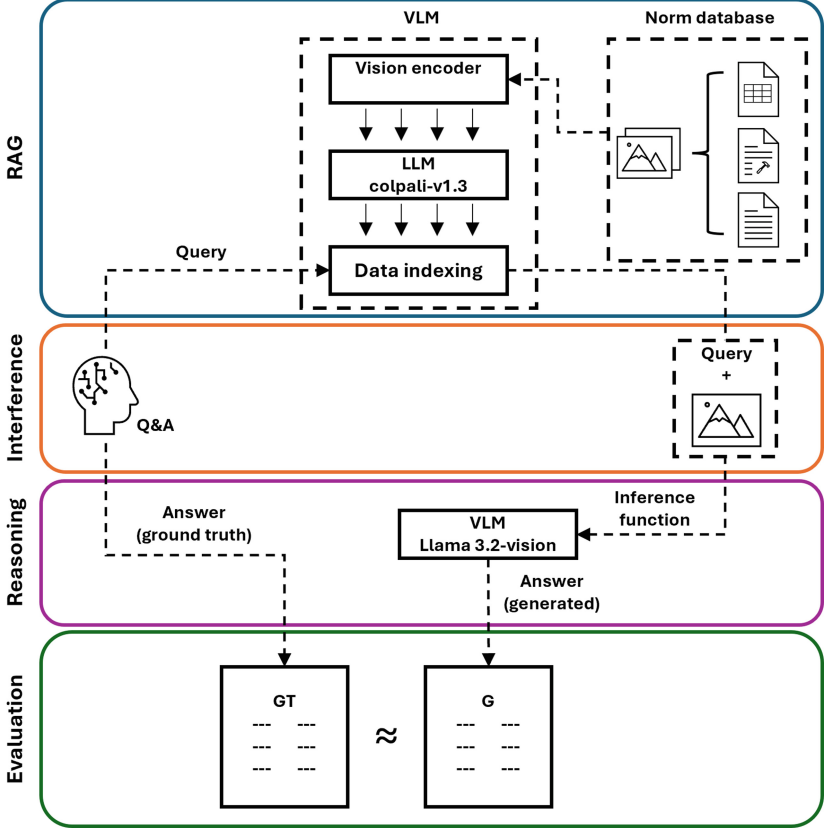


Fig. 1. Development pipeline for a conditional text-to-floor plan generator.

- Data indexing: Facilitates efficient information retrieval through vector-based retrieval for text and vision-based embedding for images.

The system retrieves relevant content based on user queries, which can include text, tables, and images, ensuring accurate document matching for complex norm-based Q/A tasks.

In the inference module, the user queries are processed by retrieving multi-modal data from the created indexed norm database and parsing the query and retrieved document (image formats of the highest-scoring pages) into the VLM model for data reasoning.

The reasoning module generates highly semantic responses to the query, guided by the content provided by the retrieval process. This module employs an LLAMA 3.2 Vision and an LLAVA 1.6 model capable of processing both text and image content to generate responses.

The model outputs the generated descriptions and answers, which are then evaluated against reference answers created along with the question (query)

development as GT benchmarks to assess the model’s accuracy. The performance evaluation compares the generated answer (G) to the GT answer to assess the chatbot’s performance in providing precise domain-specific answers.

4 Implementation

4.1 Dataset

The dataset of PDFs containing a mix of text, tables, and image content is based on active production norms derived from internal production databases that reflect real-world operational standards. The real-world sample comes from a database of nearly a thousand norm documents, the headers and footers of which follow a fixed template. The sample included for the experiments consists of 88 unique pages in English and Danish. The parameters of the dataset are described in Table 1.

Table 1. Dataset parameters

Pages	Words	Char.	Parag.	Lines	Tables	Figures
84	17.2k	88.6k	5.8k	7.9k	101	147

4.2 Vision-Based Retrieval-Augmented Generation

RAG techniques [19] enable LLMs to extract relevant information from a corpus of contextualized data for domain-specific tasks [25]. RAG is applied as an extension to an existing model [15]. For data processing, the ColPali model [7] is used to retrieve images from complex graphical PDF source files. The implementation of ColPali follows the guidelines for the implementation of RAG for LLAMA 3.2 Vision [30] which is modified to also utilize the LLAVA 1.6 vision model [11].

ColPali employs the ViT-based Qwen2-VL-7B Instruct model [1], retrieved from the Hugging Face transformers library [34].

For indexing and searching the provided documents, the Byaldi library is used along with poppler-utils for processing PDF documents [30]. Other installed dependencies are listed in Table 2.

4.3 Vision Large Language Model

The VLM used for the framework in the post-experiment reasoning module is based on the LLAMA 3.2 11b Vision model [26, 30], a multimodal version of the LLAMA 3.2 models [32] and the LLAVA 1.6 multimodal model [22]. The models run in an OLLAMA environment. A comparison of the LLAMA 3.2 11b Vision and the LLAVA 1.6 7b models is shown in Table 3. The main dependencies for the VLM implementation are shown in Table 4.

Table 2. Library dependencies for the ColPali Vision RAG implementation

Model	Version	Reference
Colpali	1.3	huggingface.co/vidore/colpali-v1.3
– PaliGemma-3B [2]	448	huggingface.co/google/paligemma-3b-mix-448
— ColBERT [16]	1.2	github.com/stanford-futuredata/ColBERT
Byaldi	0.0.7	github.com/AnswerDotAI/byaldi
Poppler-utils	0.1.0	pypi.org/project/poppler-utils
Qwen-vl-utils	0.0.10	pypi.org/project/qwen-vl-utils

Table 3. Foundation LLM-vision model parameters

Model	Version	Params	Tokens	License	Context length
LLAMA	3.2 Vision	11B	2.0T/6B	Custom	128,000
LLAVA	1.6	7B	1.2M	Apache 2.0	2,048

4.4 Performance Evaluation

The main performance parameters for the proposed framework are based on the speed of inference and the precision of the generated content. Evaluating the performance of the framework involves serving norms and standards in an enterprise context within the tool manufacturing business for high-industry processing. In this specific case, accuracy is relatively straightforward to measure, as the norms and standards are highly quantified. For this purpose, an evaluation of how well the model performs is based on visual human evaluations [4, 24]. The precision of the framework is assessed based on two operational tasks, each designed to demonstrate the capability of the proposed framework: descriptive and explanation task, and $10 \times$ “what is” Q&As. The evaluation method is inspired by the Rouge [21] evaluation metric, where a manual evaluation is performed for each generated response to evaluate precision based on similarities between generated content and technical accuracy. In task 1 the accuracy is based on counting the number of correct word sentences against the incorrect ones, based on common sense knowledge [4, 12] in relation to the production norms provided. By analyz-

Table 4. Library dependencies for VLM implementation

Model	Version	Reference
Flash-attn	2.7.4	pypi.org/project/flash-attn
Optimum	1.24.0	pypi.org/project/optimum
Auto-gptq	0.7.0	github.com/AutoGPTQ/AutoGPTQ
Bitsandbytes	0.45.3	pypi.org/project/bitsandbytes
OLLAMA	0.4.7	<code>curl -fsSL https://ollama.com/install.sh — sh</code>

ing the generated text for the performance evaluation tasks, the evaluation criteria are as follows: the model demonstrates an overall high level of semantically meaningful content (marked in green), redundant or indifferent content (marked in blue), and odd or incorrect generations that do not contribute positively to the understanding of the norm (marked in red). In the accuracy calculation (correct answers), the redundant generated text (marked in blue) is excluded from the calculation. For task 2, the accuracy is based on a subjective assessment of the generated meaningfulness, individually evaluated for each case. The results are categorized into four levels: wrong, low, medium, and high, based on the GT information found in the norm-dataset. The “what is” answers are limited to give a “precise short answer, with no extra explanation”, which is stated as a prefix for each query.

4.5 System Environment

The experiments are run on an Ubuntu 24.04.2 LTS system with a Conda environment on an RTX 4070 Ti GPU with 12GB of VRAM.

5 Results

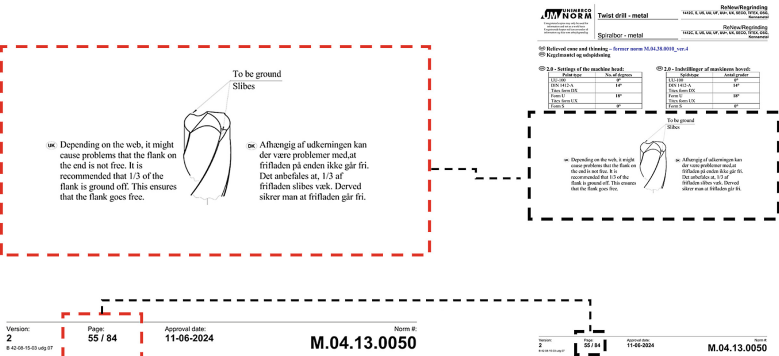
The models are tasked with generating description and explainable text based on the prompts in Table 5. Fig. 2 provides a comprehensive description of the illustration depicting a “relieved cone and the thinning process of twist drills”.

As the results in Table 5 show, the two models produce highly descriptive content, with LLAMA 3.2 Vision response with an average of 193 words per query and LLAVA 1.6 generating 303 words. Both models include a relatively high share of redundant content. However, the LLAMA 3.2 Vision model achieves an average accuracy of 88% with an inference time of 135.5 s, while the LLAVA 1.6 model achieves an accuracy of 24% with a significantly faster inference time of 20.4 s. Furthermore, while LLAVA 1.6 generates responses more quickly, it also produces a substantially higher amount of incorrect content and a lower proportion of correct information.

For the second task, Fig. 3 shows an example of the Q&A GT data information which are the foundation of the Q&A task.

Based on the page information provided, the GT answers reflect the specific content found on the relevant page, as some query terms appear on multiple pages. From Table 6, it can be seen that the LLAMA 3.2 Vision model correctly answers four of the Q&A questions with a high score, while one receives a medium score, three low and two with an incorrect response. The average inference speed for the query response for the LLAMA model is 100.1 s. In comparison, the LLAVA 1.6 model has two answers with a low assessment and eight with wrong answers, however, the LLAVA model also for this task shows a significantly faster inference at 3.7 s.

Processing inference with prompt: **Give me a full description of the illustration shown on page: 55/84**



Inference result (Processing time: 167.0 seconds):

The image shows an illustration of a twist drill, which is a cutting tool used in metalworking. The illustration features a line drawing of the twist drill with its key components labeled.

****Key Components:****

- **Cutting Edge:**** The cutting edge is located at the tip of the twist drill and is responsible for removing material from the workpiece.
- **Shank:**** The shank is the base of the twist drill that attaches to a drill press or other drilling machine.
- **Flute:**** The flute is a spiral groove on the side of the twist drill that helps to remove chips and debris from the cutting area.

****Illustration Details:****

- The illustration is a line drawing, which means it is created using straight lines rather than smooth curves.
- The twist drill is shown in profile view, with one end facing towards the viewer and the other end facing away.
- The illustration includes labels for each component of the twist drill, making it easy to identify and understand its parts.

****Context:****

- The illustration appears to be part of a larger document or manual related to metalworking or machining.
- The presence of text in a foreign language (likely Norwegian) suggests that the document may have been created for a specific industry or region.
- The illustration is likely used as a reference tool for machinists, engineers, or other professionals who need to understand the components and function of a twist drill.

Fig. 2. Result from descriptive test of image illustration based on LLAMA 3.2 Vision model reasoning.

6 Discussion

The experimental results generally demonstrate relatively high performance in the context of this highly domain-specific dataset, which could benefit from fine-tuning a RAG model to improve its semantic understanding of the data structure. However, defining an ideal evaluation metric to assess performance and establish benchmarks for such a specialized dataset has its challenges. This study can be considered a first step towards future benchmarking tasks by providing a baseline performance for such tasks. It draws on state-of-the-art methods in both the implementation of RAG and its integration with VLM for generating highly descriptive, semantically rich answers. This approach is based on multimodal data processing, which has previously been largely limited to text embedding models for both document retrieval and content generation. VLM-based models push the boundaries of local computing resources, affecting inference speed and the size of the model that can be deployed on commercial hardware. For the

Table 5. Results from explainable and descriptive test queries. The results are presented as LLAMA/LLAVA in the table.

Question	Seconds	Words	Correct	Wrong	Redund.	Score
Give me a full description of the illustration shown on page: 55	167.0/28.1	232/390	66/114	20/275	146/1	0.70/0.29
Explain Fig. 1	98.3/11.3	291/215	223/39	20/102	48/74	0.92/0.28
Explain the graph on page: 10	169.2/28.3	228/388	164/75	15/188	49/125	0.92/0.29
Explain the illustration on page: 23	157.2/16.7	184/304	116/25	4/261	64/18	0.97/0.09
Explain the illustration on page: 24	124.2/17.6	85/216	69/39	9/120	7/57	0.88/0.25
Averages	135.5/20.4	193/303	118/58	13/189	62/55	0.88/0.24

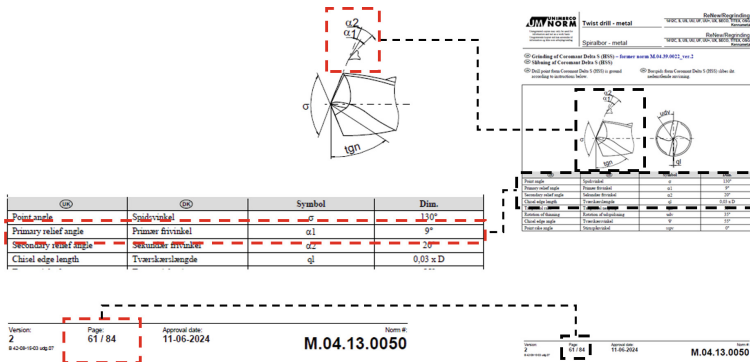


Fig. 3. Example of ground truth information from dataset

“what is” test, based on human assessment, we generally observe high performance, but with a relatively high inference time cost. However, once the models have processed an answer for a given page, the inference speed increases dramatically, up to a factor of $\times 20$, as long as the internal memory remains intact. This suggests the potential for implementing preprocessing steps to create a more efficient system in business contexts. In these experiments, the cache is cleared each time to ensure a one-shot response process. The utilized LLAVA model’s lighter structure, built on the LLAMA 2 architecture, provides a significantly faster reasoning process, but it is also seen that the generated content of this model is less precise.

From the experiments it is seen that the models sometimes draws conclusions beyond what is provided by the RAG index, while such reasoning— like noting the effects of uneven weight distribution can have a significant negative impact on hardware in production processes. However, since such reasoning is not considered part of the specific task, it is considered incorrect or outside the scope of the task.

Table 6. Evaluation of Q&A responses to “what is” questions

Question	GT	LLAMA answer	LLAVA answer	P.
What does α_1 stand for?	primary relief angle	α_1 stands for the primary relief angle.	Angle of Attack.	61
Time/acc:		102.2/high	4.4/wrong	
What is the thinning radius?	usr, 0.3mm	0.3 mm	0.6 mm	
Time/acc:		99.7/high	0.4/wrong	80
What is tgn?	Tangential rake/tangential neigung, 55°	Tangential Neigung (Tangential Angle)	Torsional Generator	
Time/acc:		102.1/medium	5.8/wrong	84
What is coolant channels?	Cooling channels inside the tool for enabling cooling in processes. Available for drills ≥ 6 mm.	Coolant channels are holes that allow for the flow of cutting fluid or coolant during the grinding process to reduce heat and improve surface quality.	The text on the image shows "coolant channels", which refers to the water or coolant passages in an engine or machine that help dissipate heat generated during operation.	
Time/acc:		90.3/low	7.5/low	22
What is Chisel edge angle?	Ψ , 35°	Value: 35°	Value: 45°	
Time/acc:		101.5/high	0.8/wrong	79
What is the meaning of the γ symbolic?	Helix angle/ spiralwinkel (DK), 1°	The gamma symbol represents Helix Angle.	Gamma in this context represents the "third" element in a row or column.	
Time/acc:		89.2/high	8.1/wrong	30
What is web diameter?	17.5%/25%, k	Value: 17.5% of the drill point diameter (D).	6 mm	
Time/acc:		103.3/low	0.8/wrong	37
What is primary sorting?	Matrix for selection of pointtype on HSS-drills, k<20%	Primary sorting is when you follow the diagram and identify the existing points visually looking for a marking of the drill type, manufacturer, etc.	Metal sorting.	
Time/acc:		112.8/low	7.6/low	41
What is point angle?	Spidsvinkel (DK), $\sigma = 140^\circ$	100°	2.9	
Time/acc:		100.6/wrong	0.9/wrong	83
What is sspv?	Point rake angle, 0°	Sspv = 0.05	Spiral shaft position valve	
Time/acc:		99.7/wrong	0.5/wrong	80
Average:		100.1 / wrong=2, low=3, medium=1, high=4	3.7 / wrong=8, low=2	

7 Future Work

Improving data with pre-processing for a more uniform structure for tables and illustrations could enhance consistency. However, the Vision-RAG approach has shown promising results in handling highly complex data structures by treating documents as vision tasks.

Fine-tuning vision models could improve their structural understanding of specific document sources, drawing further inspiration from the methods used in [7]. A training pipeline that incorporates user interaction could help develop and refine model performance. This could be achieved by implementing the model

based on RAG, allowing operators to evaluate its responses and suggest acceptance or revisions, thus creating a structure suited for future fine-tuning.

Benchmarking and evaluating with FM and fine-tuned VLM would provide better insight into its performance. However, vision-based approaches remain quite limited in terms of accessible FM models for local computing, and previous studies on benchmarking are also scarce.

8 Conclusion

Our proposed framework for norm-based data processing has shown promising results in this early implementation of a local Vision-RAG-based corporate norm chatbot, aimed at supporting businesses in a variety of domains that face challenges with highly complex document structures. The framework has the potential to facilitate a higher degree of interaction with norm databases, improving production quality and increasing accessibility to highly technical information. The reasoning interaction layer, which provides the chat interface, can serve as an explainable entry point for communicating difficult-to-understand knowledge to human operators with diverse technical backgrounds, often working in interdisciplinary teams. In the description and explanation task, the LLAMA 3.2 Vision model shows high accuracy, measured by correct and meaningful content - 88%, respectively. For the "what is" test 50% of the answers demonstrate a medium-high level of contextual meaningfulness, making the proposed framework promising for the further development of domain-specific knowledge chatbots.

References

1. Bai, J., et al.: Qwen-VL: a versatile vision-language model for understanding, localization. Text Reading Beyond **2** (2023)
2. Beyer, L., et al.: PaliGemma: a versatile 3B VLM for transfer. arXiv preprint [arXiv:2407.07726](https://arxiv.org/abs/2407.07726) (2024)
3. Chan, C.M., et al.: RQ-RAG: learning to refine queries for retrieval augmented generation. arXiv preprint [arXiv:2404.00610](https://arxiv.org/abs/2404.00610) (2024)
4. Dou, Y., Forbes, M., Koncel-Kedziorski, R., Smith, N.A., Choi, Y.: Is GPT-3 text indistinguishable from human text? Scarecrow: a framework for scrutinizing machine text. arXiv preprint [arXiv:2107.01294](https://arxiv.org/abs/2107.01294) (2021)
5. Duan, H., Yang, Y., Tam, K.Y.: Do LLMs know about hallucination? An empirical investigation of LLM's hidden states. arXiv preprint [arXiv:2402.09733](https://arxiv.org/abs/2402.09733) (2024)
6. Edge, D., et al.: From local to global: a graph rag approach to query-focused summarization. arXiv preprint [arXiv:2404.16130](https://arxiv.org/abs/2404.16130) (2024)
7. Faysse, M., et al.: ColPali: efficient document retrieval with vision language models. In: The Thirteenth International Conference on Learning Representations (2025). <https://openreview.net/forum?id=ogjBpZ8uSi>
8. Gao, L., Ma, X., Lin, J., Callan, J.: Precise zero-shot dense retrieval without relevance labels. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1762–1777 (2023)

9. Gao, Y., et al.: Retrieval-augmented generation for large language models: a survey. arXiv preprint [arXiv:2312.10997](https://arxiv.org/abs/2312.10997) (2023)
10. Guo, Z., Xia, L., Yu, Y., Ao, T., Huang, C.: LightRAG: simple and fast retrieval-augmented generation. arXiv (2024)
11. Haotian, L.: Repo: LLaVA 7B model. <https://github.com/haotian-liu/LLaVA> (2023)
12. Hendrycks, D., et al.: Measuring massive multitask language understanding. arXiv preprint [arXiv:2009.03300](https://arxiv.org/abs/2009.03300) (2020)
13. Hoang Tran: which is better, retrieval augmentation (RAG) or fine-tuning? Both. <https://snorkel.ai/blog/which-is-better-retrieval-augmentation-rag-or-fine-tuning-both/> (2024)
14. Holtzman, A., Buys, J., Du, L., Forbes, M., Choi, Y.: The curious case of neural text degeneration. arXiv preprint [arXiv:1904.09751](https://arxiv.org/abs/1904.09751) (2019)
15. IBM Technology: RAG vs. Fine Tuning (2024). <https://www.youtube.com/watch?v=00Q0G84kq3M>
16. Khattab, O., Zaharia, M.: ColBERT: rfficient and effective passage search via contextualized late interaction over BERT. In: Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, pp. 39–48 (2020)
17. Martineau, K.: What is retrieval-augmented generation? <https://research.ibm.com/blog/retrieval-augmented-generation-RAG?ref=blog.getdataring.com> (2023)
18. Kolesnikov, A., et al.: An Image is worth 16×16 words: transformers for image recognition at scale. In: Google (2021)
19. Lewis, P., et al.: Retrieval-augmented generation for knowledge-intensive NLP tasks. Adv. Neural. Inf. Process. Syst. **33**, 9459–9474 (2020)
20. Li, Y., Wang, C., Jia, J.: LLaMA-VID: an image is worth 2 tokens in large language models. In: European Conference on Computer Vision, pp. 323–340. Springer (2024). https://doi.org/10.1007/978-3-031-72952-2_19
21. Lin, C.Y.: Rouge: a package for automatic evaluation of summaries. In: Text Summarization Branches Out, pp. 74–81 (2004)
22. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 26296–26306 (2024)
23. Liu, N.F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., Liang, P.: Lost in the middle: how language models use long contexts. Trans. Assoc. Computinguistics **12**, 157–173 (2024)
24. Lystbæk, M., Giannakidis, A., Beliatas, M.J., Olsen, M.: Removing unwanted text from architectural images with multi-scale deformable attention-based machine learning. In: 2023 IEEE International Conference on Imaging Systems and Techniques (IST), pp. 1–5. IEEE (2023). <https://doi.org/10.1109/IST59124.2023.10355658>
25. Martineau, K., Explainable, A., Generative, A.: What is retrieval-augmented generation? IBM Res. Blog **22** (2023)
26. Meta: LLaMA 3.2-vision model card. https://github.com/meta-llama/llama-models/blob/main/models/llama3.2/MODEL_CARD_VISION.md (2024)
27. Meta: Repo: Llama 3.2 11B-vision. <https://huggingface.co/meta-llama/Llama-3.2-11B-Vision> (2024)
28. Meta: Repo: Llama-models. <https://github.com/meta-llama/llama-models?tab=readme-ov-file> (2025)
29. Rawte, V., Sheth, A., Das, A.: A survey of hallucination in large foundation models. arXiv preprint [arXiv:2309.05922](https://arxiv.org/abs/2309.05922) (2023)

30. Ryan Ong: Llama 3.2 vision with RAG: a guide using Ollama and ColPali. <https://www.datacamp.com/tutorial/llama-3-2-vision-rag> (2025)
31. Sanseviero, O., Cuenca, P., Passos, A., Whitaker, J.: Hands-On generative AI with transformers and diffusion models. “O’Reilly Media, Inc.” (2024)
32. Touvron, H., et al.: Llama 2: open foundation and fine-tuned chat models. arXiv preprint [arXiv:2307.09288](https://arxiv.org/abs/2307.09288) (2023)
33. Vaswani, A.: Attention is all you need. In: Advances in Neural Information Processing Systems (2017)
34. 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, pp. 38–45. Association for Computational Linguistics (2020). <https://doi.org/10.18653/v1/2020.emnlp-demos.6>
35. Xu, Z., Jain, S., Kankanhalli, M.: Hallucination is inevitable: an innate limitation of large language models. arXiv preprint [arXiv:2401.11817](https://arxiv.org/abs/2401.11817) (2024)



Artificial Intelligence Versus Food Wastage in Bakeries

Elias Pimenidis¹ , Yahia Abdelrahman¹, and Anastasios Panagiotis Psathas² 

¹ University of the West of England, Frenchay Campus, Bristol BS16 1QY, UK
elias.pimenidis@uwe.ac.uk, yahia2.abdelrahman@live.uwe.ac.uk

² Department of Civil Engineering-Lab of Mathematics and Informatics (ISCE), Democritus
University of Thrace, 67100 Xanthi, Greece
anspatha@civil.duth.gr

Abstract. Time series forecasting for sales is well established in the business world. However, the nature of the data and often the domain area of the application can yield unexpected inconsistencies and constraints. Machine learning tools can be utilized to enhance the accuracy of predictions and support effective forecasting. This work presents the case of a small chain of bakeries where machine learning supported forecasting aims to tackle food wastage, enhance profit margins, and contribute to a more sustainable community.

Keywords: Machine Learning · Sales forecasting · Time Series Forecasting · Bakeries · Food Wastage · Sustainability

1 Introduction

In January 2025 the United Kingdom government announced its AI Opportunities Plan, aiming at ramping up AI adoption across the UK to boost economic growth, provide jobs for the future and improve people's everyday lives [1]. While the Department of Science, Innovation and Technology of the UK Government is developing its long-term plan for the UK's AI infrastructure needs, backed by a 10-year investment commitment, funding initiatives target more specific needs within the business sectors that could have a more immediate impact on society. Small businesses across the country will be helped to boost their productivity and efficiency through AI tools set to be trialed. In January 2025 the government announced the launch of 120 funded projects to support small businesses in engaging AI to enhance their efficiency. Among those projects there are ones supporting bakers to use AI to cut food waste – and thereby protect profit margins. The aim is to accurately predict sales and forecast how much each product needs to be made daily [2]. Minimizing food wastage could induce lower raw material demands that could eventually lead to price drops in the medium to long-term allowing households and individuals a stronger purchase power for foods that provide better nutrition.

Beyond the business impact such projects have a strong societal and sustainability focus. The United Kingdom, despite being one of the world's leading economies, is seeing an alarmingly increasing number of people relying on food banks to feed themselves and their families [3]. Food wastage can be seen as accentuating this problem as it contravenes societal ethics. At the same time, such projects can be seen as conversant with the United Nations (UN) sustainability Goals that were set in 2015. Among the 17 Goals there is a strong focus on protecting vulnerable individuals, with Goal 2 targeting hunger and more specifically aims at ending hunger by achieving food security and nutrition, and sustainable agriculture. The UN recognizes that high food prices are a key factor in driving communities and whole countries to hunger [4]. Combating and controlling food wastage can help control food price rises and supporting communities addressing hunger issues.

The work discussed here is based on a project working with a small bakery chain in Bristol, United Kingdom, to provide stable prediction of demand for products that have a short shelf life. Accurate prediction can reduce food waste and improve profit margins within the company. The first phase of the work that is discussed here was carried out between August 2024 and March 2025. This phase of the work was funded by the Made Smarter West of England initiative [5]. The work is ongoing with the next phase planned for the summer of 2025.

2 The Nature of Data at the Core of the Case Study

Artificial Intelligence tools are usually based on having access to large volumes of data that allow for fine-tuned forecasting via stable time series. Bakeries, like any retail business, produce sales data that are in principle suitable for applying time series forecasting models. This is partly true with the case studied here. Our small chain of independent bakeries operates three retail outlets, and each bakery produces its own sales data that over time provides a time series. Therefore, relevant forecasting models can be applied and offer potentially accurate forecasts of sales that will in turn offer the business the opportunity to make more targeted orders of foods which in turn will lead to less wastage at the end of the day.

The graphs below (Figs 1 and 2) show the fluctuation of data in terms of sales of coffees and that of sandwiches per day across all branches of the bakery chain.

As can be easily seen the two sets of data yielding the above graphs vary considerably as to their behaviour, and the range of values. This suggests that different models will be required to predict future sales of these two groups of products. This is further emphasized as the nature of the two products is different. Coffees are made at the premises and on demand, while their raw materials have a much longer shelf life. On the contrary sandwiches require preparation away from the point of sales and both the raw materials used, as well as the product have a much shorter shelf life of usually a day. A further challenge is that the volume of sandwiches sold is rather low and thus predictions as to expected sales can be more difficult.

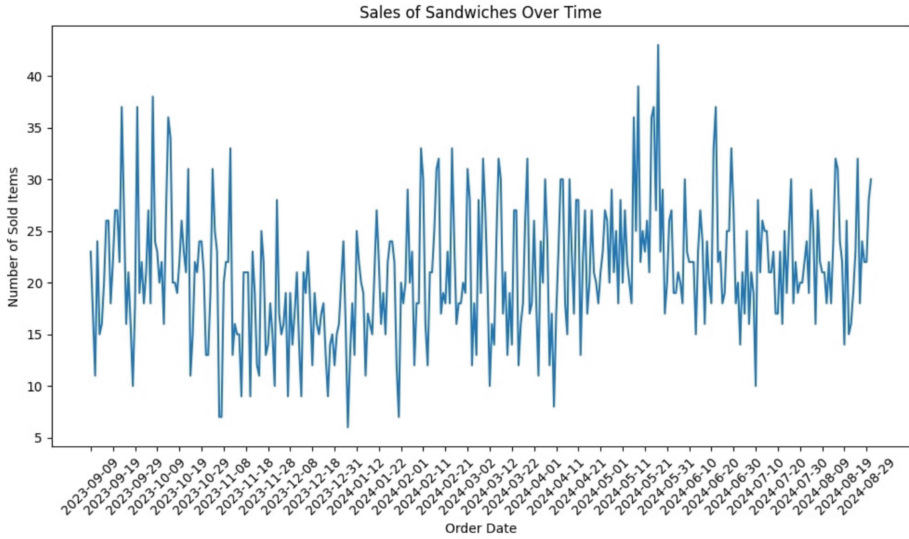


Fig. 1. Sales of Sandwiches over time – all bakeries

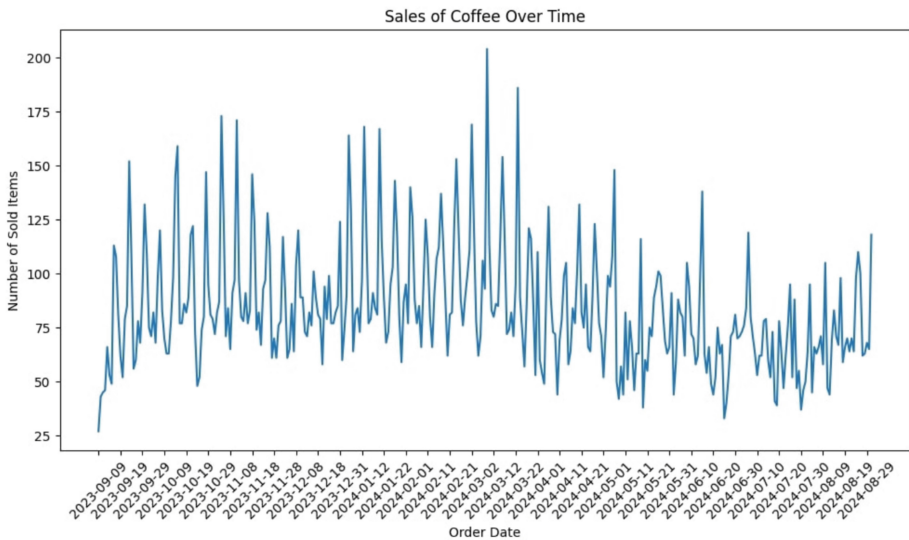


Fig. 2. Sales of Coffee over time – all bakeries

2.1 Methodology

The sales data was obtained from a system used to process the sales in each of the outlets. This allowed data to be collected per different products for each of the three outlets. Given the challenges due to low volumes of sales of different types of foods the research focused on experimenting with the most popular product as sold at the

busiest of the three outlets. This was further justified by the fact that each outlet has a different business model, dictated by each location and the type of clientele it attracts. Furthermore, some outlets might not have all the food items that are typically found in other outlets. In forecasting we are particularly keen to explore seasonality, i.e. repeating short-term fluctuation, and trend, long-term change in the time series model.

Prophet has been chosen as the forecasting model for this work. Its benefits include ability to add holiday effect by country/or even state easily, as special events/holidays have impact on sales, so by incorporating them into the model that can differentiate between normal days versus special events/holidays. Furthermore, Prophet can handle missing data as well as outliers automatically. It also can handle time series with complex seasonality, something that is relevant to our case here since each outlet has its own business characteristics and constraints. It is easy to use and is open source [6].

3 Related Work

The business world nowadays considers accurate prediction of sales as an essential task that can directly impact their ability to formulate effective plans, particularly in response to changing market trends, while also enabling them to optimize inventory management. An added advantage is normally enhanced customer satisfaction through timely product availability and possibly better prices through passing on part of the savings incurred through better utilization of business resources, to the customers. Most historic sales data is obtained in the form of time series and literature offers abundant choice of models for providing time series analysis. The authors in [7] highlight the importance of understanding the variations within time series data, particularly the influence of seasonal patterns, something that is directly relevant to bakeries as to the choice of products that could be offered, the expected volumes of sale, and timing of such sales related to weather, and other human controlled activities (such as festivals or sporting events) that occur regularly.

In [8] the author discusses constraints in relation to the data used in time series forecasting and while their work focuses on sales of alcoholic beverages in the USA, there are similarities between this work and ours presented here. Sales of alcohol present more significant fluctuations compared to other food categories and these induce more constraints. The authors performed machine learning with Prophet, an open-source forecasting software package specializing in dealing time series released by Facebook and is much more effective in handling such data. This is particularly encouraging for our work as the data set used displays similarities in nature to our data. Our own data though has the further challenge of having less historic cycles as can be witnessed in the next section of this paper.

A further work whose data displays similar variations and constraints as ours is that discussed in [9]. The authors stress the importance of preprocessing data analysis to identify the dominant elements of the time series to ensure the most important contributing attributes of the data set are considered for processing. This corresponds with the approach adopted in our work and strengthens our confidence in the outcome.

Finally, the authors in [10] confirm the suitability of the use of Prophet as an appropriate method for our data analysis. Being mainly an extension of the generalized linear model using nonlinear smoothing functions. The advantage of the Prophet method over the diversity of statistical and time series forecasting methods such as autoregressive integrated moving average (ARIMA), seasonal autoregressive integrated moving average (SARIMA), fractional autoregressive integrated moving average (FARIMA), is the approach of the analyst-in-the-loop. This allows users to apply their domain knowledge about the dataset to the forecasting algorithm without having any knowledge of the statistical methods working from within. In our case the expertise of the business professional and future user is utilized to impact the development of the model with domain expert knowledge that would otherwise evade the span of expertise of the data analysts. This can have a very positive impact on the quality of the predictions obtained as it allows expert fine tuning that cannot be interpreted through the data.

4 Results and Discussion

Following a preprocessing of that sales data over a period of twelve months the focus of the work turned to the highest selling products within the main two categories of the sales, mainly “Coffee Latte” and “Tuna Focaccia”. Despite the different nature of the products the sales data indicates similar behaviour with very few diversions from the mean value of sales instances daily, as shown in Figs. 3 and 4.

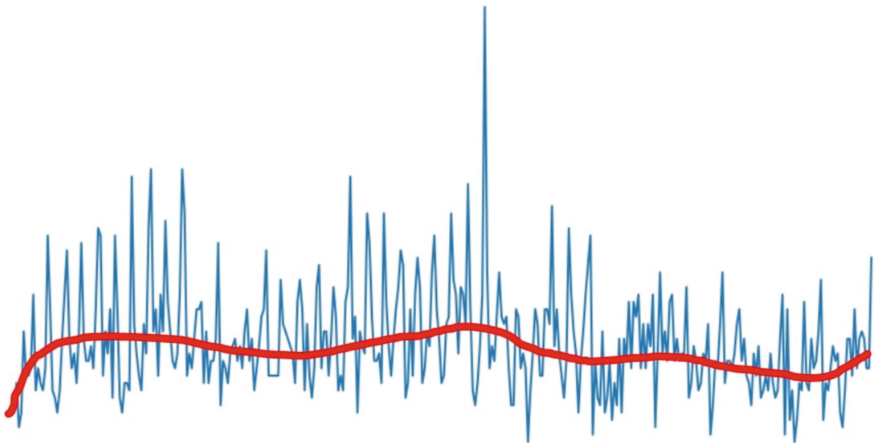


Fig. 3. Sales of Coffee Latte over time

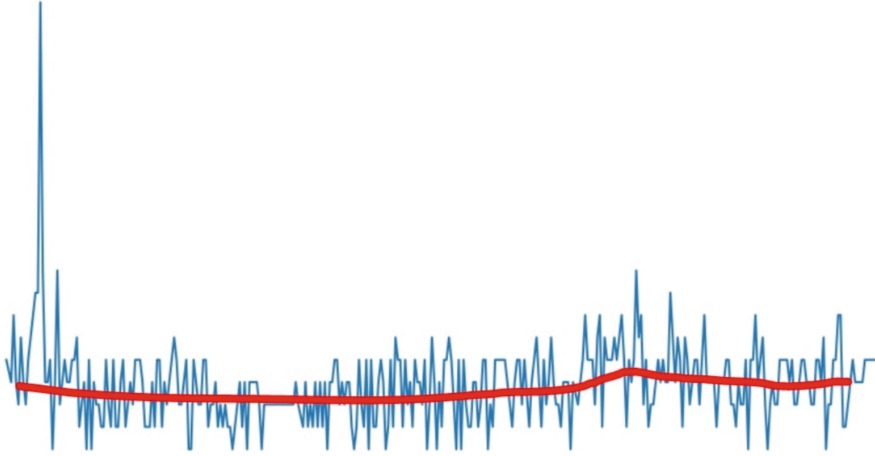


Fig. 4. Sales of Tuna Focaccia over time

To attempt to refine the series as shown in Figs. 3 and 4, we have used Prophet to add holiday effect, to explore the impact special events/holidays have can have on sales, so by incorporating them into the model can differentiate between normal days versus special events/holidays. The model was trained with no holiday effects first vs with holiday effect afterwards. The time of the data set was “Start Date is 2023–09-09 and End Date is 2024–08-31”. The model was trained on 70% of the dataset and then it was utilized to predict the rest 30% of the values, which are then compared to the actual values for the remaining 30% of the original dataset.

4.1 The Prophet Forecasting Model

Prophet uses a decomposable time series model with three main model components: trend, seasonality, and holidays. They are combined in the following equation:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon t \quad (1)$$

where:

- $g(t)$: piecewise linear or logistic growth curve for modeling non-periodic changes in time series.
- $s(t)$: periodic changes (e.g. weekly/yearly seasonality).
- $h(t)$: effects of holidays (user provided) with irregular schedules.
- εt : error term accounts for any unusual changes not accommodated by the model.

MAPE - Mean Absolute Percentage Error was obtained (the higher the value the worse the model is, the lower the value the better the model).

Mean Absolute Percentage Error (MAPE): 38.36% obtained is acceptable but not necessarily very good. This was achieved without considering the holiday effect on the time series for Coffee Latte.

The next step involved incorporating the holiday effect with the MAPE improving by just under 2% (36.69%). Although this was an improvement, this was not a significant one. The next step though offered us the opportunity to explore the full value of Prophet by following Hyperparameters in Prophet and their effect:

```
from sklearn.model_selection import ParameterGrid
params_grid = {'seasonality_mode': ('multiplicative', 'ad-
ditive'),
               'change-
point_prior_scale': [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7],
               'holi-
days_prior_scale': [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7],
               'n_changepoints' : [100, 150, 200, 300, 400],

               'weekly_yes_no_': [True, False],

               }
grid = ParameterGrid(params_grid)
cnt = 0
for p in grid:
    cnt = cnt+1

print('Total Possible Models', cnt)
```

Total 980 Models.


```

import random

model_parameters = pd.DataFrame(columns =
['MAPE', 'Parameters'])
for p in grid:
    test = pd.DataFrame()
    print(p)
    random.seed(0)
    train_model = Prophet(changepoint_prior_scale =
p['changepoint_prior_scale'],
                        holidays_prior_scale = p['holidays_prior_scale'],
                        n_changepoints = p['n_changepoints'],
                        seasonality_mode = p['seasonality_mode'],
                        weekly_seasonality=p['weekly_yes_no_'],
                        interval_width=0.95)
    train_model.add_country_holidays(country_name='UK')
    train_model.fit(Training_dataset)
    forecast = train_model.predict(forecast_date)
    df_eval = Testing_dataset.merge(forecast[['ds',
'yhat']], on='ds', how='inner')

    mape = mean_absolute_percentage_error(df_eval['y'],
df_eval['yhat']) * 100 # Convert to percentage

    print(f"Mean Absolute Percentage Error (MAPE):
{mape:.2f}%")
    print('Mean Absolute Percentage Error (MAPE)-----
-----', mape)
    model_parameters = pd.concat([model_parameters,
pd.DataFrame([{'MAPE': mape, 'Parameters': p}]), ignore_index=True)

```

We have managed to improve the model, from 36% to 30% via selecting the correct values for hyperparameters.

There are two key factors here.

Additive: states that season effect is increased by a constant rate no matter how the overall sales are going, e.g. if we have additive seasonality effect in our timeline, and we sell ice cream, we expect to see that ice cream sales would increase EACH SUMMER by 50 units for example, even if the overall sales are terrible, but each summer our sales would increase by 50.

Multiplicative: Relies more on the overall sales, every December, sales increase by a certain percentage (e.g., 20%). This means that if your sales are higher, the increase will be bigger. If your sales are lower, the increase will be smaller.

4.2 Looking at the Accuracy Graphically

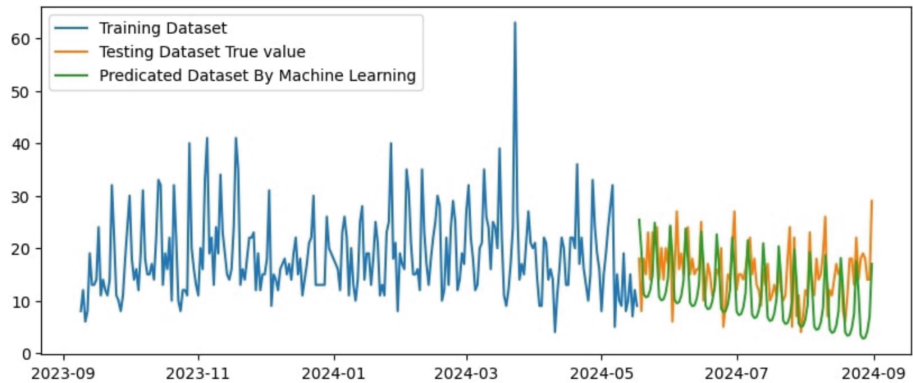


Fig. 5. Training Data Set Coffee Latte

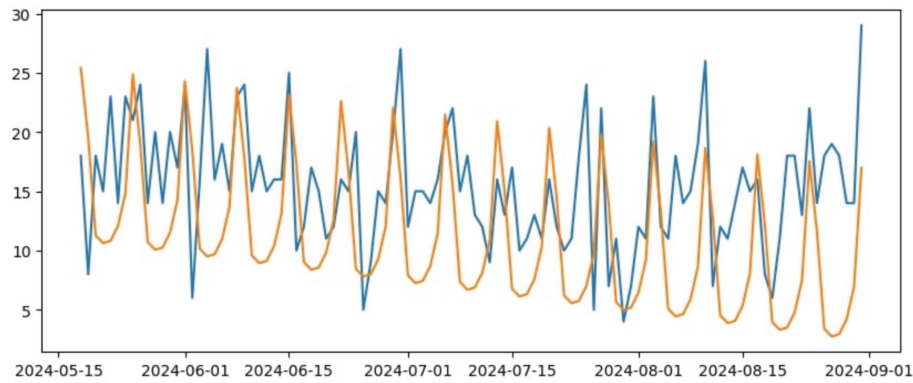


Fig. 6. Forecasting Coffee Latte Sales

Upon visual inspection of Figs. 5 and 6 it appears that the model has captured the general seasonality. The model’s accuracy appears to decrease from 2024-08-15 onwards. This is possibly due to the nature of time series problems and indicates that the model may need to be updated regularly.

To explore room for further improvement the work considered the following options, (i) Sliding Window Approach and (ii) Rolling Forecast Origin.

For the sliding window approach the model is trained on a fixed-length window of past observations, when new data arrives, the oldest data point is removed, and the latest

one is added (hence, the window “slides”). It is smaller compared to Rolling Forecast Origin, so training is quicker. One of the disadvantages is that it might lose some useful information from historical data, as it removes parts of the historic data over time.

Rolling Forecast Origin works similarly to Sliding Window, but we do not remove any data from our model, and when new data is provided, it is simply appended into the model. This could potentially enhance model accuracy, although it is computationally more expensive. For both approaches there is the assumption that there will be new data made available to update the model on a regular basis.

For the purposes of this work, it was decided to adopt Rolling Forecast Origin, starting with the model pretrained with 4 weeks of data, predicting the following week, getting next week’s true values and append to the model, then trained the model on 5 weeks data, and so forth.

The Mean Absolute Percentage Error (MAPE) achieved was 34.96%

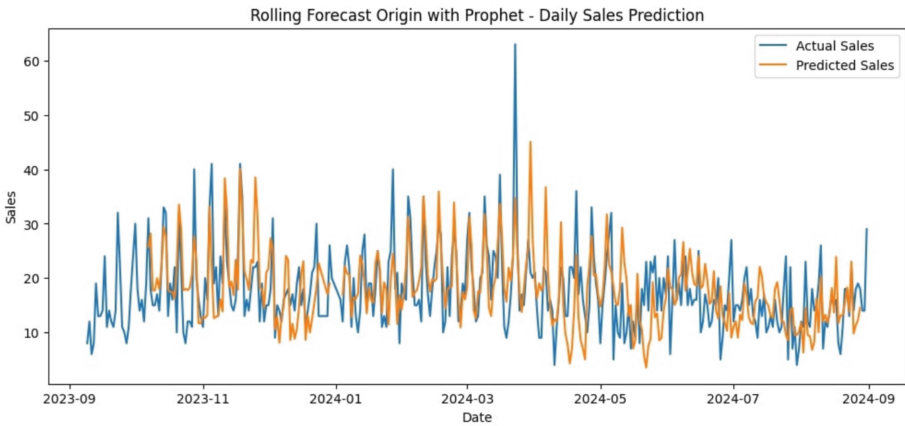


Fig. 7. Rolling Forecast Origin with Prophet – Daily Sales Prediction Coffee Latte

As seen in Fig. 7, the model appears to be more accurate than the previous one. The reason is that following a prediction for a week, once that week passes the actual data for that week, is used to update the model. Thus, the model keeps up with actual results as they evolve. Training is fine-tuned and accuracy is improved. Seeking possible ways to increase the accuracy of the rolling forecast origin, we increase the number of weeks the initial model was trained on, instead of 4 weeks, to 8 weeks and then apply rolling forecast.

The Mean Absolute Percentage Error (MAPE) improved to 31.66%. Figure 8 below shows the resultant graphical representation of the forecast. The graph demonstrates a good fit between actual and predicted levels of sales. There is certainly room for improvement, but the case presented and discussed here has its limitations and constraints that could limit the accuracy achieved. Considerable enhancements at the level of accuracy achieved have been demonstrated, despite a rather limited and possibly unstable data set used. Further work is required to improve the desired level of accuracy in predicted sales. Such predictions can be enhanced by improving the consistency of the data set used and

expanding the length of time for which data is available. Using Prophet provides a basis upon which user experience can interfere with the predictive model through a human expert and enhance the accuracy levels appropriately [11].

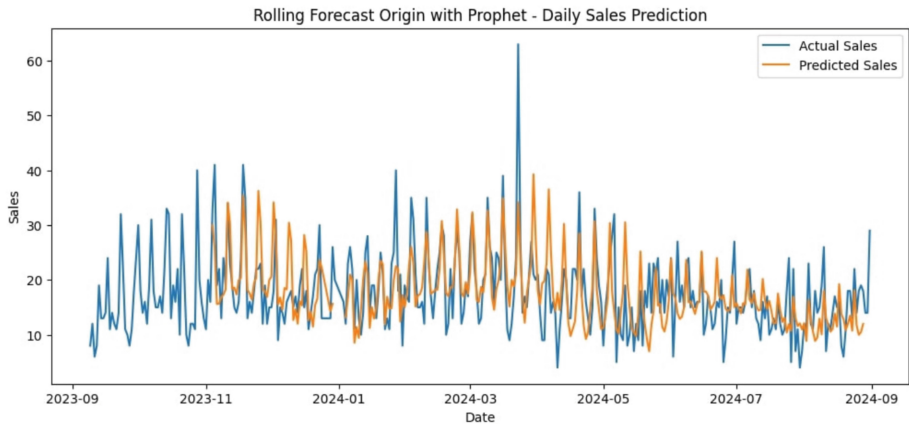


Fig. 8. Rolling Forecast Origin with Prophet – extended training period

5 Conclusions and Further Work

The work presented here has shown the potential of utilizing machine learning based tools to enhance traditional time series forecasting in successfully predicting future sales in bakeries. The quality of the data used here, and its consistency can affect the attainment of the desired outcome. Traditional time series forecasting has hidden biases and can lead to results that could yield large errors. Using an environment like Prophet can offer the benefit of enhancing a prediction by providing parameter selection in models that will improve the accuracy levels that are currently encountered.

The work is ongoing and further analysis with enhanced models, refined selection of datasets and other methods of enhancing results are being considered. Finally, the impact of capacity planning will be considered in attempting to enhance the accuracy of forecasting in a sensitive and very competitive business environment. This we expect to support effective profit margin improvements and the support of sustainability initiatives that counter food wastage.

Acknowledgments. The work presented here was funded by Made Smarter West of England in Collaboration with the University of the West of England. The authors would like to express their gratitude to Ross Gibbens of Pinkmans for his valuable collaboration and support.

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

References

1. Department for Science Innovation & Technology, Independent Report, AI Opportunities Action Plan. <https://www.gov.uk/government/publications/ai-opportunities-action-plan/ai-opportunities-action-plan>. Accessed 14 Jan 2025
2. Department for Science Innovation & Technology Government puts AI to work for bakers, road workers and more, Government puts AI to work for bakers, road workers and more - GOV.UK. Accessed 14 Jan 2025
3. Bull, R., Miles, C., Newbury, E., Nichols, A., Weekes, T., Wyld, G.: Hunger in the UK Report, The Trussell Trust, 10 August 2023, <https://www.trussell.org.uk/publications/hunger-in-the-uk>. Accessed 22 Jan 2025
4. United Nations Department of Economic and Social Affairs: Sustainable Development – Goal 2End hunger, achieve food security and improved nutrition and promote sustainable agriculture (2023). <https://sdgs.un.org/goals/goal2>. Accessed 22 Jan 2025
5. <https://www.westofengland-ca.gov.uk/what-we-do/innovation/made-smarter/>. Accessed 31 Mar 2025
6. <https://facebook.github.io/prophet/>. Accessed 31 Mar 2025
7. Boukrouh, I., Idiri, S., Tayalati, F., Azmani, A., Bouhsaien, L.: A hybrid approach for sales forecasting: combining deep learning and time series analysis. *Int. J. Eng.* **38**(4), 859–870 (2025)
8. Kuang, Y.: Time series data forecasting: alcohol sales prediction based on prophet. *Highlights Sci. Eng. Technol.* **85**, 597–601 (2024). <https://doi.org/10.54097/x446dk79>
9. Rahman, M.T., Khatun, E., Asha, A.M., Alam, M.J.: Time series forecasting of agricultural products sale using deep learning. In: 14th International Conference on Computing Communication and Networking Technologies (ICCCNT), pp. 1–6, Delhi, India (2023). <https://doi.org/10.1109/ICCCNT56998.2023.10307538>
10. Boye1, P., Ziggah, Y.Y., Agyarko, K.: A short-term prediction model for the number of registered motor vehicles using Facebook prophet forecasting approach. *Eng. Technol. J.* **9**(12), 5650–5658 (2024)
11. Prophet forecasting for scale. <https://research.facebook.com/blog/2017/2/prophet-forecasting-at-scale/>. Accessed 31 Mar 2025



Assistive X for Amyotrophic Lateral Sclerosis (ALS)

Radhika Khatri[✉], Samin Rizwan, Faryal Khan, Sudais Yasin, Faisal Alvi,
and Qasim Pasta

DSSE, Computer Science, Habib University, Karachi, Pakistan
{rk06365,sr06989,fk06676,sy06541}@alumni.habib.edu.pk,
{faisal.alvi,qasim.pasta}@sse.habib.edu.pk

Abstract. Amyotrophic lateral sclerosis (ALS), a progressive neurological disorder impacting motor neurons crucial for voluntary muscle movement, poses significant challenges for affected individuals. In Pakistan, an estimated 3.5 per 100,000 people grapple with this debilitating condition. The primary focus of our project is to address the communication difficulties faced by ALS patients due to muscle weakness and loss of motor function. We propose the development of an intuitive application that integrates eye-gaze technology with a comprehensive “Phrase Bank”, complemented by voice output capabilities. This innovative approach allows patients to select phrases and images by simply gazing at them on a screen, significantly reducing the physical effort required for communication, particularly as ALS progresses. Recognizing the importance of familiar phrases in everyday communication, our application includes a dynamic “Phrase Bank”. This feature acts as a readily accessible repository of commonly used phrases and sentences, offering patients a collection of vital expressions to convey their needs, emotions, and thoughts. Navigating through this collection becomes an effortless task for users, ensuring the application remains both intuitive and efficient. While the precise cause of ALS remains elusive, with genetics believed to play a role in some cases, our project takes a proactive approach, focusing on improving the lives of individuals with ALS by providing innovative tools to enhance communication. This aims to enable a more fulfilling and connected life despite the challenges posed by the disease.

Keywords: Amyotrophic lateral sclerosis (ALS) · neurological disorder · motor neurons · communication technology · eye-gaze technology · Phrase Bank · innovative solution · quality of life · genetic factors

1 Introduction

Amyotrophic Lateral Sclerosis (ALS), or Lou Gehrig’s disease, is a progressive neurodegenerative disorder affecting motor neurons, leading to loss of muscle control and impaired communication. While cognitive abilities remain intact,

80% to 95% of patients face severe communication barriers, causing frustration and isolation. In Pakistan, with a prevalence of 3.5 per 100,000, assistive communication technologies are scarce, leaving patients reliant on basic non-verbal methods like gestures.

2 Literature Review

Assistive communication technologies (ACTs) help ALS patients interact with their environment. This review highlights advancements and challenges in eye-tracking systems and brain-computer interfaces (BCIs).

Eye-Tracking Technologies: Eye-tracking technologies utilize users' eye movements to control cursors and select commands on a computer interface. Systems like those developed by Tobii Dynavox have revolutionized communication options for ALS patients by allowing them to use eye gaze as an input method for speech-generating devices and computers [1]. These technologies rely on infrared cameras to track the reflection from the user's retinas and determine the gaze direction accurately. While eye-tracking devices are invaluable for patients with limited mobility, they are not devoid of challenges. The equipment can be costly and cumbersome, limiting users' mobility and requiring frequent adjustments to maintain accuracy, particularly in different lighting conditions or as the disease progresses [2].

Brain-Computer Interfaces: BCIs represent a frontier in assistive technology by enabling direct communication through brain signals, bypassing the need for physical movement. The most common BCI systems used for ALS are based on electroencephalography (EEG), which records brain activity via electrodes placed on the scalp. The potential of BCIs in assisting ALS patients has been demonstrated through various studies; however, the practical application of these systems is often limited by their need for extensive calibration, susceptibility to noise and artifacts, and the cognitive load they impose on the user [3]. Moreover, BCIs tend to be expensive and require specialized knowledge to operate, which can be a significant barrier to widespread adoption.

Despite these advances, both eye-tracking systems and BCIs face significant integration challenges. One major issue is compatibility-many of the developed systems do not seamlessly integrate with other technologies that patients may use, such as household electronics and mobile platforms [4]. Furthermore, the physical setup required for some of these technologies may not be feasible in all living environments, particularly in settings without professional caregiving support.

In countries like Pakistan, where the literacy rate is relatively low, the language barrier poses an additional challenge for the deployment of advanced ACTs. Many systems are designed for users who are literate and fluent in international languages, typically English, which limits their accessibility for the local population. This gap necessitates the development of user interfaces and instructional content in local languages and potentially using symbols or intuitive

Despite progress, both technologies face integration challenges, often lacking compatibility with household electronics [4]. In Pakistan, low literacy rates and language barriers further limit adoption, necessitating localized, intuitive interfaces [5].

3 Methodology

3.1 Data Collection

The success of our data collection efforts for this project was significantly enhanced through strategic collaborations with local medical centers like **Liaquat National Hospital**, **Synapse Pakistan**, and **Ziauddin Hospital**, along with ALS support groups, and universities with robust neurology departments.

The survey specifically inquired whether the patients' vocal abilities were intact and requested detailed descriptions of their alternative communication methods, such as technological devices, gestures, muffled noises, or graphic tablets. Additionally, it sought information on the **primary language of communication**, the **most frequently used phrases by the patients**, and their **daily usage frequency**. To further validate the data's authenticity and the patients' consent for participation, doctors were also required to sign a consent form.

3.2 Data Cleaning

We received **135 phrases** from the survey, with **20 in Urdu** and **115 in English**. After translating these phrases between English and Urdu and providing Roman Urdu transliterations, we cleaned the data by removing duplicates, uninterruptible responses, and similar phrases. This process resulted in **90 unique phrases**. For analysis, we created word clouds in English and Roman Urdu to visualize phrase consistency and frequency. To assist patients with reading difficulties, we developed visual aids to convey phrase meanings through imagery, enhancing accessibility. This thorough approach has been crucial in tailoring our project to the communication needs of ALS patients and improving our application's usability.



Fig. 1. The word clouds indicating consistency in English and Roman Urdu responses.

3.3 Software Design Specification (SDS)

Functional Features. This section outlines the functional features for caretakers, doctors, patients, and the admin in both online and offline modes of our web-based communication application.

Account Management

- Create Account: Caretakers or doctors can register a patient by entering a name, email, and password, requiring an internet connection. The system saves the account and allows login when online.
- Login: Caretakers, doctors, or admins log in using email and password. Successful login directs them to the category page when connected to the internet.
- Logout: Users can log out, which clears cookies and redirects them to the landing page when connected.

Category and Phrase Management

- Category View and Selection: Logged-in users can view and select predefined categories offline or online. Patients use eye movement to select a category, leading them to the specific phrase bank.
- Phrase View and Selection: Users can view phrases in a phrase bank and patients can select phrases using eye movement. The system then generates Urdu audio output of the selected phrase.

Profile Management

- Profile Settings: Doctors and caretakers can modify settings, personal, and medical information, and deactivate accounts online. Admins can also modify and deactivate accounts.

System Architecture

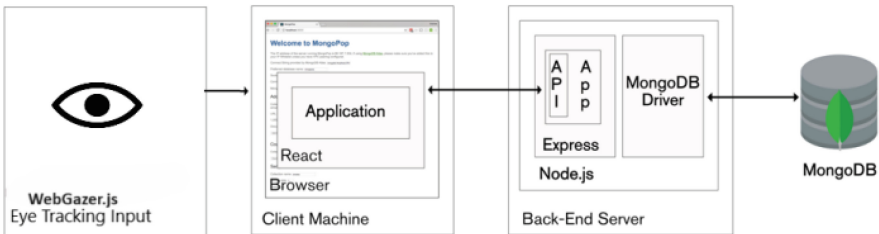


Fig. 2. High-level architecture diagram. - Description of components and their interactions. - Hardware and software requirements.

Eye Gaze Tracking Algorithm. We are using **WebGazer.js** to serve the purpose of **eye-tracking**. With the use of regular cameras, **WebGazer.js** is an eye tracking library that allows users to infer website visitors' gaze locations in real time. It works by tracking how users interact with a webpage, using this information to self-calibrate its eye tracking model and create a relationship between the positions of eyes on screen and eye features. **WebGazer.js**, which is fully written in JavaScript, integrates seamlessly with any website looking to improve user experience and comprehension. It runs completely inside the client's browser, removing the need to send video data to other servers, and it needs the user's express permission to access their webcam [6].

Key Features include:

- Real-Time Gaze Prediction: This feature allows you to predict where visitors are gazing on a webpage in real-time.
- Uses regular webcams that are frequently found in desktop and laptop computers, negating the need for additional hardware.
- Self-Calibration: Based on user inputs like clicks and cursor movements, the eye tracking model automatically modifies and improves.

We have also made use of **handsfree.js**. **handsfree.js** is a JavaScript library that enables hands-free interaction with web applications using computer vision techniques. It allows users to control and navigate web interfaces without the need for traditional input devices like keyboards or mice. Instead, **handsfree.js** utilizes the webcam feed from the user's device to recognize gestures, such as hand movements, head gestures, facial expressions, and most notably, blinks. One of the primary features of **handsfree.js** is blink detection. By monitoring the user's eye movements through the webcam, **handsfree.js** can detect when the user blinks and interpret these blinks as input commands. This functionality allows users to trigger actions, navigate through menus, make selections, and interact with content solely through their blinks.

Key Features include:

- Webcam Access: Handsfree.js uses the getUserMedia API to access the user's webcam stream directly from the browser, enabling real-time video capture without additional plugins.
- Image Processing: Handsfree.js applies image processing techniques, such as feature extraction and edge detection, to analyze each frame from the webcam feed.
- Computer Vision Algorithms: It utilizes computer vision algorithms, including pattern recognition and machine learning, to interpret the webcam feed and recognize gestures.
- Blink Detection: Blink detection is implemented by analyzing temporal changes in pixel intensity and motion patterns, using thresholds or heuristics to distinguish blinks from other facial movements.
- Gesture Recognition: Handsfree.js supports various gesture recognition features, including head motions, hand gestures, and facial expressions, each requiring specific models or algorithms for accurate detection.

Our 3rd library is **p5.js**. **p5.js** is a JavaScript library based on the widely used Processing language and environment, it extends the creative coding possibilities to the web rather than only desktop applications.

Key Features include:

- Web-Based: p5.js is fully web-based, in contrast to the desktop-based original Processing environment. This implies that you won't need any other software or plugins to develop and execute p5.js sketches in your web browser.

These 3 libraries are the stem of our project as we have combined them together and integrated them into our own application along with some refinements to serve our own purpose. Using the blink detection feature from **hands-free.js** and eye tracking feature from **WebGazer.js**, we devised a function that pauses, resumes, and selects the displayed phrases via blinking.

3.4 User Interface (UI) Design

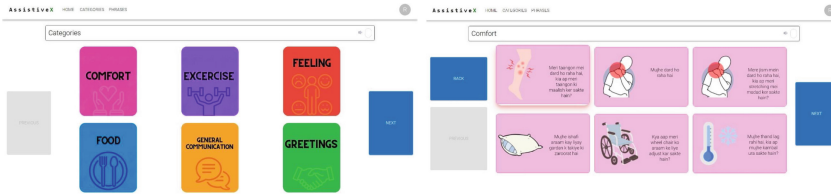


Fig. 3. Categories View and Phrases View of our Application.

3.5 Testing Parameters and Quality Assurance

Testing was conducted directly with patients, facilitated by healthcare professionals consulted for the study. Using comprehensive test parameters, the functionality and performance of each feature was rigorously validated. This approach ensured that the system meets the highest standards of accuracy and effectiveness.

Our testing phase involved six tasks for evaluation, each derived from our extensive literature review on eye-tracking technologies and their application in enhancing communication for individuals with speech impairments. These tasks were designed to thoroughly assess various aspects of our application, including but not limited to:

- Task 1: Eye Tracking Functionality Assessment
 - Objective: Evaluate the accuracy and responsiveness of the eye gaze tracking system across various predefined screen locations.

- Instructions for Participants: Patients were asked to look at each of the target areas on the screen when instructed. These areas included the top left, top right, bottom left, bottom right, extreme left middle, extreme right middle, top middle, and bottom middle of the screen.
- Task 2: Lighting Conditions and Eye Gaze Tracking Performance
 - Objective: Determine the effectiveness and comfort of using the eye gaze tracking system under different types of lighting conditions.
 - Instructions for Participants: Patients were asked to use the eye gaze tracking system under different types of lighting such as bright light (like natural light), soft light (like from a lamp), and mixed light, and on the basis of this eye gaze tracking performance was evaluated.
- Task 3: Assessing the Impact of Shadows on Eye Gaze Tracking
 - Objective: Understand how shadows cast on the patient’s face from various directions impact the performance of the eye gaze tracking system.
 - Instructions for Participants: We used a torch to cast shadows on different parts of the patient’s face, such as the right side, left side, and directly above. We aimed to see if these shadows affect how well the eye gaze tracking system works.
- Task 4: Head Pose and Distance from Screen Assessment
 - Objective: Evaluate the system’s tolerance to different head poses and distances from the screen, identifying the practical limits for accurate gaze tracking.
 - Instructions for Participants: 1) Patients were asked to turn their head to specific positions and then try to use the eye gaze tracking system. Angles assessed were: Head Pose (Roll, Pitch, Yaw Angles) 2) Patients were asked to use the system from different distances from the screen.
- Task 5: Display Variability and Platform Movement Assessment
 - Objective: Assess how various display sizes and resolutions influence the accuracy of the eye gaze tracking system, while simultaneously evaluating the system’s performance across static and dynamic platforms.
- Task 6: Performance Evaluation Under Various User Conditions
 - Objective: Assess the eye gaze tracking system’s adaptability and performance when users wear glasses, have occlusions, droop their eyelids, or blink rapidly.

Each task was accompanied by a detailed survey, developed to capture thorough feedback on the application’s performance and user experience. In addition to the primary survey, we also created a version for patients who are able to complete the form on their own. This allows them to directly provide feedback on the system’s responsiveness, user interface, and usability. The integration of professional medical advice, rigorous participant recruitment protocols, and a comprehensive testing framework highlights our commitment to developing a web application that is not only technologically advanced but also practical, user-friendly, and ethically responsible. We have completed the initial testing phase with the minimum required number of patients, which is three, but we are continuing to carry out testing with additional patients to obtain more comprehensive results. This ongoing effort is aimed at further refining our understanding and enhancing the effectiveness of our project.

3.6 Testing Analysis

We conducted tests of our eye-tracking application on three patients to evaluate its performance across different conditions and tasks. Below is an analysis of how our system responded to each patient, who all have varying degrees of communication and movement difficulties:

Participants:

- Patient 1: Male with attention deficit disorder
- Patient 2: Male with muscle dystrophy
- Patient 3: Female with cerebral palsy (CP)

The primary objective of Task 1 was to assess the accuracy and responsiveness of the eye gaze tracking system in detecting and following the gaze across various predefined locations on the screen. This evaluation was crucial to determine how well the system could facilitate communication for users with different physical disabilities. The patients were asked to look at screen targets for gaze tracking that included top, bottom, left, right, and center positions, each critical for a comprehensive assessment of the system’s functionality across the screen. In this task, Patient 1 exhibited a competent level of interaction with the eye gaze tracking system. He was able to successfully look at each target area as requested, suggesting that the system is responsive, overall for him the responsiveness was rated as “Moderate”. This indicates that the system can handle minor attention fluctuation without losing tracking efficacy. Patient 2 demonstrated an impressive ability to engage with all the targeted areas on the screen, rating the system’s responsiveness as “Fast”. This result is particularly encouraging as it suggests that the system is capable of providing efficient support even for users with physical disabilities such as muscle dystrophy, which can significantly impair movement control. Patient 3, the child with CP, encountered significant challenges with the system. She was only able to effectively engage with the top left and bottom middle areas of the screen, with all other positions being problematic, resulting in a “Very Slow” responsiveness rating. This was mainly due to the child’s disorder as the child was not usually responsive to things, but the little response she showed in our application was a positive sign. The difficulties faced by the patients highlights the need for the system to be more adaptable and sensitive to users with severe motor restrictions and suggests potential calibration issues that might not be adequately tailored to the unique needs of users with acute conditions like CP, indicating a critical area for development.

Furthermore, the objective of Task 2 was to evaluate how different lighting conditions influence the effectiveness and user comfort when interacting with the eye gaze tracking system. This evaluation is crucial in understanding environmental adaptability of the technology, especially for users with varied and specific needs. In this task, Patient 1 experienced ease of use under most lighting conditions, including bright, soft, and typical indoor environments, with only mixed lighting proving to be slightly challenging. This patient noted that varying the lighting conditions to be bright or normal light generally improved the tracking

performance, suggesting that the system might be using environmental light to enhance sensor accuracy. Patient 2 found the eye gaze tracking system easy to operate under bright and normal indoor lighting but faced difficulties with soft and mixed lighting conditions. Despite these challenges, this patient observed improvements in tracking accuracy when lighting conditions were altered, indicating some adaptive capabilities of the system that might need further optimization for consistency across all environments. Patient 3 again encountered significant challenges under all tested lighting conditions, consistently rating the ease of use as “Very Hard”. This patient reported a deterioration in system performance with changing lighting, highlighting a pronounced sensitivity to environmental conditions that could severely limit usability for individuals with more severe impairments. From this, we can conclude that further development should focus on ensuring that the system can maintain high performance regardless of environmental light changes, thus supporting a wider range of users in diverse settings.

Moreover, the objective of Task 3 was to understand how shadows cast on the patient’s face from various directions impact the performance of the eye gaze tracking system. We used a torch to cast shadows on different parts of the patient’s face, such as the right side, left side, and directly above. The goal was to determine if these shadows affected how well the eye gaze tracking system worked. The results showed that shadows significantly impacted the tracking effectiveness across all patients. Each patient reported difficulties when shadows were casted from any direction, noting that it made the system much more difficult to use. Specific problems included the cursor moving more slowly and not following the patient’s gaze accurately. In some cases, patients also reported increased eye strain or discomfort, which further detracted from the user experience. These findings suggest that our system needs to improve its accuracy and reliability under varying lighting conditions. Implementing more robust shadow compensation algorithms could mitigate these issues, ensuring the system remains functional and user-friendly even in less-than-ideal lighting conditions. Further testing with different light intensities and angles may also help refine these algorithms, providing a more adaptive and resilient eye tracking solution.

The objective of Task 4 was to evaluate how well the eye gaze tracking system could follow the user’s gaze with different head orientations. The results varied significantly among the patients, indicating the system’s current limitations in effectively handling diverse head movements. Both Patient 1 and Patient 2 experienced difficulties with almost all head positions, except when looking directly at the screen or upwards. These limitations suggest that the system struggles with tracking when the user’s gaze is not aligned directly with the sensor, highlighting a potential area for improvement in eye gaze sensor sensitivity and gaze prediction algorithms. The results for Patient 3 are particularly concerning, as they indicate a stark decrease in functionality, severely limiting the usability of the system for individuals with similar conditions. From these results, we can conclude that future work should focus on ensuring broader usability and

effectiveness, especially for users with physical conditions that prevent maintaining a steady, direct gaze. Enhancements in tracking algorithms, potentially incorporating machine learning techniques or more adaptive sensor technologies, will be essential to address these issues. This will improve the system's overall responsiveness and reliability, making it more accommodating for a wider range of users.

In Task 5, we assessed how various display sizes and resolutions influenced the accuracy of the eye gaze tracking system, while also evaluating the system's performance across static and dynamic platforms. The findings revealed that the system for all patients did not perform well on smaller screens like those of phones or tablets, with comparatively better performance on laptop screens. The sharpness of the display was crucial, as clearer screens significantly enhanced tracking accuracy. Moreover, movement, such as walking with the device or turning the head, generally made tracking worse, indicating that the system's current capabilities are limited when used in motion.

Finally, Task 6 focused on assessing the eye gaze tracking system's adaptability and performance when users wore glasses, had occlusions, or when their eyelids drooped or they blinked rapidly. The results were mixed. Although none of the patients wore glasses or had occlusions, rapid blinking or drooping eyelids significantly impaired the system's ability to track gaze accurately, often rendering it unusable. This highlights the importance of further refining the system to accommodate these common conditions more effectively.

From these evaluations, it is clear that while the eye gaze tracking system shows promise, several areas require improvement to ensure it can effectively support all users, especially those with severe disabilities or in less than ideal conditions. The system performs well under ideal conditions, but its performance deteriorates as conditions become varied or more complex. To address these challenges and enhance the system's robustness and adaptability, further research and development are essential.

3.7 Conclusion and Future Work

In summary, the development of our web application represents a major achievement in tackling the communication obstacles encountered by ALS patients. Through the integration of phrases, visuals, and voice output capabilities, along with enabling communication via eye gaze technology, our application offers a user-friendly solution for individuals with restricted mobility to convey their thoughts and feelings effectively. The ability to select phrases and receive voice output in the user's preferred language, such as Urdu, promotes accessibility and inclusivity for a wide range of individuals. We have carefully designed our application to guarantee it meets strict criteria for accuracy, performance, and security. Moving forward, we envision our project making a meaningful impact on the lives of ALS patients, empowering them to communicate with greater ease and independence. Our future works includes:

- Improving scalability: We aim to enhance the scalability of our application, ensuring seamless performance and usability across a range of devices such as mobile phones and tablets, as well as various user scenarios.
- Improving Eye Tracking Algorithm Accuracy: We are committed to enhancing the accuracy of our eye tracking algorithm to maximize its benefits for ALS patients. Currently, the eye tracking deviation error stands at 2.26 degrees, and we aim to significantly reduce this error rate. Through rigorous testing and optimization efforts, we will refine the algorithm to ensure precise and reliable selection of phrases via eye tracking, thereby improving the overall user experience and effectiveness of our application.

References

1. Fager, D.J., et al.: Use of augmentative and alternative communication strategies by individuals with amyotrophic lateral sclerosis. *J. Med. Speech-Lang. Pathol.* **21**(3), 291–309 (2013)
2. Caligari, S., et al.: Eye-tracking communication devices in amyotrophic lateral sclerosis: impact on disability and quality of life. *ALS and Frontotemporal Degeneration* **14**(7–8), 546–552 (2013)
3. Wolpaw, J., Wolpaw, E.W. (eds.): *Brain-Computer Interfaces: Principles and Practice*. Oxford University Press, Oxford (2012)
4. Fager, C., Burnfield, S.: Patients with ALS can use sensorimotor rhythms to operate a brain-computer interface. *Neurology* **77**(5), 392–397 (2011)
5. Ahmad, A., et al.: Addressing the language and literacy barrier in Pakistan using ICT. *Asian J. Educ. e-Learning* **4**(2), 92–99 (2016)
6. Papoutsaki, A., Sangkloy, P., Laskey, J., Daskalova, N., Huang, J., Hays, J.: WebGazer: scalable webcam eye tracking using user interactions. In: *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 3839–3845. AAAI Press, New York (2016). <https://webgazer.cs.brown.edu/>



Autonomous Navigation in Swarm of UAVs Using Spatio Temporal Data and Constrained-Reinforcement Learning

Abhudaya Shrivastava^(✉), Christos Petridis, Marijana Vacic,
and Zoran Obradovic

Department of Computer and Information Sciences,
Temple University, Philadelphia, PA 19122, USA
{abhudaya.shrivastava,zoran.obradovic}@temple.edu

Abstract. Obstacle avoidance in aerial robotics remains a critical challenge, particularly in environments with uncertain terrain and weather conditions. This study introduces a Constrained Q-Learning model that leverages spatio-temporal data from LiDAR and OctoMap to achieve Zero-Shot Execution (ZSE) for autonomous Unmanned Aerial Vehicle (UAV) navigation in unseen environments, eliminating the need for iterative training. Experimental evaluations are conducted using a high-fidelity simulator across three environments: random forests, clustered forests, and metropolitan areas, under varying obstacle densities and flight velocities. The proposed model demonstrates a 100% success rate, achieving average flight times of 45 s for slow velocities (below 2.5 m/s) and 34 s for fast velocities (above 2.5 m/s). Comparative analysis with 90 Human-to-Computer (HTC) flights (slow and fast velocities), conducted by three pilots under identical conditions, shows the proposed model reduces flight time by 33% (1.5 times faster) while enhancing path optimization. Additionally, the model matches the path selection efficiency of a standard Q-Learning approach without requiring iterative training, highlighting its robustness and scalability for autonomous UAV navigation in complex environments and GPS-denied locations.

Keywords: Reinforcement Learning (RL) · Constrained Reinforcement Learning (CRL) · Zero-Shot Execution (ZSE) · Human-To-Comuter (HTC) · Human Computer Interaction(HCI) · Autonomous navigation · Computer Vision (CV) · OctoMap

1 Introduction

Aerial robotics, driven by UAV advancements, has expanded into applications like disaster response, environmental monitoring, and urban planning. UAVs leverage spatio-temporal data for enhanced navigation, optimizing flight paths in complex environments. Reinforcement Learning (RL) further improves autonomy by enabling UAVs to adapt in real-time, mitigating the impact of noisy or incomplete sensor data.

This paper introduces a Constrained-RL (CRL) model for swarm UAV path planning and obstacle avoidance using computer vision (3D LiDAR and Octants) to obtain obstacle locations in various environments. The CRL model integrates Q-learning with heuristic constraints, resulting in zero-shot execution (ZSE), which optimizes the swarm UAVs' flight time and path selection to reach the destination without requiring training. Furthermore, a comparative analysis highlights the model's advantages in path optimization, collision avoidance, and flight time efficiency over state-of-the-art RL models and the Human-To-Computer (HTC) approach based on Human-Computer Interaction (HCI).

The proposed framework enables ZSE, reducing computational overhead while improving real-time decision-making due to ZSE-CRL framework. This can be used as an advantage in various applications including but not limited to disaster response, fire detection, and pollution monitoring, where rapid and adaptive UAV navigation is critical, often with limited prior data, making traditional navigation models inefficient and less adaptable [1]. By eliminating dependence on iterative training, this approach outperforms existing RL and human-to-computer (HTC) approaches, ensuring greater reliability, efficiency, and adaptability in unseen environments.

2 Related Work

Autonomous UAV navigation in complex and dynamic environments has gained significant attention, driven by advancements in multi-agent reinforcement learning (MARL), deep reinforcement learning (DRL), and evolutionary algorithms. Shi et al. [2] proposed a MARL approach with Adaptive Dimensionality Reduction (ADR) to enhance spatial data processing and coordination. Sheng et al. [3] improved DRL-based navigation with dynamic reward functions and a novel state-space representation, while Meng et al. [4] introduced a Cooperative Co-Evolutionary Algorithm (CCEA) for scalable swarm operations. Wang et al. [5] addressed real-time collision-free navigation with the APPA-3D algorithm, and Zhang et al. [6] optimized UAV routing in Mobile Edge Computing (MEC) environments.

Chenyuan He et al. [7] developed a spatiotemporal data-driven framework for UAV path planning in wind fields, significantly reducing decision time. Prior studies, including Chao Yan et al. [8], Bingze et al. [9], and Ronglei Xie et al. [10], achieved navigation improvements but required extensive training episodes or were constrained to specific scenarios.

This paper introduces the Zero Shot Execution-Constrained Reinforcement Learning (ZSE-CRL) framework, which integrates Q-learning with constraint-aware heuristic planning to enhance real-time UAV navigation without iterative training. Unlike prior models, ZSE-CRL leverages spatiotemporal data obtained using 3D LiDAR and OctoMap for immediate decision-making, improving adaptability and computational efficiency in unseen environments. Comparative evaluations against HTC and RL approaches highlight ZSE-CRL model's superiority

in flight time optimization, path accuracy, and decision-making efficiency, making it a robust solution for UAV swarm autonomous operations.

3 Material and Methods

This section outlines the real-time UAV swarm navigation framework, detailing methodologies for path planning, obstacle avoidance, and computational efficiency in complex and unseen environments.

3.1 Framework: Spatio-Temporal Data, OctoMap, and CRL

The framework combines spatio-temporal data and OctoMap to enhance CRL for UAV swarm autonomous navigation, enabling adaptability, optimized path planning, and efficient operation in complex environments (Fig. 1).

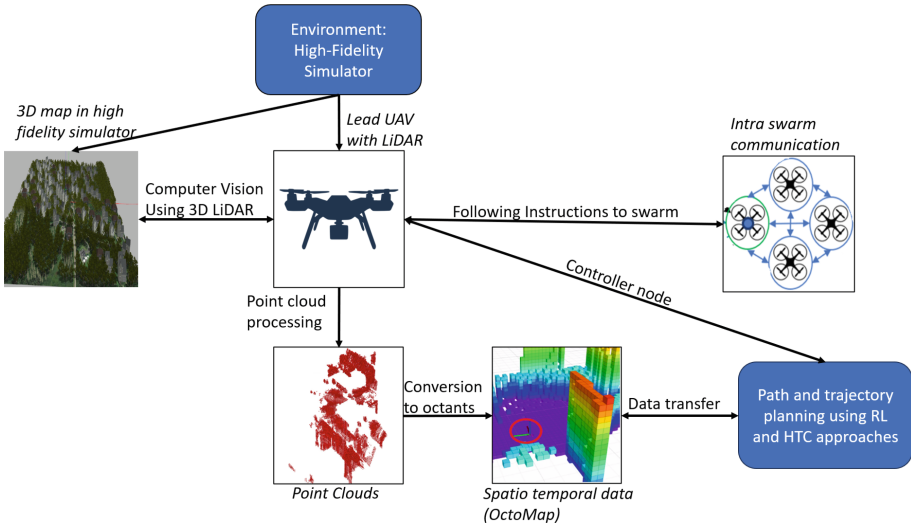


Fig. 1. The framework utilizes spatio-temporal data and octo-mapping to develop a reinforcement learning algorithm for autonomous UAV swarm navigation.

Figure 1 shows the framework employed for autonomous navigation. The explanation of each part of the framework is as follows:

Geometry Synthesis and High-Fidelity Simulator: For high-fidelity simulation, 3D environment maps (Tables 1, and 2) are generated in Gazebo. The

most common shapes used to construct 3D buildings and trees are shown below, along with their influence on the planned paths.

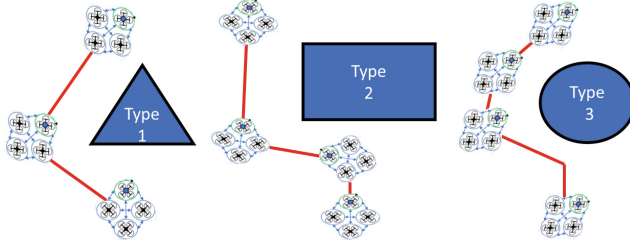


Fig. 2. Influence by shapes on path planning.

Figure 2 emphasizes the significance of spatial awareness in effective path planning, where the geometry of shapes affects the types of paths requiring need for geometry synthesis. This presents a challenge for real-time data processing and mapping in autonomous navigation using the ZSE-CRL model.

Computer Vision, Spatio-Temporal Data, and Path Planning: As shown in Fig. 1, obstacle point clouds are converted into octants to enhance computer vision for the swarm, optimizing path planning. This process is further detailed in Fig. 3.

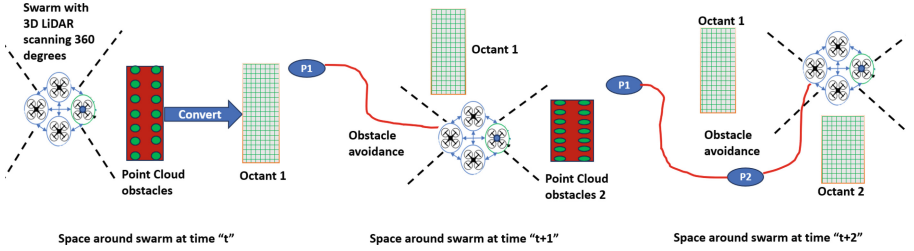


Fig. 3. Computer Vision: Lead UAV based octants mapping for obstacle avoidance and detection in different spaces and times.

Figure 3 illustrates the use of computer vision by octants mapping [21] instead of point clouds for UAVs internal mapping. The lead UAV performs a 360-degree LiDAR scan and employs a probabilistic approach to convert point clouds into a memory-efficient 3D grid-based OctoMap, as seen in octants 1 and 2. This enhances the efficiency of the ZSE-CRL model by enabling faster obstacle mapping and path computation from time t to $t+2$ [21]. The next section explores how optimized path planning using the ZSE-CRL framework complements real-time spatiotemporal data.

3.2 Spatio-Temporal Path Planning

Reinforcement Learning (RL) enables agents to optimize decision-making through trial and error, maximizing cumulative rewards. While traditional RL seeks an optimal policy without constraints, Constrained Reinforcement Learning (CRL) incorporates safety, performance, and resource limitations to ensure adherence to predefined boundaries. The ZSE-CRL framework enhances spatio-temporal awareness by utilizing real-time updates from the UAV's LiDAR-based OctoMap. As the lead UAV scans its surroundings, it converts point clouds into octants, improving memory efficiency and obstacle representation over time. These dynamically updated octants provide a structured probabilistic map, aiding informed decision-making for navigation [21]. By integrating spatial and temporal data, CRL enables UAVs to adapt their path planning in real-time while maintaining efficiency and constraint adherence. This ensures optimized trajectories and effective obstacle avoidance, enhancing autonomous navigation. The internal process works as:

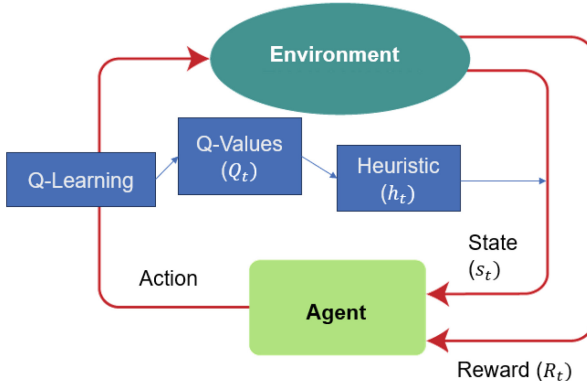


Fig. 4. The proposed reinforcement learning framework for autonomous navigation for a single UAV (one agent) and swarms of UAVs (multi-agent).

Figure 4 illustrates the CRL framework, which combines **Q-learning** with **heuristics**. Agents make decisions using **Q-values**, which are influenced by heuristic, and update their **state** at time t after each **action**. These state updates reflect the agent's position and environment at time t , influencing subsequent decisions. The agents then receive **rewards** based on their performance, which are used to further refine their decision-making process over time. This framework allows the agents to adapt dynamically to changes in their environment at each time step.

The key innovation of how heuristics work and produce a ZSE approach is explained and visualized using Fig. 5 below.

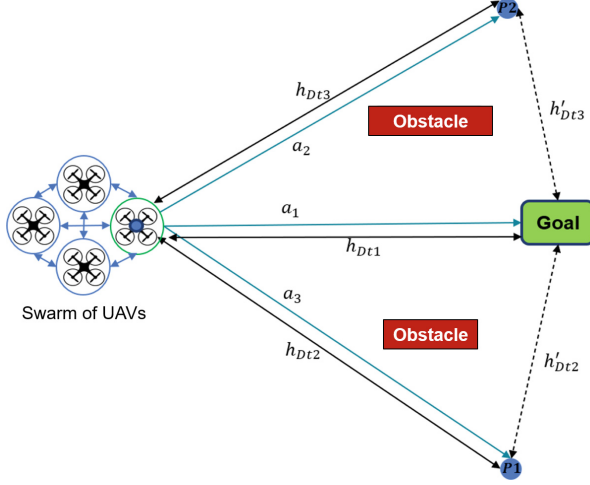


Fig. 5. Visualization of heuristic-based path selection, showing obstacles, points (P_1 , P_2), heuristic distances at time step t (h_{Dt1} , h_{Dt2} , h_{Dt3} , h'_{Dt1} , h'_{Dt2} , h'_{Dt3}), and actions (a_1 , a_2 , a_3), highlighting the optimal path to the goal.

3.3 Zero Shot Execution (ZSE)

The heuristic at time t guides the swarm’s decision-making by modeling path planning as nodes in a graph, evaluating distances D for each node as h_{Dt1} , h_{Dt2} , h_{Dt3} , h'_{Dt1} , h'_{Dt2} , and h'_{Dt3} while avoiding obstacles P_1 and P_2 . The swarm evaluates three actions: a_1 , a_2 , and a_3 , with a_1 being optimal due to the shortest distance h_{Dt1} . Actions a_2 and a_3 lead via obstacles P_1 and P_2 , respectively. In real-world scenarios, multiple minimum distances are evaluated based on spatiotemporal data obtained from computer vision, connecting different points to form the optimal path.

This method eliminates iterative training, ZSE with CRL, allowing the swarm to adapt autonomously in real-time autonomous navigation (Fig. 1). To explore the ZSE-CRL framework’s potential, it is crucial to examine its computation in terms of action space, hyperparameter sensitivity, rewards, and policies.

3.4 Computation Under ZSE-CRL

Since the ZSE-CRL model is part of RL structure, the key configurations are:

- **Actions and Hyperparameters:** The ZSE-CRL model exhibited no sensitivity to hyperparameter changes due to its use of a greedy deterministic policy (discussed in the results section). The comparison RL model, hyperparameters selection followed experimentation within the range 0.0001 to 1.5, with the testing values chosen due to the negative impact of others on performance. The learning rate was tested at 0.0001, 0.001, and 0.01, while other hyperparameters were fixed: $\gamma = 0.95$, $\epsilon_{\text{init}} = 0.8$, $\epsilon_{\text{min}} = 0.05$, and $\epsilon_{\text{decay}} = 0.98$ (s). Actions were defined in eight cardinal and diagonal directions.

- **Rewards and Policies:** The reward system consists of multiple components to guide UAV navigation. **Destination Reward** grants rewards of +10 upon reaching the destination. **Move Penalty** assigns -1 if not exploring, while **Collision Penalty** applies -1 for obstacle collisions. Conversely, **No Collision Reward** grants +1 for safe movement. To promote vertical navigation, **Y-Coordinate Change Reward** provides +1 for changing the y-coordinate. The **Avoid Points Penalty** imposes -1 for remaining at the same x and y coordinates. Moving closer to the goal is +1 reward with **Closer to Goal Reward**, whereas **Penalty for Not Getting Closer** deducts -1. Lastly, **Bubble Rebound Penalty** assigns -1 for having an obstacle in the defined bubble radius also called a dynamic safety radius around UAV/UAVs (Figs. 1, 2, 3, and 4).

Deriving from the general Q-learning equation, the ZSE-CRL approach incorporates a heuristic term, yielding the following result at time step t :

$$Q(S_t, a_t) = Q(S_t, a_t) + \alpha \cdot \left(r - \gamma \cdot \min_{a'_t} Q(S'_t, a'_t) - Q(S_t, a_t) \right) \quad (1)$$

From Eq. 1 and Fig. 5, the heuristic is defined as:

$$\min Q(S'_t, a'_t) = h_{D_t} \quad (2)$$

Thus, as the distance to the goal increases, the Q-value decreases, leading to action selection based on:

$$a_{selected_t} = \operatorname{argmax}_{a_t} (Q(S_t, a_t)) \quad (3)$$

Selecting the highest Q-value for action a_t in state S_t at time t ensures optimality, but obstacles and safety constraints may lead to invalid options. To address this, three policies are introduced to improve decision-making:

The **Rule-Based Policy with Constraint** filters actions based on distance to obstacles, ensuring $d(s', \text{obstacle}) > \text{threshold}$. The **Filtered Stochastic Policy** allows stochastic action selection from valid options using heuristics. The **Heuristic-Guided Deterministic Policy** selects the optimal action $a^* = \operatorname{argmax}_{a \in A_{\text{valid}}} Q(s, a)$. The **Reactive Policy with Constraint (Bubble Rebound)** triggers a **Rebound Action** when the UAV's distance to an obstacle $d(s_t, \text{obstacle}) \leq r_{\text{safe}}$, enabling evasive maneuvers for collision avoidance of the swarm (Figs. 3 and 5).

3.5 Lead UAV and Intra-swarm Communication

The following section explains the leader-offset UAV communication for path computation, obstacle avoidance, and mapping. Using the Leader-Follower Model [14], the lead UAV communicates with Relay UAVs (three offset UAVs in this study), which then relay information to the follower UAVs, optimizing communication and reducing the load on the lead UAV for efficient swarm coordination [14].

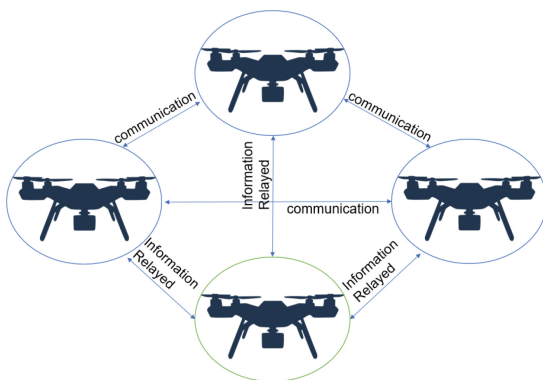


Fig. 6. A resilient robust network for communication between the UAVs in the swarm and relaying information from the lead to the offset UAVs.

As shown in Fig. 6. This communication framework also enhances fault tolerance by providing redundant communication paths and dynamic reassignment of relay roles, ensuring consistent updates in complex environments. The framework in the UAV swarm operates through a hierarchical structure for efficiency and scalability. The *Lead UAV* transmits critical state information to the selected *Relay UAVs* [15], which then forward it to neighboring *Follower UAVs* within their range [16]. This reduces the communication load on the lead UAV by avoiding direct communication with every follower. Redundancy is ensured through multiple *Relay UAVs*, enhancing fault tolerance as others can take over in case of failure [18]. Furthermore, the lead UAV can dynamically reassign relay roles as needed, optimizing mobility and system reliability [19]. This approach minimizes bandwidth usage and scales efficiently with larger swarm sizes [20].

4 Setup of the Study

The setup comprises of deploying UAVs type x500 in high fidelity simulator. Using the UAVs type x500 on described environments in the Table 1, and 2 for swarm of UAVs. The visualization of these environments is as follows:

Table 1. 3D Views of Random Forests maps

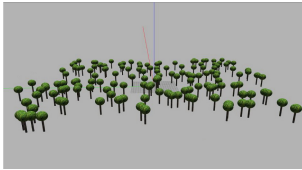
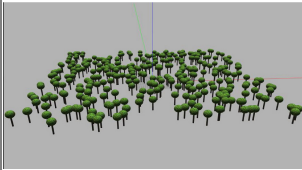
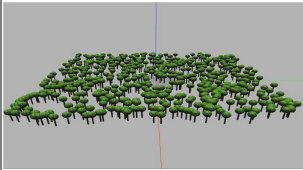
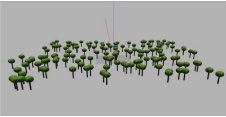
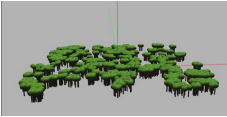
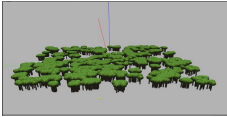
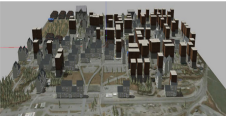
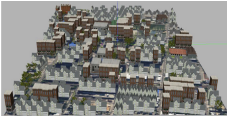
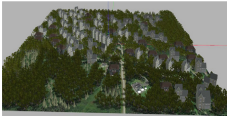
Low-Density	Medium-Density	High-Density
Random Forests	Random Forests	Random Forests
		

Table 2. 3D Views of Clustered Forests and Metropolitan maps

Low-Density	Medium-Density	High-Density
Clustered Forests 	Clustered Forests 	Clustered Forests 
Anchorage Downtown Alaska 	North Central Philadelphia Pennsylvania 	Eagle River Suburbs Alaska 

5 Results

The results are divided into two subsections: (5.1) Comparison of CRL and HTC approaches, considering psychological factors in decision-making, and (5.2) Evaluation of flight time optimization vs. cost, and the reliability of ZSE-CRL compared to state-of-the-art models.

5.1 ZSE-CRL and HTC Approaches

The ZSE-CRL is a non-iterative RL model for autonomous navigation, while the HTC approach involves human-computer interaction. To compare, three pilots performed 90 flights at slow (speed < 2.5 m/s) and fast (speed > 2.5 m/s) speeds across the environments in Tables 1 and 2. One example of decision-making from these flights is as follows:

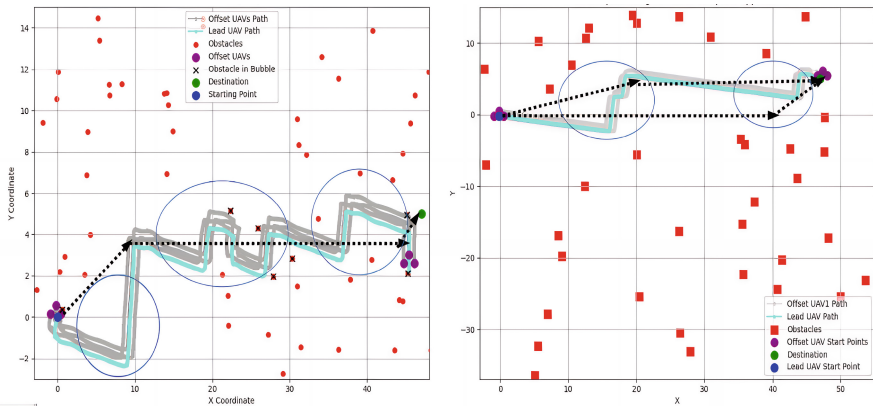
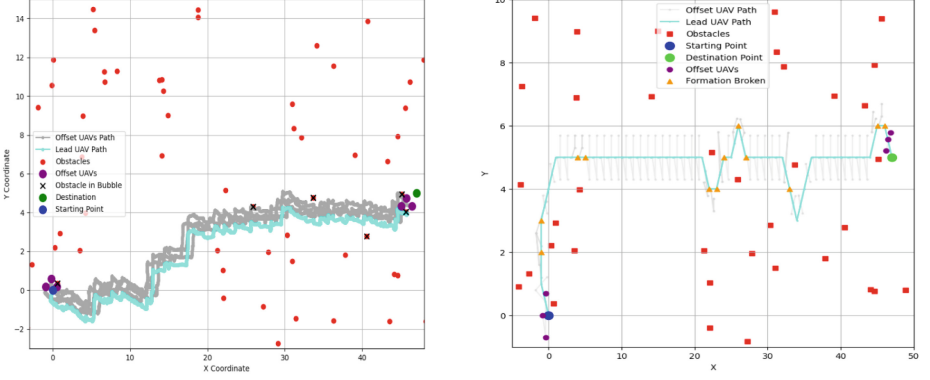
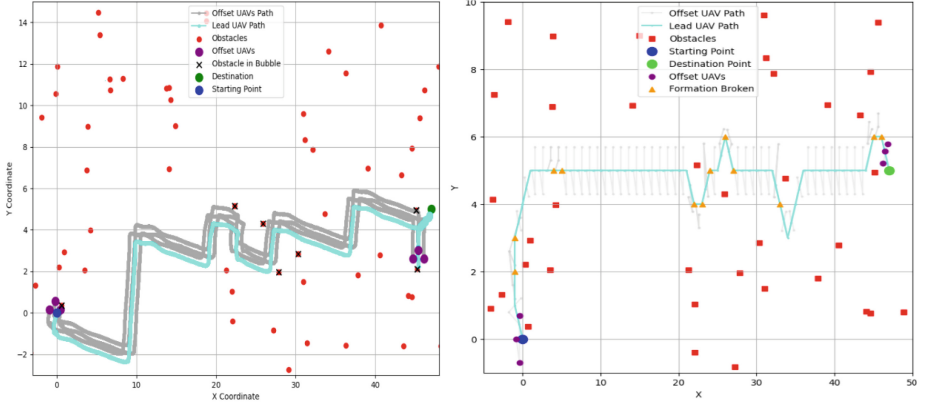


Fig. 7. HTC in random forest (left panel) and North Philadelphia (right panel).



(a) Slow flight in medium-density random forest overview using HTC (left panel) vs. CRL (right panel, zoomed in).

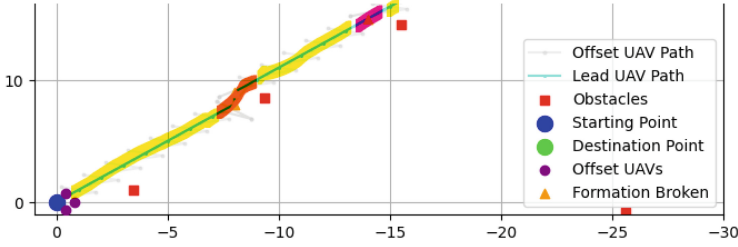


(b) Fast flight in medium-density random forest overview using HTC (left panel) vs. CRL (right panel, zoomed in).

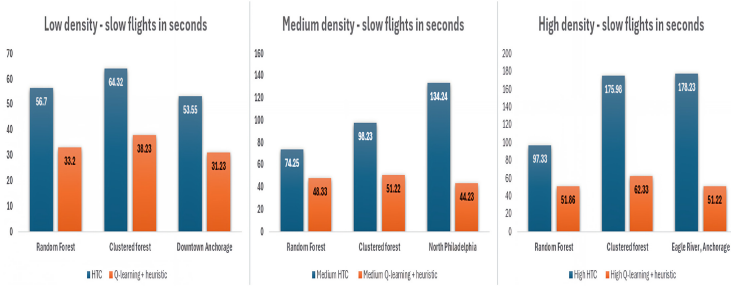
Fig. 8. Comparison of HTC and CRL in different environments and flight conditions.

Figure 7 illustrates the decision-making process in path selection, where the left and right panels reveal alternative, more optimal routes overlooked by pilots. When comparing the ZSE-CRL framework to the HTC approach (left panel), the decision-making strategy becomes evident. Figures 8a and 8b further emphasize the advantages of the ZSE-CRL model, which exhibits straight paths with sharp, well-defined turns, in contrast to the erratic turns observed in the HTC model. This trend remains consistent across various environments, demonstrating the ZSE-CRL's superiority in autonomous navigation and optimal path selection. A comparative analysis of flight time optimization with the HTC approach follows.

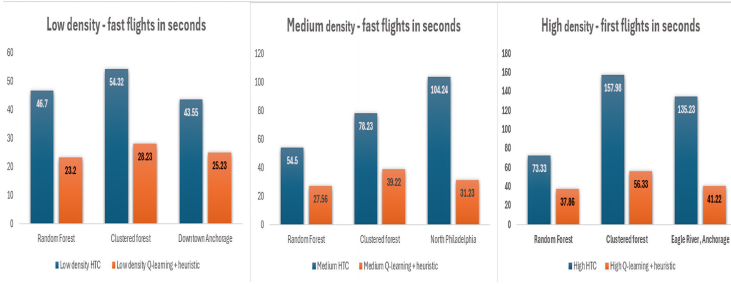
Time Optimization: The time optimization analysis of the ZSE-CRL model is explained and compared as described in the figures.



(a) A random ZSE-CRL generated path illustrating velocities. Yellow represents a straight path, orange indicates swarm turning, and pink marks formation break.



(b) Average flight time under slow velocity across 45 experiments in different densities and environments.



(c) Average flight time under fast velocity across 45 experiments in different densities and environments.

Fig. 9. Comparison of HTC (blue) and ZSE-CRL (orange) across different environments and flight conditions. (Color figure online)

To calculate the total flight time, we use the relationship between distance, velocity, and time, where time is the ratio of distance to velocity. Specifically, the time taken for a UAV to travel a given path is determined by:

$$\text{Time (t)} = \frac{\text{Distance (d)}}{\text{Velocity (V)}}$$

Figure 9a illustrates how straight paths maintain velocities of 2.0 m/s (fast) and 1.5 m/s (slow), while turns and formation breaks are assigned a velocity

of 0.5 m/s. By applying this structured velocity assignment, the swarm is able to avoid high-speed banking collisions and internal interactions. Performance is then compared to the HTC approach vs the ZSE-CRL model.

Furthermore, as seen in Figs. 9b and 9c, the ZSE-CRL model (which combines Q-learning with heuristic) completes the flights in approximately 1.5 times less time than the HCI based HTC approach, showcasing its enhanced efficiency in path selection, decision-making, and overall flight time optimization.

5.2 ZSE-CRL Vs State-Of-The-Art Models

Based on the path selection and time optimization results from Figs. 8a, 8b, 9a, and 9b, ZSE-CRL excels in selecting optimal paths and optimizing flight time. To further validate its effectiveness, we compare the proposed model with an RL model using the framework in Figs. 1 and 6, as well as other popular approaches from similar case studies as described in related work section.

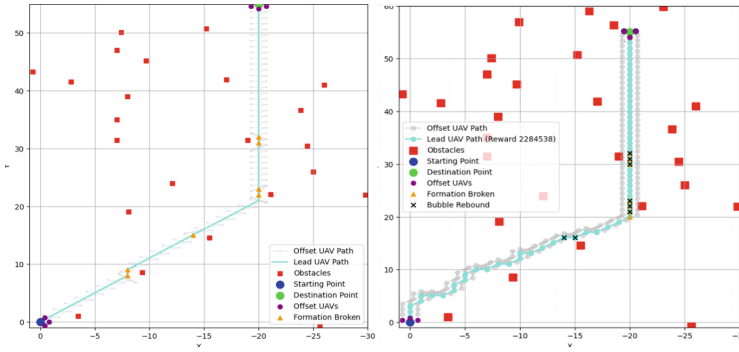


Fig. 10. Reliability for optimal path selection: ZSE-CRL model (left) vs. Q-Learning with 2500 learning episodes (right).

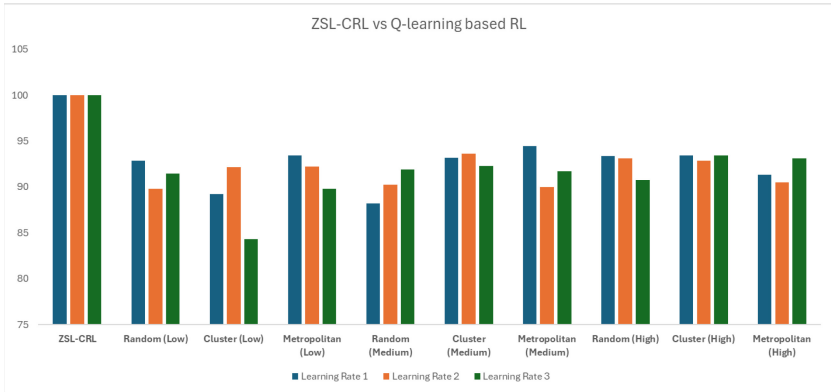


Fig. 11. Success rates for ZSE-CRL vs. Q-learning model (2500 learning episodes).

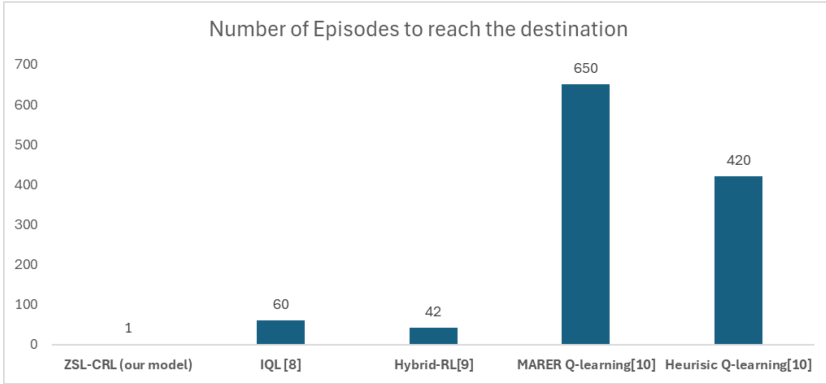


Fig. 12. ZSE-CRL: Number of episodes to reach the destination vs. other researched models.

Fig. 10 demonstrates that the ZSE-CRL model outperforms Q-learning by successfully completing the task in a single attempt with minimal formation breaks and a straighter path selection, eliminating the need for multiple learning episodes. This efficiency allows it to effectively avoid bubble rebounds and unnecessary turns. Figures 11 and 12 further validate its superiority, highlighting a 100% success rate and the ability to reach the goal within a single episode outperforming other comparable models.

6 Conclusion

The ZSE-CRL model, developed using Q-learning and heuristics, enables UAV swarms to autonomously navigate with 100% success rates. The study highlights the importance of environmental context in navigation and emphasizes the need for advancements in reinforcement learning and intra-swarm communication. A key shortcoming is the model's current limitation to static environments. Future work is focused on incorporating dynamic obstacles, environmental disturbances, and reducing reliance on leader UAVs. Expanding the model's adaptability will be essential as UAV swarm applications evolve in more complex terrains, which we are actively working towards.

Acknowledgments. This research was sponsored by the U. S. Army Engineer Research and Development Center (ERDC) and was accomplished under Cooperative Agreement Number W9132V-23-2-0002. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Engineer and Development Center (ERDC) or the U.S. Government.

References

1. Chen, X., et al.: Wildland fire detection and monitoring using a drone-collected RGB/IR image dataset. *IEEE Access* **10**, 121301–121317 (2022)
2. Shi, H., Zhao, Z., Chen, J., Zhou, M., Liu, Y.: Enhancing unmanned aerial vehicle path planning in multi-agent reinforcement learning through adaptive dimensionality reduction. *Drones* **8**, 521 (2024)
3. Sheng, Y., Liu, H., Li, J., Han, Q.: UAV autonomous navigation based on deep reinforcement learning in highly dynamic and high-density environments. *Drones* **8**, 516 (2024)
4. Meng, Q., Qu, Q., Chen, K., Yi, T.: Multi-UAV path planning based on cooperative co-evolutionary algorithms with adaptive decision variable selection. *Drones* **8**, 435 (2024)
5. Wang, J., Zhao, Z., Qu, J., et al.: APPA-3D: an autonomous 3D path planning algorithm for UAVs in unknown complex environments. *Sci. Rep.* **14**, 1231 (2024)
6. Zhang, S., Songwen, G., Zhou, Y., Shi, L., Jin, H.: Energy efficient resource allocation of IRS-assisted UAV network. *ERA* **32**(7), 4753 (2024)
7. He, C., Wan, Y., Xie, J.: Spatiotemporal scenario data-driven decision for the path planning of multiple UASs. *SCOPE: ACM*, pp. 7–12. NY, USA, New York (2019)
8. Yan, C., Xiaojia, X.: A path planning algorithm for UAV based on improved Q-learning. In: 2nd ICRAS 2018, pp. 1–5 (2018)
9. Xia, B., Mantegh, I., Xie, W.-F., Framework, H.: For UAV motion planning and obstacle avoidance: integrating deep reinforcement learning with fuzzy logic: CoDIT. Vallette, Malta 2024, 2662–2669 (2024)
10. Xie, R., Meng, Z., Zhou, Y., Ma, Y., Wu, Z.: Heuristic Q-learning based on experience replay for three-dimensional path planning of the unmanned aerial vehicle. *Sci. Progress* **103**(1) (2020)
11. Yan, L., Dai, J., Zhao, Y., Chen, C.: Real-Time 3D mapping in complex environments using a spinning actuated LiDAR system. *Remote Sens.* **15**, 963 (2023)
12. Fischer, C., Sester, M., Schön, S.: Spatio-temporal research data infrastructure in the context of autonomous driving. *ISPRS Int. J. Geo-Inf.* **9**, 626 (2020)
13. Hornung, A., Wurm, K.M., Bennewitz, M., et al.: OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Auton. Robot.* **34**, 189–206 (2013)
14. Clerigues, D., Wubben, J., Calafate, C.T., et al.: Enabling resilient UAV swarms through multi-hop wireless communications. *JWCN* **2024**, 39 (2024)
15. Burdakov, O., Doherty, P., Holmberg, K., Kvarnström, J., Olsson, P.-M.: Relay positioning for unmanned aerial vehicle surveillance*. *IJRR* **29**(8), 1069–1087 (2010)
16. Tao, C., Liu, B.: Online autonomous motion control of communication-relay UAV with channel prediction in dynamic urban environments. *Drones* **8**, 771 (2024)
17. Zhang, H., Zhang, G., Yang, R., Feng, Z., He, W.: Resilient formation reconfiguration for leader-follower multi-UAVs. *Appl. Sci.* **13**, 7385 (2023)
18. Ning, N., Zhou, S., Bao, W., Li, X.: A study on the maximum reliability of multi-UAV cooperation relay systems. *Sensors* **24**, 2886 (2024)
19. Duan, T., Wang, W., Wang, T., Li, X.: Role assignment mechanism of unmanned swarm organization reconstruction based on the fourth directed motif. *Sensors* **22**, 8799 (2022)
20. Chen, X., Tang, J., Lao, S.: Review of unmanned aerial vehicle swarm communication architectures and routing protocols. *Appl. Sci.* **10**, 3661 (2020)
21. Baca, T., et al.: The MRS UAV system: pushing the frontiers of reproducible research real-world deployment, and education with autonomous unmanned aerial vehicles. *J. Intell. Robot. Syst.* **102**, 26 (2021)



Data-Driven BLEVE Overpressure Prediction Using Explainable Machine Learning

Zifan Shi¹, Qilin Li¹(✉), Ling Li¹(✉), Yanda Shao², and Hong Hao^{2,3}

¹ Discipline of Computing, Curtin University, Perth, Australia
{Qilin.li, L.Li}@curtin.edu.au

² Centre for Infrastructural Monitoring and Protection, Curtin University, Perth, Australia

³ Earthquake Engineering Research and Test Centre, Guangzhou University, Guangzhou, China

Abstract. Boiling Liquid Expanding Vapour Explosion (BLEVE) is a high-energy explosion that generates intense blast loads, posing significant safety risks to infrastructure and public safety. Accurate overpressure prediction is essential for developing effective safety measures and emergency response strategies. Traditional empirical models fail to capture complex nonlinear interactions, while Computational Fluid Dynamics (CFD) methods, though accurate, are computationally expensive and impractical for large-scale and real-time applications. This study develops a neural network-based machine learning model for BLEVE overpressure prediction, offering an efficient alternative to traditional approaches. To improve model interpretability and reliability, Shapley Additive Explanations (SHAP) is employed, identifying key features that influence BLEVE overpressure. Using SHAP insights, an optimized model is constructed by selecting the most significant features, enhancing both accuracy and generalizability. The proposed approach is validated on both FLACS simulation data and experimental datasets, achieving a Mean Absolute Percentage Error (MAPE) of 2.93% on simulation test data and 47.62% on experimental data—significantly outperforming empirical methods. By integrating explainability into the predictive framework, this model not only delivers superior accuracy but also provides critical insights into the underlying factors influencing BLEVE overpressure, making it a practical and trustworthy tool for explosion risk assessment.

Keywords: Machine Learning · Explainable AI · BLEVE · Blast Loading

1 Introduction

High-pressure gases and other flammable materials are now ubiquitous in industry, but their use brings serious safety risks. Explosions during production, storage, and transport continue to cause heavy casualties and structural damage [1–3]. Accurate overpressure prediction is therefore critical for risk mitigation, safety design, and emergency planning [4]. Boiling-liquid expanding-vapour explosions (BLEVEs) rank among the most destructive events [5–7]. A BLEVE occurs when a pressurised vessel containing super-heated liquid fails, producing a blast wave, fireball, and possible secondary explosions. The resulting overpressure is governed by vessel geometry, fluid properties, and

ambient conditions [8, 9]. Capturing these coupled effects is essential for reliable hazard assessment.

Traditional BLEVE overpressure prediction relies on empirical equations derived from experimental data or past incidents. These methods are computationally efficient but struggle with nonlinear interactions, limiting their accuracy in complex scenarios [10]. CFD offers a more detailed approach, solving partial differential equations (PDE) to model explosion dynamics [11]. While CFD provides high accuracy, its high computational cost makes it impractical for real-time applications, particularly in industrial safety contexts requiring rapid assessments [12].

Machine learning (ML) has emerged as a powerful alternative [13], leveraging data-driven approaches to integrate multiple features and capture complex interactions. Many studies have demonstrated that deep learning models, such as Transformers and Graph Neural Networks (GNNs), achieve high predictive accuracy in BLEVE scenarios while maintaining computational efficiency [14]. However, a major limitation of many ML models is their lack of interpretability, often referred to as the “black-box” problem [15, 16]. This reduces trust in high-risk applications, where understanding feature contributions is crucial for ensuring model reliability.

To address these challenges, this study develops an explainable ML model for BLEVE overpressure prediction, incorporating Shapley Additive Explanations (SHAP) to improve model transparency and reliability. By analyzing feature importance through SHAP, the model is further optimized using the most influential features, enhancing predictive accuracy while maintaining efficiency. The proposed model achieves an R^2 of 99.80% and a reduced Mean Absolute Percentage Error (MAPE) of 2.93% on FLACS simulation data and 47.62% on experimental data—significantly outperforming traditional empirical methods. It highlights the effectiveness of integrating explainability with ML, ensuring both accuracy and interpretability in high-risk explosion scenarios. The key parameters of this study are summarized in Table 1.

Table 1. Nomenclature of key parameters for BLEVE Overpressure Prediction.

Notation	Description	Notation	Description
Tank _p	Failure Pressure (bar)	G _H	Tank Height with Gas (m)
R _L	Liquid Ratio (%)	B _H	BLEVE Height (m)
Tank _w	Tank Width (m)	e	Specific Energy (MJ/m ³)
Tank _L	Tank Length (m)	E	Total Energy (MJ)
Tank _H	Tank Height (m)	P _s ⁺	Positive Peak Pressure (bar)
Tank _v	Tank Volume (m ³)	P _O	Ambient Pressure, 1.013 (bar)
T _v	Vapour Temperature (K)	β	Energy Conversion Factor
T _L	Liquid Temperature (K)	D	Standoff Distance (m)
Status	Subcooled or Superheated	$\bar{R}$	Scaled Distance (m)

2 Numerical Modelling

2.1 FLACS Simulation

FLACS is a widely validated CFD tool for explosion simulations, particularly BLEVE overpressure prediction [17, 18]. It employs three-dimensional modeling and advanced computational techniques to simulate overpressure, impulse, and shockwave propagation. Using Favre-averaged transport equations and the Reynolds-averaged Navier–Stokes (RANS) model, FLACS integrates the ideal gas law and turbulence models (e.g., $k - \epsilon$) for accuracy. Shockwave propagation is handled via Euler equations and flux-corrected transport schemes.

2.2 Generation of Simulated Data

To generate a comprehensive dataset for model training and validation, we simulated 1,000 BLEVE cases (500 butane, 500 propane) in FLACS. Eleven key variables (Table 2) were varied systematically to capture interactions among tank geometry, fluid properties, and environmental conditions. Tanks followed European LPG standards—rectangular, $\leq 10 \times 3 \times 3$ m, BLEVE height ≤ 2.2 m [19]. Failure pressure spanned 5 bar to each fluid’s critical pressure (37 bar for butane, 42 bar for propane); liquid fill ranged 10%–90%. Standoff distance was sampled every meter from 5 m to 40 m, yielding 36 pressure points per case and 36,000 records in total. Simulations ran for two months on a desktop PC. Pre-processing removed anomalies and scaled features to unit variance for model compatibility. Compared with limited experimental datasets, this synthetic set covers a wider operational envelope, improving generalisation, mitigating over-fitting, and supporting robust real-world BLEVE overpressure prediction.

Table 2. Summary of 11 input variables of the BLEVE dataset.

Notation	Range	
	Butane	Propane
Tank _p	5–37	5–42
R _L	10–90	10–90
Tank _W	0.4–3	0.4–3
Tank _L	0.4–10	0.4–10
Tank _H	0.4–3	0.4–3
B _H	0–2	0–2
G _H	0.2–2.6	0.2–2.6
T _V	283–573	283–453
T _L	283–425	283–369
Status	Superheated or Saturated	Superheated or Saturated
D	5–40	5–40

3 Methodology

3.1 Methodology Overview

This study adopts a structured workflow for explainable BLEVE overpressure prediction (Fig. 1). A baseline ML model is first trained, after which SHAP supplies local (single-case) and global (dataset-wide) explanations of feature influence. Insights from the global analysis guide feature selection and model refinement, boosting both accuracy and interpretability. The optimised model is then validated against FLACS simulation and experimental data to verify reliability. The next sections describe model development (Sect. 3.2), SHAP-driven analysis and optimisation (Sect. 3.3), and evaluation results (Sect. 4).

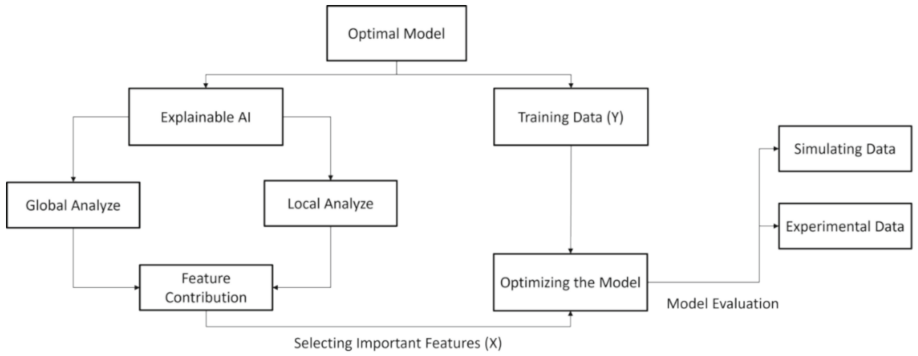


Fig. 1. Methodology flowchart.

3.2 Machine Learning Model

This study employs a Multi-Layer Perceptron (MLP) model to predict BLEVE overpressure. MLP, a fully connected neural network with nonlinear activation functions, effectively captures complex relationships in numerical datasets [20]. The dataset lacks sequential or graph-based dependencies, aligning well with MLP's ability to model intricate relationships without requiring positional or structural encoding [11]. MLP provides an optimal balance between simplicity, computational efficiency, and predictive accuracy, making it an appropriate choice for this study. The MLP architecture is formulated in Eq. (1):

$$\begin{aligned}
 \text{MLP}(X) &= \text{Linear}(\text{MLPBlock} \dots \text{MLPBlock}(X)) \\
 \text{MLPBLOCK}(X) &= \text{Dropout}(\text{ReLU}(\text{Linear}(X)))
 \end{aligned} \tag{1}$$

This study employs the ReLU activation function to prevent vanishing gradients and ensure stable convergence, while dropout regularization mitigates overfitting, enhancing generalization [21]. The MLP model consists of an input layer, two hidden layers, and an output layer. It processes 11 numerical features, modeling nonlinear interactions without

sequential dependencies. This architecture optimally balances computational efficiency and predictive accuracy for BLEVE overpressure regression.

Numerical features are standardized to a mean of zero and a standard deviation of one, ensuring model compatibility and stable training. Hyperparameter tuning with Optuna explored layers, neurons, dropout, learning rates, and weight decay [22]. The dataset is split into 80% for training and 20% for testing, with 20% of the training set used for validation. Early stopping with 32 epochs patience is applied to prevent overfitting. Table 3 summarizes the hyperparameter tuning space.

Table 3. The hyperparameter tuning space of MLP models.

Hyperparameter	Search space	Distribution
Learning rate	[7e−4, 9e−4]	log uniform
Weight decay	[1e−8, 1e−3]	log uniform
Number of layers	[2, 4]	int uniform
Number of neurons	[128, 512]	int uniform
Dropout probability	[0.0, 0.2]	uniform

The final configuration of the MLP model includes standard normalization with 11 input features. The model consists of two layers with 359 and 226 neurons per layer, a dropout rate of 0.1475, and is designed for a regression task. It uses a batch size of 512 for training. The learning rate is set to 7.73×10^{-4} with AdamW [23] optimizer, which enhances generalization and convergence stability by decoupling weight decay from gradient updates [24]. The model is trained for up to 1×10^9 epochs with an early stopping patience of 32, and a weight decay of 4.32×10^{-6} . Using these settings, 100 trials are performed and the best configuration is selected.

3.3 Explainable Machine Learning

This study employs SHAP (SHapley Additive exPlanations), an explainable artificial intelligence (xAI) method based on cooperative game theory. SHAP assigns each feature a Shapley value—its average marginal contribution across all feature subsets—thus ranking feature importance fairly and consistently [25]. We employ the model-agnostic Kernel Explainer, which estimates these values via perturbation-based weighted linear regression [26].

In BLEVE blast analysis, tank dimensions, liquid ratios, and explosion distances exhibit highly nonlinear impacts on explosion loads. In this study, the force plot is used for local analysis, while the waterfall plot is employed for global analysis. A force plot moves from the baseline $E[f(X)]$ to the prediction $f(x)$ through the summed Shapley values ϕ_i [27], as shown in Eq. (2):

$$f(x) = E[f(x)] + \sum_{i=1}^n \phi_i \quad (2)$$

The baseline prediction, $E[f(X)]$, is the expected value of the model output over a background dataset, serving as the reference point for the prediction. This study employed K-means clustering to extract representative background points to improve computational efficiency [28].

Figure 2 contrasts SHAP contributions for a low-pressure case (a) and a high-pressure case (b). In Fig. 2a, the long standoff distance ($D = 34$ m) dominates with a large negative effect, while the small tank width (0.6 m) and height (0.8 m) further lower pressure by limiting gas volume. A longer tank ($L = 5.8$ m) offsets these reductions, boosting overpressure. Lower gas-filled height and failure pressure also suppress the blast, whereas the superheated liquid state (status 1) adds energy, contributing positively. In Fig. 2b, the shorter distance ($D = 5$ m) sharply elevates pressure, and higher values of gas-filled height, failure pressure, and tank length intensify the load. Tank width remains 0.6 m in both cases, consistently restraining volume. The subcooled liquid state (status 0) now acts negatively, underscoring the liquid phase's role in modulating energy release.

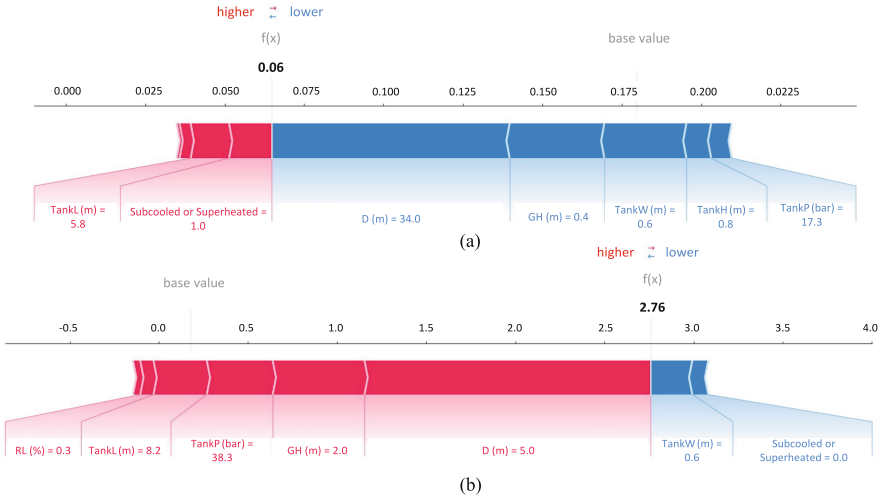


Fig. 2. Local SHAP analysis.

To evaluate the MLP model's predictive logic for BLEVE overpressure, a global SHAP analysis was conducted using a bar plot (Fig. 3). By aggregating feature importance across the dataset, SHAP provides a global perspective on key factors influencing model performance [29].

Figure 3 ranks features by their contributions, with Standoff Distance (D) having the highest impact, confirming its dominant role in pressure attenuation. Tank Length ($Tank_L$), Tank Height with Gas (G_H), and Tank Failure Pressure ($Tank_P$) also significantly influence predictions. $Tank_L$ affects tank volume, shaping energy release patterns, while G_H represents available energy within the tank, directly impacting overpressure. Higher $Tank_P$ may delay rupture, allowing greater energy accumulation and altering explosion dynamics.

In contrast, Tank Width (Tank_W), Tank Height (Tank_H), BLEVE Height (B_H), Liquid Ratio (R_L), and Liquid State contribute less globally (Fig. 3). The limited variability of Tank_W, Tank_H, constrained by European LPG tank standards [19] which specify maximum dimensions of $10 \times 3 \times 3$ m and limit BLEVE height to 2.2 m, reduces their overall impact. While Liquid State affects energy release, its influence remains minor, and R_L, when extreme, may indirectly impact pressure peaks.

Vapor Temperature (T_V) and Liquid Temperature (T_L) exhibit low contributions in global analysis and local analysis, and thus were removed, allowing the model to focus on learning the nonlinear relationships between high-contribution features and overpressure predictions.

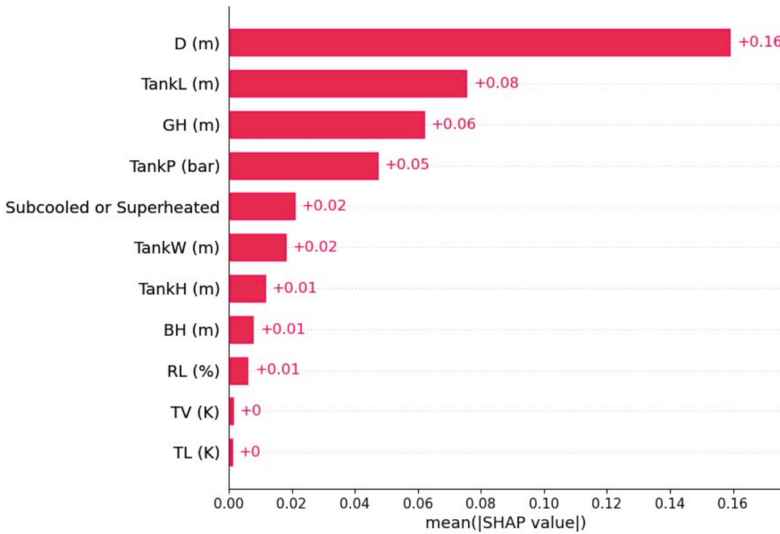


Fig. 3. Global SHAP analysis.

The top nine features from the global SHAP analysis (Fig. 3) were selected as input variables for model training. The original target values (Y) from the same BLEVE dataset were used as ground truth. The optimized MLP model retained standard normalization but used nine input features instead of eleven.

4 Experimental Results

This experiment first conducted preliminary testing using the FLACS simulation test set, which includes 7,200 samples (3,613 Butane and 3,587 Propane). The FLACS test set and real experimental data remained entirely unseen during training, preventing any prior exposure for the models. To further enhance reliability, 37 real experimental samples from Birk and Johnson [30–32] were introduced for validation, as detailed in Table 4. This ensured a robust evaluation of the MLP model and the optimized model.

Table 4. Summary of BLEVE experimental testing data.

Experiments	Fluid	Liquid status	Distance (m)	Case No
01–01(Birk)	Propane	Subcooled	10, 20, 30, 40	1, 2, 3, 4
01–02(Birk)	Propane	Superheated	20, 30, 40	1, 2, 3
01–03(Birk)	Propane	Superheated	10, 20, 40	1, 2, 3
01–04(Birk)	Propane	Superheated	40	1
01–05(Birk)	Propane	Superheated	20, 30, 40	1, 2, 3
02–01(Birk)	Propane	Superheated	10, 20, 30, 40	1, 2, 3, 4
02–02(Birk)	Propane	Superheated	10, 20, 30, 40	1, 2, 3, 4
02–03(Birk)	Propane	Subcooled	10, 20, 30, 40	1, 2, 3, 4
02–04(Birk)	Propane	Superheated	20, 30, 40	1, 2, 3
BG-1(Johnson)	Butane	Superheated	25, 50	1, 2
BG-2(Johnson)	Butane	Superheated	25, 50	1, 2
BG-3(Johnson)	Butane	Subcooled	25, 50	1, 2
BG-4(Johnson)	Butane	Subcooled	25, 50	1, 2

In the FLACS simulated test set, the MLP model achieved R^2 of 99.61% and MAPE of 5.24% (Fig. 4a). On the real experimental dataset, the MAPE is 58.18%, MSE is 0.0003, as shown in Fig. 4b. The larger error stems from safety and environmental constraints that limit BLEVE scale; overpressures are near zero, making tiny absolute differences appear proportionally large. The model remains accurate for the larger overpressures that dominate risk. For reference, the empirical correlation of Laboureur [33] exceeds 70% error, underscoring the superior accuracy of the machine-learning approach.

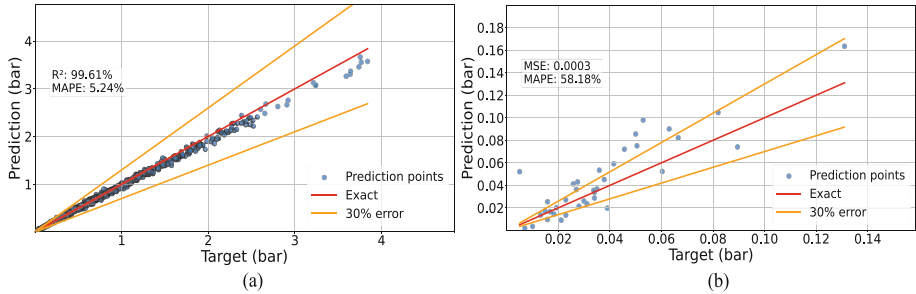


Fig. 4. MLP performance evaluation.

The optimised MLP incorporates nine SHAP-selected inputs (Fig. 3) and three hidden layers of 317, 417, and 184 neurons. Dropout was reduced to 0.0161, and AdamW employed with a learning rate of 7.19×10^{-4} and weight decay of 1.23×10^{-4} ; batch

size, training epochs, and early-stopping rules were unchanged to ensure fair comparison. On the FLACS simulation test set the revised network improves the baseline R^2 from 0.996 to 0.998 and lowers MAPE from 5.24% to 2.91% (Fig. 5a). On the 37-sample experimental set, MAPE falls from 58.18% to 47.62% (Fig. 5b), an 11% relative gain despite the prevalence of near-zero overpressures. These results demonstrate that SHAP-guided feature selection and mild regularisation sharpen the model's ability to capture the nonlinear drivers of BLEVE overpressure and sustain accuracy across both simulated and real scenarios.

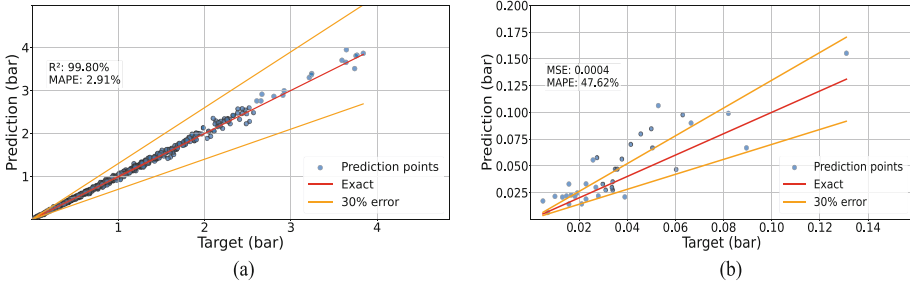


Fig. 5. Evaluation of the optimized model.

To further evaluate the accuracy of the optimized model, this study compares it with an existing BLEVE overpressure prediction model. First, the method proposed by Hemmatian et al. [10] is used to calculate specific energy, total mechanical energy, TNT equivalent mass, and scaled distance. This approach, based on simplified thermodynamic assumptions, primarily depends on liquid ratio (R_L) and vapor temperature at failure (T_V) to estimate specific energy, as defined in Eq. (3) for propane and Eq. (4) for butane.

$$e = 43.97 - 213.9 \times R_L - 0.152 \times T_V + 1.349 \times R_L \times T_V - 0.0004361 \times T_V^2 - 0.002045 \times R_L \times T_V^2 + 1.55 \times 10^{-6} \times T_V^3 \quad (3)$$

$$e = 21.32 - 87.2 \times R_L - 0.136 \times T_V + 0.4765 \times R_L \times T_V + 0.0001885 \times T_V^2 - 0.0005805 \times R_L \times T_V^2 + 9.693 \times 10^{-8} \times T_V^3 \quad (4)$$

Once the specific energy e is determined, the total mechanical energy E released by the tank is calculated using Eq. (5), where tank volume ($Tank_V$) is derived from its length, width, and height. The TNT equivalent mass is then obtained using Eq. (6), with an explosion energy conversion factor $\beta = 0.4$, as validated by Hemmatian's research. The TNT equivalent mass, expressed in kg TNT, serves as a critical parameter for predicting BLEVE overpressure. After obtaining the TNT equivalent mass, Eq. (7) is used to calculate the scaled distance, where D represents the distance from the explosion center to the observation point in meters.

$$E = e \times Tank_V \quad (5)$$

$$m_{TNT} = \frac{\beta \times E * (MJ) \times 10^3}{4680} \quad (6)$$

$$\bar{R} = \frac{D}{(m_{TNT})^{\frac{1}{3}}} \quad (7)$$

The overpressure can be calculated using the Eq. (8) empirical relation from Kinney and Graham [32] for high explosives:

$$\frac{P_s^+}{P_o} = \frac{808 \left[1 + \left(\frac{\bar{R}}{4.5} \right)^2 \right]}{\sqrt{1 + \left(\frac{\bar{R}}{0.048} \right)^2} \times \sqrt{1 + \left(\frac{\bar{R}}{0.32} \right)^2} \times \sqrt{1 + \left(\frac{\bar{R}}{1.35} \right)^2}} \quad (8)$$

As shown in Fig. 6(a), the Hemmatian-Kinney equation performs poorly on the FLACS test set, with an R^2 of only 12.07%, indicating an extremely poor fit to the test data. Additionally, the MAPE reaches 46.98%, reflecting a significant difference between the predicted and actual values. Performance worsens on the 37-sample experimental set (Fig. 6b): MAPE climbs to 67.84% and MSE to 8×10^{-4} —well above the baseline MLP (58.18%, 3×10^{-4}) and the optimised MLP (47.62%, 4×10^{-4}). These large errors highlight the empirical equation's inability to capture the complex, nonlinear factors governing BLEVE overpressure.

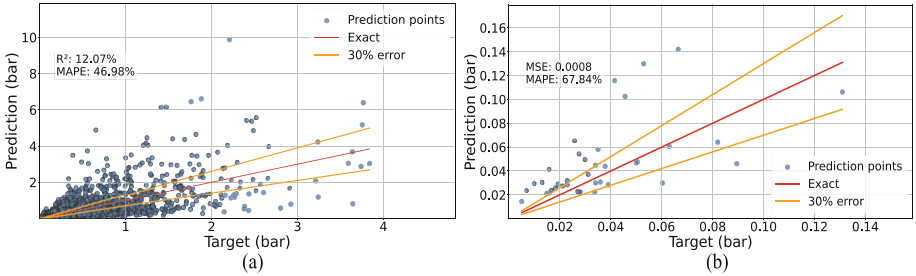


Fig. 6. Hemmatian and Kinney equation performance evaluation.

Tables 5 and 6 summarize the quantitative results for easy comparison. A further analysis reveals that the features identified through SHAP global analysis closely align with those used in traditional empirical equations, including Standoff D (Distance), R_L (Liquid Ratio), and $Tank_V$ (Tank Volume). Notably, $Tank_V$ (Tank Volume) is directly influenced by $Tank_L$ (Tank Length), $Tank_H$ (Tank Height), and $Tank_W$ (Tank Width), which matches the feature selection in the optimized model. This consistency further validates SHAP's capability in capturing key feature relationships embedded in traditional BLEVE prediction models.

Moreover, SHAP-based feature selection has identified many critical features that traditional empirical equations fail to account for, such as $Tank_P$ (Tank Failure Pressure), B_H (BLEVE Height), and Gas Status (Subcooled or Superheated). This allows the model

Table 5. FLACS test set evaluation results.

Metric	MLP Model	Optimized Model	Hemmatian and Kinney Method
R ²	99.61%	99.80%	12.07%
MAPE	5.24%	2.91%	46.98%

Table 6. Experimental data set evaluation results.

Metric	MLP Model	Optimized Model	Hemmatian and Kinney Method
MSE	0.0003	0.0004	0.0008
MAPE	58.18%	47.62%	67.84%

to develop a more precise and comprehensive understanding of BLEVE overpressure dynamics. By focusing on the most influential features, the optimized model learns key nonlinear relationships more effectively, leading to further improvements in prediction accuracy and generalization.

5 Conclusion

This study proposes an interpretable machine learning approach for BLEVE overpressure prediction, addressing the limitations of empirical and CFD methods. While empirical models struggle with nonlinear interactions and CFD is computationally costly, the MLP model provides an efficient and scalable alternative. Using SHAP-based feature selection, the model was optimized to focus on key predictive factors, reducing MAPE from 58.18% to 47.62% on experimental data—an 11% improvement. This highlights the power of explainable AI in enhancing both accuracy and interpretability. The method outperforms empirical models while being far more efficient than CFD, making it a practical solution for explosion risk assessment. Future work will improve model adaptability to extreme conditions, incorporating physical constraints and expanding feature selection for better generalization. This approach can extend beyond BLEVE overpressure to parameters like negative pressure and impulse, offering insights into explosion dynamics and supporting emergency safety measures and risk mitigation strategies.

References

1. Jo, Y.-D., Ahn, B.J.: Analysis of hazard areas associated with high-pressure natural-gas pipelines. *J. Loss Prev. Process Ind.* **15**(3), 179–188 (2002)

2. Li, J., et al.: Calculation of BLEVE energy and overpressures inside a tunnel using analytical and CFD methods. *Tunn. Undergr. Space Technol.* **120**, 104263 (2022)

3. Cheng, R., et al.: Prediction of BLEVE-induced response of road tunnel using Transformer network with modified self-attention (SAMT). *Eng. Struct.* **314**, 118415 (2024)

4. Yokohama, H., et al.: Vehicle borne improvised explosive device (VBIED) characterisation and estimation of its effects in terms of human injury. *Int. J. Protect. Struct.* **6**(4), 607–627 (2015)
5. Birk, A., Cunningham, M.: The boiling liquid expanding vapour explosion. *J. Loss Prev. Process Ind.* **7**(6), 474–480 (1994)
6. Wang, Y., Li, J., Hao, H.: Development of efficient methods for prediction of medium to large scale BLEVE pressure in open space. *Process Saf. Environ. Prot.* **161**, 421–435 (2022)
7. Eckhoff, R.K.: Boiling liquid expanding vapour explosions (BLEVEs): a brief review. *J. Loss Prev. Process Ind.* **32**, 30–43 (2014)
8. Laboureur, D., et al.: A closer look at BLEVE overpressure. *Process Saf. Environ. Prot.* **95**, 159–171 (2015)
9. Luo, W., et al.: Multi-factor coupling analysis and identification of tanks BLEVE. *Process Saf. Environ. Prot.* **177**, 880–890 (2023)
10. Hemmatian, B., Casal, J., Planas, E.: A new procedure to estimate BLEVE overpressure. *Process Saf. Environ. Prot.* **111**, 320–325 (2017)
11. Li, J., et al.: Prediction of BLEVE blast loading using CFD and artificial neural network. *Process Saf. Environ. Prot.* **149**, 711–723 (2021)
12. Wendt, J.F.: *Computational Fluid Dynamics: An Introduction*. Springer, Cham (2008)
13. Li, Q., et al.: A multi-task machine learning approach for data efficient prediction of blast loading. *Eng. Struct.* **326**, 119577 (2025)
14. Li, Q., et al.: Machine learning prediction of BLEVE loading with graph neural networks. *Reliab. Eng. Syst. Saf.* **241**, 109639 (2024)
15. Carter, A., Imtiaz, S., Naterer, G.: Review of interpretable machine learning for process industries. *Process Saf. Environ. Prot.* **170**, 647–659 (2023)
16. Petch, J., Di, S., Nelson, W.: Opening the black box: the promise and limitations of explainable machine learning in cardiology. *Can. J. Cardiol.* **38**(2), 204–213 (2022)
17. Middha, P.: Development, Use, and Validation of the CFD tool FLACS for Hydrogen Safety Studies (2010)
18. Li, Q., et al.: Prediction of BLEVE loads on structures using machine learning and CFD. *Process Saf. Environ. Prot.* **171**, 914–925 (2023)
19. Kadatec, LPG ROAD TANKERS & ISO TANK CONTAINERS (2017)
20. Gorishniy, Y., et al.: Revisiting deep learning models for tabular data. *Adv. Neural. Inf. Process. Syst.* **34**, 18932–18943 (2021)
21. Srivastava, N., et al.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
22. Akiba, T., et al.: Optuna: a next-generation hyperparameter optimization framework. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2019)
23. Kingma, D.P.: Adam: a method for stochastic optimization. *arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)* (2014)
24. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. *arXiv preprint [arXiv:1711.05101](https://arxiv.org/abs/1711.05101)* (2017)
25. Mangalathu, S., Hwang, S.-H., Jeon, J.-S.: Failure mode and effects analysis of RC members based on machine-learning-based SHapley Additive exPlanations (SHAP) approach. *Eng. Struct.* **219**, 110927 (2020)
26. Lundberg, S.M., Lee, S.-I.: A unified approach to interpreting model predictions. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
27. Ekanayake, I., Meddage, D., Rathnayake, U.: A novel approach to explain the black-box nature of machine learning in compressive strength predictions of concrete using Shapley additive explanations (SHAP). *Case Stud. Constr. Mater.* **16**, e01059 (2022)

28. Yuan, H., et al.: An empirical study of the effect of background data size on the stability of SHapley Additive exPlanations (SHAP) for deep learning models. arXiv preprint [arXiv:2204.11351](https://arxiv.org/abs/2204.11351) (2022)
29. Zhou, X., et al.: An interpretable model for the susceptibility of rainfall-induced shallow landslides based on SHAP and XGBoost. *Geocarto Int.* **37**(26), 13419–13450 (2022)
30. Johnson, D., Pritchard, M., Wickens, M., Large scale catastrophic releases of flammable liquids. British Gas Report to the CEC, Contract EV4T, 14 (1990)
31. Birk, A., Van der Steen, J.: On the transition from non-BLEVE to BLEVE failure for a 1.8 m 3 propane tank (2006)
32. Birk, A., Davison, C., Cunningham, M.: Blast overpressures from medium scale BLEVE tests. *J. Loss Prev. Process Ind.* **20**(3), 194–206 (2007)
33. Laboureur, D., et al.: BLEVE overpressure: multiscale comparison of blast wave modeling. *Process Saf. Prog.* **33**(3), 274–284 (2014)



ESN Architectures for Industrial Process Modelling to Develop Digital Twins

Raúl González-Herbón¹✉, Serafín Alonso¹, Miguel A. Prada¹,
Ignacio Díaz², Manuel Domínguez¹, and Juan J. Fuertes¹

¹ SUPPRESS Research Group, Escuela de Ingenierías, Universidad de León, Campus de Vegazana, 24007 León, Spain

{rgonzh,saloc,ma.prada,mdomg,jj.fuertes}@unileon.es

² Electrical Engineering Department, University of Oviedo, Edif. Departamental Oeste 2, Campus de Viesques s/n, 33204 Gijón, Spain

idiaz@uniovi.es

<https://suppress.unileon.es>

Abstract. Digital twins have emerged as a powerful tool for industrial process monitoring and optimization, requiring accurate and computationally efficient models to capture system dynamics. This paper presents the implementation of Echo State Networks (ESNs) for modelling complex industrial systems with non-linear behavior. The study is conducted on a pilot plant representing an industrial process, where four variables (level, pressure, flow, and temperature) are controlled and estimated using ESN models. To assess the impact of different architectures, each model is tested with various configurations, including deep networks, feedback integration, and a combination of both. The results demonstrate that the effectiveness of these architectures depends on the dynamic characteristics of each variable. The most notable improvement is observed in process temperature estimation, where feedback significantly enhances performance. The results highlight that the combination of both, feedback mechanisms and deeper architectures, can improve the prediction of variables with slower dynamics and higher inertia.

Keywords: Digital twin · Industrial process · Deep Learning · Echo State Networks · Data-based modelling

1 Introduction

In recent years, industrial digitalisation has deeply transformed industrial processes. This transformation is driven by the integration of different technologies such as digital twins that allow the simulation and optimisation of physical systems in real time [13]. These tools offer great value by virtually representing industrial processes and facilitating their dynamic analysis [7]. In this context, digital twins have emerged as essential components within the Industry 4.0 paradigm to improve performance optimisation [17].

The development of digital twins requires accurate models of the industrial systems or processes they represent, capable of capturing the dynamics and behaviour of the real system [19]. However, adverse conditions in industrial environments can significantly affect the accuracy of the models, introducing complex non-linearities due to components such as pumps, valves, or variable frequency drives, as well as noise and errors generated by sensors and measuring instruments. In these scenarios, accurately modelling all these features is challenging. Therefore, data-driven models are positioned as a more suitable solution to address the complexity of these real industrial processes, overcoming the limitations of approaches based solely on simpler mathematical or physical principles [16].

However, to ensure that the digital twins can operate in synchrony with the physical systems, it is essential that the underlying models are also capable of real-time operation. This allows effective bidirectional communication between the real system and its digital replica, enabling tasks such as process prediction, control and optimisation [2]. In this context, the need arises to choose models with a simple or computationally inexpensive architecture, such as Echo State Networks (ESN). These models allow complex dynamics to be processed efficiently and quickly, adapting especially to systems with rapid changes in their states [11].

For these reasons, this paper proposes the use of Echo State Networks to model a non-linear industrial system with the aim of developing a digital twin. For this purpose, we experiment with different ESN architectures, including deep configurations and the incorporation of feedback connections in the network, evaluating their impact on the model's performance. The system is modelled by estimating four variables with different dynamics, which allows us to analyse the capacity of ESNs to capture and predict complex behaviour in industrial systems with different characteristics.

The rest of the paper is structured as follows: a literature review of digital twins and ESNs in industrial systems is presented in Sect. 2. Next, the system used and the methodology adopted to develop the experiments are described in Sect. 3. In Sect. 4 the results are presented and discussed. Finally, the Sect. 5 describes the conclusions and the future lines of the work.

2 Literature Review

The concept of the digital twin, initially introduced by Michael Grieves as part of Product Lifecycle Management (PLM) strategies, has evolved significantly over the past two decades [8]. Originally envisioned as a virtual representation of physical assets that continuously mirrored their real-world counterparts, the digital twin has become a fundamental part of Industry 4.0, driven by advancements in IoT, artificial intelligence, and data analytics [6]. This integration enables real-time data exchange and synchronization between physical and virtual systems, enhancing the ability to monitor, optimize, and predict system performance [9].

Building on these advancements, the adoption of machine learning techniques has become an essential part of enhancing the capabilities of digital twins, particularly in modelling and predicting the non-linear and dynamic behaviors of complex systems [3]. These approaches are necessary for accurately simulating industrial processes, where traditional physics-based models may fall short due to the inherent complexities and non-linearities of real-world environments [10]. Among these methods, Echo State Networks have emerged as an efficient data-driven solution, capable of handling the temporal dynamics required for continuous system modelling while maintaining computational efficiency [15]. This makes ESNs particularly suitable for developing robust digital twins tailored to the needs of industrial applications.

ESNs have been successfully employed in the energy sector for tasks such as forecasting the energy content of steelwork by-product gases, demonstrating their ability to handle real industrial data with high nonlinearity [5]. Moreover, in anomaly detection, ESNs combined with autoencoders have been applied to industrial IoT systems, enabling efficient and accurate detection of faults in multivariate time-series data while meeting constraints of edge devices [4].

Similarly, in the field of predictive control, ESNs have been integrated into non-linear model predictive control (MPC) frameworks, providing accurate system predictions while maintaining a low computational load, as seen in applications like controlling discharge pressure in refrigeration systems [1].

Also, ESNs have demonstrated comparable or even superior performance to other advanced models in tasks of dynamic system modelling and time-series prediction. In [14], ESNs were used to estimate key variables within the sulfur recovery unit (SRU), outperforming other recurrent neural network models such as Long Short-Term Memory (LSTM) networks. The ESN approach demonstrated superior accuracy and computational efficiency, making it particularly suitable for dynamic modelling in complex processes.

3 Methodology

3.1 Non-linear Industrial System

The system used is a pilot plant representing an industrial process in a controlled environment but representative of real applications as shown in Fig. 1. The system is composed of three circuits: the main or process circuit and two auxiliary circuits.

The main or process circuit consists of the circulation of a fluid between two tanks at different heights (main and storage tank), where four variables of different dynamics can be controlled: the flow rate, the pressure, the temperature of the circuit and the level of the upper tank. These variables have been measured through sensors FT, PT, TT01 and LT, respectively. This fluid is driven by a centrifugal pump (P01) controlled by a frequency drive. In addition, there is a pneumatic valve (V01) that regulates the flow rate and an electro-pneumatic valve that interconnects both tanks.

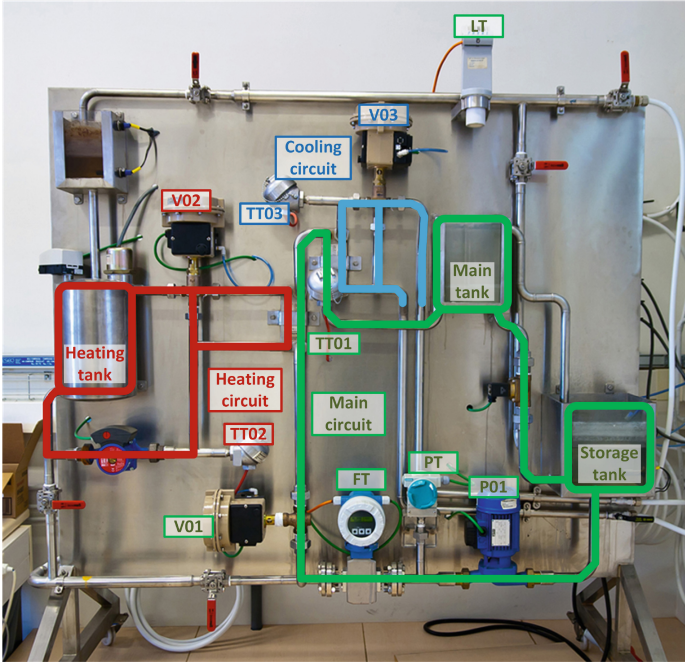


Fig. 1. Non-linear industrial system

The process temperature is controlled by the auxiliary circuits. On the one hand, there is a heating circuit, where electrical resistances are used to increase the temperature of the auxiliary circuit and, by means of a heat exchanger, the thermal energy is transferred to the main process. The flow of hot water into the heat exchanger is regulated by a three-way valve (V02). On the other hand, the cooling circuit uses an external cooler to reduce the temperature of the auxiliary circuit and, like the heating circuit, a plate heat exchanger and a three-way valve (V03) are used to transfer the thermal energy to the main process. In both circuits, the temperature has been measured by sensors TT02 and TT03 respectively.

3.2 ESN-Based Architectures

A fundamental requirement for digital twins of non-linear dynamical systems is the ability to evolve their dynamic state from the current time step to the next without requiring additional input for the state transition. To meet this requirement, Echo State Networks (ESNs) are employed as they excel in modelling complex, time-dependent processes with minimal computational cost and robust dynamic representation. The modelling framework consists of four separate ESN, each dedicated to predicting one of the target variables: level (L), pressure (P), flow rate (Q), and process temperature (T01). The input features

for the first three models (level, pressure, and flow rate) are the operational states of the process pump and process valve (P01, V01). For the temperature model, additional inputs are included, comprising the control states of the auxiliary circuit valves and the temperatures of the heating and cooling circuits (T02, T03, V02, V03). Table 1 shows the inputs and outputs for each model. Furthermore, various architectural configurations are explored for each ESN, including networks with feedback connections, deep architectures, and deep architectures with feedback. Those configurations are shown in Fig. 2.

The first architecture evaluated is the shallow ESN, whose configuration is presented in Eq. 1. In this configuration, the weight matrices W_{in} and W_{res} are randomly generated, while W_{out} is the only trainable one in the model. The function f corresponds to the *tanh* activation function, and the parameters α and ν represent the *leaking rate* and *bias*, respectively.

$$\begin{aligned} x(k) &= (1 - \alpha)x(k - 1) + \alpha f(W_{in}u(k) + W_{res}x(k - 1) + \nu) \\ y(k) &= W_{out}x(k) \end{aligned} \quad (1)$$

The next architecture is the shallow ESN with feedback, which includes a connection between the output and the reservoir layer as presented in Eq. 2. This connection is presented in [12] and allows the ESN to no longer rely on fixed random input-driven dynamics to construct the output; instead, the dynamics are adapted to the task. In this configuration, the trained readouts are fed back to the reservoir through the W_{fb} matrix, thereby altering the dynamics during training.

$$\begin{aligned} x(k) &= (1 - \alpha)x(k - 1) + \alpha f(W_{in}u(k) + W_{res}x(k - 1) + W_{fb}y(k - 1) + \nu) \\ y(k) &= W_{out}x(k) \end{aligned} \quad (2)$$

The deep architecture evaluated is presented in Eq. 3. The difference from the shallow architecture is that reservoir is composed of n layers, where the W^1 is the input weight matrix, and for $n \geq 1$, W^n is the inter-layer connections from the layer $(n-1)$ to layer n .

$$\begin{aligned} x^1(k) &= (1 - \alpha)x^1(k - 1) + \alpha f(W^1u(k) + W_{res}x^1(k - 1) + \nu) \\ x^n(k) &= (1 - \alpha)x^n(k - 1) + \alpha f(W^n x^{(n-1)}(k) + W_{res}x^n(k - 1) + \nu) \\ y(k) &= W_{out}(x^1(k), \dots, x^n(k)) \end{aligned} \quad (3)$$

The final architecture is a mixture of both deep ESN and feedback connections. In this configuration, feedback is introduced in the first layer of the reservoir, as shown in Eq. 4.

$$\begin{aligned} x^1(k) &= (1 - \alpha)x^1(k - 1) + \alpha f(W^1u(k) + W_{res}x^1(k - 1) + W_{fb}y(k - 1) + \nu) \\ x^n(k) &= (1 - \alpha)x^n(k - 1) + \alpha f(W^n x^{(n-1)}(k) + W_{res}x^n(k - 1) + \nu) \\ y(k) &= W_{out}(x^1(k), \dots, x^n(k)) \end{aligned} \quad (4)$$

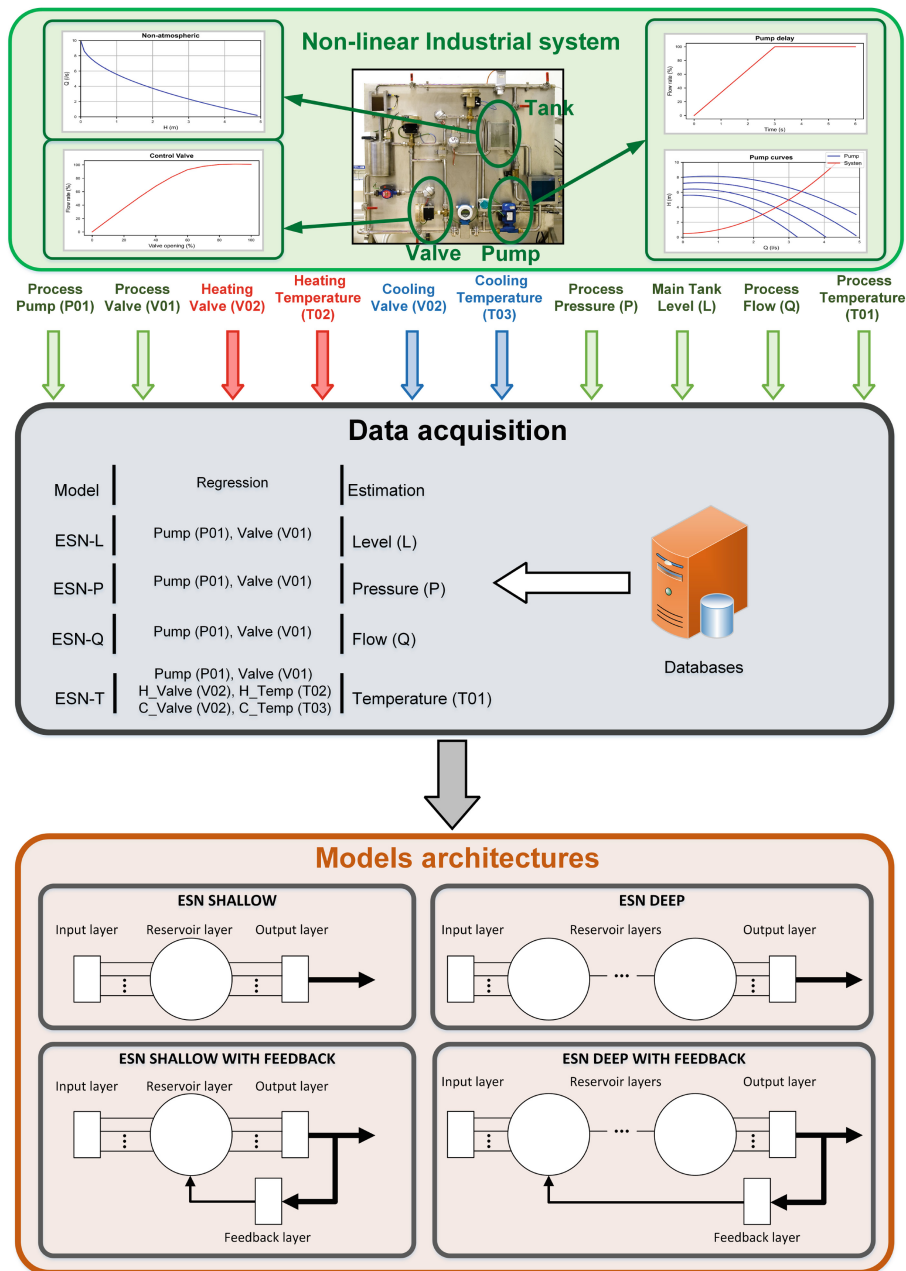


Fig. 2. Methodology

The configuration and programming of the ESNs has been done using the *reservoirpy* library [18]. To optimize the hyperparameters of the Echo State

Table 1. Inputs and outputs of the models

ESN model	Inputs	Output
ESN-L	P01, V01	L
ESN-P	P01, V01	P
ESN-Q	P01, V01	Q
ESN-T	P01, V01, V02, V03, T02, T03	T01

Networks, a random search approach was employed using the *reservoirpy* library. For each architectural configuration, 500 distinct sets of hyperparameters were sampled and tested to identify the optimal combination. The hyperparameters and their respective ranges for the different ESN architectures are detailed in Table 2. These ranges were selected to cover the widest possible spectrum of values. The performance of each training-validation configuration was evaluated using the Root Mean Square Error (RMSE) metric, while the testing process was evaluated with RMSE and the coefficient of determination R^2 .

Table 2. Hyperparameters search

Hyperparameters	Values
Reservoir units	300–1500
spectral radius	0.1–1.1
leaking rate	0.01–1
input scaling	0.1–1
input connectivity	0.1–1
ridge	0.00001–0.01
reservoir connectivity	0.05–0.7
bias	0.0001–0.5
layers (for deep)	3–9
inter laying scaling (for deep)	0.1–1

3.3 Generating Data

Experiments were carried out on the physical system where a continuous industrial process with specific characteristics is emulated to generate training data. These experiments were designed to generate complex dynamics and various scenarios, reflecting typical conditions of an industrial environment. In the system, two PI controllers operating simultaneously were implemented. One of them regulates the fluid level in the process tank, while the other one controls the process fluid temperature. These two variables exhibit oscillations within defined ranges,

designed to reproduce variations typical of a real process. The process level is adjusted between values of 30% to 90%, with increments of 15%. Similarly, the fluid temperature varies between 20 °C and 35 °C, with 5 °C changes. These variations introduce dynamic and transient scenarios that allow capturing the characteristics of the system over time.

Data were collected with a sampling rate of 125 ms using a multithreaded application developed in Python, which interacts directly with the system’s PLC controller. This configuration ensures reliable real-time data acquisition, providing a dataset that includes interactions between controlled variables and system responses under different configurations. The dataset used for training and testing the models was acquired during 10 h of experimentation, providing a detailed representation of the system’s behaviour. The dataset was divided into three subsets with 70% for training, 15% for validation and 15% for testing. In addition, the data was resampled at 1-s intervals and a *MinMaxScaler* was applied to normalise all data to a range between 0 and 1, ensuring that each variable contributed equally to the model.

4 Experimental Results and Discussion

In Tables 3 and 4, the optimal hyperparameters configuration of the different ESN-based architectures is presented. The results show that incorporating feedback into the network tends to increase the leaking rate, but only when such an increase actually improves prediction performance. Otherwise, as observed in flow rate prediction, the leaking rate remains lower, which aligns well with expectations given that higher leaking rates intensify the influence of the feedback connection as presented in Eq. 2 and Eq. 4. Furthermore, the leaking rates identified reflect the underlying system dynamics, with slower processes (such as level and temperature) exhibiting lower leaking rates, and faster processes (such as flow rate and pressure) showing higher values

Table 3. Optimal hyperparamters of shallow ESNs

Hyperparameters	No feedback				Feedback			
	ESN-L	ESN-P	ESN-Q	ESN-T	ESN-L	ESN-P	ESN-Q	ESN-T
Reservoir units	300	500	500	500	300	500	500	600
spectral radius	0.5	0.9	0.9	1.1	0.9	0.5	0.65	0.5
leaking rate	0.01	1	0.5	0.01	0.1	0.5	0.6	0.05
input scaling	1	0.5	1	0.1	1	1	0.5	0.1
input connectivity	0.5	0.75	0.75	0.5	0.5	0.5	0.5	0.75
ridge	0.00001	0.0001	0.0001	0.01	0.00001	0.00001	0.00001	0.0001
reservoir connectivity	0.7	0.5	0.4	0.1	0.5	0.4	0.7	0.05
bias	0.0001	0.5	0.1	0.1	0.0001	0.0001	0.5	0.1

Table 4. Optimal hyperparameters deep ESNs

Hyperparameters	No feedback				Feedback			
	ESN-L	ESN-P	ESN-Q	ESN-T	ESN-L	ESN-P	ESN-Q	ESN-T
Reservoir units	500	500	500	1500	900	500	300	1500
spectral radius	0.1	0.9	0.8	0.7	0.1	0.3	0.9	0.7
leaking rate	0.05	0.9	0.5	0.01	0.1	1	0.5	0.05
input scaling	0.5	0.5	1	0.1	0.5	0.1	1	0.5
input connectivity	0.5	0.5	0.1	0.1	0.5	0.5	0.5	0.1
ridge	0.00001	0.0001	0.00001	0.00001	0.00001	0.00001	0.00001	0.00001
reservoir connectivity	0.5	0.5	0.7	0.7	0.6	0.1	0.1	0.5
bias	0.1	0.1	0.01	0.0001	0.1	0.5	0.01	0.0001
layers	7	7	5	3	5	5	5	9
inter laying scaling	1	0.5	1	0.1	1	1	1	0.1

Table 5. Results on the test dataset

Shallow ESN								
	No feedback				Feedback			
	Level	Pressure	Flow	Temp.	Level	Pressure	Flow	Temp.
RMSE	0.0078	0.0059	0.0046	0.0526	0.0059	0.0059	0.0060	0.0287
R2	0.9989	0.9992	0.9991	0.8241	0.9994	0.9986	0.9985	0.9476
Deep ESN								
	No Feedback				Feedback			
	Level	Pressure	Flow	Temp.	Level	Pressure	Flow	Temp.
RMSE	0.0047	0.0058	0.0045	0.0449	0.0031	0.0066	0.0063	0.0225
R2	0.9996	0.9992	0.9992	0.8718	0.9998	0.9989	0.9984	0.9678

The four different architectures proposed in the methodology (shallow and deep, with and without feedback) have been tested for each variable to be predicted (level, pressure, flow rate and temperature process). Table 5 summarizes the performance of the ESN on the test dataset. The results highlight significant variations depending on the architecture and the inclusion of feedback, particularly for the temperature variable.

As shown in Table 5, the shallow ESNs without feedback yielded high accuracy for level, pressure, and flow rate predictions, with very good RMSE and R^2 values close to 0 and 0.999, respectively. These results are in line with the good baseline performance of ESNs for these variables, leaving a narrow improvement margin. Similarly, deep ESNs without feedback provided similar or slightly better results for these variables. However, for temperature, the shallow ESNs without

feedback showed significantly higher RMSE and lower R^2 , indicating that this configuration is not able to capture the dynamics of this variable well.

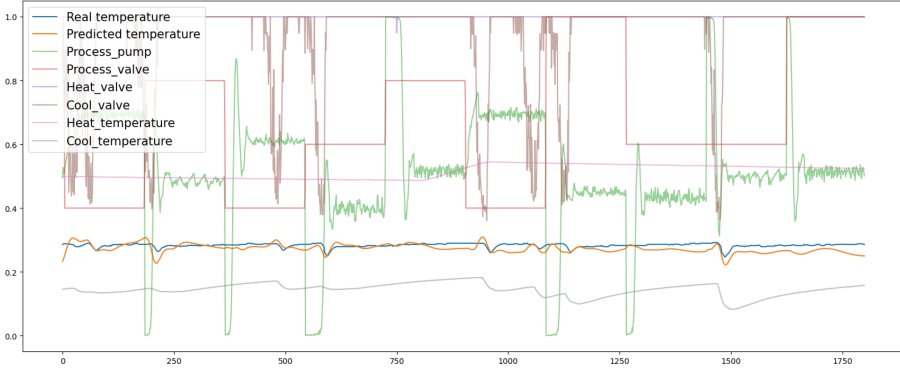


Fig. 3. Prediction of temperature (shallow ESN without feedback)

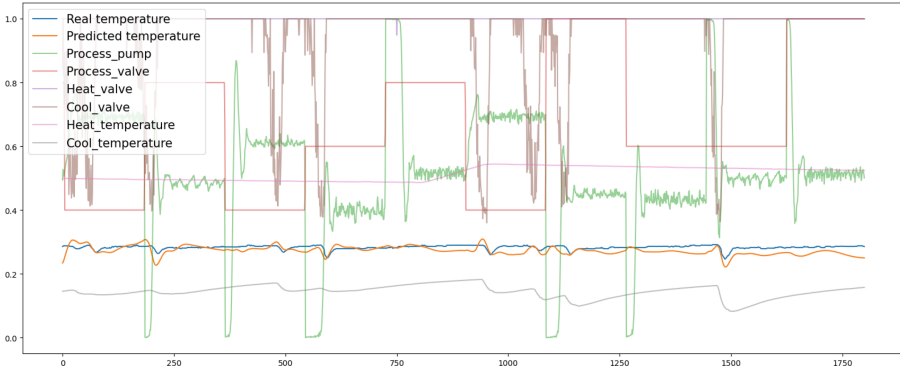


Fig. 4. Prediction of temperature (deep ESN with feedback)

The inclusion of feedback in the networks produced mixed results. For pressure and flow rate, feedback had minimal or no impact on performance, as these variables already achieved near-optimal accuracy without feedback. But for level, feedback configuration slightly improved the performance, as the dynamics of this variable contain some inertia that is reinforced with the configuration. Conversely, the impact of feedback was most evident in temperature predictions. In shallow ESNs, introducing feedback reduced RMSE from 0.0526 to 0.0287 (nearly doubling the prediction accuracy) and increased R^2 from 0.8241 to 0.9476. This improvement highlights the importance of feedback for capturing

the slower, inertia-driven dynamics of temperature. Deep ESNs with feedback further enhanced temperature predictions, achieving an RMSE of 0.0225 and an R^2 of 0.9678. These improvements are further illustrated in Fig. 3, which shows the temperature prediction using the shallow ESN, and in Fig. 4, which displays the enhanced results achieved by the deep ESN with feedback.

5 Conclusions

This paper presents the implementation of ESNs for modelling complex industrial systems with non-linear dynamics for the development of digital twins. An industrial pilot plant is used for the experiments. Four variables with different dynamics are controlled: level, pressure, flow, and temperature. An ESN model was developed to estimate each variable of the system. Additionally, each model was tested with different architectures, including deep networks, feedback integration, and a combination of both (deep networks with feedback), to analyze the impact of each configuration on the ability to model the system's dynamic behaviour.

The results depend on the architecture and the inclusion of feedback, in specific for the temperature variable. Generally speaking, the prediction of each variable shows good values, especially the estimation of the pressure and flow rate of the process circuit. For the estimation of these two variables, neither the deep architectures nor the inclusion of feedback vary the results sufficiently to justify their implementation. Especially in the case of deep architectures, which generate a more complex architecture.

On the other hand, to obtain the dynamics of the main tank level, both deep networks and feedback connection improve the results since it is a variable with slower dynamics and some inertia. The best results are obtained by combining deep networks with feedback. The most significant improvement resulting from the introduction of feedback is observed in the process temperature variable. The network with feedback outperforms even the deep architecture without feedback, reinforcing the idea that systems or variables with higher inertia benefit more from feedback mechanisms in capturing their dynamic behavior.

Future work will focus on implementing these models for the development of digital twins. To achieve this, Echo State Networks (ESNs) will be compared with other low-computational-cost approaches, such as Sparse Identification of Non-linear Dynamics (SINDy), as well as with more traditional models including Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRU), to evaluate their relative effectiveness in modelling industrial processes. Additionally, an analysis will be conducted to determine if the number of models can be reduced without losing accuracy, transitioning from separate models for each system component, through an intermediate option of grouping variables with similar dynamics, to a single global model. This approach would simplify implementation and facilitate integration into digital twin architectures.

Acknowledgements. This work was supported by grant PID2020-117890RB-I00 funded by MCIN/AEI/10.13039/501100011033.

References

1. Barancelli Schwedersky, B., Costa Flesch, R.C., Sirino Danguì, H.A., Arrigoni Iervolino, L.: Practical nonlinear model predictive control using an echo state network model. In: 2018 International Joint Conference on Neural Networks (IJCNN). IEEE (2018). <https://doi.org/10.1109/IJCNN.2018.8489446>
2. Birk, W., Hostettler, R., Razi, M., Atta, K., Tammia, R.: Automatic generation and updating of process industrial digital twins for estimation and control - a review. *Front. Control Eng.* (2022). <https://doi.org/10.3389/fcteg.2022.954858>
3. Chakraborty, S., Adhikari, S.: Machine learning based digital twin for dynamical systems with multiple time-scales. *Comput. Struct.* (2021). <https://doi.org/10.1016/j.compstruc.2020.106410>
4. De Vita, F., Nocera, G., Bruneo, D., Das, S.K.: A novel echo state network autoencoder for anomaly detection in industrial IoT systems. *IEEE Trans. Industr. Inf.* (2023). <https://doi.org/10.1109/TII.2022.3224981>
5. Dettori, S., Martino, I., Colla, V., Speets, R.: Deep echo state networks in industrial applications. In: Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, 5–7 June 2020, Proceedings, Part II 16. Springer (2020). https://doi.org/10.1007/978-3-030-49186-4_5
6. Fuller, A., Fan, Z., Day, C.: Digital twin: enabling technologies, challenges and open research. *IEEE Access*, 108952–108971 (2019). <https://doi.org/10.1109/ACCESS.2020.2998358>
7. González-Herbón, R., González-Mateos, G., Rodríguez-Ossorio, J., Domínguez, M., Castro, S.A., Fuertes-Martínez, J.J.: An approach to develop digital twins in industry. *Sensors* (2024). <https://doi.org/10.3390/s24030998>
8. Grieves, M., Vickers, J.: Digital twin: mitigating unpredictable, undesirable emergent behavior in complex systems. *Transdiscip. Perspect. Complex Syst. New Findings Approach*. (2017). https://doi.org/10.1007/978-3-319-38756-7_4
9. Jones, D., Snider, C., Nassehi, A., Yon, J., Hicks, B.: Characterising the digital twin: a systematic literature review. *CIRP J. Manuf. Sci. Technol.* (2020). <https://doi.org/10.1016/j.cirpj.2020.02.002>
10. Kapteyn, M.G., Knezevic, D.J., Huynh, D., Tran, M., Willcox, K.E.: Data-driven physics-based digital twins via a library of component-based reduced-order models. *Int. J. Numer. Meth. Eng.* (2022). <https://doi.org/10.1002/nme.6423>
11. Kong, L.W., Weng, Y., Glaz, B., Haile, M., Lai, Y.C.: Reservoir computing as digital twins for nonlinear dynamical systems. *Chaos: Interdiscip. J. Nonlinear Sci.* (2023). <https://doi.org/10.1063/5.0138661>
12. Lukoševičius, M.: A practical guide to applying echo state networks. In: *Neural Networks: Tricks of the Trade: Second Edition*. Springer, Cham (2012). https://doi.org/10.1007/978-3-642-35289-8_36
13. Min, Q., Lu, Y., Liu, Z., Su, C., Wang, B.: Machine learning based digital twin framework for production optimization in petrochemical industry. *Int. J. Inf. Manag.* (2019). <https://doi.org/10.1016/J.IJINFOMGT.2019.05.020>
14. Patanè, L., Xibilia, M.G.: Echo-state networks for soft sensor design in an SRU process. *Inf. Sci.* (2021). <https://doi.org/10.1016/J.INS.2021.03.013>
15. Rodríguez-Ossorio, J.R., Morán, A., Alonso, S., Pérez, D., Díaz, I., Domínguez, M.: Echo state networks for anomaly detection in industrial systems. *IFAC-PapersOnLine* (2023). <https://doi.org/10.1016/j.ifacol.2023.10.1836>

16. Sun, Q., Ge, Z.: A survey on deep learning for data-driven soft sensors. *IEEE Trans. Industr. Inf.* (2021). <https://doi.org/10.1109/TII.2021.3053128>
17. Tao, F., Zhang, H., Liu, A., Nee, A.: Digital twin in industry: state-of-the-art. *IEEE Trans. Industr. Inf.* (2019). <https://doi.org/10.1109/TII.2018.2873186>
18. Trouvain, N., Pedrelli, L., Dinh, T.T., Hinaut, X.: Reservoirpy: an efficient and user-friendly library to design echo state networks. In: *International Conference on Artificial Neural Networks* (2020). https://doi.org/10.1007/978-3-030-61616-8_40
19. Wagg, D., Worden, K., Barthorpe, R., Gardner, P.: Digital twins: state-of-the-art and future directions for modeling and simulation in engineering dynamics applications. *ASCE-ASME J. Risk Uncertain. Eng. Syst. Part B: Mech. Eng.* (2020). <https://doi.org/10.1115/1.4046739>



Harnessing Machine Learning for Rain Induced Landslide Detection and Analysis

Shelly Gupta^{1,3} , Hussain Otudi^{1,2,3} , Ameen Abdel Hai^{1,3} ,
Rafaa Aljurbua^{1,3} , Jovan Andjelkovic^{1,3} , Abdulrahman Alharbi^{1,2,3} ,
and Zoran Obradovic^{1,3}

- ¹ Center for Data Analytics and Biomedical Informatics, Computer and Information Science Department, Temple University, Philadelphia, PA, USA
{shelly.gupta,hussain.otudi,aabdelhai,rafaa.aljurbua,
jovan.andjelkovic,adalharbi,zoran.obradovic}@temple.edu
- ² Department of Computer Science, College of Engineering and Computer Science, Jazan University, Jazan, Saudi Arabia
- ³ Department of Computer Science, College of Computer, Qassim University, Buraydah, Saudi Arabia

Abstract. Landslides pose significant risks to infrastructure, ecosystems, and human lives, making accurate prediction crucial for disaster preparedness and mitigation. We integrate multimodal environmental data to enhance landslide prediction using machine learning. Specifically, we combine temporal weather data from ASOS, static vegetation data from NLCD, static soil composition data from SOLUS100, and temporal soil attributes from ERA5-Land to estimate landslide probability within a 5 km radius of ASOS weather stations across six U.S. states. We frame this as a multiclass classification problem, predicting high, low, or no landslide probability. Given the inherent imbalance in landslide occurrence, we explore various techniques such as SMOTE oversampling, class-weighted training, and dimensionality reduction to improve model performance. Our results indicate that XGBoost trained on SMOTE-balanced, PCA-reduced data incorporating all four datasets achieves the highest macro F1-score of 0.70. Analysis of feature importance reveals that significant predictors span all datasets, highlighting the necessity of integrating diverse environmental variables. Additionally, we conduct state-wise and seasonal comparisons to assess regional variations in model effectiveness. This research demonstrates the potential of multimodal data fusion and machine learning in landslide forecasting, paving the way for more robust and interpretable predictive models for natural hazard assessment.

Keywords: Landslides · Multimodal · Machine Learning · Feature selection

1 Introduction

Landslides triggered by heavy rainfall pose a significant threat to public infrastructure, resulting in substantial human, material, and economic losses world-

wide [12]. Detecting these shallow, rainfall-induced landslides is critical for effective disaster management and risk mitigation. These events are influenced by a complex interplay of natural factors, including topography, geology, soil composition, and climatic conditions. Understanding the multimodal mechanisms driving landslide formation is, therefore, essential for developing predictive models and mitigation strategies.

Machine learning and deep neural networks have been shown to be useful in many geo-based applications, specifically landslides. Many works have focused on change detection-based algorithms [7], but it is resource-intensive. It requires both pre-event and post-event images, which can be difficult to obtain in a timely manner, particularly in regions with persistent cloud cover or adverse weather conditions. Similar aerial images have been used to detect landslides, but they have proven ineffective in areas with heavy vegetation. In Minnesota, with 35% forest cover, and in Wisconsin, with 48.98% forest cover [1], a significant portion of the land is hidden from aerial view, thus making imaging ineffective. Even with high-resolution imagery, the pixel size may not be sufficient to detect smaller landslides, which tree canopies can hide. Some studies have utilized ground-based measurements to assess landslide occurrences. These approaches leverage environmental and geotechnical data, such as soil properties, precipitation levels, and terrain characteristics, to predict landslide susceptibility [13]. However, most of these studies focus on regions outside the United States and often lack a detailed analysis of the factors influencing their models' decision-making processes. Understanding which environmental variables contribute the most to landslide predictions is crucial for improving the interpretability and reliability of the model, particularly in diverse regions.

In this study, we integrate multimodal data sources to enhance landslide prediction by leveraging both static and temporal environmental variables. Specifically, we incorporate temporal weather data from the Automated Surface Observing Systems (ASOS) [4], static vegetation data from the National Land Cover Database (NLCD) [11], static soil composition data from the Soil Landscapes of the United States (SOLUS) [16], and temporal soil attributes from ERA5-Land [15]. By combining these diverse datasets, we aim to develop a robust machine learning framework to estimate landslide probability within a 5 km radius of ASOS weather station coordinates across six U.S. states: Minnesota, North Dakota, South Dakota, Iowa, Michigan, and Wisconsin. We frame this as a multiclass classification problem, where each instance is categorized into one of three classes: high probability of landslide, low probability of landslide, or no landslide. However, since landslides are rare events, our dataset is highly imbalanced, which presents challenges for predictive modeling. To address this, we explore various imbalance mitigation techniques and analyze their impact on classification performance. In this work, we make the following key contributions:

- We integrate diverse data sources, combining meteorological, soil, and vegetation attributes to improve landslide prediction accuracy.
- We employ multiple traditional and deep learning models to classify landslide probability while addressing challenges posed by data imbalance.

- We conduct a dimension reduction analysis to assess whether reducing feature space improves classification performance.
- We analyze model explainability by identifying the most influential features contributing to landslide prediction.
- We compare model performance across different states and seasons to evaluate whether localized models outperform general predictive models.

Through this comprehensive approach, we aim to provide insights into the key environmental factors influencing landslides and assess the effectiveness of different modeling strategies for handling highly imbalanced geospatial data.

2 Related Works

Landslide prediction has been explored using various methodologies, including traditional machine learning, deep learning, and heuristic-based approaches. However, many of these studies rely solely on image-based models, overlooking the integration of multimodal environmental data or focusing on geographically constrained regions, limiting their generalizability.

Several studies have employed machine learning techniques on aerial imagery for landslide detection. For instance, traditional machine learning models were used but primarily focused on images, overlooking other environmental factors such as soil composition, vegetation, and meteorological data [10]. While effective in certain regions, this approach lacks robustness in areas with dense vegetation or limited image availability due to cloud cover or adverse weather conditions.

Other studies have focused on heuristic-based models rather than machine learning. In particular, Weather and Research Forecast (WRF) models were employed alongside geomorphological features to detect landslides [20]. However, that approach only considered two classes (landslide and no landslide) and did not incorporate machine learning techniques, limiting the ability to adapt to complex, nonlinear relationships within the data.

Deep learning models have also been used for landslide prediction, particularly in image-based approaches. PlanetScope, Sentinel-2 imagery, and ALOS-PALSAR2 elevation models were leveraged to predict landslides using convolutional neural networks (CNNs) [19]. However, this method is both costly and ineffective in regions with dense forest cover, such as Wisconsin, where tree canopies obscure terrain features critical for landslide detection.

Hydrological models have also been explored, but their applicability remains uncertain across different geographic regions. A hydrological model was developed for landslide prediction in Chile [9], but the findings may not generalize well to other regions, including the Midwestern United States, due to differences in terrain, climate, and geological features. The challenges of generalizing landslide prediction models across different geographical areas were further emphasized, highlighting the need for localized datasets and models tailored to specific environmental conditions [8]. Additionally, the prior study lacked soil moisture measurements and relied on traditional data collection methods instead of real-time IoT sensors, which could improve predictive accuracy. Multimodal studies

have been shown to help in various weather-based problems [2, 17, 18]. Future advancements in landslide modeling, such as those proposed by [14], suggest integrating IoT sensor networks to enhance data granularity and timeliness.

Some research has explored alternative approaches, such as clustering and early warning systems. [6] investigated landslide clustering, but their work had limited application as a real-time warning system and did not analyze the explainability of contributing factors. Similarly, an initial landslide detection model was developed and provided insights into contributing factors [5]. Still, the study was confined to Seattle, Washington, and did not account for vegetation data, which plays a critical role in stabilizing slopes and influencing landslide susceptibility. Our work bridges the gap between traditional and deep learning approaches by proving that multimodal data fusion enhances prediction accuracy and generalizability across different geographic regions.

3 Data

In this study, we integrate multimodal data from multiple sources to enhance the accuracy of landslide detection. Specifically, we incorporate temporal weather data from the Automated Surface Observing Systems (ASOS), static vegetation data from the National Land Cover Database (NLCD), static soil composition data from the Soil Landscapes of the United States (SOLUS100), and temporal soil attributes data from ERA5-Land. Each of these datasets contributes unique and complementary information relevant to landslide susceptibility.

Our objective is to predict the probability of landslides using labels derived from the NASA Global Landslide Nowcast database, which provides high-resolution landslide hazard predictions. By integrating diverse data sources, we aim to capture environmental factors influencing landslide occurrence. The following subsections detail each data source, including its attributes and spatial and temporal resolution.

3.1 NASA Global Landslide Nowcast Dataset

The NASA Global Landslide Nowcast, generated by the Landslide Hazard Analysis for Situational Awareness (LHASA) model, identifies regions with a high likelihood of landslide occurrence on a daily basis [3]. LHASA integrates satellite-based precipitation estimates with a landslide susceptibility map to assess risk. While the model has the capability to run every 30 min, the dataset used in this study consists of a daily archive derived from past runs, covering latitudes from 60°N to 60°S.

For this analysis, we retrieved data from 2018 to 2020. Given a specific coordinate, we extracted relevant raster files and identified all data points within a 5 km radius, using the Haversine distance formula to account for Earth's curvature. The Haversine formula is given by:

$$d = 2R \arcsin \left(\sqrt{\sin^2 \left(\frac{\Delta\varphi}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\Delta\lambda}{2} \right)} \right) \quad (1)$$

where d is the great-circle distance between two points, R is Earth's radius (approximately 6371 km), φ_1, φ_2 are the latitudes of the two points in radians, λ_1, λ_2 are the longitudes of the two points in radians, $\Delta\varphi = \varphi_2 - \varphi_1$ and $\Delta\lambda = \lambda_2 - \lambda_1$. After obtaining the relevant points, we calculated the landslide probability by dividing the total number of labeled landslide occurrences by the total number of labels. A probability greater than 0.5 was classified as a high probability of landslide occurrence, while a probability between 0 and 0.5 was considered low. If no occurrences were detected, the area was classified as having no landslide risk.

3.2 Automated Surface Observing Systems Dataset

This study seeks to detect landslides occurring within a 5 km radius of an Automated Surface Observing Systems (ASOS) weather station. ASOS is the primary surface weather network in the United States, providing critical meteorological data that are essential for aviation safety [4]. The network consists of over a thousand weather stations distributed nationwide. For this analysis, we identified ASOS sensors located in Minnesota, Wisconsin, North Dakota, South Dakota, Iowa, and Michigan. We extracted various meteorological variables, including air temperature, dew point temperature, wind speed (in knots), one-hour precipitation, pressure altimeter, wind gust speed (in knots), sky altitude at various levels, ice accretion over 1, 3, and 6 h, and peak wind gust speed.

The ASOS data is recorded at varying temporal resolutions, ranging from 1-minute to 5-minute intervals. Given the focus of this study on landslide detection at the end of each day, it was necessary to aggregate the data to a daily resolution. This aggregation was performed by calculating each attribute's mean, standard deviation, minimum, and maximum values. The daily summary statistics provide a standardized representation of the weather conditions over time, aligning the temporal scale of the weather data with that of the landslide occurrences.

3.3 ERA5-Land Dataset

ERA5-Land is a high-resolution reanalysis dataset that consistently represents land surface variables over time with a spatial resolution of approximately 9 km [15]. The dataset offers vertical coverage extending from 2 m above the surface to a soil depth of 289 cm and is available at an hourly temporal resolution. For this study, we collected 28 meteorological and land surface attributes per location. These attributes include forecast albedo, snow layer temperature, soil temperature at various depths, surface temperature, skin reservoir content, dew point temperature, and air temperature at 2 m above the surface. Additionally, we extracted wind speed in different directions, surface pressure, leaf area index, soil water volume at multiple depths, and snow-related variables such as cover, mass, and depth. Furthermore, we incorporated lake-related parameters, including depth, temperature, and the thickness of the uppermost lake layer.

For this study, we acquired ERA5-Land data for the six selected states and refined the dataset by retaining only the data points located within a 5 km

radius of ASOS weather stations. To ensure consistency with the temporal scale of the analysis, the hourly data was aggregated to a daily resolution. This was achieved by computing each attribute’s mean, standard deviation, minimum, and maximum values, providing a comprehensive statistical representation of daily variations in land surface and meteorological conditions.

3.4 Soil Landscapes of the United States 100-Meter Dataset

The Soil Landscapes of the United States (SOLUS) dataset provides high resolution (100 m) maps documenting 20 key soil properties [16]. These properties include bulk density (oven dry), calcium carbonate content, cation exchange capacity, clay content, coarse sand, electrical conductivity (saturated paste), effective cation exchange capacity, fine sand, gypsum, medium sand, pH, rock content, total sand content, sodium adsorption ratio, silt content, soil organic carbon, very coarse sand, very fine sand and depth to bedrock.

The dataset is stored as raster files and covers the conterminous United States. Soil properties are documented at multiple depths, specifically at 0, 5, 15, 30, 60, 100, and 150 cm. For this study, we extracted soil property data for locations within a 5 km radius of ASOS weather stations. To maintain consistency with other datasets, we aggregated these attributes using the same approach, computing each property’s mean, standard deviation, minimum, and maximum values.

3.5 National Land Cover Database

The National Land Cover Database (NLCD) provides comprehensive land cover classification and tree canopy cover data for the United States, offering area coverage statistics at the county level for 20 land cover categories [11]. These categories include open water, perennial ice/snow, developed land (ranging from open space to high intensity), barren land, various forest types (deciduous, evergreen, and mixed), shrubland, grassland, pasture/hay, cultivated crops, and wetland types (woody wetlands and emergent herbaceous wetlands). To integrate NLCD data into this study, we identified the county corresponding to each ASOS weather station and mapped the respective land cover attributes accordingly.

4 Experimental Setup

We integrate multiple datasets, including soil composition data (SOLUS100), vegetation data (NLCD), soil attributes (ERA5), and weather data (ASOS), to detect landslides within a 5 km radius of an ASOS weather station. The weather stations are distributed across six states: Minnesota, North Dakota, South Dakota, Wisconsin, Michigan, and Iowa. Initially, we identified 360 weather stations; however, we narrowed our analysis to locations that experienced at least one landslide event between 2018 and 2020, ultimately reducing the dataset to 43 stations.

We performed a temporal split, using all records from 2018 and 2019 for training and those from 2020 for testing. We also partitioned the training set, allocating 80% for model training and 20% for validation. We frame this task as a multiclass classification problem, where each instance is assigned one of three labels: ‘0’ (no landslide), ‘1’ (low probability of landslide), and ‘2’ (high probability of landslide). A key challenge in this study is the class imbalance in the dataset: out of a total of 46,904 samples, 43,864 instances (93.51%) are labeled as ‘0’, 2,970 instances (6.33%) are labeled as ‘1’, and only 70 instances (0.14%) are labeled as ‘2’.

In this study, we experiment with various machine learning models to assess their performance in detecting landslides. The models tested include ensemble methods such as Random Forest, LightGBM, and XGBoost, as well as simpler linear models like logistic regression and Support Vector Classifier (SVC). Random Forest, an ensemble learning technique, is known for its robustness and ability to handle non-linear relationships, while LightGBM and XGBoost are gradient-boosting methods that are effective in handling large datasets with complex interactions. Logistic regression, a widely used linear classifier, provides a benchmark for performance in simpler, less complex datasets. SVC is considered for performing well with high-dimensional data and small sample sizes.

In addition to these traditional machine learning algorithms, we also train deep learning models, specifically neural networks, to explore their ability to capture non-linear patterns in the data. These neural networks are regularized using kernel methods to prevent overfitting, dropout techniques are employed to randomly drop units during training, improving generalization. The ReLU activation function is used for introducing non-linearity, which helps the model learn complex representations, and batch normalization is applied to accelerate training by normalizing the inputs to each layer. This combination of regularization and activation functions allows the model to learn from the data effectively while preventing overfitting and ensuring convergence.

Given the inherent class imbalance in our dataset, where landslides are rare events, we employ several techniques to address this issue and improve the model’s ability to predict the minority classes correctly. We first utilize the Synthetic Minority Over-sampling Technique (SMOTE), which generates synthetic instances for the minority class by interpolating between existing minority class samples. This method helps balance the distribution of the classes without losing any information from the majority class. We also experiment with SMOTEEN, a hybrid approach that combines both oversampling of the minority class and undersampling of the majority class. This hybrid technique aims to provide a more balanced dataset while avoiding the loss of potentially important majority class data. Finally, we incorporate class weights into the training process, where the each class’s weight is inversely proportional to its frequency in the training data. This method adjusts the model’s learning process, ensuring that it places greater emphasis on the minority classes during training.

To further improve model performance and computational efficiency, we explore dimensionality reduction techniques due to the diverse nature of the

data sources and the high number of features involved. Principal Component Analysis (PCA) is used to reduce the dimensionality of the data by projecting it onto a lower-dimensional space that captures the most variance in the dataset. Similarly, Singular Value Decomposition (SVD) is employed to decompose the data matrix and reduce its rank, which can help capture the essential information while discarding noise. We train models using both PCA- and SVD-reduced data, and the original feature set to compare the impact of dimensionality reduction on model performance. These techniques help mitigate overfitting and computation time issues, especially in high-dimensional datasets.

4.1 Evaluation Metrics

Given the imbalanced nature of our dataset, where most samples are labeled as ‘no landslide’ (class 0) and the landslides are much less frequent, we assess the performance of our models using the macro F1 score. The macro F1 score is a valuable metric in imbalanced classification problems, as it computes the F1 score for each class independently and then averages them without considering the class distribution. This allows us to equally evaluate the performance across all classes (no landslide, low probability of landslide, and high probability of landslide), ensuring that the model’s performance on the minority classes is adequately reflected.

In addition to the macro F1 score, we also report the false alarm rate and hit rate. The false alarm rate (FAR) is defined as the proportion of non-landslide instances (class 0) that are incorrectly predicted as landslides (class 1 or class 2). It can be calculated using the following formula:

$$\text{False Alarm Rate} = \frac{\text{False Positives (FP)}}{\text{False Positives (FP)} + \text{True Negatives (TN)}} \quad (2)$$

The hit rate (HR), on the other hand, quantifies the proportion of actual landslides (classes 1 and 2) that the model correctly identifies. It is calculated using the formula:

$$\text{Hit Rate} = \frac{\text{True Positives (TP)}}{\text{True Positives (TP)} + \text{False Negatives (FN)}} \quad (3)$$

Both metrics provide valuable insight into the model’s ability to detect landslides while minimizing the number of incorrect predictions.

5 Results

We conducted extensive experiments to assess how dataset combinations, imbalance handling, dimensionality reduction, and model choice affect landslide prediction. We tested each setup with and without oversampling and dimension reduction and reported the best model and configuration for each dataset combination. As shown in Table 1, integrating all four datasets (ASOS, ERA5, NLCD,

Table 1. Landslide detection in six states (2020) using 2018–2019 data. Performance comparison with oversampling, dimensionality reduction, and ML models.

Data	Oversampling Technique	Dimension Reduction	Best Model	Macro F1 Score
ASOS	SMOTE	-	Random Forest	0.63
ERA5	-	-	XGBoost	0.62
ASOS+NLCD	SMOTE	-	Random Forest	0.67
ERA5+NLCD	-	SVD	XGBoost	0.64
ASOS+SOLUS	-	-	XGBoost	0.66
ERA5+SOLUS	SMOTE	PCA	XGBoost	0.64
ASOS+ERA5	-	-	XGBoost	0.64
ASOS+ERA5+NLCD	-	PCA	XGBoost	0.66
ASOS+NLCD+SOLUS	SMOTE	PCA	XGBoost	0.68
ERA5+NLCD+SOLUS	SMOTE	-	Neural Net	0.61
ASOS+ERA5+NLCD+SOLUS	SMOTE	PCA	XGBoost	0.70

SOLUS) achieved the highest Macro F1 Score of 0.70, highlighting the value of multimodal environmental data. Our findings indicate that as the number of datasets increases, it leads to a higher number of input features. In these cases, SMOTE and PCA become particularly useful in improving classification performance. XGBoost without PCA achieved an F1 macro of 0.68, whereas without oversampling, it reached 0.64. Furthermore, our results consistently show that XGBoost outperforms other models, making it the most effective choice for landslide prediction across different dataset configurations. The highest-performing model (XGBoost trained on the full dataset with SMOTE and PCA) achieves a Macro F1 Score of 0.70. XGBoost approach is able to capture complex interactions between environmental variables better than simpler models. Interestingly, neural networks underperformed in our experiments, likely due to high-class imbalance. Even though we used weighted cross entropy and trained on synthetically oversampled data, our experiments showed that neural networks could not generalize well. Our experiments demonstrate that combining multiple data sources, leveraging oversampling techniques, and applying dimensionality reduc-

Table 2. Individual class performance. Hit rate measures correctly detected positives, while false alarm rate measures incorrectly detected negatives as positives.

Metric	No landslide	Low probability	High probability
Hit Rate	96.5%	52.2%	76.5%
False Alarm Rate	2.7%	54.7%	43.5%

tion contribute to enhanced predictive accuracy. These findings underscore the importance of multimodal data integration and preprocessing to improve landslide prediction models.

To further evaluate the best model’s performance (XGBoost with SMOTE and PCA trained on 4 datasets combined), we conduct a class-wise analysis of hit and false alarm rates, as shown in Table 2. For the “No Landslide” class, the model achieves a high hit rate (96.5%) and a low false alarm rate (2.7%), indicating strong performance in identifying non-landslide events. This suggests that the model is well-calibrated for detecting stable conditions but may be biased toward the majority class due to dataset imbalance. The hit rate for the “Low Probability” class is 52.2%, meaning nearly half of the low-probability landslides are missed. The false alarm rate (54.7%) is high, indicating that many events are incorrectly classified as low probability when they belong to other classes. For the “High Probability” class, the model performs better, with a hit rate of 76.5%, meaning it detects most of the high-risk landslide cases. However, the false alarm rate of 43.5% suggests that many non-landslide events are misclassified as high probability, leading to potential over-warning situations. From a disaster management perspective, these results are acceptable if the primary goal is to predict high-risk landslides, as the model successfully captures most major events. Additionally, the low detection of low-probability landslides (HR = 52.2%) means that smaller-scale events are often overlooked, which could be a concern in regions where even minor landslides pose significant risks.

Next, we investigate which features contribute the most to the landslide detection, and the results are shown in Table 4. For this experiment, we analyze the best-performing model, XGBoost with SMOTE and PCA. After training the model, we analyze the principal components that had the highest influence on predictions and backtrack their contributions to the original attributes. The results reveal that important features originate from all four datasets, demonstrating that an integrated approach leveraging multiple data modalities enhances predictive performance. Notably, land cover attributes from NLCD and soil composition features from SOLUS appear frequently, highlighting the strong influence of terrain and soil properties on landslide susceptibility. Additionally, meteorological attributes from ERA5 and ASOS, such as horizontal wind speed, lake bottom temperature, and cloud layer height, emphasize the role of atmospheric conditions in triggering landslides. These findings reinforce the importance of combining static environmental features and dynamic weather patterns to improve landslide risk assessment, ultimately supporting the development of more accurate and reliable predictive models (Table 3).

To assess individual feature contributions without dimensionality reduction, we train XGBoost on the original dataset with class weights. The results, shown in Table 4, reveal that while some features remain consistent with the PCA-transformed model, such as evergreen forest area (NLCD) and coarse sand percentage (SOLUS), others differ. Notably, features like precipitation amount (ASOS), soil moisture content (ERA5), and soil chemistry such as pH and gypsum content (SOLUS) emerges, indicating that training on raw data high-

Table 3. Top features ranked by importance from the model trained on the PCA-reduced dataset, listed in decreasing order.

Feature	Source
Area of Evergreen Forest	NLCD
Lake bottom temperature	ERA5
Fine sand% (5cm depth)	SOLUS
Horizontal speed of air (mean)	ERA5
Horizontal speed of air (max)	ERA5
Very Coarse sand% (15cm depth)	SOLUS
Developed, Low Intensity	NLCD
Areas dominated by shrubs	NLCD
Area of Barren Land (Rock/Sand/Clay)	NLCD
Thickness of the uppermost layer of inland water bodies	ERA5
Height of the lowest cloud layer	ASOS
Coarse sand% (5cm depth)	SOLUS
Area of Open Water	NLCD
Area of Pasture/Hay	NLCD

lights additional factors that PCA de-emphasized. Despite variations in individual features, key environmental patterns persist. Water availability in soil remains critical, as reflected in soil moisture, precipitation, and soil water volume features, while soil composition plays a significant role. Additionally, soil temperature variations at different depths (ERA5) emerge as important factors, further emphasizing the role of subsurface conditions in landslide susceptibility. Crucially, significant features continue to originate from all four datasets, reinforcing the importance of an integrated approach. While PCA enhances feature abstraction, training on original data with class weights preserves explicit environmental variables, ensuring interpretability in landslide risk assessment. These results highlight how different modeling strategies influence feature selection while maintaining consistency in broader predictive patterns.

In the next experiment, we evaluate landslide detection performance across six states using the macro F1 score, comparing three model setups: state-specific models trained individually for each state, an all-states model trained on data from all six states, and a model trained on all states except Minnesota (MN). Minnesota is the only state with three classes, while all other states have two classes. The results in Table 5 show that state-specific models generally perform well, as they are tailored to the unique characteristics of each region. However, the all-states model improves performance in MN (0.68 vs. 0.66 in the state specific model) by providing additional training examples for the underrepresented class. In contrast, states with only two classes experience a decline in performance with the all-states model, as the introduction of an additional class

Table 4. Top features ranked by importance from the model trained on the original dataset, listed in decreasing order.

Feature	Source	PCA Top Feature?
Area of Evergreen Forest	NLCD	Yes
1 h precipitation amount (mean)	ASOS	No
Volume of water in soil layer 1 (0–7 cm)	ERA5	No
Silt content% (30 cm depth)	SOLUS	No
Oven-dry bulk density (60 cm depth)	SOLUS	No
Amount of rain intercepted by foliage and water from dew (mean)	ERA5	No
Sodium Adsorption Ratio (5 cm depth)	SOLUS	No
Air temperature	ASOS	No
Temperature of the surface of the Earth (std)	ERA5	No
Soil temperature at 28–100 cm (std)	ERA5	No
Soil pH(60 cm depth)	SOLUS	No
Gypsum content (100 cm depth)	SOLUS	No
Coarse sand% (5 cm depth)	SOLUS	Yes
Soil pH(0 cm depth)	SOLUS	No
Coarse sand% (15 cm depth)	SOLUS	Yes

creates confusion. Notably, excluding MN from training enhances performance in certain two-class states, such as North Dakota, where the macro F1 increases from 0.76 to 0.81. These results indicate that class distribution differences across states impacts model performance and a single model trained on all states may not be optimal for every region.

Next, we compare the performance of seasonal models with that of the overall model. To achieve this, we categorize the data into four seasons: Summer (June–

Table 5. Performance comparison between state-specific models and an overall model trained on all states.

State	Classes	State Model	All-States Model	Model Excluding MN
Minnesota (MN)	3	0.66	0.68	-NA-
Iowa (IA)	2	0.72	0.70	0.73
North Dakota (ND)	2	0.76	0.68	0.81
South Dakota (SD)	2	0.63	0.68	0.65
Michigan (MI)	2	0.67	0.67	0.68
Wisconsin (WI)	2	0.74	0.73	0.73

Table 6. Performance comparison between season-specific all-states models and an overall all-states model.

Season	Time period	Seasonal model	Overall model
Summer	Jun-Aug	0.56	0.72
Fall	Sep-Nov	0.47	0.49
Winter	Dec-Feb	0.50	0.50
Spring	Mar-May	0.64	0.65

August), Winter (December–February), Fall (September–November), and Spring (March–May). The results of this comparison are presented in Table 6. Our analysis indicates that all seasonal models are affected by temporal splitting. Except for Winter, the overall model consistently outperforms the seasonal models. In the Winter months, performance remains unchanged due to the limited number of landslide occurrences in the testing year. Only two landslides were recorded in these months, and neither model successfully identified them.

6 Conclusion

In this study, we develop a machine learning framework for landslide prediction by integrating multimodal environmental datasets, combining static and temporal attributes from ASOS, NLCD, SOLUS, and ERA5-Land. By framing landslide detection as a multiclass classification problem, we evaluate various machine learning models while addressing the challenges of data imbalance through oversampling techniques such as SMOTE and class-weighted training. Our results demonstrate that integrating all four datasets enhances predictive performance, with XGBoost trained on SMOTE-balanced, PCA-reduced data achieving the highest macro F1-score.

Our analysis highlights the importance of both meteorological and land-based attributes, with features from all datasets contributing significantly to model predictions. But despite achieving strong overall performance, our class-wise evaluation reveals limitations in detecting low-probability landslides, emphasizing the trade-off between reducing false alarms and capturing rare events. We provide insights into how environmental conditions impact landslide susceptibility by comparing models across states and seasons. This work underscores the value of integrating diverse data sources and applying advanced machine learning techniques to predict natural disasters more effectively. Future work could explore further improvements in the model in generalization, including using additional remote sensing data and dynamic aggregation of features.

Acknowledgments. This research was sponsored by the U.S. Army Engineer Research and Development Center (ERDC) and was accomplished under Cooperative Agreement Number W9132V-23-2-0002. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing

the official policies, either expressed or implied, of the U.S. Army Engineer Research and Development Center (ERDC) or the U.S. Government.

References

1. Most Forested States 2024 worldpopulationreview.com. <https://worldpopulationreview.com/state-rankings/most-forested-states>
2. Aljurbua, R., Alshehri, J., Gupta, S., Alharbi, A., Obradovic, Z.: Early prediction of power outage duration through hierarchical spatiotemporal multiplex networks. In: Complex Networks and Their Applications XIII: Proceedings of The Thirtieth International Conference on Complex Networks and Their Applications: COMPLEX NETWORKS 2024-Volume 3, p. 320. Springer, Cham (2024)
3. Amatya, M., Medwedeff, W., Clark, M.K.: Data-driven landslide nowcasting at the global scale
4. ASOS Network: Iowa environmental mesonet, Iowa State University (2022). <https://mesonet.agron.iastate.edu/ASOS/>. Accessed 31 May 2022
5. Baum, R.L., Godt, J.W.: Early warning of rainfall-induced shallow landslides and debris flows in the USA. *Landslides* **7**, 259–272 (2010)
6. Benz, S.A., Blum, P.: Global detection of rainfall-triggered landslide clusters. *Nat. Hazard.* **19**(7), 1433–1444 (2019)
7. Coluzzi, R., et al.: Rapid landslide detection from free optical satellite imagery using a robust change detection technique. *Sci. Rep.* **15**(1), 4697 (2025)
8. Ebrahim, M., et al.: Exploring time series models for landslide prediction: a literature review. *Geoenviron. Disast.* **11**(1), 1–20 (2024)
9. Fustos, I., Abarca-del Río, R., Mardones, M., González, L., Araya, L.: Rainfall-induced landslide identification using numerical modelling: a southern Chile case. *J. S. Am. Earth Sci.* **101**, 102587 (2020)
10. Fustos-Toribio, I., Manque-Roa, N., Vásquez Antipan, D., Hermosilla Sotomayor, M., Letelier Gonzalez, V.: Rainfall-induced landslide early warning system based on corrected mesoscale numerical models: an application for the Southern Andes. *Nat. Hazard.* **22**(6), 2169–2183 (2022)
11. Homer, C., et al.: Completion of the 2011 national land cover database for the conterminous united states-representing a decade of land cover change information. *Photogrammetric Eng. Remote Sens.* **81**(5), 345–354 (2015)
12. Iizuka, S., Xuan, Y., Kondo, Y.: Impacts of disaster mitigation/prevention urban structure models on future urban thermal environment. *Sustain. Urban Areas* **19**, 414–420 (2015)
13. Kang, J., Wan, B., Gao, Z., Zhou, S., Chen, H., Shen, H.: Research on machine learning forecasting and early warning model for rainfall-induced landslides in Yunnan Province. *Sci. Rep.* **14**(1), 14049 (2024)
14. Kuradusenge, M., Kumaran, S., Zennaro, M.: Rainfall-induced landslide prediction using machine learning models: the case of Ngororero district, Rwanda. *Int. J. Environ. Res. Publ. Health* **17**(11), 4147 (2020)
15. Muñoz-Sabater, J., et al.: Era5-land: a state-of-the-art global reanalysis dataset for land applications. *Earth Syst. Sci. Data* **13**(9), 4349–4383 (2021)
16. Nauman, T., et al.: Data from: soil landscapes of the united states 100-meter (solus100) soil property maps project repository (2024)

17. Otudi, H., Gupta, S., Albarakati, N., Obradovic, Z.: Classifying severe weather events by utilizing social sensor data and social network analysis. In: *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining*, pp. 64–71 (2023)
18. Otudi, H., Gupta, S., Obradovic, Z.: Leveraging diverse data sources for enhanced prediction of severe weather-related disruptions across different time horizons. In: *International Conference on Engineering Applications of Neural Networks*, pp. 220–234. Springer, Cham (2024)
19. Salam, R., Pla, F., Ahmed, B., Painho, M.: A convolutional neural network-based approach for automatically detecting rainfall-induced shallow landslides in a data-sparse context. *Nat. Hazards Res.* (2024)
20. Wang, H., Zhang, L., Yin, K., Luo, H., Li, J.: Landslide identification using machine learning. *Geosci. Front.* **12**(1), 351–364 (2021)



Leveraging Computational Geometry for Data Augmentation in Medical Flow Fields Classification

Riccardo Margheritti^{1,2} , Andrea Schillaci¹ , Carlotta Pipolo³ ,
Maurizio Quadrio¹ , and Giacomo Boracchi¹ 

¹ Politecnico di Milano, 20133 Milano, MI, Italy
riccardo.margheritti@mail.polimi.it

² Politecnico di Torino, 10129 Torino, TO, Italy

³ Unita' di Otorinolaringoiatria ASST Santi Paolo e Carlo, 20133 Milano, MI, Italy

Abstract. The application of Machine Learning (ML) to Computational Fluid Dynamics (CFD) has gained significant attention due to its potential in speeding up simulations and approximating numerical solutions of physical equations. Still, the dominant role of ML, which is to infer expert labels that cannot be calculated from explicit equations, has received much less attention in CFD. A major challenge in this direction is the scarcity of large, annotated datasets required to train robust ML models for flow field classification. In this work, we address the problem of training a ML model to classify CFD flow fields, inferring pathologies affecting the human upper airways. We propose a novel data augmentation method to address this limitation which involves an automated pipeline to extract CFD-ready surfaces from CT scans and a computational geometry technique to generate synthetic training samples. By defining deformation functions for specific pathologies on a reference surface and mapping these to healthy anatomical surfaces, we create a large and diverse training set with minimal expert supervision. This method allows for the generation of a dataset with high anatomical variability and well-defined labels, enhancing the model's ability to generalize to unseen geometries. We demonstrate that a Neural Network (NN) can accurately classify two common nasal pathologies, septal deviation and turbinate hypertrophy, achieving strong performance on real pathological patient data despite being trained solely on synthetic samples.

Keywords: Data Augmentation · Computational Fluid Dynamics · Machine Learning · Functional Mapping · Nasal Pathology Classification

1 Introduction

The use of Machine Learning (ML) in Computational Fluid Dynamics (CFD) has rapidly increased due to improved computational power and the potential of deep learning in fluid mechanics [1]. ML techniques are widely used to estimate complex non-linear input-output relationships in motion equations, enhancing or even replacing traditional CFD simulations. As a result, much work has focused

on improving turbulence modeling [2], flow field reconstruction using Convolutional Neural Networks (CNNs) [3], and reducing computational costs with physics-informed deep learning [4]. However, the primary use of ML, which is to infer labels that cannot be computed from equations (but rather defined by expert observations), has been poorly investigated in CFD. This is largely due to the difficulty in collecting CFD data and the vast amount of information contained within a single CFD flow field, which makes it impractical to feed directly into a ML model. As a result, training a Neural Network (NN) with CFD input becomes particularly challenging.

In this work, we address the classification of nasal pathologies, specifically, *septal deviation* and *turbinate hypertrophy*, by leveraging CFD simulations of airflow in the human upper airways. Our approach is motivated by the work of Schillaci et al. [5], which demonstrated the potential of combining ML and CFD for classifying parametric deformation in simplified 3D shapes of the nasal cavities. However, the simplified geometries used in [5] fail to capture the anatomical complexity and variability of human airways, limiting their applicability in clinical practice. To overcome this limitation, we introduce a novel data augmentation method that generates a realistic and labeled training set of CFD data based on real anatomical surfaces extracted from CT scans.

Data augmentation methods [6] are widely recognized as a standard approach to address data scarcity. These techniques expand the training set by generating additional data, typically through either data warping or oversampling. However, traditional augmentation methods used in image classification [7] cannot be directly applied to CFD data, as they might violate physical laws and boundary conditions. Our method integrates computational geometry tools such as functional correspondence [8, 9], to transfer deformation functions defined by ENT experts to healthy anatomical surfaces, simulating various pathological conditions with different severity levels and locations. This approach allows us to produce a large dataset with minimal expert supervision while preserving anatomical realism and ensuring well-defined pathology labels.

We apply our method to train a binary classifier to identify 2 nasal pathologies: turbinate hypertrophy and septal deviation. The training set is entirely generated with our augmentation method, starting from the CT scans of 7 healthy individuals. Overall, 315 surfaces are produced using deformation functions corresponding to different severity levels and locations of pathologies. We test a classifier exclusively trained on our synthetic samples both on synthetic samples and on a set of 10 CFD simulations from real patients, and we show that the classifier can successfully generalize and correctly identify pathologies.

2 Related Work

The application of machine learning (ML) in fluid mechanics has grown significantly over the past decade, with a surge of interest in integrating ML with Computational Fluid Dynamics (CFD) [1, 10]. Traditionally, ML models are used to approximate or accelerate fluid simulations by modeling complex non-linear

relationships in fluid dynamics, often bypassing traditional numerical solvers. A primary example is the use of physics-informed neural networks to replace the Navier-Stokes equations with ML models, significantly speeding up simulations [4, 11]. Another research direction involves enhancing turbulence models through ML techniques. Ling et al. [12] applied deep learning to improve the Spalart-Allmaras turbulence model [13], while Fukami et al. [3] used convolutional neural networks (CNNs) for super-resolution tasks, reconstructing high-fidelity flow fields from lower-resolution simulations. These works primarily focus on improving the accuracy and efficiency of CFD simulations by predicting traditional fluid dynamics quantities. In contrast, our approach aims to infer high-level labels, such as medical diagnoses, from CFD flow fields, a less explored area in ML-CFD research. The work by Schillaci et al. [5] is one of the few attempts to combine ML with CFD for classification tasks. However, their study was limited to simplified geometries and toy examples, whereas our method targets real-world clinical scenarios by generating realistic, labeled datasets through data augmentation.

Data augmentation in CFD poses unique challenges due to the strict physical laws and boundary conditions governing flow fields. Traditional augmentation methods, such as geometric transformations, often fail to produce physically consistent results. Wu et al. [14] addressed this issue using Generative Adversarial Networks (GANs) to create synthetic flow fields, while Abucide-Armas et al. [15] leveraged the similarity principle of fluid dynamics to maintain physically consistent flow behavior across different scales by matching the Reynolds number. These approaches, however, primarily focus on augmenting the flow field in simplified scenarios, which limits their applicability to complex medical contexts where the effects of anatomical pathologies on internal flows must be accurately captured. Our method differs by applying data augmentation directly to the anatomical geometries rather than the flow fields. By deforming healthy anatomical surfaces according to expert-defined pathology models, we generate a large and diverse training set with clear pathology labels. This strategy ensures that the augmented data remains physically realistic and allows the ML model to learn meaningful associations between anatomical deformations and their impact on fluid dynamics, improving the classification of nasal pathologies in clinical settings.

3 Problem Formulation

We address the problem of training a classifier K capable of identifying nasal pathologies by analyzing flow fields obtained from CFD simulations. The classifier K operates as follows:

$$K : F \rightarrow Y, \quad Y = \{y_j\}_{j=1, \dots, M}, \quad (1)$$

where F represents the space of CFD-simulated flow fields, and Y is a set of predefined pathology labels, with $M = 2$ corresponding to the labels for *turbinate hypertrophy* and *septal deviation*.

Each CFD simulation F_i is computed from a surface $S_i \subset \mathbb{R}^3$ extracted from a CT scan, and consists of a matrix stacking in each column the 3D coordinates and the fluid dynamics quantities of a cell:

$$F_i[:, j] = [x_j, y_j, z_j, p_j, u_j, v_j, w_j]^T \in \mathbb{R}^7, \quad j = 1, \dots, n_i, \quad (2)$$

where (x_j, y_j, z_j) are the spatial coordinates, (u_j, v_j, w_j) are the components of the velocity vector, p_j is the pressure, and n_i is the number of cells of the simulation domain Ω . In our application, n_i can easily range in the order of $n_i \approx 10^7$.

The training set for the model K consists of labeled pairs:

$$\{(F_i, y_i)\}_{i=1, \dots, N}, \quad (3)$$

where $y_i \in Y$ is the ground truth label associated with each flow field F_i . Our goal is to train K to accurately predict y_i based on features extracted from F_i , enabling robust and automatic classification of nasal pathologies.

4 Method

Our goal is to develop a method that can efficiently generate a labeled and realistic training set of annotated CFD flow fields, minimizing the need for expert supervision of numerous anatomies, which is both costly and time-intensive. Our solution is illustrated in Fig. 1.

First, in collaboration with ENT experts, we define a *reference* surface $R \subset \mathbb{R}^3$ representing a healthy nose, along with a set of M *surface deformation functions* $\{(\delta_j, y_j)\}_{j=1, \dots, M}$, as shown in the top left corner of Fig. 1. Each deformation δ_j modifies R to create a variant R_j^* , which is affected by the pathology y_j . Deformation functions are defined by ENT experts, who manually modify the healthy nose R to obtain a surface that is compliant with the nasal cavities of a patient affected by the pathology y_j , as in an inverse *virtual surgery* procedure. We then use R and $\{(\delta_j, y_j)\}_{j=1, \dots, M}$, together with a computational geometry technique, to generate a large training set of annotated anatomies starting from a small set $\{\tilde{S}_i\}_{i=1, \dots, N}$ of N surfaces extracted from the CT scans of *healthy* individuals. In particular, the key idea of our method is that each surface from a healthy patient can be considered a deformed variant of R , thus we adopt functional maps [16], an advanced shape registration technique, to establish a point-to-point mapping between R and each $\tilde{S}_i$. This mapping is used to automatically apply deformation δ_j to S_i (the cleaned version of $\{\tilde{S}_i\}$, manually removed of some spurious geometry), resulting in a new surface $S_{i,j}^*$, which is a variant of S_i affected by the pathology y_j . By repeating this procedure for all the surfaces of healthy patients and all the pathologies, we obtain a set of surfaces $\{S_{i,j}^*\}_{i=1, \dots, N; j=1, \dots, M}$ with their corresponding labels. We use these surfaces to run $N \times M$ CFD simulations $\{F_{i,j}\}_{i=1, \dots, N; j=1, \dots, M}$ which form the training set of our classifier.

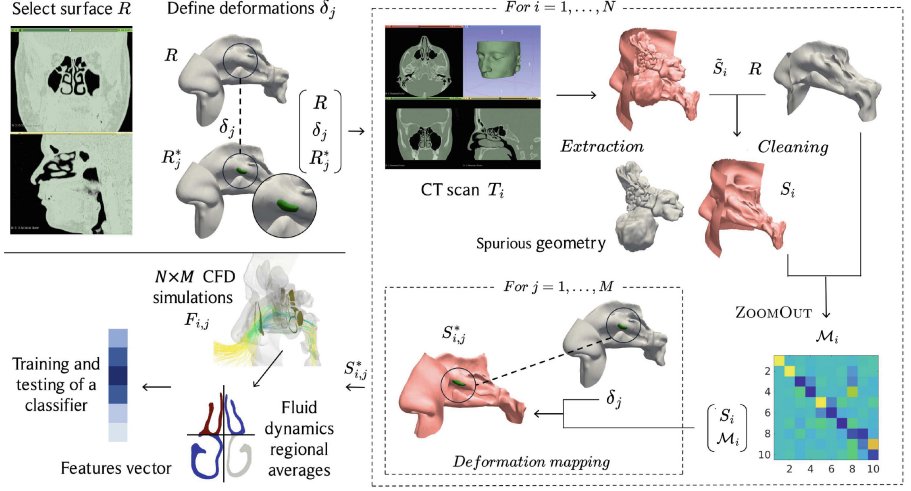


Fig. 1. A schematic representation of our method. The top-left corner illustrates the selection of the reference nose R and the definition of deformation functions $\{\delta_j\}_{j=1,\dots,M}$, which transform R into pathological variants R_j^* . On the right, we show the generation of synthetic pathological surfaces $S_{i,j}^*$. Surfaces $\tilde{S}_i$ are extracted from the N healthy CT scans T_i by applying a threshold. Experts manually remove spurious parts (e.g., paranasal sinuses) from $\tilde{S}_i$, producing the cleaned surface S_i . The bottom-right corner displays the mapping $\mathcal{M}_i$, enabling the transfer of δ_j from R to S_i , resulting in $S_{i,j}^*$. Finally, CFD simulations on $S_{i,j}^*$ are used to train and test the classifier, as shown in the bottom-left corner.

Our method allows us to automatically produce a large set of surfaces $\{S_{i,j}^*\}$, each associated with a specific pathology. These surfaces are then used to perform CFD simulations, generating a wide and labeled training set for ML models. The anatomical variability of the healthy subjects, combined with the flexibility of applying different deformation intensities and combinations, significantly increases the size and diversity of the dataset. The effectiveness of our method is evaluated by training a classifier on the synthetic CFD data and testing it on real patient anatomies.

4.1 Reference Surface Selection and Deformation Functions

A critical component of our augmentation method is the selection of a high-quality *reference surface* $R \subset \mathbb{R}^3$ representing a healthy nasal anatomy. ENT experts from *ASST Santi Paolo e Carlo* in Milan reviewed a CT scan database to identify a well-defined, healthy scan with high resolution. The surface of R , as well as the surfaces $\{\tilde{S}_i\}$, are extracted from CT scans $\{T_i\}$ using a segmentation process that isolates the internal airways [17]. The segmentation process can be

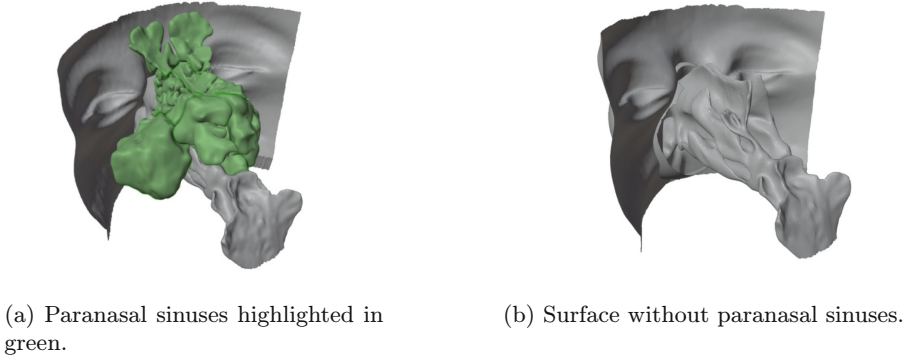


Fig. 2. The reference surface before and after the removal of paranasal sinuses, optimizing it for CFD simulations.

challenging, particularly in small regions where nasal mucus may accumulate and increase the local density, leading to irregularities and disconnections in the surface. These small errors need to be evaluated and corrected by experts. The extracted surfaces are further refined by manually removing unnecessary regions [18], namely the paranasal sinuses, which contribute minimally to airflow dynamics and unnecessarily increase the computational cost of CFD simulations (Fig. 2). Despite these manual interventions being very time-consuming, it is important to stress that this is limited to just 7 healthy patients out of the 315 synthetic samples produced by our augmentation technique. The remaining synthetic pathological samples are automatically generated from these cleaned surfaces, avoiding further segmentation and manual refinement.

To generate synthetic pathological surfaces, we define a set of *surface deformation functions* $\{(\delta_j, y_j)\}_{j=1, \dots, M}$ over the reference surface R . Each deformation function $\delta_j : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is manually designed by ENT surgeons to simulate specific nasal pathologies, such as *septal deviation* and *turbinate hypertrophy*. A *septal deviation* involves bending the nasal septum, potentially obstructing airflow, while *turbinate hypertrophy* results in enlarged tissue along the nasal walls, leading to nasal obstruction. Figure 3 illustrates how these deformations can impact the reference geometry R . The deformation functions map each point (x, y, z) of the reference surface to a displacement vector $(\Delta x, \Delta y, \Delta z)$, allowing us to transform the healthy geometry R into a pathological variant R_j^* . By varying the location and severity of these deformations, we generate multiple pathological surfaces with minimal expert intervention. This approach not only provides a robust and diverse dataset but also ensures that each generated surface has a clear and well-defined pathology label. The ability to combine and control the intensity of these deformations further enhances the diversity of the training set, contributing to better generalization of the classification model.

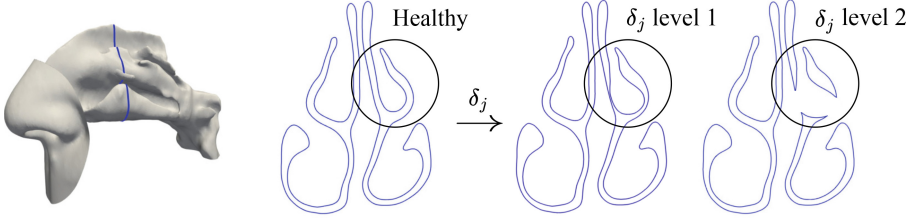


Fig. 3. Left: section of the reference surface used to visualize the pathology. Center: healthy section. Right: simulated turbinate hypertrophy with two levels of severity as defined by deformation functions.

4.2 Mapping Deformations from R to S_i

Our data augmentation method transfers deformation functions $\{\delta_j\}_{j=1,\dots,M}$, defined by ENT surgeons on the reference surface R , to healthy anatomical surfaces S_i of different patients. Each deformation function δ_j simulates a specific pathology y_j , transforming R into a pathological variant R_j^* . Instead of redefining deformations for every surface, we use *functional mapping* to automatically apply δ_j to any healthy cleaned surface S_i , significantly reducing the need for manual intervention. The process involves estimating a point-to-point correspondence $\mathcal{M}_i : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ between R and S_i such that $S_i = \mathcal{M}_i(R)$, that allows us to map each deformation function directly to the target surface. This mapping is achieved using ZOOMOUT [19], which aims to preserve the anatomical consistency of the deformations despite geometric differences between R and S_i . The deformed surfaces $S_{i,j}^*$ are computed as:

$$S_{i,j}^* = S_i + \delta_j(\mathcal{M}_i^{-1}(S_i)), \quad \forall i = 1, \dots, N; \quad j = 1, \dots, M. \quad (4)$$

This approach generates a set of $N \times M$ annotated surfaces for CFD simulations. Additionally, we can control the severity of pathologies by scaling deformations or combining multiple functions as follows, further enhancing the diversity of the training set:

$$S_{i,(j_1,j_2)}^* = S_i + \alpha \delta_{j_1}(\mathcal{M}_i^{-1}(S_i)) + \beta \delta_{j_2}(\mathcal{M}_i^{-1}(S_i)). \quad (5)$$

This flexibility in generating synthetic data contributes to robust model training, enabling accurate classification of nasal pathologies.

4.3 CFD Simulations

We run numerical simulations to obtain the flow fields inside $\tilde{S}_{i,j}$ using OpenFOAM, an open-source CFD toolbox. The simulations employ the *Large Eddy*

Simulation (LES) technique under the assumption of incompressible flow, capturing the complex turbulent structures within the nasal cavities. The computational domain is defined by enclosing the nostrils with a spherical surface. This is a standard practice in CFD to simulate the boundary condition of an ideally infinite external ambient. The simulations replicate a steady-state inspiration at a flow rate of 16 l/min, ensuring consistent physiological conditions. Approximately 12 million mesh cells are used per simulation, leveraging parallel computing with 96 cores on the Galileo supercomputing system. We maintain the same boundary conditions across all synthetic and real samples to ensure consistency and comparability of the results among simulations. The simulations produce detailed flow fields containing spatial coordinates, velocity components, and pressure values, which are subsequently processed for feature extraction.

5 Experiments

To demonstrate the effectiveness of our approach, we train a classifier using CFD flow fields $\{F_{i,j}\}$ computed from the set of surfaces $\{S_{i,j}^*\}$. In light of the study conducted in [5] on simplified nose shapes, we extract six transversal sections (right and left sides) from each surface $S_{i,j}^*$. Regional averages of the velocity $\mathbf{U}$ are then computed to build the feature vector fed to the classifier. The classification performance is assessed in two settings: *i*) we evaluate the classifier on synthetic samples using a Leave One Patient Out Cross Validation (LOPOCV); *ii*) we test the model on CFD data from a set $\{D_i\}_{i=1,\dots,10}$ of real, *pathological*, and *never-seen-before* patients. Each patient D_i is extracted from a CT scan and undergoes the same surface extraction and cleaning (Sect. 4.1), and CFD simulation procedures as healthy samples. We aim to demonstrate that a classifier trained solely on manually deformed data can effectively identify pathologies in real and anatomically diverse cases.

5.1 Training Set Generation

We generate the training set by applying our method to the CT scans of 7 healthy individuals and then perform CFD simulations for each surface $S_{i,j}^*$. These CT scans are provided by *ASST Santi Paolo e Carlo* and have been diagnosed as *healthy* by ENT experts. Before processing, all scans were verified for high quality and resolution to ensure accurate surface extraction and CFD simulations.

For both septal deviation and turbinate hypertrophy, ENT experts manually defined deformation functions $\{(\delta_j, y_j)\}$ at different locations and with two levels of severity, as described in Sect. 4.1. This process yielded six variations of septal deviation and nine of turbinate hypertrophy. Additionally, combining deformations of the same pathology at different locations and severities (as shown in 5) resulted in 12 combinations of septal deviations and 18 combinations of turbinate hypertrophy. The initial set of 7 individuals was thus expanded to a total of 315 pathological samples $S_{i,j}^*$ with clearly defined labels. CFD simulations were then

Table 1. Summary of the datasets used in our experiments, distinguishing between hypertrophies (Hyp) and septal deviations (Dev), as well as the severity level of pathologies.

			Dataset Name			
			$S1$	$S2$	S	D
Samples	Hyp	Lev 1	35	-	189	5
		Lev 2	-	28		
	Dev	Lev 1	21	-	126	5
		Lev 2	-	21		
	Tot	Lev 1	56	-	315	10
		Lev 2	-	49		

performed as outlined in Sect. 4.3. Healthy samples were excluded since the focus of this study is on classifying pathological cases.

To clarify the structure of our dataset and simplify the notation, we denote as $S1$ and $S2$ the subsets of $S_{i,j}^*$ containing surfaces affected by a single pathology of severity level 1 and 2, respectively, and we name S the complete set that extends $S1$ and $S2$ with combined deformations. We also denote D as the set $\{D_i\}_{i=1,\dots,10}$ of real, pathological, and never-seen-before patients. These sets are used to validate the classifier performance on real anatomical data. Table 1 summarizes all datasets used in our experiments, categorizing samples by pathology type and severity level.

5.2 Feature Extraction

Due to its large size, CFD flow fields cannot be directly fed to a NN. Therefore, we perform feature extraction to reduce the dimensionality of the input. Following the approach of [5], we compute *regional averages* of the velocity magnitude $|\mathbf{U}|$ over six transversal sections of each surface $S_{i,j}^*$, including the olfactory region boundaries and four equally spaced sections along the nasal cavity (see Fig. 4).

To account for varying mesh densities and to ensure a consistent representation of flow characteristics, regional averages are computed as area-weighted means:

$$\overline{|\mathbf{U}|}_n = \frac{\sum_i |\mathbf{U}|_i A_i}{\sum_i A_i}, \quad (6)$$

where A_i is the area of cell i , and the index i spans all cells in region $\mathcal{R}_n$. This approach ensures that larger cells, which cover more significant portions of the anatomical region, contribute proportionally to the final feature values.

Regional averages are calculated separately for the left and right halves of each section, therefore, each CFD simulation F_i is reduced to a vector of 12 regional averages, providing a compact and informative input for the NN.

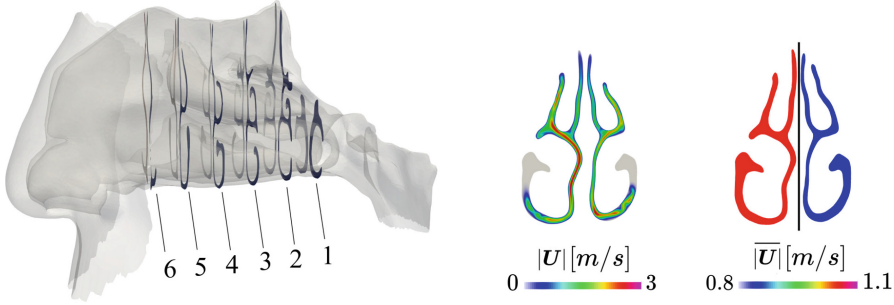


Fig. 4. Sections used for computing feature vectors. The left-hand side shows the positions of the 6 sections we considered. On the right-hand side, the module of the velocity is averaged on the left and right half-sections.

5.3 Model Training

We train a model to classify feature vectors composed of regional averages of fluid dynamics quantities. The classifier K takes as input a vector of 12 features, namely, regional averages of $|\mathbf{U}|$ described in Sect. 5.2.

The classifier is implemented as a multilayer perceptron (MLP) with a standard architecture comprising four hidden layers of 60, 40, 20, and 10 neurons, respectively, totaling 4261 trainable parameters. The output layer consists of a neuron with sigmoid activation where 0 indicates hypertrophy and 1 indicates septal deviation. Other neurons use the ReLU activation function.

Each input feature is standardized to have zero mean and unit variance on the training set. The model is trained using binary cross-entropy as the loss function for up to 1000 epochs, with early stopping based on validation loss. During training, 85% of the samples are used for training and 15% for validation.

5.4 Results

We evaluate our classifier using two experimental settings: (i) Leave One Patient Out Cross-Validation (LOPOCV) on synthetic data generated through our augmentation method, and (ii) testing on a set D of 10 real pathological patients, with expert-confirmed diagnoses of either hypertrophy (5 patients) or septal deviation (5 patients).

Leave One Patient Out Cross Validation (LOPOCV). The LOPOCV approach involves iteratively excluding all synthetic samples derived from a single healthy individual during training and using them as the test set, eventually, we average the inference performance over all the patients. We rely on the LOPOCV to evaluate the model's ability to generalize to unseen synthetic samples produced by our augmentation method. As shown in Fig. 5, the classifier K achieves 94% accuracy on hypertrophies and 84% on septal deviations, resulting in an overall accuracy of 89%. Accurate classification performance of

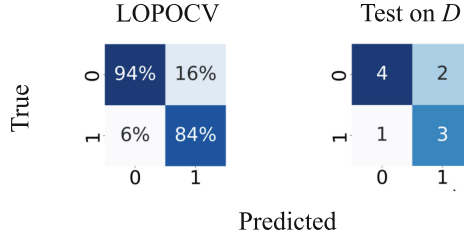


Fig. 5. Confusion matrices for the classifier K . Left: LOPOCV on synthetic data. Right: Test on real patient data. Labels 0 and 1 correspond to hypertrophy and septal deviation, respectively.

Table 2. Precision, Recall and F_1 -score in the LOPOCV for both classes: hypertrophy and septal deviation.

<i>Pathology</i>	<i>Hypertrophy</i>			<i>Septal Deviation</i>		
<i>Metric</i>	Precision	Recall	F_1 -score	Precision	Recall	F_1 -score
<i>Score</i>	0.899	0.941	0.919	0.906	0.841	0.872

the LOPOCV, denotes that the classifier trained on synthetic data successfully learned how deformations δ_j impact CFD data, and that K can properly generalize when the anatomy changes. The confusion matrix indicates that most classification errors occur when predicting septal deviations, suggesting that this pathology may not always produce distinct flow differences. The higher accuracy in identifying hypertrophies (94%) aligns with clinical expectations, as these conditions often cause significant anatomical obstructions and distinct airflow patterns, making them easier to classify. The solid performance of $|U|$ highlights that velocity magnitude is an effective indicator of pathological changes in the nasal airflow.

Table 2 reports the mean values of *precision*, *recall*, and F_1 -score for both hypertrophy and septal deviation computed during the LOPOCV procedure. Given a binary classification setting, and denoting by:

TP = true positives, FP = false positives, FN = false negatives,

these metrics are computed as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \quad F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

The results indicate that hypertrophies (class 0) are associated with a higher recall and F_1 -score, suggesting that the model is generally able to detect most true cases of this condition, likely due to the more distinct effects they produce on the flow field. Conversely, septal deviations (class 1) exhibit a slightly higher

precision, meaning that when the model predicts a septal deviation, it is more likely to be correct. However, the lower recall for class 1 implies that some true cases are missed, possibly due to the more subtle and variable effects of septal deviation on the airflow field. Overall, these results suggest that while the model is more sensitive to hypertrophies, it achieves higher classification precision for septal deviations.

Test on Real Patients. The test on real patients is conducted using a NN having the same architecture as in the previous experiment, but trained on all the synthetic data in S and tested only on data from a set D of real pathological individuals. When applied to D , the classifier trained exclusively on synthetic data correctly classifies 7 out of 10 cases, showing good generalization capabilities. The confusion matrix (Fig. 5) reveals consistent results with the LOPOCV, showing that hypertrophies are generally easier to detect, with 4 out of 5 correct classifications, while septal deviations exhibit more variability in prediction accuracy. This behavior underscores the clinical complexity of septal deviations, where mild cases might not induce significant changes in the airflow, leading to potential misclassification. The performance on real pathological patients shows the real potential of our augmentation method: given a large anatomical variability, and considering that we apply our augmentation method to a set of only 7 healthy patients, our classifier performs well on real patients, underscoring the method’s ability to generalize to real cases even when trained exclusively on synthetic data.

Overall, these results validate our augmentation method’s effectiveness in generating synthetic samples that not only enhance model performance in synthetic scenarios but also ensure reliable generalization to real clinical cases. This demonstrates the potential of combining CFD with ML to support ENT specialists in diagnosing nasal pathologies more accurately.

6 Discussion and Conclusions

In previous work, Schillaci et al. [5] demonstrated the ability of neural networks to classify CFD flow fields, but their tests were restricted to simplified scenarios due to the lack of training data. Our work extends this approach to a more realistic clinical setting, focusing on the classification of nasal pathologies in human upper airways using only CFD data. We present a data augmentation method that generates a large and diverse set of labeled CFD data by applying expert-defined deformations to a limited number of healthy CT scans. This approach significantly reduces the need for extensive manual labeling while maintaining anatomical realism, a critical advantage in medical applications where annotated datasets are often scarce.

Our experiments focused on two key evaluations: *i*) Leave One Patient Out Cross-Validation (LOPOCV) on synthetic data, and *ii*) testing on a set of 10 real patients with confirmed diagnoses of either hypertrophy or septal deviation. Our classifier trained exclusively on synthetic data achieved strong performance,

with up to 89% accuracy in LOPOCV and correctly identifying pathologies in 7/10 real pathological patient cases. The classification performance demonstrates that a model trained on augmented synthetic data can effectively generalize to real clinical scenarios, despite never encountering real pathological samples during training. The consistency of the results across both synthetic and real datasets confirms the effectiveness of our augmentation method in generating high-quality training data. This approach holds the potential to support ENT specialists in diagnosing nasal pathologies through non-invasive CFD analysis, enhancing diagnostic accuracy and treatment planning.

It is important to stress that creating the real patient dataset D required significantly more effort compared to the synthetic dataset of 315 samples, as the former involved the manual interventions described in Sect. 4.1. This highlights the efficiency of our proposed augmentation method, which automates the generation of synthetic pathological samples, drastically reducing manual intervention while preserving data quality and label reliability.

In conclusion, our method establishes a practical and effective connection between CFD data and real clinical scenarios, offering a practical solution for training NNs when labeled data is limited. Future work will focus on expanding the dataset by including a broader range of pathologies to improve the generalization of the model. We also plan to design new features that better capture the fluid dynamic behaviors associated with specific clinical conditions, enhancing the model's predictive power. Additionally, we will explore advanced interpretability techniques to provide deeper insights into which features contribute most significantly to the classification of pathologies, ultimately supporting clinical decision-making with more transparent and explainable models.

Acknowledgments. The Authors are grateful to A. Bulfamante for the time spent as ENT expert. Computing time has been provided by the Italian CINECA HPC Center in Bologna under the ISCRA-C grant AICFD-23. This publication is part of the project PNRR-NGEU which has received funding from MUR-DM 351/2022. MQ acknowledges support from the PRIN 2022 project OpenNOSE, cod. 2022BYA5AF, CUP D53D2300343006.

References

1. Brunton, S., Noack, B., Koumoutsakos, P.: Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52**, 477–508 (2020)
2. Duraisamy, K., Iaccarino, G., Xiao, H.: Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* **51**(1), 357–377 (2019)
3. Fukami, K., Fukagata, K., Taira, K.: Assessment of supervised machine learning methods for fluid flows. *Theoret. Comput. Fluid Dyn.* **34**(4), 497–519 (2020). <https://doi.org/10.1007/s00162-020-00518-y>
4. Raissi, M., Yazdani, A., Karniadakis, G.E.: Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. *Science* **367**(6481), 1026–1030 (2020)

5. Schillaci, A., Quadrio, M., Pipolo, C., Restelli, M. and Boracchi, G.: Inferring functional properties from fluid dynamics features. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 4091–4098 (2021)
6. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. *J. Big Data* **6**(1), 60 (2019)
7. Yang, S., Xiao, W., Zhang, M., Guo, S., Zhao, J., Shen, F.: A survey, Image data augmentation for deep learning (2022)
8. Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., Guibas, L.: Functional maps: a flexible representation of maps between shapes. *ACM Trans. Graph.* **31**(4), 30,1–30,11 (2012)
9. Ovsjanikov, M.: Computing and processing correspondences with functional maps. In: *ACM SIGGRAPH 2017 Courses*, pp. 1–62 (2017)
10. Brenner, M.P., Eldredge, J.D., Freund, J.B.: Perspective on machine learning for advancing fluid mechanics. *Phys. Rev. Fluids* **4**(10), 100501 (2019)
11. Pin, W., et al.: Navier–stokes generative adversarial network: a physics-informed deep learning model for fluid flow generation. *Neural Comput. Appl.* **34**(14), 11539–11552 (2022)
12. Ling, J., Kurzwaski, A., Templeton, J.: Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *J. Fluid Mech.* **807**, 155–166 (2016)
13. Spalart, P., Allmaras, S.: A one-equation turbulence model for aerodynamic flows. In: *30th Aerospace Sciences Meeting and Exhibit*, pp. 439–445. American Institute of Aeronautics and Astronautics, Reno, NV (1992)
14. Haizhou, W., Liu, X., An, W., Lyu, H.: A generative deep learning framework for airfoil flow field prediction with sparse data. *Chin. J. Aeronaut.* **35**(1), 470–484 (2022)
15. Abucide-Armas, A., Portal-Porras, K., Fernandez-Gamiz, U., Zulueta, E., Teso-Fz-Betoño, A.: A data augmentation-based technique for deep learning applied to CFD simulations. *Mathematics* **9**(16), 1843 (2021)
16. Melzi, S., Ren, J., Rodola, E., Sharma, A., Wonka, P. and Ovsjanikov, M.: Zoomout: spectral upsampling for efficient shape correspondence. *ACM Trans. Graph.* **38**(6), 155,1–155,14 (2019)
17. Quadrio, M., et al.: Effects of CT resolution and radiodensity threshold on the CFD evaluation of nasal airflow. *Med. Biol. Eng. Comput.* **54**(2), 411–419 (2016)
18. Jin, H.H., Fan, J.R., Zeng, M.J., Cen, K.F.: Large eddy simulation of inhaled particle deposition within the human upper respiratory tract. *J. Aerosol Sci.* **38**, 257–268 (2007)
19. Cosmo, L., Rodola, E., Masci, J., Torsello, A., Bronstein, M.M.: Matching deformable objects in clutter. In: *Fourth International Conference on 3D Vision (3DV)*, pp. 1–10 (2016)



Machine Learning Analysis of Dissolved Oxygen at the “G. Giannouli” Hydrological Station Greece

Nichat Kiourt^(✉), Lazaros Iliadis^{ID}, Christos Akratos, and Antonis Papaleonidas^{ID}

Department of Civil Engineering, Democritus University of Thrace, Xanthi, Greece
{nkiourt, liliadis, cakratos, papaleon}@civil.duth.gr

Abstract. Dissolved oxygen (DO) is the amount of O₂ that is present in water and it is found in molecular state. Water bodies receive oxygen from the atmosphere and from aquatic plants. Running water, such as that of a swift moving stream, dissolves more oxygen than the still water of a pond or lake. Dissolved O₂ levels change according to the temperature, air pressure and salinity. It is consumed not only by the reduction of substances such as *sulphides*, *nitrites* and *iron ions*, but also by the respiration of microorganisms and the oxidative decomposition of organic substances by aerobic microorganisms. The aim of this research is the analysis and classification of DO at the Hydrological Station *G. Giannouli* (HSG) of the *Pinios* river in central Greece, with the development of robust *Machine Learning* models. The results clearly demonstrate that these models can successfully assign five labels based on the thresholds established by the Norwegian Water Research Institute namely: “Poor”, “Deficient”, “Moderate”, “Good” and “High”.

Keyword: Machine Learning · Classification · Dissolved Oxygen in water

1 Introduction

1.1 Problem Definition - Aim of This Research

The concentration of Dissolved Oxygen in water is an undeniable indicator of the viability of river ecosystems. The development of most forms of life (animal, plant, fungi, protists and bacteria) presupposes its presence. Therefore, DO is the of crucial importance, and it is the expression of its self-purification capacity. In aquatic ecosystems, DO is much less than its equivalent in the atmosphere [1]. The main reason must be considered the difficulty of its dissolution in the aquatic environment. The two important sources that supply the aquatic environment with O₂ are the atmosphere and the plant organisms that photosynthesize [1]. The concentration of Dissolved Oxygen in water depends on the following [2]:

- a) The higher the temperature of water, the lower its oxygen content.
- b) The amount of organic matter decomposed in the water.
- c) The presence or absence of plants (microscopic and macroscopic), which can carry out photosynthesis.

d) The degree of light penetration, which depends on the depth.

Fish and other animals that live in water need O_2 to live. Natural lakes and rivers generally cannot have large percentages of dissolved oxygen. If its levels are too low, fish and other animals may die. The chances of fish deaths are even higher if the fish live in a polluted or over productive system. Things are even more difficult in polluted systems where we have intensive industrial development, intensive use of fertilizers and discharges of human waste that can contaminate water more quickly, causing the fish to suffocate and die [1].

The standards set by the Norwegian Institute for Water Research (NIVA) have been used to estimate the adequacy of Dissolved Oxygen measured by automatic stations in mg/L [3, 4]. Based on the DO concentration, five quality classes have been defined namely: *High, Good, Moderate, Deficient, Poor* which is measured. The classification boundaries are described on Table 1 below [5].

Table 1. NIVA Classification in 5 quality classes, based on the concentration of DO in water, on a daily basis

NIVA classes of DO in water					
	HIGH	GOOD	MODERATE	DEFICIENT	POOR
DO (mg/L)	>9	[6.4, 9]	[4–6.4)	[2–4)	<2

Dissolved Oxygen (DO) measurement in liquids is carried out with special instruments. The dissolved Oxygen concentration of “*Healthy*” water should fall in the interval [6.4 – 9] measured in mg/L. The cooler the water the higher the DO, while respective the concentration in warmer polluted waters is lower.

The aim of this research is to offer robust *Machine Learning* (ML) models [6] to rationally classify and evaluate DO concentrations in river *Pinios* and its Hydrological station of “*Gianouli*” [20] located in the prefecture of *Larisa* in Central Greece. The determination of the respective classes is done based on the Norwegian Institute for Water Research (NIVA) standards. This will enable not only the local authorities but also the public to know the quality of the available water resources in the specific area of study.

1.2 Literature Review

This section presents a comprehensive literature review on the application of Machine Learning in modeling of water dissolved oxygen. Balahaha Fadi Ziyad Sami et al., 2023, proposed a reliable machine learning model to predict DO in Fei-Tsui Reservoir, Taiwan, for better water quality monitoring [7]. Salim Heddami et al., 2020, used long short-term memory (LSTM) deep learning applications to predict dissolved oxygen in rivers [8].

Eda G. et al., 2019, employed ML algorithms to predict DO concentrations based on the relatively easy measurements of water temperature, pH and conductivity in the rivers [9]. Olyae E., et al., 2017, used three different AI methods namely: *Multi-Layer*

perceptron (MLP) *Radial Based function* (RBF) and *Support Vector Machines* (SVM) optimized by *Linear Genetic Programming* – LGP, for the case of *Delaware River* at Trenton, USA [10].

Csabragi A. et al., 2017, used ANN to forecast DO content in the Hungarian section of the river Danube [11]. Shuguang Li et al. 2023, used SVM regressors, MLP, random forest (RF), and gradient boosting (GB) to estimate daily DO concentration in the Tualatin River [12].

Sani Isah Abba et al., 2020, employed four different Machine Learning (ML) ensemble techniques for modeling dissolved oxygen concentration models, by employing Long Short-Term Memory Neural Networks (LSTM), Extreme Learning Machines (ELM), Hammerstein-Weiner (HW) and General Regression Neural Network (GRNN) for modeling DO concentration of Kinta river in Malaysia [13].

Abul Abrar Masrur Ahmed et al., 2023, used Multivariate Adaptive Regression Spline (MARS) to forecast DO concentration using a hybrid model coupled with Maximum Overlap Discrete Wavelet Transformation (MODWT) as a feature decomposition approach, for the case of Surma River using a set of water quality hydro-meteorological variables. The proposed hybrid model was compared with numerous machine learning methods, namely Bayesian Ridge Regression (BRR), k-Nearest Neighbors (k-NN), Kernel Ridge Regression (KRR), Random Forest (RF), and SVR [14]. Moon et al., 2022, utilized RF, GB, and AdaBoost models. The results demonstrated that the AdaBoost model showed better predictions of DO compared to the other methods [15].

To the best of our knowledge, this paper presents one of the first Machine Learning oriented research efforts on DO classification, for the *Giannouli Hydrological Station* in river *Pinios* of central Greece.

2 Data – Are of Research

Data was selected from the Pinios river which is the largest river in Thessaly with a total length exceeding 200 km [21]. It originates from the mountain range of Pindos, crosses the Thessalian plain, meets Larissa and passing the foothills of Mount Olympus, turns east and finally flows into the Aegean Sea, creating its Delta near the town of “*Stomio*”. It is one of the most important wetlands in Greece, as it has riparian forests, rich fauna and extensive sand dunes in its Delta. It belongs to the NATURA network and has been designated as a Landscape of Outstanding Natural Beauty.

Pinios river is very important for the entire prefecture of Larissa. In addition to supplying water to settlements, it is used to irrigate thousands of acres with various crops’ cultivation. The hydrological station G. Giannouli, is found on the bridge of the Pinios river that connects the town of G. Giannouli with the city of Larissa.

Data used in this research was found and obtained by the Open *Hydrosystem Information Network* (OpenHi.net) website [16]. Publicly available data is stored on a website, with the aim of an open information infrastructure for the country’s surface water, with free access to relevant hydrological, environmental and geographic data. The Hydrosystems Open Information Network is a component of the national research infrastructure “*Hellenic Integrated Monitoring, Forecasting and Technology System of Seas and Surface Waters*” (HEIMO_FOTES³) [17]. The infrastructure is developed by a broad

research group, involving 14 agencies (Universities and Research Centers) led by the Hellenic Marine Research Center (HEMARC) [18] and focusing on monitoring both surface waters and the marine environment.

The dataset considered, contains 29,269 daily measurements from August 2019 to the end of December 2021 (collected on a daily basis, half an hour apart) at the Hydrological Station G. Giannouli [19], of the *Pinios* river in *Larissa*, Greece. The data consists of the following characteristics:

Independent features:

- Electrical water conductivity ($\mu\text{S}/\text{cm}$),
- Water Temperature (Temp ($^{\circ}\text{C}$)),
- Water level (cm),
- Potential of Hydrogen (pH (-)).

Depended feature:

- Dissolved Oxygen, DO (mg/L).

It is worth noting that regarding the area of study, the DO concentration was less than 2 mg/L (<2) in 1,875 cases (Poor Quality). Additionally, in 1,960 data records DO values ranged between (2–4] mg/L (Deficient Quality), whereas it ranged between (4–6.4] mg/L (Moderate Quality) in 5,586 records. Moreover, in 13,882 records, DO was found in the interval (6,4–9] mg/L (Good Quality) and finally, in 5,966 records DO concentration was higher than (>9) mg/L (High Quality) .

In the first two cases, the water was deemed *unsuitable*, due to the reduced concentration of DO, which offered a hostile environment for fish and other forms of life. On the contrary, in the last three cases, due to the increased values of dissolved oxygen in the water, it was considered satisfactory, in favor of the viability of most forms of life in the study area (Fig. 1).



Fig. 1. Giannouli hydrological station, in Larisa prefecture [19].

3 Dissolved Oxygen Machine Learning Modeling

There are several modes of Learning namely Supervised Machine Learning (SML) Unsupervised Machine Learning (UML), Semi Supervised ML (SSML) and Reinforcement ML (RML) [22]. This research is based on SML.

3.1 Data Preprocessing

The first stage of this research was Data pre-processing (D_PreP) which was carried out using the MS-Excel software. D_PreP was necessary, as the original data suffers from various kinds of inconsistencies, which are related to the existence of conflicting values, to the existence of the coding scheme, to the fields' naming, to the employed measurement units, as well as to the existence of missing values, i.e. randomly fluctuating data without meaningful content. These problematic records are called “dirty-data” and the process of dealing with them is called “data cleansing”. Data pre-processing includes *Data cleaning and Class Development* [24].

The MS-Excel environment was used to calculate the maximum value, minimum value and standard deviation of dissolved oxygen. The standard deviation was used to correct the normal distribution in the entire data set. In a normal distribution, almost all data falls within three standard deviations of the mean.

Development of classes: This modeling effort examines the five classes of dissolved oxygen in water as defined by the *Norwegian Water Research Institute*, namely “Poor”, “Deficient”, “Moderate”, “Good” and “High”.

- The “Poor” class contains all values less than 2 mg/L
- The “Deficient” class contains all values between [2–4) mg/L
- The “Moderate” class contains all values between [4–6.4) mg/L
- The “Good” class contains all values between [6.4–9) mg/L
- The “High” class contains all values higher than 9 mg/L

3.2 Development of Machine Learning Models

After pre-processing, data was imported into MATLAB and an attempt was made to classify the dissolved oxygen, with the help of various Machine Learning algorithms. The following classifiers were used: *Decision Trees, Naïve Bayes, Support Vector Machines, k-Nearest Neighbors* and *Ensembles of Classifiers*.

Decision trees are among the most well-known classification or categorization models. The decision tree is a Graph with the classic tree structure where we distinguish: an initial node (the root), the internal nodes, the leaves (the external nodes), directed edges, control conditions located on the directed edges. Each internal node corresponds to a feature or characteristic that is used to further divide the tree. Edges leaving the root or each internal node are assigned to a check condition based on the separator [26].

The Naive Bayes algorithm, a probabilistic classifier based on Bayes' theorem, has many applications in various fields due to its simplicity and efficiency. It is widely used in classification tasks, where it efficiently handles large datasets and high-dimensional feature spaces. The *Naive Bayes* algorithm is a classification technique under the assumption that all features predicting the target value are independent of each other. It calculates

the probability of each category and then selects the one with the highest probability [27].

Support Vector Machines use supervised learning models to solve complex classification, regression, and outlier detection problems by performing optimal data transformations that determine boundaries between data points, based on predefined classes, labels, or outputs [28].

The *k-Nearest Neighbors* (k-NN) classifier is one of the simplest classifiers. When given an item to classify, it finds the item in the training data that is most similar to the new item and produces its label. The k-NN algorithm is a non-parametric, supervised learning classifier that uses proximity to make classifications or predictions about the clustering of a single data point. It is one of the most popular and simplest classification and regression classifiers used in machine learning today [29].

Ensemble classifiers train a set of classifiers for the same data set and finally combine the results of each. The most common methods of constructing set classifiers are as follows [24]:

1. *Bagging*: The training set is resampled to create a set of smaller datasets, and then a classifier is trained with each one. When an unknown sample needs to be classified, it is first categorized by all the classifiers and the final decision is made by voting among them [24].
2. *Boosting*: Here, a set of classifiers is trained again with the difference that each sample carries a weight (according to its “importance” in the set it belongs to. This weight is renewed iteratively, each sample that is incorrectly categorized by the classifier increases its weight for the next classifier, while otherwise it decreases it. Thus, each classifier is required to give more weight to samples that have so far eluded correct categorization. The final decision is made by voting, possibly weighted by some evaluation metric [24].

3.3 Models' Assessment Indices

An evaluation metric in the field of Artificial Intelligence (AI) and Machine Learning (ML) is a quantitative measure utilized to assess the performance of a machine learning model. These metrics are crucial as they provide a standardized approach for evaluating the effectiveness, efficiency, and accuracy of the model in making predictions or classifications based on input data [23]. Evaluation metrics can be broadly categorized into different types based on the nature of the machine learning task, such as classification, regression, clustering, and ranking. Each task type has specific metrics that are most appropriate for evaluating the performance of models designed to solve that specific task [23]. In the present study, a classification model was conducted. Classification tasks involve predicting discrete labels or categories for given inputs. Common evaluation metrics for classification models include:

Accuracy: The ratio of correctly predicted instances to the total number of instances. While it is a simple and intuitive metric, it may not be suitable for imbalanced datasets.

Precision: The ratio of true positive predictions to the total number of predicted positives. Precision is crucial when the cost of false positives is high.

Recall (or Sensitivity): The ratio of true positive predictions to the total number of actual positives. Recall is important when the cost of false negatives is significant.

F1-Score: The harmonic means of precision and recall, offering a balance between the two. It is particularly useful when the dataset is imbalanced.

ROC Curve: A metric that assesses the trade-off between the true positive rate and the false positive rate at various threshold values. The ROC curve provides a visual representation of the model's performance across all thresholds. It represents the probability that a randomly selected positive example is ranked higher than a randomly selected negative example.

4 Results

The following Tables 2–6 present the performance of each classifier, based on the model evaluation indices used in this classification research.

Table 2. Classification results for Decision Trees via the one versus all approach

Decision Trees Classifier					
	Precision	Recall	Specificity	Accuracy	F1-Score
Class 1	1.000	1.000	1.000	1.000	1.000
Class 2	1.000	1.000	1.000	1.000	1.000
Class 3	1.000	1.000	1.000	1.000	1.000
Class 4	1.000	1.000	1.000	1.000	1.000
Class 5	1.000	1.000	1.000	1.000	1.000

Table 3. Classification results for Naïve Bayes via the one versus all approach

Naïve Bayes Classifiers					
	Precision	Recall	Specificity	Accuracy	F1-Score
Class 1	0.980	0.990	0.998	0.998	0.990
Class 2	0.970	0.980	0.998	0.997	0.980
Class 3	0.970	0.990	0.993	0.992	0.980
Class 4	0.990	0.980	0.990	0.984	0.980
Class 5	0.970	0.980	0.991	0.991	0.980

Table 4. Classification results for Support Vector Machines via the one versus all approach

Support Vector Machines Classifiers					
	Precision	Recall	Specificity	Accuracy	F1-Score
Class 1	0.990	1.000	0.999	0.997	1.000
Class 2	1.000	0.990	0.998	0.997	1.000
Class 3	1.000	1.000	1.000	1.000	1.000
Class 4	1.000	1.000	1.000	1.000	1.000
Class 5	1.000	1.000	1.000	1.000	1.000

Table 5. Classification results for K-Nearest Neighbor via the one versus all approach

K-Nearest Neighbor Classifiers					
	Precision	Recall	Specificity	Accuracy	F1-Score
Class 1	0.970	0.970	0.998	0.996	0.970
Class 2	0.930	0.940	0.995	0.991	0.940
Class 3	0.960	0.960	0.991	0.985	0.960
Class 4	0.980	0.980	0.993	0.980	0.980
Class 5	0.970	0.980	0.994	0.990	0.980

Table 6. Classification results for Ensembles Classifiers via the one versus all approach

Ensembles Classifiers					
	Precision	Recall	Specificity	Accuracy	F1-Score
Class 1	1.000	1.000	1.000	1.000	1.000
Class 2	1.000	1.000	1.000	1.000	1.000
Class 3	1.000	1.000	1.000	1.000	1.000
Class 4	1.000	1.000	1.000	1.000	1.000
Class 5	1.000	1.000	1.000	1.000	1.000

Though all algorithms had an excellent performance, the Bagged trees of Ensembles Classifiers reached an accuracy equal to 100%. Figure 2 presents the Confusion Matrix for the *Ensembles Classifiers Bagged trees* algorithm.

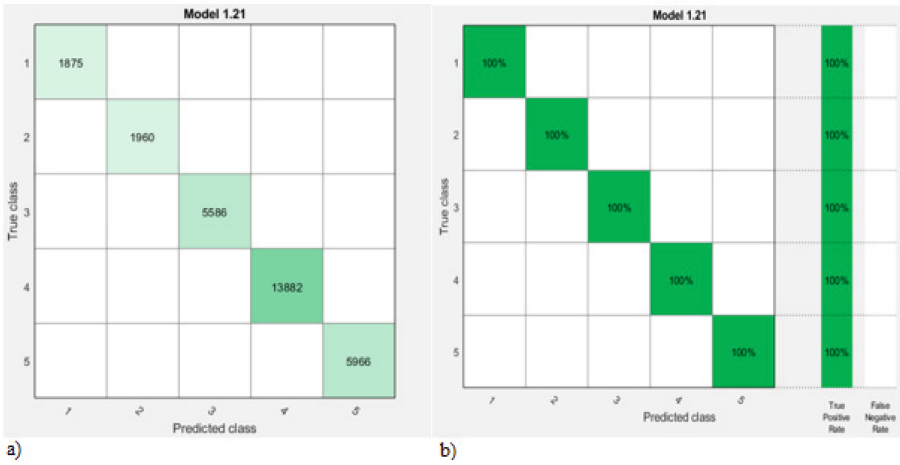


Fig. 2. Modeling with the *Bagged trees of Ensembles Classifiers* algorithm. Confusion Matrices with the classification accuracy of the data in 5 classes (a. considering the number of records b. considering the percentages of the records).

In Fig. 2, we notice that in all five classes the algorithm was able to correctly identify all cases. Also, in the Fig. 3 below, we observe that the ROC curve has an AUC equal to 1.

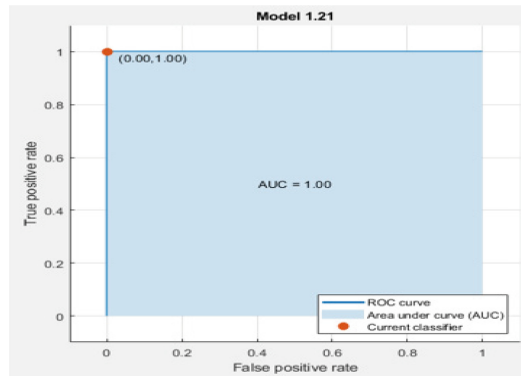


Fig. 3. ROC Curve obtained for the Bagged trees of Ensembles Classifiers algorithm.

The following Fig. 4 presents the performance for all algorithms.

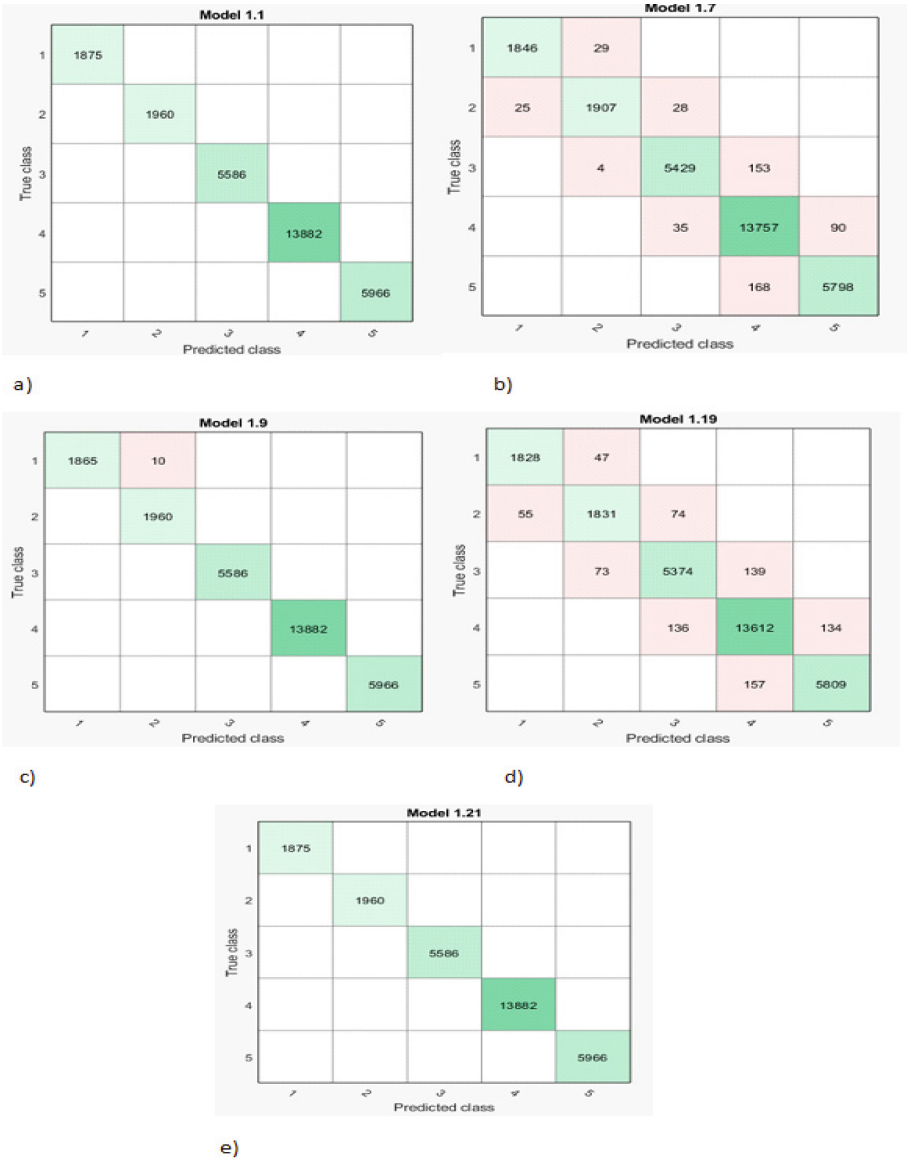


Fig. 4. Modeling with a) Decision Trees Classifier, b) Naïve Bayes Classifiers, c) Support Vector Machines Classifiers, d) K-Nearest Neighbor Classifiers, e) Bagged trees of Ensembles Classifiers algorithm.

Figure 5 below presents the ROC Curves for all algorithms used.

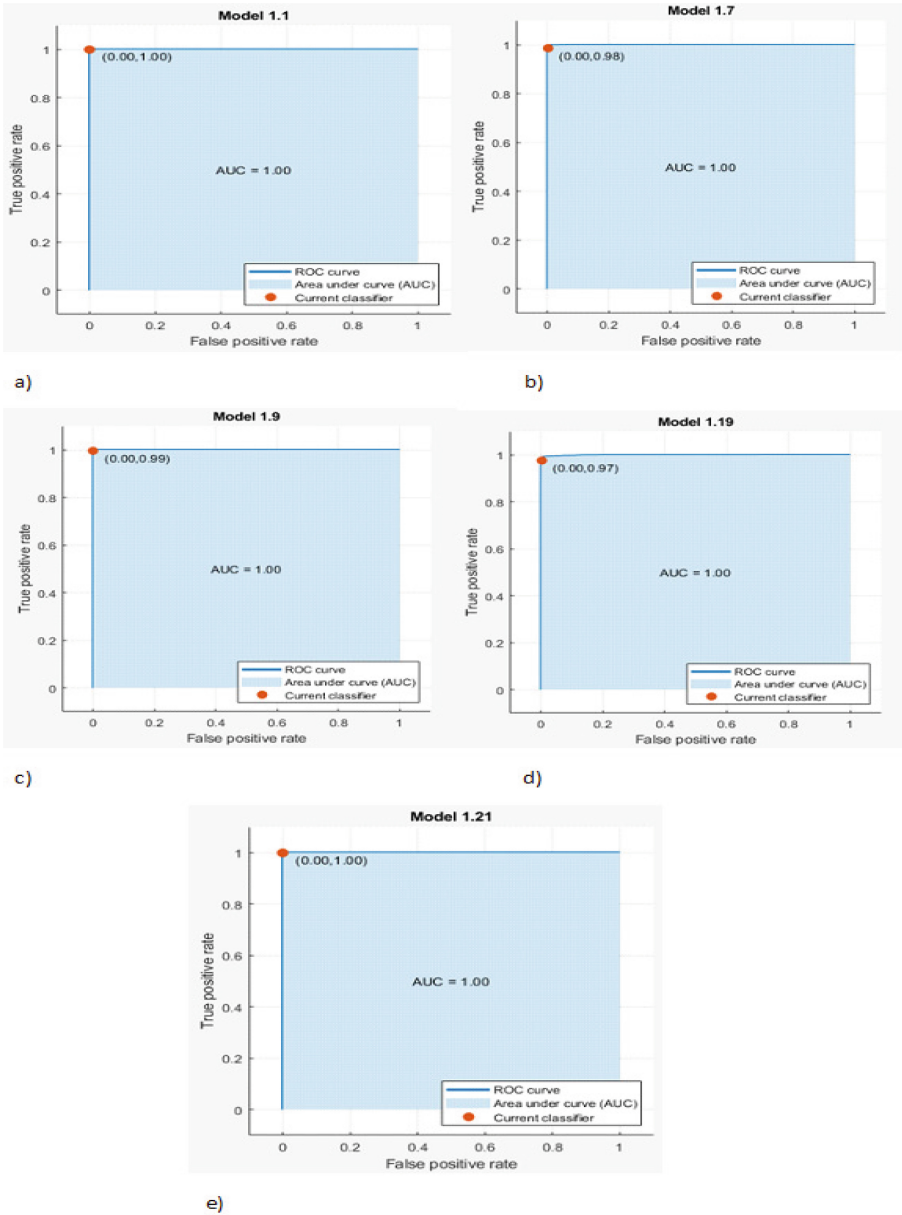


Fig. 5. ROC Curve a) Decision Trees Classifier, b) Naïve Bayes Classifiers, c) Support Vector Machines Classifiers, d) K-Nearest Neighbor Classifiers, e) Bagged trees of Ensembles Classifiers algorithm.

5 Discussion-Conclusions

In the present research, an attempt was made to classify the water dissolved oxygen of the Hydrological station “*Giannoulis*” of the *Pinios* river in Larissa, with the help of various machine learning algorithms. The performance of each algorithm was examined using the indices used in multiclass classification (e.g., Precision, Recall, Specificity, Accuracy, the confusion matrix and the ROC curve). Our extracted results showed a slightly better performance in the *Bagged trees* method of the *Ensembles Classifiers* classification algorithms compared to the other classification algorithms, with an accuracy close to 100%.

Dissolved Oxygen is constantly consumed during respiration by plants and animals in water, so knowing its levels is very important. According to the processed 3,835 data records in the Hydrological station of G. Giannouli of the *Pinios* river, (under an average temperature of 21 °C) we found that water Dissolved Oxygen levels were quite low in 13.10% of the data records, whereas in 19.09% of the records (5,586 measurements) they were below average. It should be considered that the higher the temperature of the water, the lower the dissolved oxygen content. The 32.19% percent of records with Low oxygen in a period of 28 months (without considering the temperature problem) must be considered and assessed by the competent authorities, because it rings the bell about the low respiratory function of living organisms such as fish, which will suffocate and die, resulting in the destruction of the entire aquatic ecosystem. Finally, another important finding is that any changes that occur in the dissolved oxygen of the water are not due to natural causes, but to various sources of pollution caused by human activities. More responsible actions are required by the local authorities.

References

1. Tousis L.: Graduate Dissertation. The importance of oxygen dissolved in water for breeding of aquatic animal organisms. Variations in duration of the day (2007). Retrieved from <https://core.ac.uk/download/pdf/132800514.pdf>.
2. National Technical University of Athens. Concentration of dissolved oxygen (DO). (2019)
3. Cardoso A.C., Duchemin J., Magoarou P., Premazzi G. Criteria for the identification of fresh-water subject to eutrophication. Their implem. Nitrates Urban Waste Water Direct. EUR 19810 EN, EU – JRC, 87 (2001)
4. Norwegian Institute for Water Research (NIVA). Research institute for water and the environment (2001). Retrieved from <https://www.niva.no/en>.
5. Hellenic Centre for Marine Research (HCMR). Dissolved Oxygen Quality Index. (2019)
6. Georgouli, A.: Machine Learning (chapter 4). Artificial Intelligence. Kallipos (open academic publishing) (2015). <https://repository.kallipos.gr/handle/11419/3382>,
7. Balahaha F. Z. S., S. D. L., Ali N. A., Ming F. C., Muhammad A. M., Asep S., Balahaha H. Z. S., Jee K. W., Ahmed H. B., Ahmed El-S.: Machine learning algorithm as a sustainable tool for dissolved oxygen prediction: a case study of Feitsui Reservoir, Taiwan (2022). Retrieved from <https://www.nature.com/articles/s41598-022-06969-z>

8. Salim H., Sungwon K., Ali D. M., Mohammad Z.K., Anurag M., Ahmed E., Ozgur K.: Predicting dissolved oxygen concentration in river using new advanced machines learning: Long-short term memory (LSTM) deep learning. Retrieved from <https://www.sciencedirect.com/science/article/abs/pii/B9780323898614000312> (2021)
9. Eda G., Mehmet Y., Erdal K.: Machine Learning Application of Dissolved Oxygen Prediction in River Water Quality (2019). Retrieved from https://www.researchgate.net/publication/332237144_Machine_Learning_Application_of_Dissolved_Oxygen_Prediction_in_River_Water_Quality
10. Olyae E., Abyaneh H. Z., Mehr A. D.: A comparative analysis among computational intelligence techniques for dissolved oxygen prediction in Delaware River Geosci. Front. (2017). Retrieved from <https://www.sciencedirect.com/science/article/pii/S1674987116300469>,
11. Csabragi A., Molnar S., Tanos P., Kovacs J.: Application of artificial neural networks to the forecasting of dissolved oxygen content in Hungarian section of the river Danube. (2017) Retrieved from https://www.researchgate.net/publication/311897715_Application_of_artificial_neural_networks_to_the_forecasting_of_dissolved_oxygen_content_in_the_Hungarian_section_of_the_river_Danube
12. Shuguang L., Sultan N. Q., Shahab S. B., Rasoul A., Hao-Ting P., Saeid M.: Explainable machine learning models for estimating daily dissolved oxygen concentration of the Tualatin River (2023). Retrieved from <https://www.tandfonline.com/doi/full/https://doi.org/10.1080/19942060.2024.2304094>
13. Abba, S.I., et al.: Hybrid machine learning ensemble techniques for modeling dissolved oxygen concentration. Retrieved from (2020). <https://doi.org/10.1109/ACCESS.2020.3017743>
14. Ahmed, A. A. M., Jui, S. J. J., Chowdhury, M. A. I., Ahmed, O., Sutradha, A.: The development of dissolved oxygen forecast model using hybrid machine learning algorithm with hydro-meteorological variables (2023).
15. Moon, J., Lee, J., Lee, S., Yun, H.: Urban river dissolved oxygen prediction model using machine learning. Retrieved from (2022). <https://doi.org/10.3390/w14121899>
16. OpenHi.net.: Open Hydrosystem Information Network. Retrieved from <https://openhi.net/>
17. HIMIOFoTS.: Hellenic Integrated Marine Inland Water Observing, Forecasting and offshore Technology System. Retrieved from <https://www.himiofots.gr/el>
18. HCMR.: Hellenic Centre for Marine Research. Retrieved from <https://www.hcmr.gr/el/>
19. HIMIOFoTS.: Hellenic Integrated Marine Inland Water Observing, Forecasting and offshore Technology System. Retrieved from <https://system.openhi.net/stations/1545/>
20. Psilopoulos Kostas.: Town of Giannouli. (2011) Retrieved from http://users.sch.gr/ntinos_psilop/index.php?option=com_content&view=article&id=52:giannouli-xtes-simera&catid=41:giannouli&Itemid=109
21. Greek Travel Pages.: Pineios river of Thessalia (2003). Retrieved from <http://www.gtp.gr/PineiosRiver>
22. Geeksforgeeks.: Types of Machine Learning (2023). Retrieved from <https://www.geeksforgeeks.org/types-of-machine-learning/>
23. EITC/AI/GCML.: Google Cloud Machine Learning. (2024) Retrieved from <https://el.eitca.org/artificial-intelligence/eitc-ai-gcml-google-cloud-machine-learning/first-steps-in-machine-learning/the-7-steps-of-machine-learning/what-is-an-evaluation-metric/>
24. Tarun A.: Advanced Ensemble Classifiers (2019). Retrieved from <https://towardsdatascience.com/advanced-ensemble-classifiers-8d7372e74e40>
25. Kallipos.: Data Preprocessing (2015). Retrieved from <https://repository.kallipos.gr/handle/11419/1234>

26. Geeksforgeeks.: Decision Tree in Machine Learning (2024). Retrieved from <https://www.geeksforgeeks.org/decision-tree-introduction-example/>
27. Sunil R.: Naive Bayes Classifier Explained: Applications and Practical Problems. Retrieved from <https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/>,
28. Vijay K.: What Is a Support Vector Machine? Working, Types, and Examples. (2024). Retrieved from: <https://www.spiceworks.com/tech/big-data/articles/what-is-support-vector-machine/>
29. Leif E. P.: K-nearest neighbor. Retrieved from http://scholarpedia.org/article/K-nearest_neighbor (2009)



Privacy-Preserving and Personalized AI Modules for E-Commerce Platforms

Arantzazu Florez Tapia^(✉), Izar Azpiroz, Ane M. Florez-Tapia,
Amaia Gil, Elena Zotova, Roger Solsona, Igor García Olaizola,
and Marco Quartulli

Fundación Vicomtech, Basque Research and Technology Alliance (BRTA), Mikeletegi
57, 20009 Donostia-San Sebastián, Spain
aflorez@vicomtech.org

Abstract. The evolution of e-commerce platforms can be characterized by the low digital maturity level of represented stores, their limited data amounts and the fragmentation of existent data. In order to overcome these challenges, we present an e-commerce extension library that collects customer sales predictions and clusterization functionalities grouped into two distinct modules: the Federated Learning and personalized module. As a proof of concept, a prototype is presented in this paper that has been developed in the post-Covid context to boost the digitalization capability of small short commerce.

Keywords: Federated Learning · Customer Segmentation & Suggestion · Market Basket Analysis · Sentiment Analysis · Sales Prediction

1 Introduction

The COVID-19 pandemic has significantly transformed global socio-economic dynamics. The resulting acceleration of digitization has impacted various sectors within the Industry 4.0 framework [1], particularly in how data is transferred and processed. In this context, ensuring data privacy [2] has become increasingly urgent due to the widespread use of sensitive information. Federated Learning (FL) has emerged as a promising technological response to current challenges, offering a way to preserve data privacy, governance, and security [3, 4], while still enabling the extraction of knowledge from distributed data. It is a machine learning paradigm that utilizes a decentralized training architecture [4], enabling collaborative model training across multiple devices or platforms—such as smartphones located in diverse geographical locations [5] or multi-platform environments [3]—without sharing raw data. The growing interest in FL is evident in ongoing initiatives aimed at federating systems that manage sensitive user data on a daily basis [3].

Among the many potential applications of FL, e-commerce stands out as a particularly relevant domain [6]. The surge in online shopping—accelerated by

the COVID-19 pandemic—has rapidly transformed the retail landscape [7]. This shift has driven significant changes in consumer behavior, as shopping habits have moved from physical to digital channels. Consequently, many traditional retailers have been compelled to adopt new technologies and expand their digital presence. The economic effects of the pandemic have extended beyond public health, influencing employment dynamics worldwide [7]. In this environment, large retailers and digital platforms have continued to dominate online sales, while many small and medium-sized businesses struggle to remain operational [8]. This highlights an urgent need for businesses of all sizes to adapt to the evolving digital economy. According to Euromonitor [9], 17% of all goods were purchased online in 2021, nearly double the figure from 2016. In 2020 alone, global online purchases increased by 24%, while store-based sales declined by 7%.

The digital transformation of the commerce sector, when aligned with appropriate machine learning tools, enables the extraction of valuable insights from user experiences [10]. Natural language processing (NLP) techniques, such as text classification [11], clustering [12], and text summarization [13], facilitate the analysis of consumer attitudes expressed through comments, reviews, and social media posts. By applying data mining techniques to customer actions, transaction records, and social sentiment, it becomes possible to characterize customer profiles and predict behavioral patterns [14]. These insights can be used, for instance, to enhance brand loyalty by identifying customers at risk of churning.

In addition, sales-focused methodologies such as product selling analysis can provide both daily and cumulative sales forecasts [15], offering valuable input for inventory planning and logistics optimization. Another important technique is Market Basket Analysis [16], which leverages association metrics to deliver actionable insights for product placement and supply chain decisions in retail environments.

Although small businesses are in crisis, new opportunities arise from all situations, and now they can benefit from digitalization. According to recent studies [17], in the following years, e-commerce and data-analysis capabilities are among the most important ones. The benefits of federation can be extended to the e-commerce domain, addressing the challenges of training models on distributed data sources while ensuring data privacy [18]. By enabling collaborative model training without exchanging raw data, FL allows e-commerce platforms to leverage the collective intelligence of multiple sources [19]. It enhances data privacy [20], reduces the risk of breaches, and facilitates collaboration [21], resulting in improved recommendation systems [22], personalized marketing, and customer churn prediction. Additionally, FL offers real-time learning, network efficiency, lower latency, and scalability benefits, empowering e-commerce platforms to unlock valuable insights and drive growth while adhering to data privacy regulations and utilizing the heterogeneity of data sources available.

This work aims to present the data privacy and interpretability e-commerce library description as well as its integration into an e-commerce prototype. The

initial stage collects the technicalities of the tailored e-commerce functionalities, which have been structured into two principal modules: the federated and personalized modules. The second stage describes the considered use case for validation purposes. The prototype dashboard visualizes the comparison and benefits of created functionalities. To this end, an e-commerce Brazilian public data set has been adapted.

2 Methodology and Specifications

Recent studies compare the barriers to digital transformation between small and medium-sized and large enterprises [23], concluding that setting up new positions to manage digital transformation poses a more intense barrier for small and medium enterprises. Distinct hindrances have been identified as barriers to digital transformation: missing skills, technical, organizational and external barriers. These obstacles include a generally low level of digital maturity, where a clear gap exists between the available technology and the actual adoption or interest observed among small businesses. The post-pandemic period has also witnessed a noticeable slowdown in digital transformation efforts within this segment.

Another significant challenge is the absence of standards and data fragmentation. Many small retailers are hesitant to join larger e-commerce platforms or to share their data due to concerns about privacy, competitiveness, or a preference for maintaining control over their own online presence. As a result, these businesses often rely solely on their own limited datasets. This scarcity of data restricts their ability to take full advantage of machine learning (ML) techniques, which typically require large volumes of high-quality data to produce accurate and reliable outcomes.

In regions where minority languages are prevalent, such as the Basque Country, the lack of linguistic standardization further complicates the extraction of shared trends and meaningful insights. Federated Learning (FL) offers a promising technological approach to these challenges. By enabling secure, privacy-preserving training across decentralized datasets, FL allows small businesses with similar characteristics to collaboratively benefit from a larger pool of data without compromising sensitive information (e.g., transaction records, customer data). This approach addresses both the issue of limited local data and the broader fragmentation in the sector.

In this paper, we present a set of ML-based functionalities designed to enhance retailers' capabilities by identifying common user behavior patterns and improving the analysis of numerical and linguistic data. These functionalities are organized into two main modules: a FL module and a customized analytics module, as illustrated in Fig. 1.

Federated Learning Module

Sales Prediction: Enables retailers to predict demand and optimize inventory management, thereby minimizing the risk of stock breakage and supporting proactive investment planning.

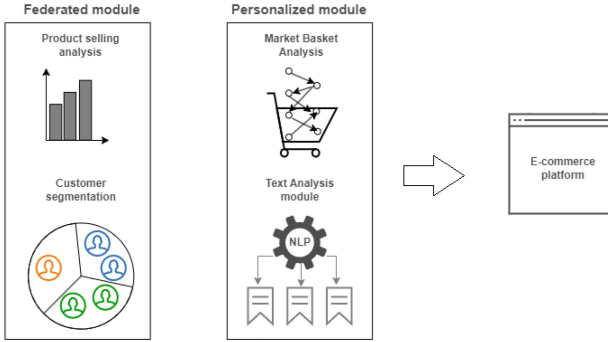


Fig. 1. Division of functionalities between federated and personalized modules

Customer Segmentation: Classification of customer profiles in order to improve the targeting of marketing strategies and tailor offerings to distinct used groups.

Customized Module

Suggestion Module: Helps to identify common patterns in customer purchases to recommend complementary products.

Text Analysis Tools: to obtain information from comments and reveal hidden tendencies in them.

2.1 Federated Prediction Module

In what follows, we describe the training workflow adopted in our horizontal FL architecture. The main goal is to train the global and federated ML model of the central service, making advance of the local training of independent devices. For this purpose, first, all devices included in the federated ecosystem must have the same initial model, initial model provided by the central server generally. Subsequently, the local models located on devices gradually are trained using their local information. The devices are then allowed to transfer the model fine-tuned parameters, not the training data, from the local devices back to the central server. This process happens across all the devices participant in the federated ecosystem. The centralized server updates its central ML model from the aggregated model weights, which is far better than the previously deployed version. Finally, the device's local models are updated to the more innovative model, which can be retrained using their data.

Product Selling Analysis. To prevent stock shortages and improve investment forecasting, two product sales prediction models have been developed. The first model estimates daily sales, providing short-term insights for operational decision-making. The second model forecasts cumulative sales over a longer period, offering a broader view of sales trends and aiding in strategic planning.

1. **Daily sales prediction:** Several machine learning prediction approaches can be applied to estimate daily sales. In this study, a decision tree-based algorithm was chosen due to its interpretability and its effectiveness when working with structured dataset. Decision trees are non-parametric, supervised learning models and are commonly used for both classification and regression tasks. The model was implemented across nine predefined product categories. Trained on historical daily sales data, it is capable of forecasting sales up to two months in advance, providing valuable insights for short-terms inventory control and operational planning.
2. **Cumulative sales prediction:** Retailers are also interested in the cumulative sales prediction, in order to better schedule the stock needed. In addition to daily forecasts, retailers benefit from cumulative sales predictions to more effectively plan stock levels and anticipate long-term demand. This is addressed through the implementation of a Customer Lifetime Value (CLV) [15,24]. CLV quantifies the expected net profit attributed to the future relationship with a customer, based on their purchasing behavior over time. It provides a probabilistic framework for forecasting individual transactions and estimating the total purchase volume across the customer base, particularly in non-contractual settings.
Research indicates that effective customer retention strategies—guided by CLV insights—can increase profit margins by 25% to 95%. Since maintaining long-term relationships with all customer types (especially unprofitable ones) is not always desirable, the primary goal is to identify and retain high-value customers. Integrating the CLV model into the platform supports more informed decisions in marketing, stock management, and overall e-commerce productivity.

Customer Characterization. Customers can be characterized based on several aspects. In this work, two clustering approaches were implemented to capture different dimensions of customer behavior. First, we characterized the monetary aspect, i.e., how much is their final payment value, which prices they are willing to pay, and how much they spend on shipping. The second clustering method builds on the same technique but shifts the focus to product-related behaviors. Specifically, it identifies customer groups associated with products that have a high likelihood of being purchased. This dual perspective supports both stock planning and more effective ad targeting, allowing retailers to tailor their strategies based on the profile of customers they aim to attract.

1. **Customer segmentation:** Customers are segmented using a clustering approach that considers multiple variables, such as product price, shipping costs, total transaction value, and payment method. By grouping customers with similar purchasing behaviors, valuable insights can be extracted regarding spending patterns and economic profiles. For this purpose, the K-Means clustering algorithm is employed. A predefined number of clusters, k , is established, each represented by a centroid—either a real or conceptual point that denotes the center of a cluster. The algorithm assigns each data point to

the nearest centroid by minimizing the within-cluster sum of squares, thus forming groups of customers with similar characteristics.

2. **Customer behavioral prediction:** This module performs sales predictions based on the results of the prior customer segmentation phase. By integrating specialized Customer Lifetime Value (CLV) libraries, the system is able to forecast purchasing trends and assess customer loyalty across different customer profiles. This enables more targeted marketing strategies and improved long-term planning based on expected customer behavior.

2.2 Personalized Modules

The Suggestion and the Text Analysis modules are not federated. In these cases, it is crucial that the training processes rely exclusively on data from individual retailers. Incorporating data from external sources could compromise the specificity and personalization of the insights generated. By isolating the training to local datasets, the resulting recommendations and text interpretations remain fully tailored to each retailer's unique context and customer base.

Suggestion Module: Market Basket Analysis: Apart from segmenting the customers, it is interesting to clusterize the purchased items. For this purpose, Market Basket Analysis (MBA) is applied [16]. This data mining technique enables retailers to identify patterns in customer purchase behavior by analyzing large datasets, such as transaction histories, to uncover product groupings and items frequently bought together. The insights derived from MBA are particularly useful for generating personalized product recommendations and for anticipating demand, thereby supporting inventory planning and enhancing the overall shopping experience.

Text Analysis Module: Text Analysis Tool is implemented with novel natural language processing techniques, which allows automatically analysing and making predictions based on textual data from reviews, comments, product descriptions and other textual data. The implemented tools are language-agnostic, enabling the processing of multilingual data, including minority languages such as Basque. This feature is particularly important for ensuring the applicability of the developed functionalities in local platforms operating in multilingual contexts where Basque is an official language. Furthermore, the underlying technologies are open-source and can be deployed on the client's infrastructure at relatively low cost, facilitating integration into existing systems without significant investment.

- 1 **Text classification:** In the current prototype, the text classification module can assign a sentiment score to a user's review, from 1 to 5, where 1 is negative and 5 is positive. The prediction model is trained on the Brazilian e-commerce public dataset, the closest example of data for our tool's needs. The model is based on a fine-tuned Transformer architecture [11], which enables the

development of high-performing deep learning models even with relatively small datasets. This is particularly relevant for low-resource languages such as Basque, where manually annotated corpora are limited or unavailable. The sentiment classification model is adaptable and can be fine-tuned to accommodate specific domains and label categories required by the end-user, thereby increasing its applicability across different e-commerce contexts.

- 2 **Text clustering:** The text clustering module aims to detect hidden patterns in the unstructured textual data and categorize them. It is implemented with BERTopic framework¹ which has k-means and HDBSCAN [25] algorithms applied to BERT embeddings (multidimensional vectors). We use a multilingual Transformer language model LabSE [12], pretrained to detect similarities in multilingual texts. The main advantage of this model is the capability of encoding texts in any language, including Basque. The module is adjustable with the following parameters: the number of clusters and the number of keywords to represent each cluster, and it can be displayed by month to detect the changes in the users' behavior and tendencies.
- 3 **Text summarization:** Text summarization module resumes a long text into a short one to facilitate the reading and extracting the most significant information. The algorithm PageRank [13] is used for an extractive summarization system. The ranking algorithm evaluates each sentence in the document and extracts the most important ones, representing the whole text best, omitting sentences with no valuable information. The user can select the number of sentences in the summary.
- 4 **Text statistics:** This module extracts quantitative features from textual data, including word frequencies, collocation frequencies, and other statistical measures. These metrics provide a foundational understanding of the text's structure and usage patterns, serving as a basis for further linguistic or semantic analysis.
- 5 **Automatic translation:** Automatic translation module is an additional functionality to overcome language limitations, augment the data in a specific language, and facilitate understanding. Open-source system Marian NMT [26] was used to translate the examples from Portuguese to English.

3 Use Case: Local E-Commerce Prototype

The COVID-19 pandemic crisis exposed critical challenges faced by small retailers, particularly their low levels of digital maturity and their reluctance to out-source or share private data. Despite the availability of numerous e-commerce platforms, many retailers continue to struggle with limited digital adoption and fragmented data practices. Enhancing e-commerce platforms with intelligent functionalities can offer substantial benefits. The integration of machine learning techniques enables features such as personalized customer recommendations, internal process optimization, and improved understanding of customer behavior

¹ <https://maartengr.github.io/BERTopic/index.html>.

through data-driven insights. Moreover, by leveraging FL, it becomes possible to capture broader trends across the retail sector without compromising data privacy. This decentralized approach allows machine learning models to benefit from collective knowledge across multiple retailers while ensuring that sensitive data remains local.

As a proof of concept, the socio-economic context of the Basque Country (northern Spain) has been used to illustrate a plausible application scenario. In this hypothetical scenario, each region is assumed to have a central node that gathers insights from surrounding establishments, enabling them to benefit from aggregated knowledge across all three regions without directly sharing raw data. This federated architecture ensures privacy while promoting collaboration. Furthermore, the same structure could be applied at a more granular level—allowing individual stores to gain visibility into broader sectoral trends without exposing their own data. The functionalities integrated into the current prototype correspond to the two main modules described in the previous section and summarized in Fig. 1: sales prediction, customer segmentation, suggestion module, and text analysis tools.

To demonstrate the potential of these tools, we used an existing open-source dataset from a Brazilian e-commerce platform [27] as the basis for creating a synthetic dataset. This involved adapting the original geographic distribution and product categorization to reflect the Basque Country context. Specifically, the georeferenced positions were randomly reassigned to locations within the three provinces that comprise the Basque Country, as illustrated in Fig. 2. To simulate a federated scenario among the three Basque territories, the dataset was proportionally divided based on the relative size of each province. The resulting synthetic dataset contains approximately 110,000 product sale transactions across nine product categories: health & beauty, menage, clothes, home appliances, sports, tools, kids, grocery, and transport. The dataset includes transaction timestamps spanning the years 2016 to 2018, along with multiple features that provide a multidimensional view of the sales process: order status, pricing, payment and freight performance, customer location, product attributes, and user-generated reviews.

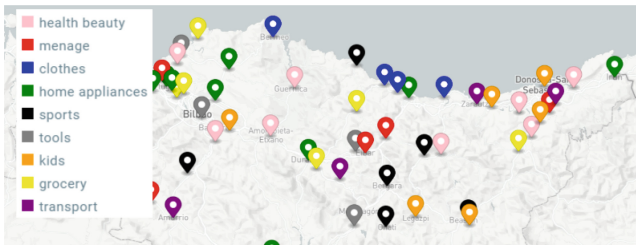


Fig. 2. Basque Country synthetic e-commerce data geographic distribution and categorization

For the implementation of this work, we employed federated-learning-lib, an open-source library developed by IBM [28]. This Python-based framework is designed for FL in enterprise environments and allows for the integration of advanced features. It is framework-agnostic—compatible with various machine learning libraries—and supports multiple training topologies (e.g., centralized aggregator) and communication protocols. The library accommodates a wide range of learning paradigms, including supervised, unsupervised, and reinforcement learning, and supports both classical machine learning models and deep neural networks. In this study, we extended the IBM FL library to implement the Product Sales Prediction and Customer Segmentation services, described in Sects. 2.1 and 2.1, respectively. Figure 3 illustrates the two functionalities available in the Product Sales Prediction module: daily sales prediction and cumulative sales forecasting. The daily sales prediction functionality is implemented using a Decision Tree Regressor model, as shown in Fig. 3a, which also compares the predicted sales values with historical and real sales data from the grocery category. The cumulative sales prediction, on the other hand, is based on the Beta-Geometric/Beta-Binomial model [24], illustrated in Fig. 3b, where predictions are likewise compared with actual values in the grocery sector. Given the two-year historical range and the size of the dataset, both models are capable of generating reliable sales forecasts for up to two months.

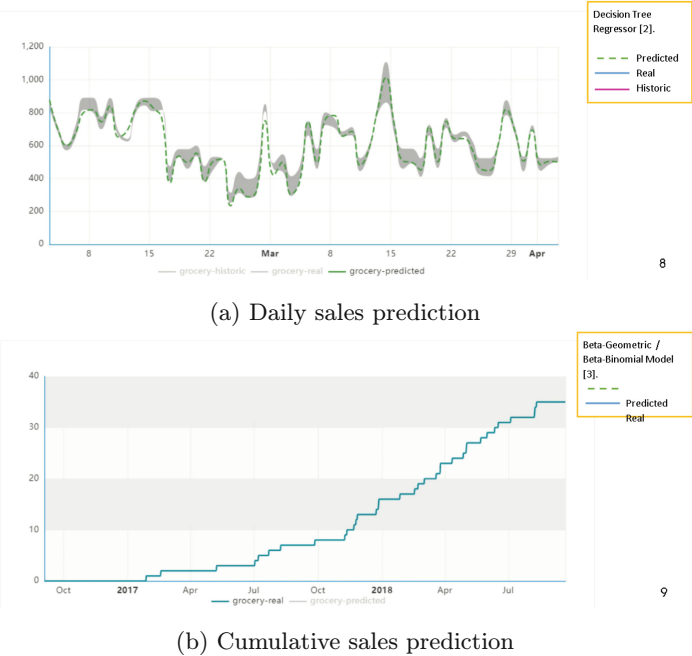


Fig. 3. Visualization of product selling prediction results

The dashboard associated with the Customer Segmentation functionalities is presented in Fig. 4. In the example shown in Fig. 4a, customers are segmented based on monetary attributes such as shipping cost, total payment amount, and product price. These clusters were generated using the K-Means algorithm. On the other hand, Fig. 4b depicts the clusterized prediction of the highest selling likelihood. Again applying the K-Means algorithm, customers are categorized according to their shopping behavior into the following profiles: evidence-seeking, high spenders, risk-averse, and persuadable customers.

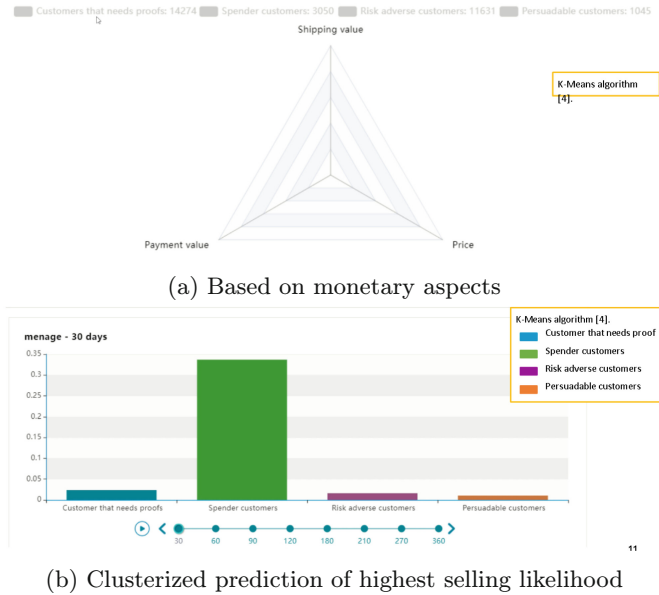


Fig. 4. Customer segmentation

Another component of the prototype is the Suggestion Module, which is based on product association metrics derived from Market Basket Analysis, as illustrated in Fig. 5. This visualization highlights the frequency with which products from different categories are purchased together. In the graph, node colors correspond to product categories, while the size of each node and the thickness of the connecting edges are proportional to the strength of the association metrics. This representation facilitates the identification of cross-category purchase patterns, which can be used to generate data-driven product recommendations.

The final set of figures (Figs. 6 and 7) illustrates the functionalities associated with the Text Analysis Tool module. Figure 6a displays the clustering of user comments written in Portuguese, grouped into five distinct topics. The visualization is generated using a dimensionality reduction technique that enables the spatial distribution of documents for a selected month. Each cluster is defined

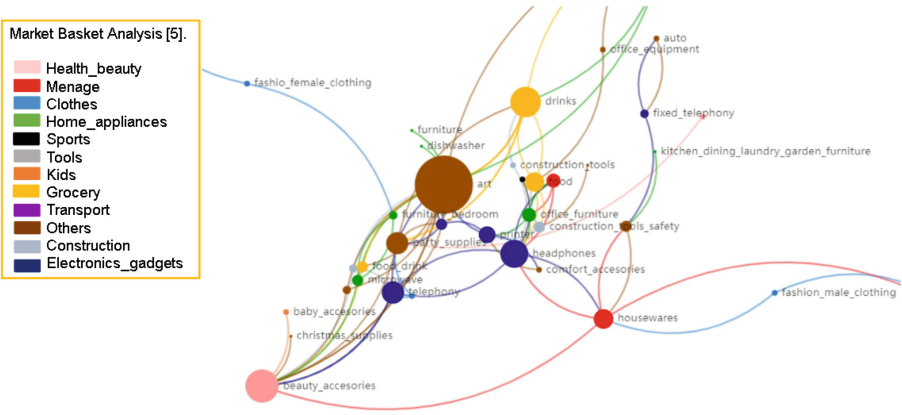
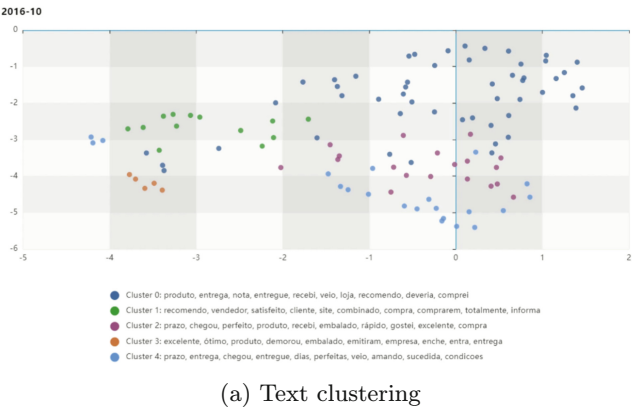
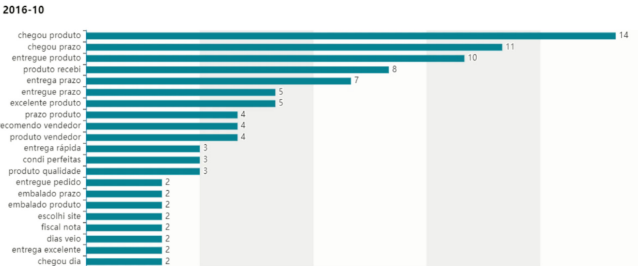


Fig. 5. Market Basket Analysis



(a) Text clustering



(b) Bigram frequency

Fig. 6. Text analysis tools

by its most representative keywords, and the relative size of each group is also visible. Performing monthly clustering allows for the detection of evolving user concerns and emerging trends over time.

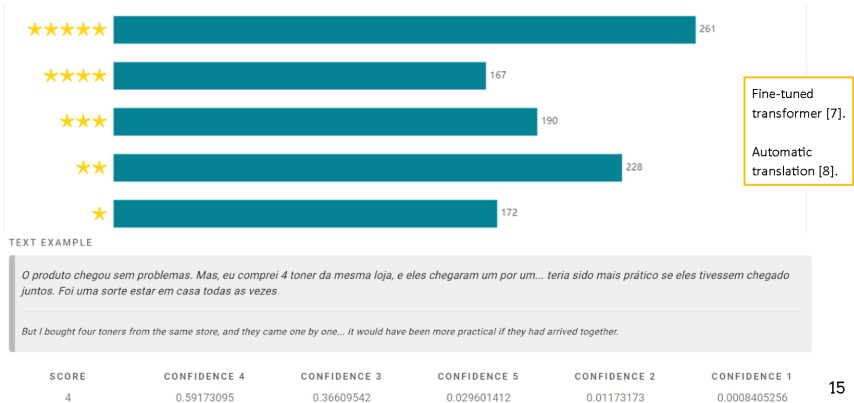


Fig. 7. Text analysis tools: Sentiment analysis

Figure 6b presents an example of quantitative text analysis, showing the frequency distribution of bigrams associated with the most frequently purchased products.

Lastly, Fig. 7 demonstrates the sentiment analysis functionality, which employs a fine-tuned Transformer model along with automatic translation. This component assigns a sentiment label to each user review, ranging from 1 (negative) to 5 (positive), along with a confidence score indicating the model's certainty. The automatic translation module supports the analysis of multilingual content, ensuring consistent interpretation even when reviews are written in foreign languages. The resulting visualizations help retailers assess product quality, customer satisfaction, and shifting sentiment patterns.

4 Conclusions

The main objective of this work was to develop a technological solution to support small retailers in improving their digital visibility a need that became particularly urgent during the COVID-19 pandemic. Several key barriers were identified: the requirement to fully digitize inventory, limitations in storage infrastructure, insufficient financial and human resources to provide an online service comparable to physical retail, and concerns that digital platforms might replace traditional labor. These challenges have contributed to a persistent lack of digital maturity among small businesses.

The nature of small shops does not allow them to meet the requirements of many giant platforms, as they do not have facilities for sufficiently large storage. In addition, the fear of sharing sensitive data with competitors has influenced small retailers not to participate, nor to engage with them. The lack of a unified data framework complicates information interoperability and exacerbates privacy concerns, ultimately restricting access to sufficient and reliable datasets.

In response to these challenges, a prototype system has been developed that offers a privacy-preserving service designed to enable data-driven decision-making without compromising sensitive commercial information. This system comprises two distinct modules, each targeting a specific analytical strategy: On the one hand, to gain knowledge from general consumption trends ensuring the security of non-sharing private data, and on the other hand, provide advance tools to explore in detail inner store trends, based on past purchasing historic. These modules collectively incorporate a suite of AI-powered tools: *Product selling analysis*, *Customer characterization*, *Suggestion module* and *Text Analysis module*.

Finally, this prototype enhanced operational efficiency, decision-making processes, and personalized services while ensuring data privacy and regulatory compliance, benefits that are extensible to other digitalized sectors such as industrial platforms and healthcare systems.

Acknowledgments. This work was supported by Vicomtech Foundation, part of Basque Research and Technology Alliance.

Data Availability. Brazilian E-Commerce Public Dataset by Olist is available at <https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce>.

Disclosure of Interests. The authors have no competing interests to declare.

References

1. Almeida, F., Santos, J.D., Monteiro, J.A.: The challenges and opportunities in the digitalization of companies in a post-COVID-19 world. *IEEE Eng. Manage. Rev.* **48**(3), 97–103 (2020)
2. Yang, Q., Liu, Y., Chen, T., Tong, Y.: Federated machine learning: concept and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **10**(2), 1–19 (2019)
3. Li, L., Fan, Y., Tse, M., Lin, K.-Y.: A review of applications in federated learning. *Comput. Ind. Eng.* **149**, 106854 (2020)
4. Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., Gao, Y.: A survey on federated learning. *Knowl.-Based Syst.* **216**, 106775 (2021)
5. Li, Z., Sharma, V., Mohanty, S.P.: Preserving data privacy via federated learning: challenges and solutions. *IEEE Consum. Electron. Mag.* **9**(3), 8–16 (2020)
6. Mahdiani, M.R.: Enhancing recommendation systems with federated learning for privacy-preserving e-commerce platforms, *Available at SSRN 5129722*
7. Roggeveen, A.L., Sethuraman, R.: How the COVID-19 pandemic may change the world of retailing. *J. Retail.* **96**(2), 169 (2020)
8. Lashgari, Y.S., Shahab, S.: The impact of the COVID-19 pandemic on retail in city centres. *Sustainability* **14**(18), 11463 (2022)
9. Evans, M.: Five e-commerce trends that will change retail in 2021. *Forbes, Business, Retail* (2021)
10. Iqbal, M.: Machine learning applications in e-commerce. *Org. Bus. Manag.* **65** (2022)

11. Zhang, J., Chang, W.-C., Yu, H.-F., Dhillon, I.: Fast multi-resolution transformer fine-tuning for extreme multi-label text classification. *Adv. Neural. Inf. Process. Syst.* **34**, 7267–7280 (2021)
12. Feng, F., Yang, Y., Cer, D., Arivazhagan, N., Wang, W.: Language-agnostic BERT sentence embedding, *arXiv preprint arXiv:2007.01852* (2020)
13. Berkhin, P.: A survey on pagerank computing. *Internet Math.* **2**(1), 73–120 (2005)
14. Cirqueira, D., Hofer, M., Nedbal, D., Helfert, M., Bezbradica, M.: Customer purchase behavior prediction in e-commerce: a conceptual framework and research agenda. In: *International Workshop on New Frontiers in Mining Complex Patterns*, pp. 119–136, Springer (2019)
15. Gupta, S., et al.: Modeling customer lifetime value. *J. Serv. Res.* **9**(2), 139–155 (2006)
16. Blattberg, R.C., Kim, B.-D., Neslin, S.A.: Why database marketing?. In: *Database Marketing*, pp. 13–46. Springer (2008)
17. Jablonska, J.: How European Shoppers Will Buy Groceries in the Next Normal. McKinsey & Co (2020)
18. Wang, H., Xie, F., Duan, Q., Li, J., et al.: Federated learning for supply chain demand forecasting. *Math. Probl. Eng.* **2022** (2022)
19. Li, J., Cui, T., Yang, K., Yuan, R., He, L., Li, M.: Demand forecasting of e-commerce enterprises based on horizontal federated learning from the perspective of sustainable development. *Sustainability* **13**(23), 13050 (2021)
20. Wang, Q., Yin, H., Chen, T., Yu, J., Zhou, A., Zhang, X.: Fast-adapting and privacy-preserving federated recommender system. *VLDB J.*, 1–20 (2021). <https://doi.org/10.1007/s00778-021-00700-6>
21. Tan, B., Liu, B., Zheng, V., Yang, Q.: A federated recommender system for online services. In: *Proceedings of the 14th ACM Conference on Recommender Systems*, pp. 579–581 (2020)
22. Durrant, A., Markovic, M., Matthews, D., May, D., Enright, J., Leontidis, G.: The role of cross-silo federated learning in facilitating data sharing in the agri-food sector. *Comput. Electron. Agric.* **193**, 106648 (2022)
23. Radicic, D., Petković, S.: Impact of digitalization on technological innovations in small and medium-sized enterprises (SMES). *Technol. Forecast. Soc. Chang.* **191**, 122474 (2023)
24. Fader, P.S., Hardie, B.G., Shang, J.: Customer-base analysis in a discrete-time noncontractual setting. *Mark. Sci.* **29**(6), 1086–1108 (2010)
25. Stewart, G., Al-Khassaweneh, M.: An implementation of the HDBSCAN* clustering algorithm. *Appl. Sci.* **12**(5) (2022)
26. Junczys-Dowmunt, M., et al.: Marian: fast neural machine translation in C++ *arXiv preprint arXiv:1804.00344* (2018)
27. Olist, Sionek, A.: Brazilian E-commerce public dataset by Olist (2018)
28. Ludwig, H., et al.: IBM federated learning: an enterprise framework white paper v0. 1, *arXiv preprint arXiv:2007.10987* (2020)



Quench Detection and Localization via Interpretable Machine Learning

Alessandro Biagiotti¹ , Dario Malchiodi¹  , Samuele Mariotto^{2,3} ,
and Lucio Rossi^{2,3} 

¹ Computer Science Department, Università degli Studi di Milano, via Celoria 18,
20133 Milano, Italy

`alessandro.biagiotti@studenti.unimi.it`, `dario.malchiodi@unimi.it`

² Physics Department, Università degli Studi di Milano,

Via Celoria 16, 20133 Milano, Italy

`{samuele.mariotto,lucio.rossi}@unimi.it`

³ Istituto Nazionale di Fisica Nucleare, Laboratorio Acceleratori e Superconduttività
Applicata, LASA, Via Fratelli Cervi, 201, 20054 Segrate, Italy

Abstract. In particle accelerator physics, superconducting magnets are crucial to achieve high-energy particle beams used in fundamental physics experiments. One of the key challenges of superconducting magnets is the transition of the superconducting material to the normal conducting resistive state—called *quench*—which could cause damage to the superconducting magnet volume, due to the large quantities of heat produced by ohmic losses in the material if the magnet is not properly protected. In this work, we train explainable machine learning models to detect the occurrence of quenches, starting from the harmonic decomposition of the magnetic field produced by the residual superconducting magnetization after a quench event and aiming at explaining the prediction process. A successful solution to this problem is a fundamental ingredient in the construction of a real-time predictive maintenance system. Indeed, interpretability paves the way to the construction of a tailored, efficient, and affordable system that only considers the relevant magnetic field harmonics. We show that quenches can be detected with high accuracy by rather simple models, and we also describe some preliminary but encouraging results on the quench localization problem.

Keywords: Quench detection · Quench localization · Machine learning applications in Physics

1 Introduction

Superconductivity is a phenomenon that has been studied since 1911 [1] and is characterized by the vanishing of electrical resistance and expulsion of external magnetic field lines once: (i) the material operating temperature is below the critical temperature T_c , (ii) the electrical current flowing through the material does not exceed the critical current density J_c , and (iii) the magnetic field acting

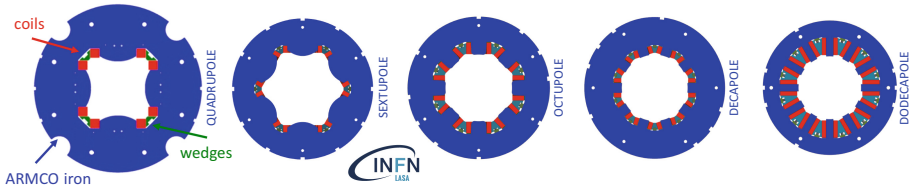


Fig. 1. High order corrector magnets cross-sections. The three principal components of the magnet assembly, superconducting coils, iron yoke and wedges of the mechanical support structure are shown.

on the material does not exceed the critical magnetic field B_c . If the material is in the superconducting state, and any of the above three conditions is not valid, then the material undergoes a transition, called *quench*, to the normal conducting state. Depending on the superconductor operating condition, a non-properly accounted quench development, especially in the field of accelerator physics, where high electrical current density is required in the superconducting material lattice, can lead to severe damages, such as destruction of the superconductive magnet or other components of the accelerator complex and disruption of the accelerator facility or physics experiments.

The analyzed quench events described in this paper have been measured from the test campaign of the 54 High Order (HO) corrector superconducting magnets developed by the Istituto Nazionale di Fisica Nucleare (INFN) Laboratorio Acceleratori e Superconduttività Applicata (LASA) laboratory within the INFN and CERN development agreement for the low- β section [2] of the High Luminosity LHC upgrade project [3]. Divided into 5 different magnetic field harmonic content (from quadrupole up to dodecapole order, see Fig. 1), the superferric magnets will be used to correct the field errors of the main strong focusing quadrupoles, responsible for the focusing of the proton beams in the ATLAS [4] and CMS [5] interaction point (IPs) experiments.

An analytical model of the reconstructed quenched superconducting coils within the superferric magnets, based on the residual magnetic field harmonic decomposition analysis after a quench event, has been proposed in [6]. The obtained results led to a general method to analyze the performance of any superconducting magnet order [7] where the superconducting coils are independent of each other, and the goal of the reconstruction analysis is the localization of the quenched portion of the magnet. Originating from this work, this paper describes the use of explainable machine learning (ML) models trained to recognize whether a superconducting magnet has undergone a quench event and which coils underwent the transition, starting from the harmonic decomposition of the magnetic field. More precisely, we focus on:

- the Quench Recognition problem, consisting in predicting whether or not a magnet is undergoing a quench, and

- the Quench Localization problem, in which the prediction is more accurate, being expressed in terms of the specific coil(s) transitioning to the normal-conducting state.

For brevity, we will use the acronyms QRP and QLP to refer to the Quench Recognition and Localization problems, respectively. A successful solution to these problems is a fundamental ingredient to develop a real-time predictive maintenance system for the HO superconducting magnets, instead of the *post*-event measurements analyzed both with the analytical model and the ML approach described here.

To the best of our knowledge, no interpretable ML models have yet been proposed to detect and locate quench occurrences within a superconductor magnet based on magnetic field measurements of the superconductor residual magnetization. State-of-the-art research is mainly based on two different approaches: the first one is aimed at leveraging ML to predict properties of superconducting materials in the realm of material science [8,9], while the latter tries to predict quench events using high-performance ML models that lack the explainability property [10,11]. This property, however, is a key factor in building the above-mentioned predictive maintenance system: indeed, knowledge about the harmonics needed to solve QRP and QLP allows a tailored construction of the hardware system devoted to detect and localize quench in real-time, jointly optimizing its cost and efficacy.

The work is organized as follows: Sect. 2 provides a detailed description of the experiment that generated the dataset utilized in our analysis, along with a discussion of the ML models employed and the overall training process. Section 3 presents our findings for QRP and QLP, respectively. In the last section, we draw some concluding remarks, highlighting the limitations of our study and outlining prospective future research directions.

2 Materials and Methods

Here we describe how we carried out our experiments: in Sect. 2.1 we provide details on the dataset construction, and illustrate its structure; Sect. 2.2 briefly describes the ML models that we have used; Sect. 2.3 outlines the process we have run for training and testing these models.

2.1 Data Collection

Following [6], the HO corrector magnets have been designed to produce five different multipolar field orders (quadrupole, sextupole, octapole, decapole, and dodecapole). Tested at cryogenic temperature, the magnets are powered to evaluate their performances and thermal stability up to the nominal operating condition. During magnet powering, spontaneous quench transitions in the superconducting coil volume can occur due to mechanical instability of the winding or stress induced by the electromagnetic Lorentz forces acting on the conductors.

Each magnet is protected using a classical resistive voltage detection system and discharged on an external dumping resistance to extract most of the stored magnetic energy from the magnet to prevent any material damage. Limited to analyzing only two separate voltage signals connected in parallel to the coils, the quench detection system can only evaluate if a quench development occurred in half of the magnet coils without the possibility to localize precisely which superconducting winding started the transition to the normal conducting state. To retrieve this information and evaluate if a possible degradation of the coil has occurred, a magnetic field measurement-based model has been proposed analyzing the residual magnetic field quality produced by the non-quenched superconducting coils in the magnet after the quench event. The original work proposed a study specific to quadrupoles, since they gave the majority of the positives (quench events), which have been used as the subject of this analysis, therefore ignoring all other configurations focusing only on the five quadrupoles of the series production from MQSXF1 to MQSXF6.

As a result of the magnets test campaign, 279 events, after the magnet discharge (quench event) or equivalent at zero operating current (non-quench event), were analyzed sampling the magnetic field in the magnet bore volume using a magnetic field rotating shaft. For each event, 60 samples of measured harmonic coefficients have been measured and averaged to obtain a single set of 15 different magnetic field harmonics coefficients which contain all the information of the produced magnetic field quality (see Fig. 2). Recalling that these coefficients are expressed as complex numbers, the information have been organized in terms of the following four attributes, in which the suffix $n \in \{1, \dots, 15\}$ is related to a specific harmonic:

- the imaginary part A_n of the n^{th} field harmonic coefficient;
- the real part B_n of the n^{th} field harmonic coefficient;
- the absolute value $|C_n|$ of the n^{th} field harmonic coefficient;
- the phase ϕ_n of the n^{th} field harmonic coefficient.

In the experiment generating data, the magnets were mounted skewed: as a consequence, the A_n feature was supposed to be highly informative since they are

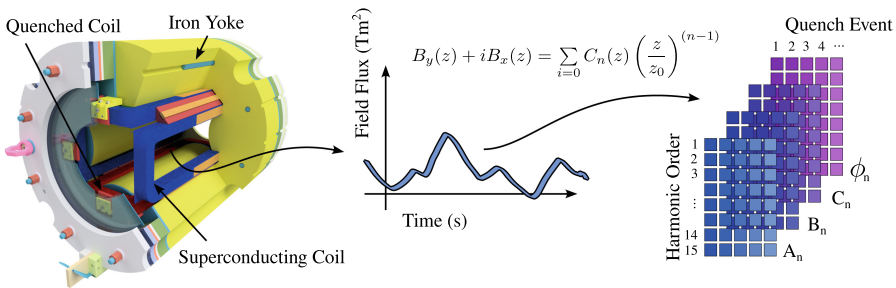


Fig. 2. Sketch of the magnetic field measurements after a quench event for the quadrupole magnet and dataset obtained for each quench event measurement.

directly correlated with the rotational position of the quenched coil in the magnet assembly. The absolute values $|C_n|$ would be a valuable feature for QRP and not for QLP since two different configurations with the same number of quenched coils but different angular positions would give the same harmonic content and different phases. Each measurement was coupled with a *label* encoded as follows:

- a simple binary flag for QRP, where 1 means that a quench has happened, while 0 characterizes the normal working conditions of the superconductor;
- four binary flags for QLP, each describing a coil and stating whether or not a quench happened within it, using the same encoding as in previous point.

We remark here that the labels for QRP were reasonably balanced: namely, the number of non-quench events was of 87 while the number of quench events was of 192; concerning QLP, on the other hand, every coil has a different distribution: as detailed in Table 1, the proportion of quench over non-quench events roughly ranges from 30% to 50%. It is important to notice that the sum of all quench events for QLP is not 192, since one quench event for QRP could have been caused by up to three different coils (e.g., coils 0, 1 and 3) quenching at the same time.

Table 1. Label distribution for the various coils in QLP.

Coil	Quench	Non-quench
0	68	211
1	96	183
2	79	200
3	73	206

In the following, we will denote by $x \in \mathbb{R}^{60}$ the generic vector gathering the values of a measurement, naming it an *item*, and by y the corresponding label. Note that $y \in \{0, 1\}$ in QRP and $y \in \{0, 1\}^4$ in QLP. Finally, a pair (x, y) will be called an *example*, so that our dataset can be modeled as a set $\mathcal{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ of examples. This dataset underwent a classical pre-processing phase, during which we evaluated, separately for each attribute: (i) the cross-correlations between the harmonics, (ii) the cross-correlation between the harmonics and the labels, (iii) the box plot of the attributes, and (iv) the distribution of data after dimensionality reduction via principal component analysis (PCA). Cross-correlation between the harmonics shows, for all attributes apart from ϕ_n , a pattern of very strong dependencies (a classic occurrence is harmonic number 2 having a very strong correlation, ≈ 1 , with its odd multiples), which made the process of subview extraction (cfr. Sect. 2.3) quite complicated, since it was necessary to find harmonics capable of having a high correlation with the label(s) while, at the same time, having a low correlation among themselves. Whenever in feature selection we were forced to decide between two or more harmonics we referred to the boxplot and chose the one that had the highest

variability. When we checked the distribution of the data after a round of PCA dimensionality reduction (we moved from 15 to 2 dimensions) we discovered that the A_n and $|C_n|$ contain clusters with a very high degree of purity, which was taken as evidence of the quality of the attributes. After this phase, the dataset has been standardized (that is, linearly transformed so that it has null mean and unit variance).

2.2 Models

Both QRP and QLP fall in the realm of supervised classification (respectively, binary and multi-label): starting from $\mathcal{D}$, a ML *model* is trained so that, when queried with an item x , it outputs a *prediction* $\hat{y}$ that is close to the corresponding label y , possibly generalizing well to unseen data. The following paragraphs briefly describe the three models that we have considered in our experiments.

Decision Trees. A decision tree (DT for short henceforth) is a classifier that hierarchically organizes simple queries in a binary tree [12]. When an item is presented to the tree, the query corresponding to its root is executed on the data, and depending on the result the process is repeated either on the left or on the right child node, continuing until a leaf is reached, and returning the corresponding prediction. Therefore, the item identifies a path in the tree, whose queries implicitly define a *decision rule* which symbolically explains the classification process. Starting from a dataset $\mathcal{D}$, a DT is trained by recursively splitting the input space, each time selecting the boolean query jointly optimizing an impurity measure for the labels of the items that result, respectively, in a positive and a negative outcome of the query itself [13]. Each time the label homogeneity of the items that satisfy (or do not satisfy) the query is maximal or quasimaximal, a leaf associated to the majority label of these items is created.

Random Forests. A Random Forest (RF) basically consists of a set of decision trees [14], each induced on a different bootstrap sample drawn with replacement from $\mathcal{D}$, and considering queries only from a random subset of the available attributes. When a RF is queried on an item x , the predictions of the various trees on the latter are aggregated through majority vote.

Support Vector Classifiers. A binary Support Vector Classifier (SVC) considers each item as a vector in the Euclidean space, and finds a decision surface separating the vectors corresponding to two classes. This is achieved by maximizing the so-called *margin*, intended as the minimal distance between the decision surface and the data images through a nonlinear mapping [15]. The related optimization process exploits the so-called *kernel trick* to automatically select this mapping within some prefixed families. SVCs can be naturally extended to the multi-label case using standard techniques [16].

2.3 Training Process

Hardware and Software Configuration. We used standard facilities, as our code did not require top-performing hardware. All experiments have been executed using Python 3.10.12 (the latest version at the time of writing), using the following libraries: scikit-learn (release 1.5.2) for data preprocessing and model training and evaluation, numpy (release 2.2.3) for efficient data storage, and Pandas (release 2.2.3) for data management. These libraries have been handled through the pip package manager (release 25.0.1), within a virtual environment.

Dataset Generation. We partitioned the measurements collected as in Sect. 2.1 and encoded in $\mathcal{D}$ into a *reduced* and a *blind test* set, which we will refer to as $\mathcal{D}_{\text{red}}$ and $\mathcal{D}_{\text{test}}$, respectively. The related sampling process was carried out by stratifying on labels, highlighting that in QRP and in some instances of QLP we had only two different labels, but in general the multi-label setting of QLP required to adapt the stratification process (see [17] for details). We used $\mathcal{D}_{\text{red}}$, which contains 250 examples, for all experiments, while the remaining data (29 examples) were kept aside and used only at the end of the process, in order to avoid a bias in the estimate of model generalization. Based on our knowledge in the field, we have considered different views on the dataset, namely using only A_n , only B_n , only $|C_n|$ and only ϕ_n . Each view was further detailed in several subviews, each focused on a different set of harmonics.

Model Selection and Evaluation. The models described in Sect. 2.2 are characterized by the presence of *hyperparameters*, that is, values to be set before the training phase. To properly fine-tune these values and jointly evaluate generalization performance, we used a 5×5 nested cross-validation scheme (CV) on the examples in $\mathcal{D}_{\text{red}}$, devoting the outer CV to model evaluation and the inner one to model selection, using a grid search over a prefixed set of values. Table 2 shows the hyperparameters for each considered model, along with the set of possible values that we tested. We applied stratification on labels, using the same strategies described before for QRP and QLP, also when generating the folds of both cross-validations. A specific evaluation scheme was adopted when building tree aggregators (cfr. Sect. 3.1), obtained by grouping already trained models. In this case, we just adopted a single 5-fold-CV for model evaluation.

Evaluation Metrics. All trained models have been tested using the following performance metrics, where, as usual, by positive (negative) item we mean an item having label 1 (0), and by true (false) positive we mean a correctly classified positive (negative) item:

- *Accuracy*: fraction of correct predictions,
- *Precision*: number of true positives over number of positive predictions,
- *Recall*: number of true positives over number of positive items,
- *F1 score*: harmonic mean between Precision and Recall,
- *Inverse recall*: number of true negatives over number of negative items,
- *ROC AUC*: area under the receiver operating characteristic curve.

Table 2. Hyperparameters which we have tuned during the model training phase, each paired with the corresponding set of considered values.

Model	Hyperparameter	Values
DT	impurity criterion	gini, entropy, log loss
	max depth	2, 3, 4, 5
	min impurity decrease	0.001, 0.01, 0.05
	max features	None, 0.5, 0.75
	min samples leaf	5, 10, 20
RF	n. of trees	2, 3, 5, 10
	+ the same hyperparameters and values of DT	
SVC	C	0.1, 1, 10, 100, 1000
	γ	scale, auto, 0.001, 0.01, 0.1, 1, 10
	degree	2, 3, 4, 5
	c_0	0, 0.1, 0.5, 1
	kernel	linear, polynomial, sigmoid, RBF

In the following figures, these metrics will be shortened as Acc, Prc, Rec, F1, Irec and RAUC. For all of them, the higher the metric value, the better the performance. The model selection phase was driven by the accuracy values, whereas all metrics were computed in the model evaluation step.

3 Results

Here we report the results of our experiments: Sect. 3.1 describes our findings for QRP, while Sect. 4 contains some preliminary outcomes for QLP.

3.1 QRP

The simplest setting for our experiments was focusing on QRP, in which a binary prediction for quenches was provided by a single DT. Figure 3 depicts the results in terms of the four best trees trained starting from the subviews described in Sect. 2. A model based on A_n emerges as the best one, reaching a 0.984 accuracy, meanwhile recognizing both quench and non-quench events with a high rate. The structure of this tree, shown in Fig. 4, is extremely simple: it exclusively refers to the second and twelfth harmonics, and has depth 2. Therefore, its decision rules are very easy to consider for an interpretation (e.g., if $A_2 \leq -1.058$ and $A_{12} \leq 0.333$, predict ‘no-quench’). These experiments naturally extend to RFs, in the hope that this more complex model might attain even higher performances (possibly aggregating trees that explore different subviews of the data), while maintaining a sufficiently explainable structure. However, Fig. 5 shows the performance of the four best forests obtained, highlighting that the best performing, based on A_n and $|C_n|$ and taking advantage of all features, is on par,

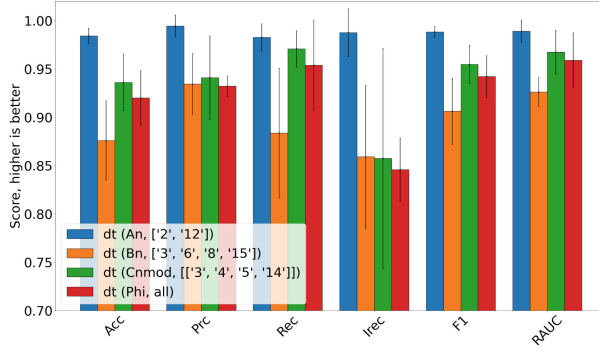


Fig. 3. Decision tree performance of the best data subviews. Each bar shows the value of an evaluation metric, marked on the X axis and averaged on the five folds of the outer CV. Error bars describe the corresponding standard deviation. The legend identifies the attributes and harmonics used by the trees.

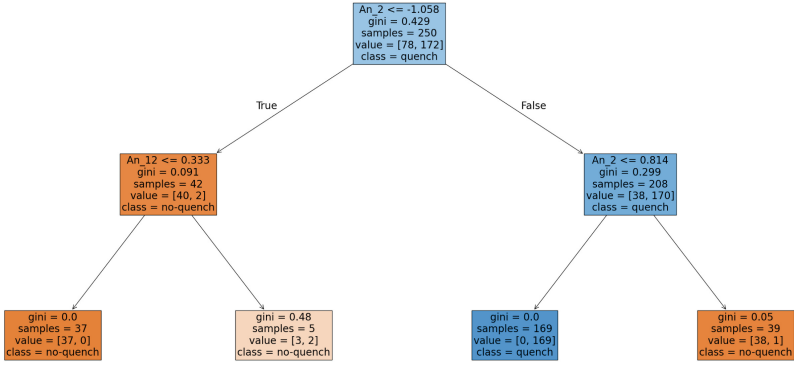


Fig. 4. Best DT for QRP, based on the A_n features of the second and twelfth harmonics. Each node is labeled with the corresponding query.

performance-wise, with the best DT. Noting that each RF in the previous experiment was generated by selecting uniformly at random among all features and all harmonics, we suspected that a more informed sampling might lead to better results. Thus we created ensembles selecting the best DTs, chosen based on their accuracy, for each data subview, keeping their structure (that is, the fine-tuned hyperparameter values) and training them on $\mathcal{D}_{\text{red}}$ using a 5-fold CV (as explained in Sect. 2.3). The prediction of such ensembles was obtained as in RF, using majority voting. We call this model *Tree Aggregator* (TA) to differentiate it from RF. We choose to work with triplets of trees (e.g., based on $A_n, B_n, |C_n|$), in order to avoid ties when aggregating their predictions. Figure 6 compares the performance of different TAs, each aggregating specific subviews, with that

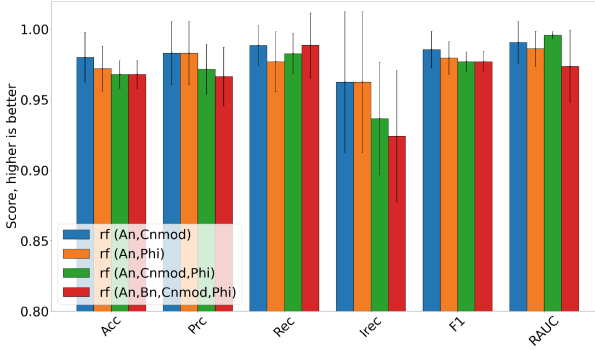


Fig. 5. Performance of the best four RF models. Same notations as in Fig. 3.

Table 3. Performances of the best TA on the blind test set $\mathcal{D}_{\text{test}}$.

Acc	Prc	Rec	F1	Irec	RAUC
0.965	0.952	1	0.976	0.889	0.944

obtained by the best RF and SVC on the same data subview¹. We considered the metrics obtained by the latter as the potentially optimal ones, due to the strong performance of this model generally reported in the literature². Indeed, SVC is almost always the best performer, but TAs obtain not so dissimilar results, and the model built using A_n , B_n , and $|C_n|$ even outperforms it in all metrics except for RAUC, probably in view of the aggregation technique that could lead to partly biased predictions. The best TA contains trees that exhibit a fairly simple structure, as shown in Fig. 7 for the B_n and $|C_n|$ trees (recall that the one based on A_n has already been shown in Fig. 4). In sum, there is a reasonable trade-off between a higher performance and a smaller interpretability. Finally, we evaluated the best TA on the blind test set $\mathcal{D}_{\text{test}}$, obtaining only an incorrect classification over 29 examples (precisely, a false positive), corresponding to the metric values detailed in Table 3, essentially confirming our initial estimate.

4 Best Models for QLP

In this section we will provide early results for QLP. Directly extending the QRP models to the multi-label setting resulted in too low a performance (≈ 0.6 accuracy). Thus we used a *divide et impera* approach, training a different model for each coil. When considered separately, DTs attained performances close to

¹ We opted for this comparison in order to level the playing field and train the models on the same data. However, we also conducted a comparison with the overall best RF and SVC, obtaining analogous results.

² Note that, in any case, SVCs are not an option for our aim, lacking a reasonable explainability of their predictions.

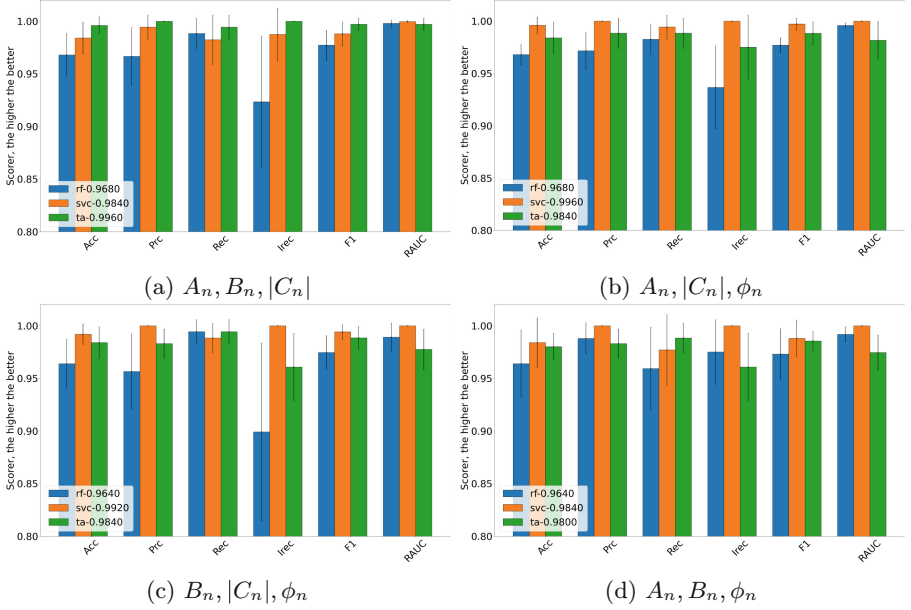
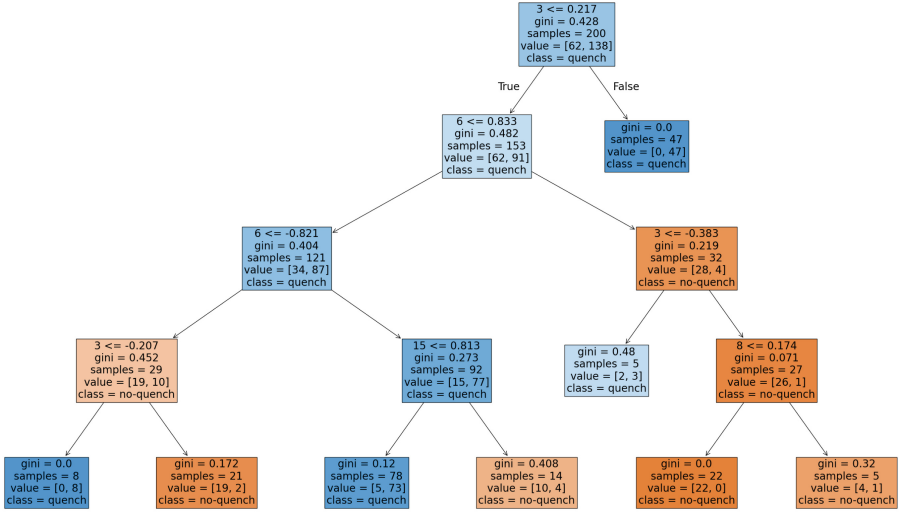
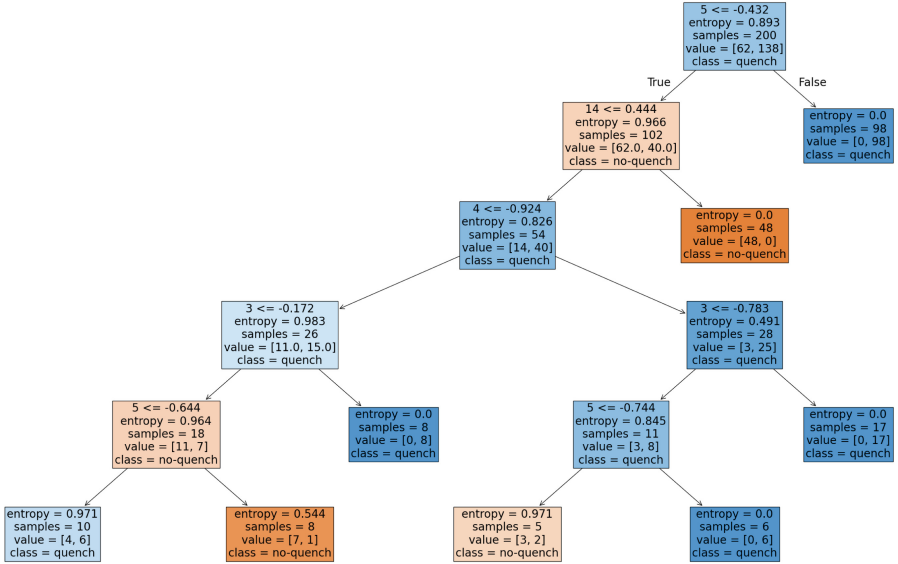


Fig. 6. Performance of the TAs trained on different triplets of features, each shown under the figures, compared with the best reference RF and SVC. Same notations as in Fig. 3.

the ones we obtained for QRP, while keeping an overall relatively simple structure. For the sake of comparison, the best models we found have an accuracy $\geq .95$, and the most frequently used feature is A_n , followed by B_n and ϕ_n . Differently from QRP, RFs scored a significant performance improvement w.r.t. DTs, but we chose to not investigate this model in depth, as its structure is too complex to be interpreted in the context: for each coil, we would have a RF, in turn containing at least three DTs. For the same reasons, we did not consider TAs. To address the multi-label problem, we aggregated the results of the four best DTs by simply concatenating their predictions, and we measured the accuracy using the Hamming distance h between labels and predictions, both intended as 4-bits words, as follows: given a label y and a prediction $\hat{y}$, the corresponding contribution to accuracy is

$$\text{HS}(y, \hat{y}) = 1 - \frac{h(\hat{y}, y)}{4} . \quad (1)$$

Performance was estimated on $\mathcal{D}_{\text{red}}$ by using a single 5-fold CV, as we did for TAs in QRP, obtaining a mean accuracy 0.948, with standard deviation of 0.0024. When recomputed on the blind test set, the mean and standard deviation are respectively equal to 0.9569 and 0.0944.

(a) Tree model trained on the subview of B_n containing harmonics 3, 6, 8 and 15(b) Tree model trained on the subview of $|C_n|$ containing harmonics 3, 4, 5 and 14**Fig. 7.** The components of the Tree Aggregator trained from the B_n (a) and $|C_n|$ (b) data views.

5 Conclusions

We considered the problem of detecting and localizing quenches in high-efficiency superconducting magnets, using explainable ML approaches that can give valuable insight for building predictive maintenance systems for these magnets. We have shown that a relatively simple ensemble of decision trees trained using an informed selection of the available features attains very good classification performances, being meanwhile open to an actual interpretability process. We also obtained some preliminary, although very promising, results on quench localization. The main limitations of our approach are essentially two. The first one, typical of the ML realm, is related to the specific quadrupole magnets for which we collected the data. In principle, our work is extensible to other types of magnets, but this would require a new data collection phase, followed by model training. The second limitation is related to the refinement of our results: further collection and training would be required even for the considered magnets, in case higher accuracies should be required. However, the corresponding increase in the size of the dataset might have an impact on interpretability.

We plan to extend our work in several directions, in addition to that related to further study QLP. First of all, we will analyze the decision rules extracted by the best DT and TA, with the aim of finding interesting physical interpretations of their behavior. In addition, we have preliminary performed some unsupervised analysis, computing a clustering from the bidimensional dataset obtained via PCA: we found that most clusters have a very high purity for both QRP and a variation of QLP, namely, focusing on identifying non-quench, single quench, and multiple quench events. This could be used as a pre-processing step for a finer classification, leading to better performance. Finally, we envisage reconsidering the detection problem, for instance, working in the realm of anomaly detection [18], or switching from a crisp to fuzzy forms of classification [19,20], in which belongingness to the positive class is expressed, for instance, through a label $y \in [0,1]$ that allows the representation of quenches having different strengths, for instance taking into account the severity of the transition (where a quench occur in more coil volume than other quench events resulting in a faster buildup of the resistive voltage).

Acknowledgments. The authors would like to thank Marco Statera from INFN Milano, responsible for the superconducting magnet development within the CERN-INFN research agreement, and Ernesto De Matteis from INFN, responsible for the magnetic measurement campaign performed during the magnet testing. Part of this work was funded by the National Plan for NRRP Complementary Investments (PNC) in the call for the funding of research initiatives for technologies and innovative trajectories in the health—project n. PNC0000003—AdvaNced Technologies for Human-centrEd Medicine (project acronym: ANTHEM).

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

References

1. Van Delft, D., Kes, P.: The discovery of superconductivity. *Phys. Today* **63**, 38–43 (2010)
2. Todesco, E., et al.: The high luminosity LHC interaction region magnets towards series production. *Supercond. Sci. Technol.* **34**, 053001 (2021)
3. Rossi, L., Bruning, O.: High Luminosity Large Hadron Collider, The: New Machine For Illuminating the Mysteries of the Universe, 2nd edn. Advanced Series on Directions in High Energy Physics, World Scientific Publishing Company (2024)
4. ATLAS: technical proposal for a general-purpose pp experiment at the Large Hadron Collider at CERN. Technical report. CERN-LHCC-94-43; LHCC-P-2, CERN, Geneva (1994)
5. Bayatian, G.L., et al.: CMS physics: technical design report volume 1: detector performance and software, Technical report. CERN-LHCC-2006-001, CMS-TDR-8-1, CERN-LHCC-2006-001, CMS-TDR-8-1, CMS-TDR-008-1 (2006)
6. Mariotto, S., Sorbi, M.: Quench position reconstruction through harmonic field analysis in superconducting magnets. *Supercond. Sci. Technol.* **35**, 015006 (2021)
7. Mariotto, S., De Matteis, E., Prioli, M., Sorbi, M., Statera, M., Valente, R.: Quench localization in the high order corrector magnets using the harmonic field method. *IEEE Trans. Appl. Supercond.* **32**, 1 (2022)
8. Matsumoto, K., Horide, T.: An acceleration search method of higher TC superconductors by a machine learning algorithm. *Appl. Phys. Express* **12**, 073003 (2019)
9. Stanev, V., et al.: Machine learning modeling of superconducting critical temperature. *NPJ Comput. Mater.* **4**, 29 (2018)
10. Hoang, D., et al.: Intelliquench: an adaptive machine learning system for detection of superconducting magnet quenches. *IEEE Trans. Appl. Supercond.* **31**, 1–5 (2021)
11. Peng, D.: Analysis of superconducting magnet quench antenna data. Technical report, Fermi National Accelerator Laboratory (FNAL), Batavia, IL (USA) (2024)
12. Quinlan, J.R.: C4.5: Programs for Machine Learning. Elsevier (2014)
13. Breiman, L., Friedman, J., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Routledge (2017)
14. Breiman, L.: Random forests. *Mach. Learn.* **45**, 5–32 (2001)
15. Cortes, C., Vapnik, V.: Support-vector networks. *Mach. Learn.* **20**, 273–297 (1995)
16. Bogatinovski, J., Todorovski, L., Džeroski, S., Kocev, D.: Comprehensive comparative study of multi-label classification methods. *Expert Syst. Appl.* **203**, 117215 (2022)
17. Szymański, P., Kajdanowicz, T.: A scikit-based Python environment for performing multi-label classification, *ArXiv e-prints* (2017)
18. Nassif, A.B., Talib, M.A., Nasir, Q., Dakalbab, F.M.: Machine learning for anomaly detection: a systematic review. *IEEE Access* **9**, 78658–78700 (2021)
19. Malchiodi, D., Pedrycz, W.: Learning membership functions for fuzzy sets through modified support vector clustering. In: Masulli, F., Pasi, G., Yager, R. (eds.) *WILF 2013. LNCS (LNAI)*, vol. 8256, pp. 52–59. Springer, Cham (2013). https://doi.org/10.1007/978-3-319-03200-9_6
20. Apolloni, B., Malchiodi, D., Valerio, L.: Relevance regression learning with support vector machines. *Nonlinear Anal.* **73**, 2855–2867 (2010)



Spatiotemporal Multiplex Network Model for Predicting Forced Outage Severity in Distribution Grids

Rafaa Aljurbua^{1,3}(✉) , Rashid Baembitov² , Christos Petridis¹ ,
Daniel Saranovic¹ , Mladen Kezunovic² , and Zoran Obradovic¹

¹ Computer and Information Science Department, Center for Data Analytics and Biomedical Informatics, Temple University, Philadelphia, PA, USA

{rafaa.aljurbua,christos.petridis,
daniel.saranovic,zoran.obradovic}@temple.edu

² Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX, USA

{bae_rashid,m-kezunovic}@tamu.edu

³ Department of Computer Science, College of Computer, Qassim University, Buraydah, Saudi Arabia

Abstract. Weather-related power disruptions present significant challenges to public infrastructure, societal well-being, and the distribution grid. Predicting outage durations in distribution grids is another challenge compared to transmission line outage durations due to distribution networks' complexity and finer granularity. While forecasting forced power outages is crucial, accurately estimating their duration is essential for timely response and mitigation measures. This study introduces the Spatiotemporal Multiplex Network (SMN-WVF), a methodology designed to predict power outage durations across varying lead times, tackling the difficulties posed by small, high-complexity spaces within distribution grids. SMN-WVF employs multiplex networks that incorporate multi-modal data across both time and space, including layers such as power outages, weather conditions, weather forecasts, vegetation, and distances between substations. We demonstrate the importance of incorporating additional layers of data sources as they are shown to help the model's predictions through gradual improvement in the macro F1 score performance.

Keywords: Forced power outage · multiplex network · spatiotemporal prediction

1 Introduction

Forced power outages (further referred to as “outages”) have a considerable impact on both the economy and the lives of residents in affected areas. The causes of such power outages can be broadly divided into four primary categories: emerging threats, accidental incidents, malicious activities, and natural events

[10]. Severe outages, particularly those in rural regions, can last for extended periods and require significant restoration time [30]. Furthermore, power system operations are typically classified into four distinct states: normal operation, alert, emergency, and extreme [13]. Previous research studied the power outage problem statistically, employing techniques such as quantile regression forests and Bayesian additive regression tree models [31]. In contrast, recent advancements in machine learning have shifted towards predictive models that utilize machine learning techniques to enhance the accuracy of power outage predictions and identify their causes [19]. Previous research also employed graph-based models to predict power outages [11, 25]. However, these studies did not leverage multiplex networks to improve predictions; instead, they were limited to using single-layer input.

Predicting power outages is essential, but predicting their duration is crucial to efficient response and mitigation actions. Statistical methods [21] and machine learning models [12] have been applied to predict the duration of power outages. However, researchers have yet to fully explore the potential of multiplex networks that integrate data from multiple sources. In our previous study [1], multiplex graphs demonstrated their effectiveness in predicting the occurrence of power outages. Based on this, our follow-up study [2] successfully used multiplex graphs to predict the duration of power outages, outperforming alternative approaches. However, neither analysis addressed the prediction of the duration of outages for the distribution grid. To address this limitation, our current study introduces SMN-WVF, a Spatiotemporal Multiplex Network model designed to predict the duration of power outages while extending prediction time intervals demonstrating a use case of a utility in Texas, U.S.A. SMN-WVF integrates data from multiple sources, including weather, forecasts, and vegetation, collected across space and time. By providing earlier warnings of outage risks, this model aims at enhancing the efficiency of outage management and improving response efforts. The key contributions of this paper are:

1. Establishing a foundational framework for improving spatiotemporal classification by incorporating multi-modal data to tackle the challenges of missing weather recordings effectively.
2. Introducing Spatiotemporal Multiplex Network methodology to predict power outage durations by employing the multi-modal approach.
3. Applying the new approach to estimate outage severity (duration) in the distribution grid by providing insights distinct from those focused on transmission line outages.

2 Related Work

Extreme weather events like severe storms (rain, snow, wind) and other catastrophic natural disasters (earthquakes, wildfires, hurricanes) can cause power outages. To improve the planning of power restoration efforts, accurately predicting the duration of power outages early is crucial for utility companies.

Various statistical approaches have been employed to estimate outage duration, including accelerated failure time regression, Cox proportional hazards regression, Bayesian additive regression trees, regression trees, and multivariate adaptive regression splines [21]. Some studies integrate statistical methods with geographic information systems (GIS) to analyze urban distribution systems affected by winter storms. For example, GIS tools were used to map repair crew data and examine the duration of outages during such events [26]. Similarly, researchers have applied Accelerated Failure Time (AFT) and Cox Proportional Hazard (CPH) models to estimate storm-induced power outage durations [18]. In another example, daily Night Time Lights (NTL) imagery data has been utilized to assess the duration of outages in Puerto Rico [7]. In addition, studies indicate that socioeconomic factors play a significant role in determining the outage duration [20]. Recently, the focus has shifted toward leveraging machine learning models to predict the duration of power outages more accurately. For instance, Random Forest-based models have been used to forecast the duration of hurricane-related outages, incorporating variables such as wind speed and duration [22]. Furthermore, machine learning techniques such as Extra Trees (ET), Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), Random Forest (RF), Gradient Boosting Regression (GBR), and Decision Tree (DT) have been applied to predict outage durations during typhoon disasters [15].

Weather data plays a crucial role in predicting power outages [9]. However, severe weather conditions often result in many missing values in weather recordings, which present challenges for predictive modeling. Our research integrates weather data with multi-modal learning techniques to address this challenge. Given the challenges of missing data, multi-modal learning techniques offer a promising solution by improving predictive accuracy [1, 5, 8, 24]. In particular, we explore multiplex networks' impact in predicting the power outage duration at the distribution grid level. The network structure is crucial for capturing complex interdependencies between different data sources, thus improving the accuracy and robustness of the predictions [3, 4]. This study introduces a novel approach that leverages multiplex network representations to predict the power outage duration for severe weather cases. Specifically, we evaluate the benefits of using a Spatiotemporal Multiplex Network that integrates data from multiple key sources: power outages, weather observations, weather forecasts, vegetation, and distance between the substations.

3 Methodology

This study investigates the effectiveness of the proposed Spatiotemporal Multiplex Network (SMN-WVF) model, which combines multiplex networks and multi-modal data, in improving the prediction of power outage duration. The SMN-WVF model estimates the duration of the predicted outages by predicting their duration. Figure 1 illustrates the pipeline of the proposed approach to building the Spatiotemporal Multiplex Network model that includes information captured at five layers from different sources. The research is divided into

three phases: (1) data collection, (2) construction of a spatiotemporal graph, and (3) development of the model that predicts the duration of expected outages. Detailed descriptions of each component of this framework are provided in the following subsections.

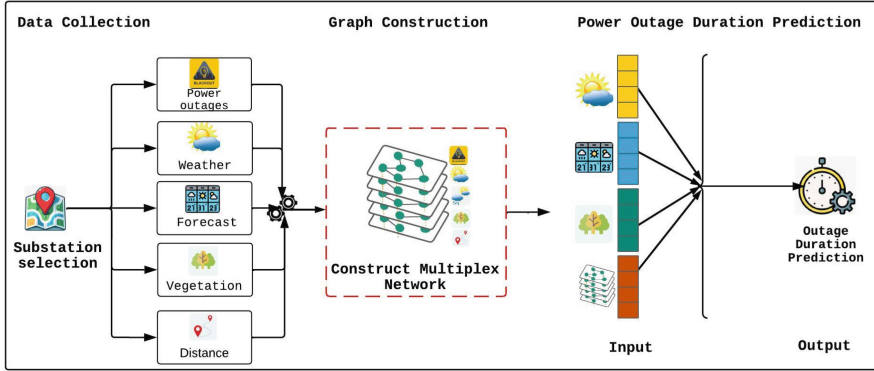


Fig. 1. The architecture of the Spatiotemporal Multiplex Network (SMN-WVF) for multi-modal prediction of power outage duration.

3.1 Data Collection

In our previous research [1], we demonstrated the advantages of integrating data from diverse sources to improve the prediction of the duration of power outages. To achieve this, we identify and gather key factors that influence the power outage duration. These factors include power outage records, weather conditions, forecasts, and vegetation data. This subsection outlines the data collection process.

1. **Power outages data:** We utilize a historical outage dataset obtained from a Texas utility company, which details when and where outages occurred from January 2018 to December 2023. To effectively use this dataset, it is crucial to correlate each outage occurrence with the corresponding substation. The dataset contains outage start and end times, outage cause codes, the nearest substation, failed equipment IDs, and repair crew comments. We employ the equipment ID and nearest substation fields to map each outage to its respective substation. Historical outage data are recorded in the local time zone. However, since most public datasets use UTC for timestamps, we converted the outage timestamps from the local time zone to UTC and linked the events with the appropriate substation ID. Our study utilizes six years of historical outage data during which the reported power outages reveal a total of 109,366, with an average duration of 101.08 min.

2. **Weather data:** Between January 2018 and December 2023, we collected weather data from Automated Surface Observing Systems (ASOS) [6]. The ASOS database includes hourly information retrieved from the ground weather station sensors, as well as data about the station's location, sky conditions, obstructions to vision, pressure, ambient temperature, wind, and precipitation accumulation for point locations. We collected 8,425,383 observations from ASOS weather stations. However, the substations are not connected to an ASOS weather station; therefore, we map each substation to the closed ASOS station using latitude and longitude. We use the Haversine formula to calculate the distance between the substation and the ASOS stations [27].
3. **Weather forecast data:** The ASOS database does not include weather forecast data. To obtain this information, we use the OpenMeteo API, providing open-source weather forecasts from national weather services [23]. OpenMeteo offers weather models with a resolution of 11 km and regional models with up to 1.5 km resolution. The database provides hourly data on various weather parameters, including temperature, relative humidity, dew point, precipitation, wind speed, and wind direction. By providing the latitudes and longitudes of the substations, we collect forecast data for each substation based on its proximity to the nearest weather model. Our focus has been on analyzing forecast information during the mentioned power outages resulting in the collection of 3,575,712 observations.
4. **Vegetation:** The landscape of a geographic location is a critical factor in predicting power outages. To account for this, we collected vegetation data from the Ecological Mapping Systems (EMS), which offers a comprehensive land cover summary for Texas. The EMS data has a spatial resolution of 10 meters per map [29].

3.2 Data Preprocessing

To assess the impact of the spatial-temporal multiplex network on predicting the duration of power outages, we use individual and a combination of modalities. These features are gathered from various sources, including weather, forecasts, vegetation, and distance between the substations. Due to the wide range of features, our feature selection eliminates irrelevant information and identifies the most relevant features that enhance the model's performance. In this process, the weather dataset includes the following features: weather station location, sky conditions, visual obstructions, pressure, ambient temperature, wind, and precipitation. We calculate the maximum, minimum, mean, and standard deviation for each selected weather feature per day and time window for each substation. The forecast dataset includes the following features: temperature, relative humidity, dew point, precipitation, wind speed, and wind direction. We also calculate the

maximum, minimum, mean, and standard deviation for each selected weather feature per day and time window for each substation.

3.3 Spatiotemporal Multiplex Network Creation

Consider G as a Spatiotemporal Multiplex Network represented by $G = (V, E, L, T)$, where $V = \{v_1, v_2, \dots, v_n\}$ denotes the set of vertices corresponding to substations, $E = \{e_1, e_2, \dots, e_m\}$ represents the set of edges, $L = \{l_1, l_2, l_3, l_4, l_5\}$ is the set of layers, and $T = \{t_1, t_2, \dots, t_k\}$ is the set of time steps. The network comprises five distinct layers: l_1 corresponds to the power outage layer, l_2 is the weather layer, l_3 represents the forecast layer, l_4 is the vegetation layer, and l_5 refers to the distance layer. In the multiplex network, all vertices across different layers represent the same entities but reflect various types of interactions among the vertices. Specifically, the vertices V represent substations. The connections between the vertices illustrate different types of interactions, with each edge e assigned a weight that is a real number greater than or equal to 1. This weight indicates the strength of the connection between vertices (v, u) in the i^{th} layer. The set of edges E in each layer l_i signifies a different type of relationship among the vertices V , as explained below:

1. **Power outage layer (E_{l_1}):** If a power outage is reported in both substations on the same date and time window, the vertices (u_{l_1}, v_{l_1}) are connected. The edge weight ω represents the shared power outages between these substations.
2. **Weather layer (E_{l_2}):** We assume that power outages are closely linked to weather conditions. As a result, vertices (u_{l_2}, v_{l_2}) are connected if they reveal similar weather properties. To quantify this similarity, we use Euclidean distance. By measuring the Euclidean distance between the weather features, we can determine the closeness of the weather conditions between the substations.
3. **Forecast layer (E_{l_3}):** Vertices (u_{l_3}, v_{l_3}) are connected if they share similar weather properties, since weather conditions have a significant impact on power outages. In this context, we use Euclidean distance to measure the similarity between the weather attributes of the two vertices.
4. **Vegetation layer (E_{l_4}):** We connect two vertices (u_{l_4}, v_{l_4}) (substations) if they exhibit similar vegetation properties. To assess this similarity, we compute the Euclidean distance between their corresponding vegetation features. This method helps illustrate the potential influence of vegetation on power outages, as comparable vegetation patterns may lead to similar outage behaviors.
5. **Distance layer (E_{l_5}):** Two vertices (u_{l_5}, v_{l_5}) (substations) are connected based on their spatial distance. To quantify this distance, we use the Euclidean distance. This measure helps capture the physical proximity between substations, which is essential as the spatial closeness of substations may influence the likelihood of shared power outage events or other related factors.

This graph serves as input for the Spatiotemporal Multiplex Network (SMN-WVF) model. At the end of each day, a new snapshot of the multiplex graph is

generated for each time window, timestamped with that day’s data, to capture the interdependencies among substations. We train the model using these daily snapshots as input, generating embeddings for the substation nodes through the proposed method, which is detailed in the following subsection. The aim is to predict the duration of a power outage at a substation and connected feeders.

3.4 Proposed Model: Spatiotemporal Multiplex Network (SMN-WVF)

This study investigates the potential of the Spatiotemporal Multiplex Network (SMN-WVF) combined with multi-modal data to improve the prediction of disruption severity expressed as the outage duration. Initially, a historical outage dataset from a Texas utility company is utilized, where outage frequency varies across different substations. Each data point is marked as 1 if a power outage occurs in a specific substation during a specific time. Next, the outages are categorized into three groups to predict outage duration, as detailed in Table 1. The model aims to predict the duration of a power outage once it has been identified. It classifies outages into predefined duration categories, as shown in Table 1. This classification aids in estimating the duration of each outage, thereby helping to understand disruption severity and improve planning and response. To enhance predictions, the model employs a modified version of Node2Vec [14] to generate substation node embeddings that incorporate multiple graph layers. We combine the structured data with unstructured data (multiplex model embeddings), creating a unified input dataset for predicting outage duration. This combined input is then processed through a Bidirectional Long Short-Term Memory (BiLSTM) network with eight layers.

Table 1. Distribution of power outage durations across three classes of duration from January 2018 to December 2023.

Class	Duration	Percentage
Class 1	Less than 1 hour	35.1%
Class 2	1 to 3 h	34.9%
Class 3	Greater than 3 h	29.9%

4 Experimental Setup

This study examines whether the multiplex network combined with a multi-modal data approach can enhance the early classification of disruption severity (duration) into one of three categories, ranging from short to very long durations. The study covers six years of historical outage data from January 1, 2018, to December 31, 2023. The training data spans from January 1, 2018, to December

31, 2021, while the testing data covers January 1, 2022, to December 31, 2023. We compare the performance of the Spatiotemporal Multiplex Network model using weather, vegetation, and forecast features. The model is optimized using the Adam optimizer [16], employing a batch size 32 and a learning rate 0.0005. The model is trained with sparse categorical cross-entropy loss, which is used for multiclass classification to improve its ability to predict outage durations across the specified classes.

The study uses a supervised machine-learning approach to predict the duration of disruptions caused by power outages. Since our classification task involves three distinct classes, we employ macro-averaging, which gives equal importance to all classes, regardless of their frequency in the dataset. Specifically, we calculate macro precision and macro recall to measure the model's false positives and false negatives rates. We evaluate the model's overall performance using the macro F1 scores for each class. Here, C denotes the total number of classes, and the macro F1 score is defined as:

$$\text{Macro F1} = \frac{1}{C} \sum_{i=1}^C 2 \cdot \frac{\text{precision}_i \cdot \text{recall}_i}{\text{precision}_i + \text{recall}_i} \quad (1)$$

5 Spatiotemporal Multiplex Network Analysis

This section presents the results of the topological structure analysis conducted on the constructed weighted multiplex networks. To gain a comprehensive understanding of these networks, we examine various centrality measures, including Degree Centrality (DC), Closeness Centrality (CC), Eigenvector Centrality (EC), Square Clustering (SCF), and the Clustering Coefficient (CF), to further explore the network's structural properties. Here, we report the average values of all these measures.

Table 2. Multiplex network topological structure. Here L = number of layer, V = number of nodes, E = number of total edges, *coupling E* = number of coupling edges, $avg(DC)$ = average degree centrality, $avg(CC)$ = average closeness centrality, $avg(EC)$ = average eigenvector centrality, $avg(SCF)$ = average square clustering, $avg(CF)$ = average clustering coefficient.

L	V	E	<i>coupling E</i>	$avg(DC)$	$avg(CC)$	$avg(EC)$	$avg(SCF)$	$avg(CF)$
5	430	14.9M	1,720	161.57	0.484	0.0169	0.482	0.898

Degree Centrality (DC) measures the connectivity of a node based on the number of edges connected to it, which provides insight into the node's significance and potential role as a hub in the network. Nodes with a higher degree of centrality are considered more central, as they are more connected than others. It is calculated as follows:

$$C_D(G) = \frac{\sum_{i=1}^{|V|} [C_D(v^*) - C_D(v_i)]}{|V|^2 - 3|V| + 2}, \quad (2)$$

where v represents a vertex in the graph G . On the other hand, Closeness Centrality (CC) reflects how close a node is to all other nodes within the network. A node with the shortest overall distance to other nodes has a high Closeness Centrality, making it a key candidate for spreading information. We calculate CC as:

$$C(v) = \frac{N - 1}{\sum_u d(u, v)}, \quad (3)$$

where N refers to the total number of nodes in the graph, and $d(u, v)$ represents the distance between the vertices u and v . Eigenvector Centrality (EC) considers the significance of a node's neighbors. In addition, it determines a node's centrality based on the centralities of its neighboring nodes. EC is shown in the following equation as follows:

$$x_v = \frac{1}{\lambda} \sum_{u \in M(v)} x_u = \frac{1}{\lambda} \sum_{u \in G} a_{v,u} x_u, \quad (4)$$

where $A = (a_{v,u})$ is the adjacency matrix of the graph G , $M(v)$ denotes the set of neighbors of node v , and λ is a constant. The Clustering Coefficient (CF) reflects the average number of edges between nodes within each node's neighborhood. Following is the calculation of the Clustering Coefficient:

$$\bar{C} = \frac{1}{n} \sum_{i=1}^n C_i, \quad (5)$$

Finally, Square Clustering (SCF) extends the traditional Clustering Coefficient by focusing on the likelihood that two neighboring nodes share a common neighbor that is not part of the original node's neighbors, thus forming a square-shaped connection [17]. It can be calculated as:

$$C(v) = \frac{\sum_{u=1}^{k_v} \sum_{w=u+1}^{k_v} q_v(u, w)}{\sum_{u=1}^{k_v} \sum_{w=u+1}^{k_v} [a_v(u, w) + q_v(u, w)]}, \quad (6)$$

where $q_v(u, w)$ denotes the number of common neighbors shared by u and w , excluding v . Table 2 presents a detailed network analysis. The graph consists of 430 nodes and 14,902,735 edges. A key characteristic of a multiplex network is the presence of coupled edges (*coupling E*), which represent the transitions of nodes between neighboring layers [28]. Networks with more coupled edges are generally denser and exhibit richer connectivity than those with few or no coupled edges. In this network, there are 1,720 coupling edges. The graph contains 86 unique nodes, with an average degree centrality of 161.57, indicating a well-connected network. The average closeness centrality is 0.484, reflecting the

average proximity between nodes in the graph. Additionally, the average eigenvector centrality is 0.0169, which suggests that nodes with higher centrality are spread out in the network. The average square clustering is 0.482, indicating moderate clustering. The clustering coefficient is 0.898, signifying a highly interconnected overall structure.

6 Results and Discussion

The results for different evaluation metrics of the proposed Spatiotemporal Multiplex Network (SMN-WVF) model using weather, vegetation, and forecast features are shown in Table 3. We emphasize the effectiveness of incorporating multi-modal learning to improve predictive performance. Each variant of the SMN model explores the contribution of specific feature combinations to the overall results. SMN-WV is the proposed model that incorporates both weather and vegetation data. In addition, SMN-WF employs both weather and forecast data. Lastly, the SMN-WVF model combines weather, vegetation, and forecast data for its predictions.

Table 3. Comparison of macro precision, macro recall, and macro F1 score of the proposed Spatiotemporal Multiplex Network (SMN-WVF) model using weather (W), vegetation (V), forecast (F) features, and the Spatiotemporal Multiplex Network. The outage duration is classified into three classes, as detailed in Table 1.

Modality	Macro precision	Macro recall	Macro F1 Score
SMN-WF	0.42	0.40	0.41
SMN-WV	0.43	0.42	0.42
SMN-WVF	0.45	0.41	0.43

We can observe that incorporating weather and vegetation (SMN-WV) yields slightly better results, with a macro F1 score of 0.42 compared to weather and forecast (SMN-WF). This suggests that vegetation can play a critical role in power outages, particularly due to incidents such as trees falling onto or touching overhead lines due to wind impacts, which are significant causes of power faults. Furthermore, the SMN-WVF model improves prediction performance by up to 2% compared to alternative models. The SMN-WVF model demonstrates strong predictive capability, consistently outperforming alternative models despite the inherent challenges of outage prediction in a small distribution grid area. Notably, it achieves a macro F1 score of 0.43. Unlike transmission networks, where outages often follow large-scale, high-impact events, distribution grids present a significantly harder prediction task due to their localized nature, smaller coverage area, and complex outage drivers. Outage durations in distribution grids are influenced by a mix of highly complex localized factors, and relying significantly on regional weather data makes accurate outage prediction

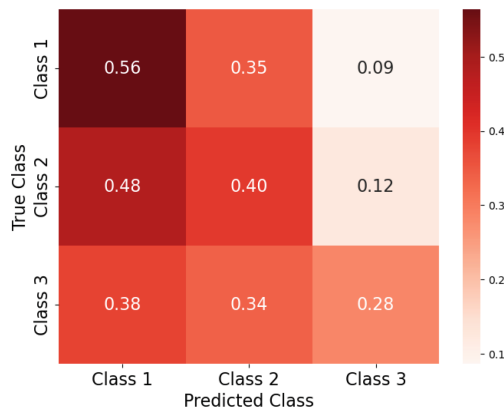


Fig. 2. The normalized confusion matrix of the SMN-WVF model, which utilizes weather, vegetation, and forecast features along with the Spatiotemporal Multiplex Network, demonstrates its predictions for outage durations. The classification is defined as follows: Class 1 represents duration from 30 min to 3 h, Class 2 represents duration from 3 and 6 h, and Class 3 represents duration greater than 6 h.

even more difficult. The multi-modal approach, integrating weather, vegetation, and forecast features, helps mitigate these challenges by capturing additional contextual information beyond weather and forecast features alone. Given the small-scale nature of the study area, even achieving a moderate macro F1 score indicates that the model is successfully generalizing across different outage conditions rather than overfitting to frequent outage patterns.

Table 4. Comparison of precision, recall, and F1 score of every class using SMN-WVF model.

Class	Precision	Recall	F1 Score
Class 1	0.44	0.56	0.49
Class 2	0.37	0.40	0.38
Class 3	0.52	0.28	0.37

We evaluate the performance of the SMN-WVF model using a confusion matrix. Figure 2 provides a breakdown of the correct and incorrect predictions. The classification categories are defined as follows: Class 1 represents the duration from 30 min to 3 h, Class 2 represents the duration from 3 and 6 h, and Class 3 represents the duration greater than 6 h. We further analyze the predictions for each class. Table 4 details the results of the SMN-WVF model per class. The results show that Class 1 achieves the highest recall (0.56) and a moderate F1 score (0.49), indicating relatively better identification of these instances. Class 2 has the lowest precision (0.37) and recall (0.40), resulting in an F1 score of 0.38.

Class 3 exhibits the highest precision (0.52) but suffers from low recall (0.28), leading to a reduced F1 score of 0.37. These results highlight the model’s tendency to favor precision over recall, particularly in Class 3, potentially leading to underestimation of certain outage durations.

The second experiment assesses the effectiveness of the SMN model using different SMN-WVF models in making predictions at earlier stages. In the context of power systems, earlier predictions yield greater benefits. Table 5 illustrates the macro F1 score achieved for the early prediction scenarios. This score reflects the ability of the SMN model to detect outages in advance. We can observe that the macro F1 score remains stable and yields similar results as the prediction time approaches the outage event.

Table 5. Performance of the proposed Spatiotemporal Multiplex Network (SMN-WVF) model using macro precision, macro recall, and macro F1 score for early outage detection, evaluated across a three-class problem formulation, as detailed in Table 1.

Lead time of power outage	Macro precision	Macro recall	Macro F1 Score
0 h	0.45	0.41	0.43
6 h	0.43	0.41	0.42

7 Conclusion

Power outages pose serious threats to residential, commercial, and industrial customers, as well as transportation, healthcare, communication, and other essential services. Making effective predictions of their occurrences and duration is paramount for strategic outage mitigation planning. This study introduces the Spatiotemporal Multiplex Network with Weather-Vegetation-Forecast (SMN-WVF) method, a novel approach designed to improve power outage duration predictions. By integrating multi-modal data and network structures, our model evaluates outage durations across different time horizons, offering insights with various lead times. Even with the challenges of geographically confined space and the difficulties of predicting the outage duration, particularly in distribution grids—which are more intricate and finer-grained than transmission line outages—our SMN-WVF method demonstrates practical results. Our multiplex network comprises multiple layers, including power outages, weather conditions, weather forecasts, vegetation properties, and substation connectivity, allowing us to capture both temporal and spatial dependencies in the outage process. We rigorously assess whether our innovative network approach enhances predictive accuracy compared to simple weather data inputs. Our method achieves a macro F1 score of 0.42, underscoring both the difficulty of the task and the need for further advancements in outage duration predictions. We observe that integrating vegetation and forecast enhances performance by providing additional context beyond weather conditions alone. These findings highlight the importance

of incorporating diverse data sources to improve outage duration predictions. Future work will focus on refining spatial feature representation and expanding alternative data sources. Also, it is worth exploring model performance while looking into node representations as a substation grouping. Looking ahead, we will evaluate our model's performance across diverse geographic regions and explore its efficacy with extended prediction intervals.

Acknowledgments. This work was funded by the National Science Foundation Award # 2125985.

References

1. Aljurbua, R., Alshehri, J., Alharbi, A., Power, W., Obradovic, Z.: Social media sensors for weather-caused outage prediction based on spatio-temporal multiplex network representation. *IEEE Access* (2023)
2. Aljurbua, R., Alshehri, J., Gupta, S., Alharbi, A., Obradovic, Z.: Early prediction of power outage duration through hierarchical spatiotemporal multiplex networks. In: *Complex Networks & Their Applications XIII: Proceedings of The Thirteenth International Conference on Complex Networks and Their Applications: COMPLEX NETWORKS 2024*, vol. 3, p. 320. Springer (2024)
3. Aljurbua, R., Gillespie, A., Alshehri, J., Alharbi, A., Albarakati, N., Obradovic, Z.: Node2vecfuseclassifier: bridging perspectives in modeling transplantation attitudes among dialysis patients. In: *2024 IEEE 12th International Conference on Healthcare Informatics (ICHI)*, pp. 113–122. IEEE (2024)
4. Aljurbua, R., Gillespie, A., Obradovic, Z.: The company we keep. Using hemodialysis social network data to classify patients' kidney transplant attitudes with machine learning algorithms. *BMC Nephrol.* **23**(1), 414 (2022)
5. Alqudah, M., Obradovic, Z.: Enhancing weather-related outage prediction and precursor discovery through attention-based multi-level modeling. *IEEE Access* (2023)
6. Automated Surface Observing Systems: NOAA's National Weather Service. <https://www.weather.gov/asos/asostech>. Accessed Aug 2024
7. Azad, S., Ghandehari, M.: A study on the association of socioeconomic and physical cofactors contributing to power restoration after hurricane maria. *IEEE Access* **9**, 98654–98664 (2021)
8. Baembitov, R., Karmacharya, A., Kezunovic, M., Saranovic, D., Obradovic, Z.: Effect of lightning features on predicting outages related to thunderstorms in distribution grids. In: *Proceedings of the 58th IEEE Hawaii International Conference on System Science (HICSS)*, pp. 2978–2987. IEEE (2025)
9. Baembitov, R., Kezunovic, M., Saranovic, D., Obradovic, Z.: Sensitivity analysis of machine learning algorithms for outage risk prediction. In: *Proceedings of the 57th IEEE Hawaii International Conference on System Science (HICSS)*, pp. 3150–3159. IEEE (2024)
10. Bompard, E., Huang, T., Wu, Y., Cremenescu, M.: Classification and trend analysis of threats origins to the security of power systems. *Int. J. Electr. Power Energy Syst.* **50**, 50–64 (2013)
11. Dokic, T., Pavlovski, M.: Spatially aware ensemble-based learning to predict weather-related outages in transmission. In: *The Hawaii International Conference on System Sciences–HICSS, Maui, Hawaii, January 2019* (2019)

12. Eskandarpour, R., Khodaei, A.: Machine learning based power grid outage prediction in response to extreme events. *IEEE Trans. Power Syst.* **32**(4), 3315–3316 (2016)
13. Fink, L.H., Carlsen, K.: Operating under stress and strain [electrical power systems control under emergency conditions]. *IEEE Spectr.* **15**(3), 48–53 (1978)
14. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 855–864 (2016)
15. Hou, H., Liu, C., Wei, R., He, H., Wang, L., Li, W.: Outage duration prediction under typhoon disaster with stacking ensemble learning. *Reliab. Eng. Syst. Safety* **237**, 109398 (2023)
16. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014), <https://arxiv.org/abs/1412.6980>
17. Lind, P.G., Gonzalez, M.C., Herrmann, H.J.: Cycles and clustering in bipartite networks. *Phys. Rev. E—Stat. Nonlin. Soft Matter Phys.* **72**(5), 056127 (2005)
18. Liu, H., Davidson, R.A., Apanasovich, T.V.: Statistical forecasting of electric power restoration times in hurricanes and ice storms. *IEEE Trans. Power Syst.* **22**(4), 2270–2279 (2007)
19. Mensah, A.F., Dueñas-Osorio, L.: Outage predictions of electric power systems under hurricane winds by bayesian networks. In: *2014 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, pp. 1–6. IEEE (2014)
20. Mitsova, D., Escaleras, M., Sapat, A., Esnard, A.M., Lamadrid, A.J.: The effects of infrastructure service disruptions and socio-economic vulnerability on hurricane recovery. *Sustainability* **11**(2), 516 (2019)
21. Nateghi, R., Guikema, S.D., Quiring, S.M.: Comparison and validation of statistical methods for predicting power outage durations in the event of hurricanes. *Risk Anal. Int. J.* **31**(12), 1897–1906 (2011)
22. Nateghi, R., Guikema, S.D., Quiring, S.M.: Forecasting hurricane-induced power outage durations. *Nat. Hazards* **74**(3), 1795–1811 (2014). <https://doi.org/10.1007/s11069-014-1270-9>
23. Open-Meteo: Open-meteo. <https://open-meteo.com/>. Accessed Sep 2024
24. Otudi, H., Gupta, S., Obradovic, Z.: Leveraging diverse data sources for enhanced prediction of severe weather-related disruptions across different time horizons. In: *International Conference on Engineering Applications of Neural Networks*, pp. 220–234 (2024)
25. Owerko, D., Gama, F., Ribeiro, A.: Predicting power outages using graph neural networks. In: *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pp. 743–747. IEEE (2018)
26. Reed, D.A.: Electric utility distribution analysis for extreme winds. *J. Wind Eng. Ind. Aerodyn.* **96**(1), 123–140 (2008)
27. Robusto, C.C.: The cosine-haversine formula. *Am. Math. Mon.* **64**(1), 38–40 (1957)
28. Škrlj, B., Renoust, B.: Layer entanglement in multiplex, temporal multiplex, and coupled multilayer networks. *Appl. Netw. Sci.* **5**(1), 1–34 (2020). <https://doi.org/10.1007/s41109-020-00331-w>
29. Texas Parks and Wildlife Department: Ecological Mapping Systems. <https://tpwd.texas.gov/landwater/land/programs/landscape-ecology/ems/>. Accessed Aug 2024
30. Wong, C.J., Miller, M.D.: Guidelines for electrical transmission line structural loading. American Society of Civil Engineers (2009)
31. Yang, F., Wanik, D.W., Cerrai, D., Bhuiyan, M., Anagnostou, E.N.: Quantifying uncertainty in machine learning-based power outage prediction model training: a tool for sustainable storm restoration. *Sustainability* **12**(4), 1525 (2020)

Machine Learning/Time Series



AI-Based Water Quality Monitoring Module for the Segura Hydrographic Confederation Platform

Izar Azpiroz¹✉, David Velásquez¹, Breno da Costa Paulo¹, Iker Landa del Barrio^{1,2}, Juan Odriozola¹, and Mikel Maiza¹

¹ Fundación Vicomtech, Basque Research and Technology Alliance (BRTA), Mikeletegi 57, 20009 Donostia-San Sebastián, Spain

iazpiroz@vicomtech.org

² School of Architecture, University of Navarra, 31009 Pamplona, Spain

Abstract. Efficient water management is essential in regions facing persistent imbalances between water supply and demand. The Segura Hydrographic Confederation has developed a digital twin platform integrating Artificial Intelligence (AI) tools for efficient water resource management. This study presents an AI-based water quality monitoring module integrated in this platform, that explores three modeling approaches: ML-based water quality predictions, DL-based forecasting, and surrogate modeling. The study demonstrates that AI-enhanced water quality monitoring can support decision-making processes.

Keywords: Water Quality Monitoring · Digital Twin · Artificial Intelligence · Hydrological Prediction

1 Introduction

Water is a natural resource that is crucial to several economic activities [1]. As water demand increases due to economic and population growth, its availability decreases, generating an imbalance between supply and demand [2]. This situation makes necessary to efficiently manage available water resources [3]. One such region facing this challenge is the Segura Hydrographic Basin, located in southeastern Spain, where the imbalance between water supply and demand is particularly evident. Although it has favorable climatological conditions, it is characterized by a scarcity of precipitation [4]. It is an environment with considerable agricultural activity [5], which is reflected in water demand, as about 87% of the resources are used exclusively for agricultural practices [6]. The mismatch between supply and demand has intensified in the last three decades, as water demand has systematically exceeded availability. As a result, a persistent water deficit has been generated, aggravated by both natural and human factors, including intensive exploitation and recurrent droughts.

1.1 Literature Review

Hydrologic decision-support systems have become essential for the effective management of water resources, providing critical estimations and predictions of hydrological and environmental conditions [2,7,8]. Their role is particularly relevant in semi-arid regions like the Segura Basin, characterized by water scarcity exacerbated by climatic variability and intensive agriculture [4,5].

Recent advances emphasize artificial intelligence (AI) and digital twin technologies to enhance operational efficiency and prediction accuracy [3]. The Segura Hydrographic Confederation (CHS) developed a digital twin platform, HidroTwin [9], integrating hydrological simulations, optimization tools, and intuitive visual interfaces for sustainable management. Within this framework, machine learning (ML) algorithms such as XGBoost and Random Forest have demonstrated robust predictive capabilities for water quality parameters [3]. Deep learning (DL) methods, including LSTM and CNN architectures, have also successfully addressed complex time-series forecasting tasks in hydrological applications [3]. Complementing these approaches, surrogate models trained on data from physical simulations have emerged as efficient tools, significantly reducing computational demands while maintaining predictive accuracy [3].

This research integrates these three AI methodologies—ML-based prediction, DL-based forecasting, and surrogate modeling—within the CHS platform, aiming to optimize water resource management decisions in the Segura Basin.

2 Materials and Methods

The Digital twin [9] of CHS is an innovative platform for decision support in water management, developed using advanced AI technologies. The present study has been financed as part of the CHS project, funded by the Center for Technological Development and Innovation, a Public Business Entity under the Spanish Ministry of Science, Innovation, and Universities.

2.1 Water Quality Data Description

The Automatic Water Quality Information System (SAICA) [10] is a nationwide Spanish system designed for continuous, real-time monitoring of inland surface water quality. It represents a significant advancement in the ongoing control of water quality, with monitoring stations strategically positioned at critical points—either due to their susceptibility to pollution discharges or their designation as protected areas, some of which include water catchments for drinking water production.

SAICA operates nine Water Quality Stations (WQSSs) integrated into the SAIH system, located at Contraparada, San Antón, Archena, Azud Ojós, Cieza, Mundo Azaraque, Azud Cenajo, Los Huertos, and Sifón Orihuela. Due to logistical and budgetary constraints, stations installed in the first phase were inactive during two periods: from October 2010 to May 2011 and from February 2016 to September 2017.

Part of data used in this study, provided as part of the Digital twin of the CHS project, covers 2015 to 2021 with hourly frequency measurements. A gap in the data exists between 2016 and 2018, corresponding to the previously mentioned inactive phase. Water quality variables from these stations include *Water Temperature*, *pH*, *Conductivity*, *Dissolved oxygen*, *Turbidity*, *Ammonium*, *SAC*, *Nitrates*, and *Phosphates*. For simplicity, this study considers only the results for *pH* and *Water Temperature* predictions, as well as *pH* forecasting, in Water Quality ML-DL applications.

To analyze water quality, meteorological data has also been examined. For completeness, the *Water Temperature* recorded at Water Quality Stations (WQSs) has been synchronized with meteorological data from nearby weather and rainfall stations. This process requires the geographical identification of the nearest meteorological and pluviometric stations based on their proximity. This association was established using the Haversine distance [11] method to determine the nearest station.

For *Water Temperature* predictions, the influence of temporal variables was included due to climatic seasonality. The inclusion of these variables resulted in the following final feature set: *Air Temperature*, *Humidity*, *Rainfall*, *Day*, *Month*, *DayYear*, and *WeekYear*.

In addition to the aforementioned data, there was provided the access to GESCAL, which is a simulation software designed for modeling the quality of water in hydrological systems [12]. GESCAL is implemented for large-scale water quality management, simulating the evolution of water quality based on various management, purification, pollution, and resource-use scenarios.

The main purpose of GESCAL is to model water quality in rivers and reservoirs, considering both physical-chemical and biological processes that affect water quality. The tool allows the simulation of multiple water quality constituents such as temperature, oxygen, nitrogen (organic, ammonium, and nitrates), phosphorus (organic and inorganic), and others. GESCAL integrates these processes with the hydraulic conditions of the system, including inflows, outflows, and sediment flows to generate a detailed simulation of water quality changes across a system.

From the GESCAL output, initially existed 696 water-quality variables (e.g., *pH*, *Conductivity*, etc.) measured across multiple stations, each station effectively functioning as an “output variable” of an specific location. On the input side, there were corresponding data for 43 “upstream” or “input” stations, each also providing monthly measurements of the same set of chemical parameters. In total, the underlying data spanned from 1999 to 2012, with approximately 156 monthly measurements per variable. The goal is to predict the water-quality variable in the output locations based on the input locations.

Given that running the full GESCAL simulations is computationally expensive, the idea is to develop a surrogate model that can replicate the results of GESCAL with minimal error but significantly reduced computational cost. Once trained, this surrogate model can provide predictions nearly instantaneously, thus removing the need for the costly computational effort required tool.

2.2 GESCAL Data Preprocessing

In large-scale hydrological systems, physical distance does not always indicate actual upstream–downstream influence; an input station might appear distant on a map yet still contribute significantly to the water quality at a particular output station. To address this complexity, it was applied correlation-based analysis to identify which inputs truly drive each output. That way, if an input station is in fact upstream (e.g., due to aquifer flow, diversion channels, or river branches), its signals (like changes in dissolved oxygen or conductivity) will correlate strongly with conditions at the output. By relying on measured data rather than purely spatial assumptions, it is possible to find out automatically which input variables matter most for each output, even if they are physically far apart.

Before training, the available data was cleaned to address issues like:

- **Outlier detection and removal:** Certain measurements displayed abnormal high values (e.g., ammonium or nitrates up to an order of magnitude larger than typical), and they could insert bias in the model. These were filtered so they would not skew the models.
- **Low-variance columns:** Many input or output stations had columns with nearly constant readings; these provide no meaningful signal for model training and were removed.
- **Correlation-based feature selection:** For each of the 696 output variables, their correlation with all possible input columns were computed. Only the subset of input columns exceeding a minimum correlation threshold—typically 50% of the highest observed correlation—was retained for each model. This procedure was particularly useful in ensuring that each output was paired only with its most informative inputs.

After cleaning, only about 270 of the 696 output variables had sufficient variance and correlation to support training a robust surrogate model. These 270 variables could be modeled effectively, and their results are explored in the next section. The remaining outputs were either constant values or had no meaningful correlation with any of the input variables, meaning that creating a meaningful model relating input and output for these cases would be nonsensical.

2.3 Evaluation Metrics

The efficiency and accuracy of distinct Water Quality models are measured using the following evaluation metrics: Coefficient of Determination $R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$, Relative Energy Error (REE) $\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n y_i^2}$, Mean Squared Error (MSE) $\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$, Mean Absolute Error (MAE) $\frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|$, Root Mean Squared Error (RMSE) $\sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}$, and Root Mean Squared Logarithmic Error (RMSLE) $\sqrt{\frac{1}{n} \sum_{i=1}^n [\log(\hat{y}_i + 1) - \log(y_i + 1)]^2}$, where y denote real observations, $\hat{y}$ the corresponding predictions and $\bar{y}$ represent the mean value.

2.4 IA-Based Water Quality Module

Effective water quality management in river basins requires robust modeling approaches to assess, predict, and mitigate potential environmental risks. Given the complexity of hydrological and ecological interactions, selecting an appropriate modeling strategy is crucial for accurate assessments and informed decision-making. To address these challenges, three distinct methodologies have been explored and implemented:

1. ML-based pH water quality predictions.
2. DL-based pH water quality forecasting.
3. Surrogate modeling of the GESCAL physical model.

Water pH and Temperature Prediction Models: The selected data for each module were first partitioned into training and validation sets, with 70% allocated for training and 30% for validation. The Automated Machine Learning EvalML library was then utilized to identify the most significant variables and to develop optimal ML forecasting models. In this section, two water quality parameters were studied. On the one hand, the **pH Predictor**. The goal of this model is to develop a regressor capable of predicting *pH* levels based on all other available water quality variables. On the other hand, the **Water temperature Predictor**. This model aims to estimate *Water Temperature* using data from meteorological and pluviometric stations. By incorporating these external environmental factors, the model enhances the accuracy of *Water Temperature* predictions. The performance of ML models is assessed using the R^2 , REE, MSE and MAE evaluation metrics.

pH Water Quality Forecasting Models: Different algorithms and libraries specialized in time series analysis for forecasting were tested, including the EvalML AutoMLSearch Time Series module, the PyCaret Time Series module, and TensorFlow. After extensive testing and model optimization, TensorFlow was found to deliver the best forecasting results. The outcomes presented in the Result section correspond to the TensorFlow models detailed in Table 1.

The selected data are then separated into training, validation and test. These are divided into 75%, 12.5%, and 12.5% respectively. Time windows of 30 days of history were then created for 15 days of prediction for all variables in the dataset. For validation purposes, only the **pH Forecaster** has been evaluated. The accuracy of the models is compared using MAE, REE and MSE accuracy metrics.

GESCAL Surrogate Model: A surrogate model is an ML-based model trained on data originally generated by a calibrated physical model. Once built, the surrogate provides near-instantaneous predictions similar to the results that would be obtained from the original model, making them useful when the corresponding counterpart is computationally expensive. In this case, GESCAL serves

Table 1. Comparison of Model Architectures

Model	Architecture
Dense Model	Feedforward NN
Dense + Dropout	Feedforward NN + Dropout
LSTM Model	Single LSTM layer (32 units)
CNN Model	1D Conv (256 filters, kernel = 3)
Advanced DNN	Dense (128x2) + Dropout
Advanced CNN	2 Conv1D + MaxPooling
Advanced LSTM	2 LSTM layers (72, 48 units)

as the reference simulation model provided by the CHS which has the benchmark of data. Based on a substantial history of previously executed simulations and in historical data. The surrogate model aims to replace the behaviour of the software GESCAL in predicting water-quality variables. This historical data enabled the development of a prototype surrogate model that achieves promising performance.

The Python library *PyCaret* for automated machine learning (AutoML) was chosen as the tool to be used. PyCaret systematically tests and tunes various regression algorithms—such as Decision Trees, Random Forest, XGBoost, and others—to find an optimal model for each target variable. The steps for each surrogate model were:

1. **Data Assembly:** Combine the selected input variables (based on correlation) and the single output variable into a training set of ~ 140 monthly samples and a validation set of ~ 16 samples (for a total of ~ 156).
2. **Model Selection and Tuning:** Use PyCaret’s AutoML pipeline to compare algorithms, performing hyperparameter optimization for promising candidates.
3. **Model Finalization:** Finalize and save the best-performing model (e.g., an XGBoost regressor or a Decision Tree), logging its performance metrics.

In the following section it is shown how these model were trained to handle each of these variables individually, as the realionships of each input-output for each locations varies significantly.

3 Results

3.1 Data Processing and Analysis

As can be seen in Fig. 1, there is no data between January 2016 and January 2018, so these periods are removed from the dataset. The reduction of continuous historical data to the period from January 2018 onwards implies a significant limitation to build AI models.

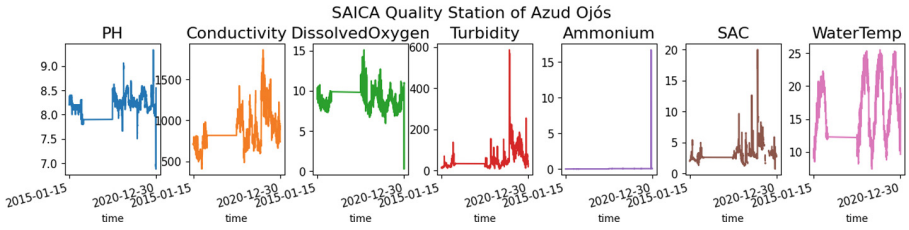


Fig. 1. History of all water quality variables (with and without outliers) for the Azud Ojós station, data period from 2015-01-15 to 2020-12-30.

The boxplots depicted in Fig. 2 illustrate the distribution of various water quality parameters at the Contraparada station, emphasizing the presence of outliers. While *Water temperature* shows minimal anomalies, *pH* exhibits extreme values at both ends. *Conductivity* and *Turbidity* display strong skewness with numerous high outliers, while *Dissolved oxygen* has deviations at very low and high concentrations. *Ammonium* and *SAC* also contain several outliers. These observations indicate significant variability in water quality.

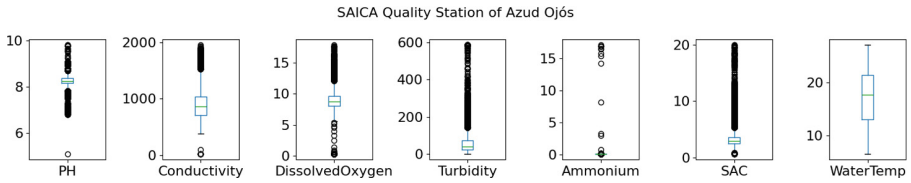
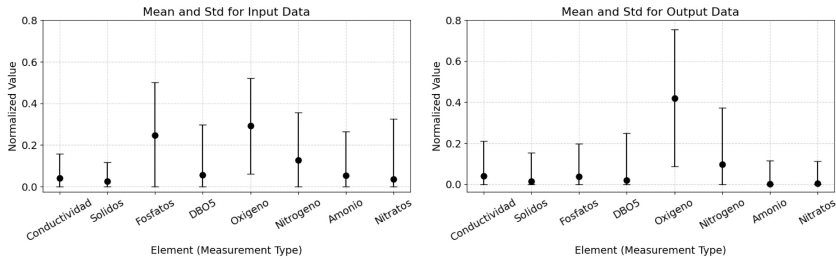


Fig. 2. Outlier plots for all variables at Contraparada station.

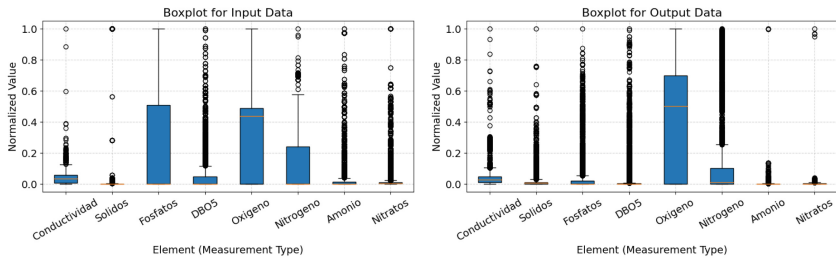
In the analysis phase, tasks such as variability/variance analysis and correlation analysis were conducted. As illustrated in Figs. 3a–3d many of the input variables of the GESCAL model exhibit very low variability, which limits the learning capacity of the machine learning models. The plot exhibited in Figs. 3a and 3b show the mean of each measurement with their respective standard deviation range, for input and output. The plots of Fig. 3c and Fig. 3d show the boxplot of these measurements. In these plots is possible to see this low variance of some elements, specially for the outputs, which are very homogeneous apart from the outliers.

3.2 Prediction Model Performance

This section summarizes the results of *pH* and *Water temperature* prediction models.



(a) Variance in input measurements. (b) Variance in output measurements.



(c) Boxplot of the input measurements separated by measurement type. (d) Boxplot of the output measurements separated by measurement type.

Fig. 3. Variance analysis of water quality parameters.

Table 2. ML pipeline R^2 performance comparison for pH prediction modeling. Cross-validation procedure is applied to obtain accuracy robust results for the R^2 evaluation metric.

Pipeline Name	R^2 mean cv score	Std Dev cv score	% better than baseline
XGBoost Regressor w/ Imputer + Select Columns	0.707987	0.003943	7.810656e+06
LightGBM Regressor w/ Imputer + Select Columns	0.618553	0.004163	6.824025e+06
Random Forest Regressor w/ Imputer	0.578479	0.011658	6.391922e+06
Decision Tree Regressor w/ Imputer + Select Columns	0.556233	0.011132	6.235779e+06
Random Forest Regressor w/ Imputer + RF Regressor	0.552635	0.022667	6.097008e+06
Baseline: Regression Pipeline	—	—	—

pH Predictor: Table 2 presents a summary of the ML models for pH prediction generated by EvalML, along with their respective performance metrics. XGBoost Regressor highlighted as the most effective algorithm from the EvalML library. According to Table 2, the model achieves an R^2 score of 0.707987 across all SAICA stations, which is considered a highly acceptable metric given that it is a single global model applied to all stations. Moreover, the XGBoost Regres-

sor exhibits the lowest standard deviation, indicating that it is the most stable model in terms of efficiency, and the percentage improvement over the baseline is significantly higher at $7 \times 10^6\%$. Further pipeline details are configured in the graph of Fig. 4. Figure 5a illustrates the predicted vs. actual pH values during the validation and test phases. The results demonstrate that the predicted values closely replicate the actual values, indicating strong model performance.

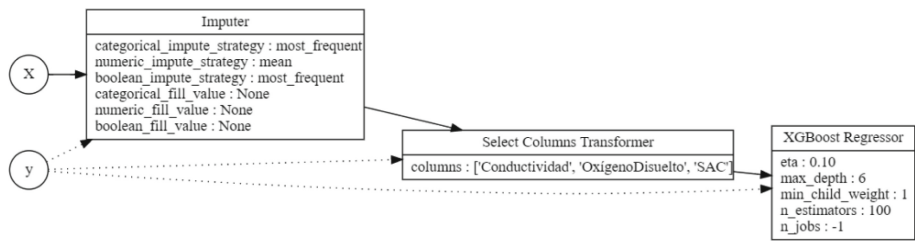


Fig. 4. Pipeline diagram of the best pH predictor model generated by EvalML.

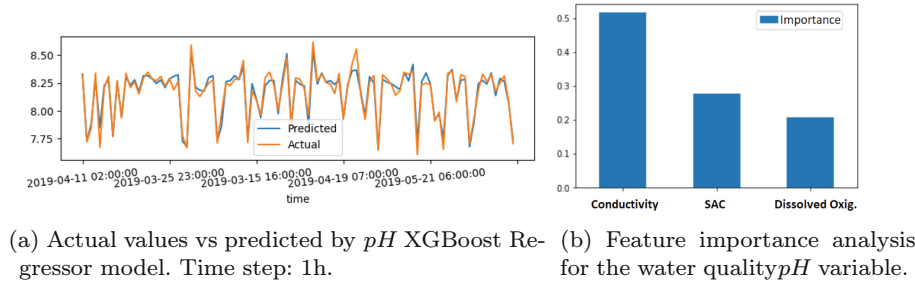


Fig. 5. pH prediction model visualization and feature importance summary. Testing evaluation results: $R^2 = 0.9524$, $REE = 10^{-6}$, $MSE = 0.0025$, $MAE = 0.0331$.

This model is capable of predicting pH at any SAICA station with historical data on *Conductivity*, *Dissolved oxygen*, and *SAC*, as these variables were identified according to importance analysis depicted in Fig. 5 as the most significant during the exploratory data analysis phase.

Water Temperature Predictor: The key variables selected for developing this model include *Air Temperature*, *Humidity*, *Precipitation*, *Day*, *Month*, *Day of the Year*, and *Week of the Year*. The resulting XGBoost Regressor model from EvalML tool is depicted in 6 achieves an R^2 score of 0.9035, indicating a strong predictive capability for *Water temperature*.

Figure 7a compares the predicted and actual *Water temperature* values in the test dataset, demonstrating the model's performance. Additionally, Fig. 7b highlights the most influential variables, showing that *Air Temperature*, *Day of the Year*, and *Humidity* have the highest impact. This confirms the cyclic nature of *Water temperature* variations throughout the year.

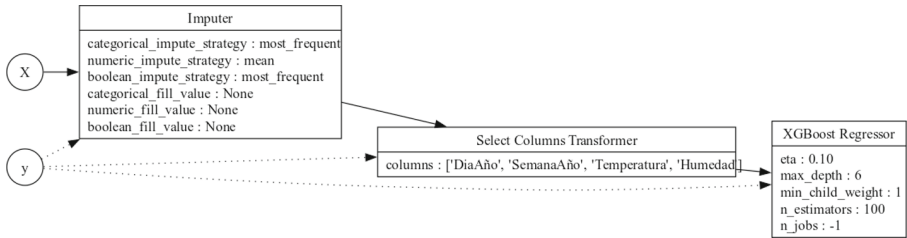


Fig. 6. Pipeline diagram of the best *Water temperature* predictor model generated by EvalML.

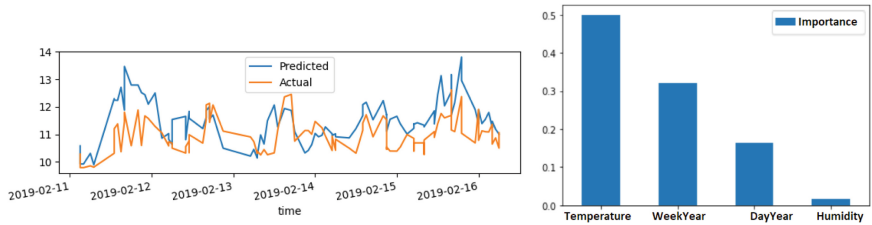


Fig. 7. *Water temperature* prediction model visualization and feature importance summary. Testing evaluation results: $R^2 = 0.9045$, $REE = 0.0066$, $MSE = 2.0384$, $MAE = 1.0806$.

3.3 *pH* Forecasting Model Performance

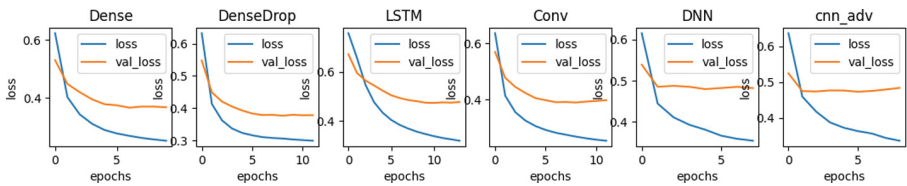


Fig. 8. Comparison of error loss in *pH* forecasting models.

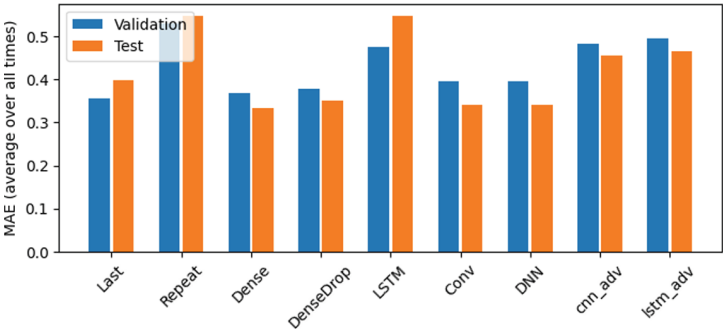


Fig. 9. *pH* forecasting model performance comparison by MAE.

The learning curves for the training and validation sets of each DL model listed in Table 1 are presented in Fig. 8. In this Figure, the LSTM, DNN, and advanced CNN models converge to a error loss value of approximately 0.35, while the Dense Drop model achieves lower loss values around 0.25. Notably, the lowest values correspond to the Dense and CNN models, with the Dense model performing the best, exhibiting slightly lower error loss values.

The performance comparison is more evident in Fig. 9, which illustrates the MAE performance metric of each model. Once again, the similarity in efficiency among the Dense, Dense Drop, CNN, and DNN models is apparent, with the Dense model displaying slightly lower MAE values. In contrast, the LSTM, advanced CNN, and advanced LSTM models demonstrate inferior performance and may be disregarded in future studies. All models, except those derived from LSTM, outperform the baseline models (last and repeated values) in terms of performance metrics.

3.4 GESCAL Surrogate Model

During the model-building process, it has been observed that when the input variables that correlate with the target variable exhibit a sufficiently high variance, the resulting models achieve strong performance metrics, with a value exceeding 0.9. Conversely, when the correlated input variables have very low variance, constructing reliable models becomes challenging.

Metrics such as R^2 , MAE, MSE, and RMSE were tracked on both the training data and held-out validation data. In many cases, R^2 exceeded 0.6–0.7, indicating that, for those variables with sufficient correlation, the surrogate model is reliably capturing the relationship between the physical inputs and the GESCAL-simulated outputs.

The results obtained from the surrogate models are highly promising, particularly in cases where the historical data represents adequate variability.

For example, Fig. 10a shows a scatter plot for a specimen with an R^2 value of 0.991, indicating that the model is very well-correlated, making it easy to train

and predict with high accuracy. The dashed line in the figure represents the fit, clearly illustrating the strong relationship between the predicted and actual values.

Figure 10b provides another example where a water quality variable is compared between simulated (with unseen data from GESCAL) and the result given by the surrogate model. This comparison highlights the effectiveness of the surrogate model in replicating GESCAL’s calculations, showing how the model generalizes the relationships between input and output variables. The area below the measured curve is **142.15**, and the area below the predicted curve is **132.54**, representing an error of **6.7%**, an acceptable result for a model trained with a dataset of around 140 entries.

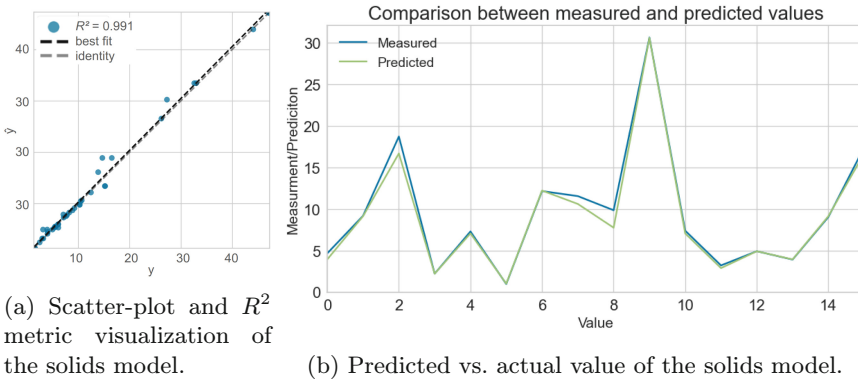


Fig. 10. Model performance for the estimation of solid presence in a specific output.

Apart from the R^2 that appears in Fig. 10a, after training and fine tuning the model, PyCaret returns a summary of the training metrics of the surrogate model, which are: **MAE** = 0.486; **MSE** = 0.676; **RMSE** = 0.822; **RMSLE** = 0.077.

In total, 270 surrogate models were successfully created from the available 696 output variables. These models demonstrate that surrogate modeling can be a viable alternative to running the full GESCAL simulations, as they are able to provide predictions after training. The strong performance of these models, particularly those with high input-output correlation and variance, further underscores their potential as a tool for water quality monitoring and management.

It is concluded that surrogate models present a promising and complementary alternative to traditional simulation software like GESCAL. Experts from the CHS are encouraged to explore these models as predictive tools for water quality, as their use could significantly enhance decision-making and forecasting capabilities within the CHS platform.

4 Discussion and Conclusion

The AI-driven water quality monitoring module developed for the Segura Hydrographic Confederation (CHS) platform demonstrates significant potential in optimizing water resource management. The study explored three key modeling approaches: ML for water quality predictions and forecasting, and surrogate modeling to enhance the computational efficiency of simulation tools.

ML models, particularly XGBoost for *pH* prediction and *Water temperature* prediction, performed well with R^2 values of 0.71 and 0.90, respectively. These models demonstrate the ability to predict water quality variables reliably using available monitoring data. Additionally, the integration of meteorological and time variables into the water temperature predictor enhanced its accuracy.

The comparison of **DL models** for *pH* water quality prediction highlights distinct strengths for each architecture. The Dense model, characterized by its simplicity and efficiency, performs best, achieving the lowest error loss and MAE values. CNNs excel at capturing short-term dependencies through local pattern extraction, while LSTM models are designed to learn long-term dependencies but exhibit inferior performance in this study. Advanced architectures show higher error values, suggesting their complexity does not necessarily translate to better predictive accuracy. Overall, the Dense and CNN models demonstrate the most promising results, making them suitable choices for real-world water quality forecasting applications.

Surrogate models, developed as alternatives to the computationally expensive GESCAL simulations, provide an computationally efficient tool for water quality management. By leveraging historical data from several years, these models replicate the results of the simulations with minimal error, reducing drastically simulation time. Although surrogate models show promising results, their performance is constrained by the quality and variance of the available data. In particular, outputs with low variance or weak correlation with inputs present challenges in model training, leading to the exclusion of several output variables from the modeling process. Nonetheless, the surrogate models that were successfully developed provide reliable predictive capabilities, enough to serve as valuable tools for real-time water quality monitoring, substituting the use of computationally expensive simulators.

5 Future Work

The prediction and forecasting models developed will be extended to the remaining water quality variables. Addressing missing data at certain stations presents a significant challenge, as key variables such as nitrates and phosphates are absent in nearly half of the water quality monitoring stations. Another critical aspect to consider is the comparison between local and global models. In this study, we focus on evaluating the efficiency of global models, particularly in the first two modules. A thorough comparative analysis of local versus global models could yield valuable insights into seasonal variations. Additionally, the geographical distribution of monitoring stations provides crucial contextual information.

Consequently, estimating missing data based on nearby stations—particularly those located upstream or downstream along the river channel—may serve as an effective interpolation methodology.

Acknowledgments. This work was supported by project GEMELO DIGITAL CHS (IN3210/2021), financed by the CDTI-CPI-2020 program of the Centro para el Desarrollo Tecnológico y la Innovación (CDTI) and run in collaboration with Confederación Hidrográfica del Segura (CHS), Grusamar, DHI, Meteobit, and Fundación Vicomtech, member of Basque Research and Technology Alliance (BRTA).

Data Availability. Part of the data will be made available on request.

References

1. Niva, V., Cai, J., Taka, M., Kumm, M., Varis, O.: China's sustainable water-energy-food nexus by 2030: impacts of urbanization on sectoral water demand. *J. Clean. Prod.* **251**, 119755 (2020)
2. Vanham, D., et al.: Physical water scarcity metrics for monitoring progress towards SDG target 6.4: an evaluation of indicator 6.4. 2 “level of water stress”. *Sci. Total Environ.* **613**, 218–232 (2018)
3. Kavya, M., Mathew, A., Shekar, P.R., Sarwesh, P.: Short term water demand forecast modelling using artificial intelligence for smart water management. *Sustain. Cities Society* **95**, 104610 (2023)
4. Lopez-Bermudez, F., Romero-Diaz, A., Martinez-Fernandez, J., Martinez-Fernandez, J.: Vegetation and soil erosion under a semi-arid mediterranean climate: a case study from Murcia (Spain). *Geomorphology* **24**(1), 51–58 (1998)
5. Puy, A., Balbo, A.: The genesis of irrigated terraces in al-Andalus. A geoarchaeological perspective on intensive agriculture in semi-arid environments (Ricote, Murcia, Spain). *J. Arid Environ.* **89**, 45–56 (2013)
6. S.H. Confederation: Hydrological plan for the demarcation of Segura 2015-2021 (2015). Accessed 05 Feb 2025
7. Muratoglu, A., Iraz, E., Ercin, E.: Water resources management of large hydrological basins in semi-arid regions: spatial and temporal variability of water footprint of the upper euphrates river basin. *Sci. Total Environ.* **846**, 157396 (2022)
8. Ródenas, M., Albacete, M.: The river Segura: reclaimed water, recovered river. *J. Water Reuse Desalin.* **4**(1), 50–57 (2014)
9. Otamendi, U., Maiza, M., Olaizola, I.G., Sierra, B., Florez, M., Quartulli, M.: Integrated water resource management in the Segura hydrographic basin: an artificial intelligence approach. *J. Environ. Manage.* **370**, 122526 (2024)
10. Hydrographic Confederation of Segura Basin: Control networks – SAICA (2025). Accessed 09 Apr 2025
11. Robusto, C.C.: The cosine-haversine formula. *Am. Math. Mon.* **64**(1), 38–40 (1957)
12. Paredes Arquiola, A., Solera, S.: Manual Técnico del Modelo GESCAL. Instituto de Ingeniería del Agua y Medio Ambiente, Universidad Politécnica de Valencia (2013). Version 2.0. Accessed 8 May 2013



An Integrated Approach for Short-Term Forecasting of Highway Vehicle Flows Based on Singular Spectrum Analysis and Artificial Neural Networks

Nikiforos Botzoris , Anastasios Panagiotis Psathas^(✉) , Antonios Papaleonidas ,
and Lazaros Iliadis 

Department of Civil Engineering-Lab of Mathematics and Informatics (ISCE), Democritus
University of Thrace, 67100 Xanthi, Greece
n.botzoris@gmail.com, {anpsatha,papaleon,liliadis}@civil.duth.gr

Abstract. Forecasting transportation demand is essential for the effective planning and operation of infrastructure and related services. Accurate predictions are critical to ensuring the efficient performance of transport systems and meeting evolving user needs. This study analyzes the key factors influencing transportation demand and explores their implications for infrastructure planning and management. The primary objective is to integrate Singular Spectrum Analysis (SSA) with Artificial Neural Networks (ANNs) for time series modeling and forecasting of transportation infrastructure demand. SSA, a non-parametric method for decomposing time series into lower-dimensional interpretable components, is employed to extract significant features such as trend, periodicities, and noise. This hybrid approach enhances model predictive accuracy while supporting efficient handling of large datasets. The methodology is applied to a real-world case study on the Egnatia Odos Motorway, focusing on the Iasmos–Komotini toll station over a four-year period. The models are trained using the hold-out validation method and evaluated based on the coefficient of determination (R^2) and Mean Squared Error (MSE).

Keywords: Machine Learning · Neural Networks · Univariate · Autoregression · toll · traffic management

1 Introduction

All transport systems, road, rail, air, and sea, operate for many years once they are designed, with limited opportunities for upgrades, which are often costly and time-consuming. This is because they operate continuously throughout the year, being of vital importance to society and the economy of each region [1]. Disruptions in transport systems, such as traffic congestion, time delays, and reduced safety, reduce social and business benefits, forcing citizens and businesses to reconsider their travel choices, multimodal connections, and supply chain operations [2]. At a global level and for the

period 2005–2030, it is estimated that a total of around 41 trillion US dollars has already been invested and will be invested to modernize transport infrastructure to meet transport demand, with most of the required investments allocated to land transport—primarily road, and to a lesser extent, rail [3]. Therefore, accurate knowledge of transportation demand is essential for the timely planning and proper sizing of the required transport infrastructure investments. This is achieved through correct forecasts which should realistically reflect the dynamics of the phenomenon of demand for passengers and freight transport [4].

Traffic load is defined as the number of vehicles passing through a specific road segment or intersection; thus, traffic load forecasting is critical for optimizing road operations [5]. Accurate forecasting is particularly important on roads with toll stations, as tolling points constitute critical infrastructure nodes that can lead to significant delays. Therefore, the timely and accurate prediction of traffic load, especially the hourly traffic volume that toll stations must handle, has direct implications for their functionality and performance, and more specifically [6]:

- Reducing vehicle waiting times through optimal toll station staffing based on accurate forecasts.
- The proper allocation of resources, such as personnel and available automated or non-automated vehicle service locations, thereby improving efficiency and reducing operating costs.
- Managing peak traffic more effectively by enabling proactive measures (e.g., temporary free passage during heavy congestion).

2 Literature Review

Various methods have been developed for analyzing and predicting road traffic load, falling into two main categories: qualitative and quantitative methods. Qualitative methods are mainly applied to long-term forecasts, mainly at the road design stage, while quantitative methods can produce highly accurate forecasts for short- and medium-term horizons, including [7]:

- Statistical models, where historical traffic data is used to predict future traffic.
- Simulation models, which simulate traffic flow by taking into account vehicle interaction and traffic dynamics in real time.
- Artificial intelligence and machine learning models, which use advanced algorithms to analyze big data and predict complex traffic patterns.

SSA is a powerful non-parametric technique that decomposes time series into interpretable components, requiring no prior assumptions about the data. It involves embedding the series into a trajectory matrix, performing singular value decomposition, and reconstructing the series through diagonal averaging. SSA is particularly adept at handling non-stationary and nonlinear data, making it suitable for complex real-world time series. In 2007, Hassani [8] applied SSA to the monthly accidental deaths in the USA dataset and compared its forecasting performance with Box-Jenkins SARIMA models, the ARAR algorithm, and the Holt-Winter algorithm. The results demonstrated that SSA provided more accurate forecasts than the other methods. In 2017, Sitohang and

Darmawan [9] compared SSA with the Autoregressive Fractionally Integrated Moving Average (ARFIMA) model for forecasting motorcycle sales in Indonesia. SSA achieved a Mean Absolute Percentage Error (MAPE) of 13.57%, outperforming ARFIMA's MAPE of 22.95%. In 2006, Marques et al. [10] utilized SSA for hydrological time series, including annual precipitation, monthly runoff, and hourly water temperature. SSA effectively extracted trend and oscillatory components and provided accurate forecasts for these components.

In 2024, Lima et al. [11] evaluated Autoregressive Integrated Moving Average (ARIMA), classical decomposition with multiple linear regression, and Holt–Winters methods on retail trade data from seven European countries. ARIMA models demonstrated superior performance, achieving lower Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) values compared to the other methods. Holt–Winters also produced accurate forecasts, serving as a viable alternative to ARIMA. In 1982, Engle [12] introduced the Autoregressive Conditional Heteroskedasticity (ARCH) model to capture time-varying volatility. In 1986, Bollerslev [13] extended it to Generalized ARCH (GARCH) to account for both autoregressive and moving average components in volatility modeling. These models have been instrumental in financial econometrics for forecasting market volatility.

In 2018, Siامي-Namini et al. [14] compared ARIMA and Long Short-Term Memory (LSTM) Networks models for economic and financial time series forecasting. LSTM models achieved significantly lower error rates, reducing errors by 84–87% compared to ARIMA, indicating superior performance in capturing complex patterns. In 2021, Lara-Benítez et al. [15] conducted an extensive study evaluating seven deep learning architectures across over 50,000 time series. Convolutional Neural Networks (CNNs) emerged as top performers, offering high accuracy with less variability and higher efficiency compared to other models. Also in 2021, Tang et al. [16] proposed a hybrid model combining SSA with LSTM (SSA-LSTM) for financial time series forecasting. The SSA-LSTM model achieved a Root Mean Squared Error of 1.17, Mean Absolute Error of 0.97, and Mean Absolute Percentage Error of 0.00317. In 2020, Oreshkin et al. introduced the Neural Basis Expansion Analysis for Interpretable Time Series Forecasting (N-BEATS). It is a deep learning architecture for univariate time series forecasting. Unlike traditional Recurrent Neural Network (RNN)-based models (e.g., LSTM), N-BEATS employs a fully connected feedforward neural network with forward and backward residual connections to decompose time series into trend and seasonality components. This allows for both high forecasting accuracy and interpretability, which are often trade-offs in deep learning models.

In this paper, the authors utilize a dataset provided by Egnatia Odos S.A., consisting of hourly vehicle crossings recorded at the Iasmos toll station on the Egnatia Motorway, spanning the period from January 1, 2014, to October 31, 2018. The objective of this study is to integrate Singular Spectrum Analysis (SSA) into time series modeling, in combination with Artificial Neural Networks (ANNs), to analyze and forecast transportation infrastructure demand (Fig. 1). SSA, a technique that decomposes time series data into lower-dimensional components, is employed to extract key features such as the underlying trend, primary and secondary periodicities, and residual noise.

The rest of the paper is organized as follows: Sect. 2 describes the area of research, Sect. 3 describes the dataset, its features and the pre-processing process, Sect. 4 provides the architecture of the proposed models, and the evaluation indices used for this research effort, Sect. 5 presents the experimental results and the evaluation of the model, and, finally, Sect. 6 concludes the research.

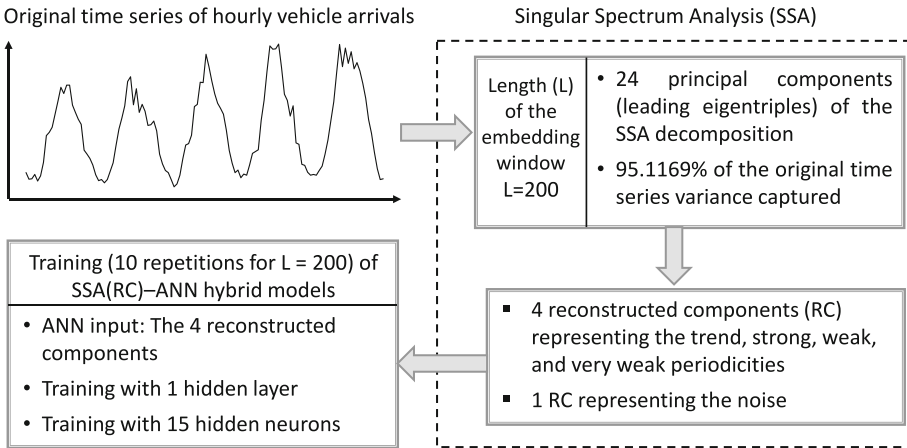


Fig. 1. The proposed hybrid methodology for forecasting hourly vehicle arrivals integrates Singular Spectrum Analysis into the training process of Artificial Neural Networks.

3 Area of Research

Motorway 2 (A2), also known as the Egnatia Motorway, is the longest in Greece, extending 670 km and forming part of European Route E90. It starts from Igoumenitsa in Epirus and ends at Kipi on the Greek-Turkish border (Fig. 2). It was constructed for the most part between 1994 and 2009, while its last sections were completed in 2014. The management body of the Egnatia Motorway maintains a website (<https://egnatia.eu/>) which provides useful information about the construction and operation of the road.

The modern Egnatia Odos was constructed near National Road 2, largely replacing it in the section between Thessaloniki and Kipi. Its name comes from the Ancient Egnatia Odos which crossed almost the same area. It also replaced segments of other national roads, including National Road 4 between Alexandria and Kozani, National Road 20 between Kozani and Siatista, National Road 15 in the Siatista - Grevena section, the Grevena - Metsovo Provincial Road in the Grevena - Metsovo section and National Road 6 in the Panagia - Igoumenitsa section.

Egnatia Motorway has two traffic directions, one towards Igoumenitsa and one towards Kipos Evros, which are separated by a median dividing island. Each direction includes two traffic lanes and one emergency lane, apart from the Kleidi Imathias - Lagkadas section which has three traffic lanes and one emergency lane. The maximum permitted speed is 130 km/h. After the completion of the Egnatia Motorway, the average

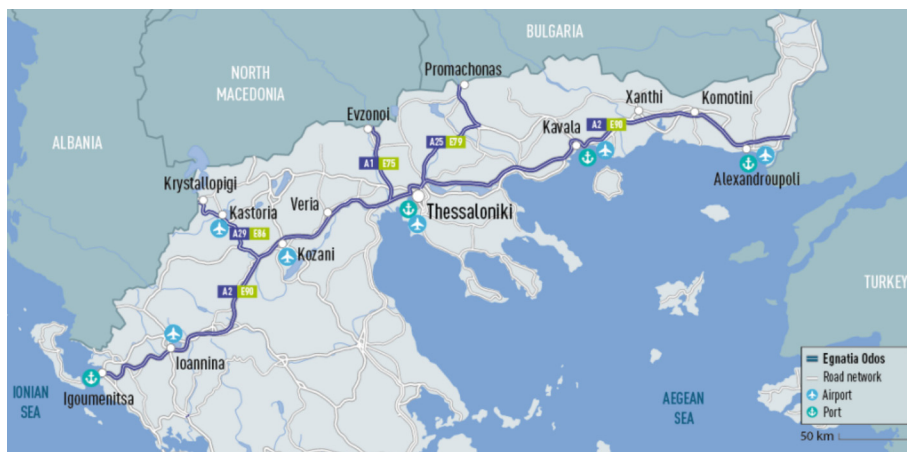


Fig. 2. Alignment of the Egnatia Motorway across Northern Greece (from Igoumenitsa to Kipi/Alexandroupoli), including key infrastructure nodes.

travel time between Igoumenitsa and Kipi was reduced from 11.5 h to 6.5 h, resulting in a time saving of 5 h. The Egnatia Motorway's area of influence and service concentrates 36% of the country's total population, 1/3 of the total gross national product, in the primary sector 1/2 of the country's total agricultural land and 2/3 of the irrigated area, in the secondary sector 41% of total employment in the country's industry and crafts as well as 51% of mining activity. Egnatia Motorway has 42 km of bridges, which constitute 6% of the total length of the axis, as well as approximately 12% of the total cost. Also, the Egnatia features 73 twin tunnels with a combined length of 50 km, representing 7% of the total road length, while their construction cost is approximately 30% of the total construction cost.

There are fourteen front toll stations operating on the Egnatia Motorway, four front stations on the vertical axes, one off-axis station, and seventeen side toll stations. At the toll stations on the Egnatia Motorway, vehicle service is usually provided either by manned service stations or (as of 1/9/2020) by automated electronic service stations, through the EgnatiaPass service (Fig. 3). In this case, a transponder installed in the vehicle assumes the vehicle's transit charge through an account linked to the service. The Iasmos toll station (Rhodopi Prefecture) is located midway between the cities of Xanthi and Komotini. It has a total of six service stations (two of which are automatic), usually operating three in each direction (Figs. 3).

As already mentioned above, the aim of this research is the prediction of traffic load and, more specifically, the number of conventional passenger vehicles passing through the Iasmos toll station. For this purpose, Egnatia Motorway granted hourly passage for conventional passenger vehicles during the period from 1/1/2014 at 00:00 to 31/10/2018 at 23:59 in both directions of Egnatia Motorway, namely towards Igoumenitsa and towards Kipi/Alexandroupoli. A total of 84,700 hourly vehicle crossings in both directions of the Egnatia Motorway will therefore be analyzed.



Fig. 3. Iasmos toll station, direction from Igoumenitsa to Kipi/Alexandroupoli.

4 Dataset

The dataset consists of finite sequences of numerical measurements (samples), automatically collected by the toll management system and represented as time series [19]. A time series of length n is an ordered sequence of values $x = x[1], \dots, x[n]$ of finite precision. A subsequence $x[a:b]$ of x is defined as follows

$$x[a : b] = (x[a], x[a + 1], \dots, x[b]), 1 \leq a < b \leq n \quad (1)$$

The data used in this research were provided by Egnatia Odos S.A. Specifically, the dataset consists of hourly vehicle crossings recorded at the Iasmos toll station. Figure 4 illustrates the evolution of hourly vehicle arrivals at the toll station during the analysis period, which spans from January 1, 2014 (00:00–00:59) to October 31, 2018 (23:00–23:59). In total, the dataset includes 42,351 hourly records of vehicle arrivals.

The stationarity test of a time series is performed using the Augmented Dickey-Fuller test statistic, which tests for the existence of a unit root. The null hypothesis H_0 of the test is that the time series Y under stationarity test presents a unit root, while the alternative hypothesis H_1 does not present a unit root and is stationary [20]. The maximum number of lags $\text{lag}_{\max}$ for testing stationarity using the augmented Dickey-Fuller test statistic will be determined using the rule [21]:

$$\text{lag}_{\max} = 12 \cdot \left(\frac{n}{100} \right)^{\frac{1}{4}} \quad (2)$$

For $n = 42,351$ measurements, $\text{lag}_{\max} = 54.34$ is obtained, therefore when applying the augmented Dickey-Fuller test, $\text{lag}_{\max} = 55$ will be obtained. The results of the augmented Dickey-Fuller test for all three cases a) the time series does not show a trend nor has a fixed term (no intercept, no trend), b) the time series has a fixed term (intercept, no trend), and c) the time series has a fixed term and shows a trend (intercept and trend) are given in Fig. 5.

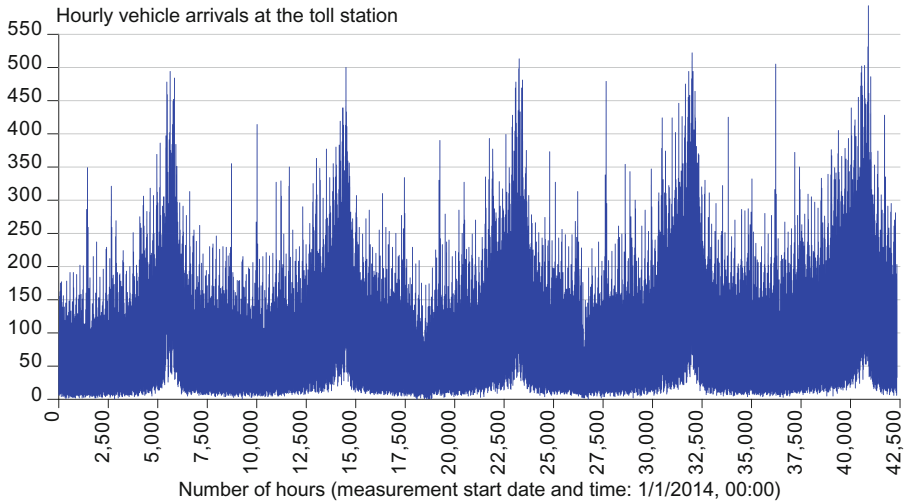


Fig. 4. Evolution of hourly vehicle arrivals at the Iasmos toll station (westbound direction toward Igoumenitsa) during the analysis period.

The results shown in Fig. 4 indicate that in all cases $p\text{-values} \leq 0.05$ apply, therefore at a significance level of 5% all null hypotheses that the time series has a unit root must be rejected in favor of the alternatives that the time series does not have a unit root and is therefore stationary. In fact, the test result is so strong that it is valid even at a significance level of 1% ($p\text{-values} \leq 0.01$). However, this is noted only for reference, as all statistical tests in this study are conducted at a 5% significance level.

5 Algorithms and Evaluation Indices

5.1 Singular Spectrum Analysis

Singular Spectrum Analysis (SSA) focuses on decomposing the original time series into a set of meaningful components. The components may represent the trend of the time series, seasonality, long-term periodicities (cycles) not related to seasonality, peculiar or extreme observations that are unusually far from the rest, and change in the variation of observations over time, and of course, structureless oscillations (time series noise). After decomposition, SSA enables the identification of key components and their characteristics, followed by the reconstruction of the original time series and potential forecasting. The application of the Eigen spectrum Analysis method includes four steps [22, 23]:

- Embedding process, in which a one-dimensional time series $Y_T = (y_1, \dots, y_t)$ is converted into a multidimensional one. The multidimensional time series $X_1, X_2, \dots, X_K$ results from the lagged L ($L \leq T/2$) vectors (lagged L vectors) $X_i = (y_i, \dots, y_{i+L-1})^T \in \mathbb{R}^L$, where $K = T - L + 1$. The vectors X_i are called trajectory vectors. The result of the above step is the $L \times K$ dimensional trajectory

Null Hypothesis: HOURLY has a unit root

Exogenous: None

Lag Length: 53 (Automatic - based on SIC, maxlag=55)

	t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic	-3.384232	0.0007
Test critical values: 1% level	-2.565017	
5% level	-1.940832	
10% level	-1.616695	

Null Hypothesis: HOURLY has a unit root

Exogenous: Constant

Lag Length: 53 (Automatic - based on SIC, maxlag=55)

	t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic	-9.556392	0.0000
Test critical values: 1% level	-3.430335	
5% level	-2.861417	
10% level	-2.566745	

Null Hypothesis: HOURLY has a unit root

Exogenous: Constant, Linear Trend

Lag Length: 53 (Automatic - based on SIC, maxlag=55)

	t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic	-10.16037	0.0000
Test critical values: 1% level	-3.958250	
5% level	-3.409908	
10% level	-3.126666	

*MacKinnon (1996) one-sided p-values.

Fig. 5. Stationarity test of the time series of hourly vehicle arrivals using the Augmented Dickey-Fuller test. The top panel corresponds to the model with no trend and no constant term, the middle panel includes constant term, the bottom panel includes both trend and constant term.

matrix $X = [X_1, X_2, \dots, X_K]$, where:

$$X = (x_{ij})_{i,j=1}^{L,K} = \begin{pmatrix} y_1 & y_2 & \dots & y_K \\ \vdots & \vdots & \ddots & y_{K+1} \\ y_L & y_{L+1} & \dots & y_T \end{pmatrix} \quad (3)$$

- Decomposition process, which first involves the construction of the $L \times L$ matrix $\mathbf{XX}^T$, to which singular value decomposition (SVD) will be applied in the next step. The $\mathbf{XX}^T$ matrix is obtained by multiplying $\mathbf{X}$ by the transpose of $\mathbf{X}^T$. This multiplication creates the symmetric $\mathbf{XX}^T$ matrix whose eigenvalues are all positive. The $\mathbf{XX}^T$ matrix will be used to calculate the correlation and covariance of the rows of the $\mathbf{X}$ matrix. This is followed by the decomposition into singular values of $\mathbf{XX}^T$ and the calculation of its eigenvalues and eigenvectors, which are represented in the form

$\mathbf{XX}^T = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^T$. The matrix $\mathbf{\Lambda}$ is a diagonal matrix containing the eigenvalues $\lambda_1, \lambda_2, \dots, \lambda_L$, such that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_L \geq 0$. The matrix $\mathbf{P}$ in turn is the diagonal matrix containing the eigenvectors $\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_L$ of $\mathbf{XX}^T$.

- Grouping eigenvectors based on their eigenvalues, in order to separate the real signal of the time series (principal components) from the noise. Usually, the largest eigenvalues correspond to significant components, while the smallest ones to noise. We choose a group ℓ ($1 \leq \ell \leq L$) of eigenvectors $\mathbf{P}_{i,1}, \mathbf{P}_{i,2}, \dots, \mathbf{P}_{i,\ell}$. Clustering consists of dividing the elementary matrices (of order 1) $\mathbf{X}_i$ into several groups based on their similarity and then summing the matrices within each group. If $I = \{i_1, i_2, \dots, i_\ell\}$ is a group, then the matrix $\mathbf{X}_I$ corresponding to group I can be defined as $X_I = X_1, X_2, \dots, X_K$.
- Reconstruction of a one-dimensional time series, calculating, as an approximation of the $\mathbf{X}$ -matrix, the matrix $\tilde{\mathbf{X}} = \|\tilde{x}_{i,j}\| = \sum_{k=1}^{\ell} \mathbf{P}_{i_k} \mathbf{P}_{i_k}^T \mathbf{X}$. The transition to the one-dimensional time series can be achieved by averaging the diagonals of the matrix $\tilde{\mathbf{X}}$.

5.2 Artificial Neural Networks

Artificial Neural Networks (ANNs) are computational frameworks inspired by the structure and functioning of the human brain. They consist of interconnected processing units called *neurons* that work collectively to solve complex tasks such as pattern recognition, classification, and forecasting. Each neuron receives input, applies a transformation (typically nonlinear), and passes the result to the next layer. ANNs are particularly well-suited for tasks involving large datasets with non-linear relationships, making them popular in fields like finance, healthcare, and time series forecasting [24].

The concept of ANNs dates back to the 1940s when McCulloch and Pitts, in 1943 [25], introduced a mathematical model of a neuron. However, practical development accelerated in the 1980s with the invention of the backpropagation algorithm by Rumelhart, Hinton, and Williams, in 1986 [26], which allowed multi-layer networks to be trained efficiently. This advancement enabled the development of multilayer perceptrons (MLPs), consisting of input, hidden, and output layers. Each layer transforms its input using weighted connections and activation functions such as sigmoid, tanh, or ReLU.

Training ANNs involves adjusting the weights of the connections between neurons to minimize forecasting errors. This is typically done using gradient descent algorithms and backpropagation, where the error from the output layer is propagated backward to update weights iteratively. The network learns from a dataset through multiple *epochs*, gradually improving its performance. Modern variants like stochastic gradient descent (SGD), Adam optimizer, and dropout regularization enhance training stability and generalization [27].

5.3 Evaluation of Deep Learning Algorithms

For the performance of the proposed approach the R^2 and MSE evaluation indices were used. The R^2 index is calculated by the following equation:

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (4)$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, $\bar{y}$ is the mean value, and N is the total number of the instances. R^2 measures how much variability in dependent variable can be explained by the model. R^2 is valued in closed interval $[0, 1]$ and a bigger value indicates a better fit between prediction and actual value. R^2 is a useful metric for evaluating how well the model explains variability in the dependent variable [17].

The Mean Square Error (MSE) measures the average magnitude of the errors between the predicted and actual values, as it penalizes larger errors more heavily by squaring the differences before averaging. It is calculated as follows:

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (5)$$

where again y_i is the actual value, $\hat{y}_i$ is the predicted value, $\bar{y}$ is the mean value, and N is the total number of the instances [18].

6 Evaluation and Experimental Results

In the SSA procedure, several variables are defined, including the length of the time series, the window length L , and the number of components which are used to determine the principal components (PC) of the original time series and the reconstructed components (RC) that contribute to its reconstruction. For the window length L , values are chosen based on the nature of the observations and the expected, mainly intuitive, time period of periodicities. Thus, the SSA method is fed for $L = 50$ and $L = 200$. The value $L = 50$ is chosen to capture periodicities that may appear in a range of up to 50 h, i.e. approximately 2 days ($= 48$ h). The value $L = 200$ now represents the range of hourly observations of a previous consecutive eight-day period ($= 192$ h), i.e. for the analysis of the 24-h values of a current day, for example, the current Wednesday, the hourly observations from the previous eight consecutive days (i.e., from Tuesday back to the Tuesday of the previous week) are taken into account. After conducting all experiments, the better results were achieved for $L = 200$ and thus, only these will be described.

For $L = 200$, the initial time series of vehicle arrivals at the Iasmos toll station (directed towards Igoumenitsa) is decomposed into 200 principal components. The eigenvalues of these components are shown in Fig. 6.

Seeking to explain at least 95% of the variation of the original time series by the principal components, it is found from Fig. 6 that the first 24 principal components (PC) explain 95.1169% of the variation of the original time series. Therefore, the next 176 ($= 200 - 24$) principal components can be considered as noise in the time series and can be grouped (integrated) into a reconstructed component. From the first 24 principal components, the 1st principal component has an eigenvalue (2.41×10^6) that is significantly larger than the rest, therefore it can be considered to represent the dominant component that reflects the long-term changes in the value of the time series, i.e. the trend. The 2nd

and 3rd principal components have relatively close eigenvalues (4.44×10^5 and 4.31×10^5 respectively) and as a pair can be considered to represent the strong periodicities of the time series. The 4th and 5th principal components have eigenvalues between 2.12×10^4 and 1.98×10^4 therefore can be considered to represent weak periodicities. Finally, all remaining principal components, from the 25th to the 200th, are considered random and treated as time series noise.

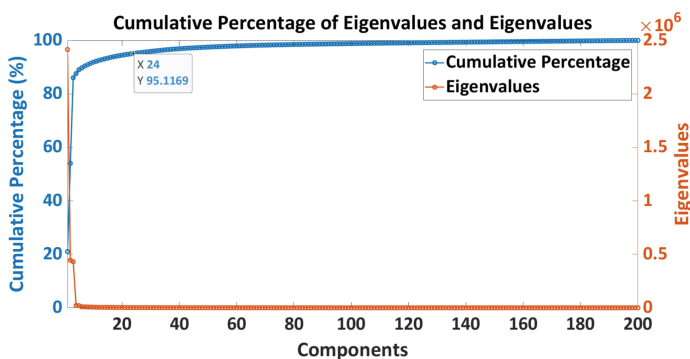


Fig. 6. Eigenvalues per principal component and cumulatively, after decomposition of the original time series with $L = 200$.

Therefore, the 1st principal component (PC), identified as a trend, alone composes the 1st reconstructed component (RC), the 2nd and 3rd PC, as strong periodicities, compose the 2nd RC, the 4th and 5th PC, as weak periodicities, compose the 3rd RC, the very weak periodicities (from the 6th to the 24th PC) compose the 4th RC, while the 25th to the 200th PC, considered as noise, compose the 5th and last RC, as it is presented in Fig. 7. The coefficient of determination (R^2) and MSE between the original time series and the four RC (excluding the noise captured in the 5th RC) are as follows: $R^2 = 0.95532$, $MSE = 300.4968$. Furthermore, Fig. 7 shows the corresponding diagram for $L = 200$ where the total four RC include the trend and the strong and weak periodicities (PC 1 to 24) and the RC incorporating the noise (PC 25 to 200).

All ANNs, including their training, testing, and evaluation procedures, were implemented on the MATLAB platform. The parameters and characteristics of the neural architecture are as follows:

- The input layer consists of four neurons, corresponding to the four RCs. The RC associated with noise was excluded from model training.
- A 70%-15%-15% split was applied for training, validation, and testing.
- The hidden layer comprises 15 neurons in a single hidden layer configuration.

ANNs were trained 10 times, and the average value was taken into consideration for the evaluation. The optimized model achieved an R^2 of 0.9607 and an MSE of 263.9394. Figure 8 presents the actual values vs predicted values (Fig. 9).

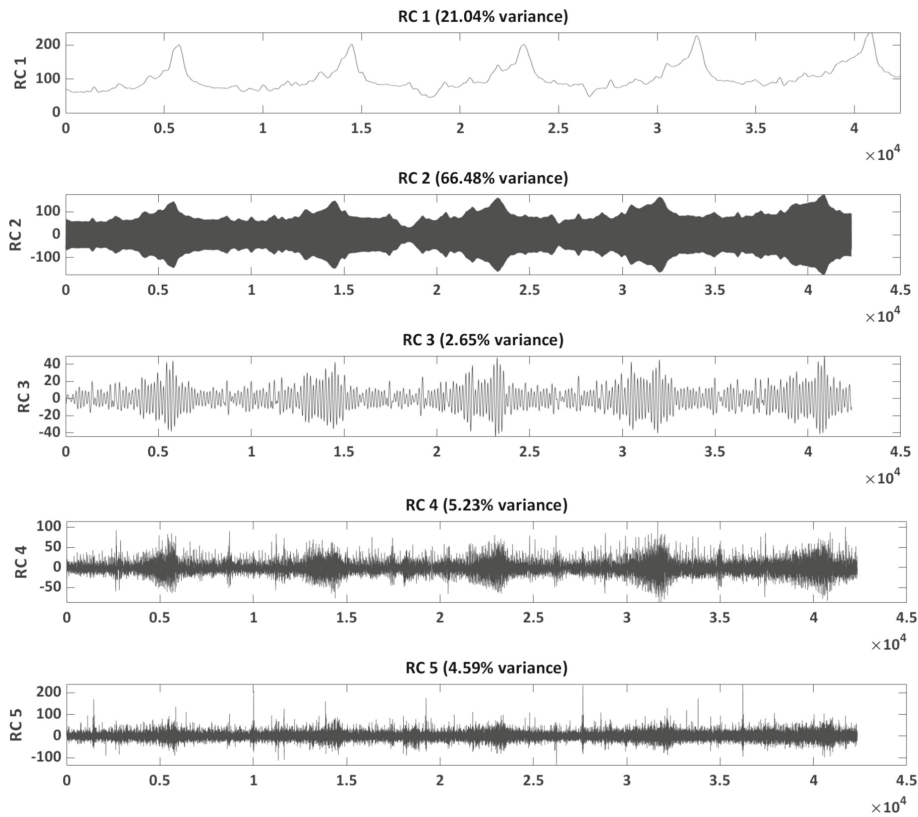


Fig. 7. The reconstructed components (RC) of the original time series for $L = 200$.

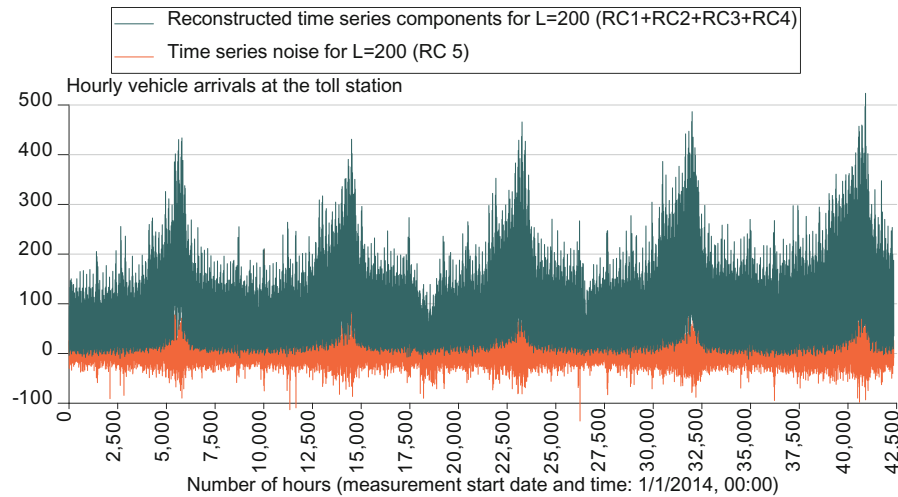


Fig. 8. The 5 reconstructed components (RC) for $L = 200$, which compose the original time series (green), times series noise (orange). (Color figure online)

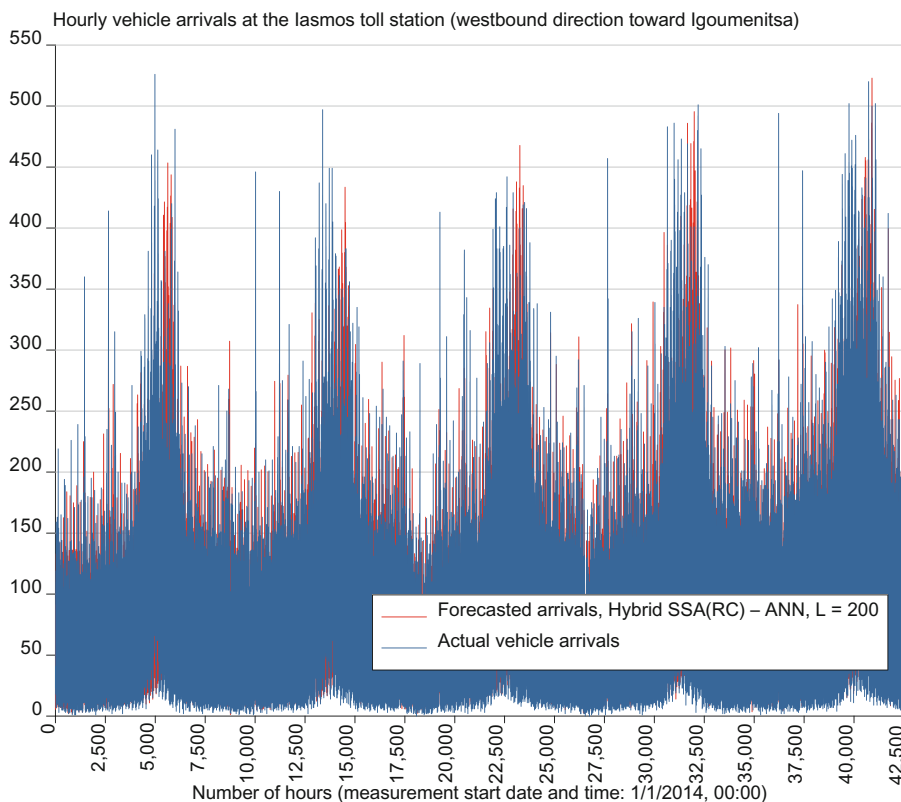


Fig. 9. Actual hourly vehicle arrivals and forecasted values using the hybrid SSA(RC) – ANN model (based on reconstructed components) at the Iasmos toll station.

7 Conclusions and Future Work

The aim of this study is to analyze the traffic load at the Iasmos toll station, and more specifically, the hourly vehicle crossings from the toll plaza towards Igoumenitsa, a total of 42,351 hourly observations were analyzed for the period from 1/1/2014 to 31/10/2018. SSA was used to decompose the time series of 42,351 hourly observations into its main features, namely trend, strong and weak periodicities, and noise. Using SSA with a window length of $L = 200$, the principal and reconstructed components of the original time series were extracted. 24 principal components were extracted that explained 95.1169% of the variance of the original time series (therefore the next 176 principal components were considered as noise). Ten hybrid SSA(RC)-NN models were trained, and the average performance across all trainings was considered. The input variables in these models were the 4 reconstructed components $L = 200$. Given the limited number of input variables (4), the neural networks included a single hidden layer with 15 neurons. Training time was under one minute. The models achieved high performance metrics: $R^2 = 0.9607$ and $MSE = 263.9394$.

Although the results were very good, there is always room for improvement. Several directions for future work are already under consideration. Data enrichment to improve the generalizability of the model. This may involve either expanding the dataset to cover a longer time span or generating synthetic data through techniques such as noise injection, rescaling, or temporal transformation. Combination of other methods with Singular Spectrum Analysis (SSA), like ARIMA, Support Vector Machines (SVM), Genetic Algorithms (GA). Another direction involves incorporating exogenous variables, such as fuel prices, GDP per capita, toll rates, and weather conditions. Improving the neural network architecture by using advanced types of ANNs, such as Long Short-Term Memory (LSTM) networks, which can store information for long periods of time, have mechanisms to control the flow of information, thus helping to preserve important information and eliminate less important ones.

References

1. Lebo, J., Schelling, D.: Design and Appraisal of Rural Transport Infrastructure: Ensuring Basic Access for Rural Communities, vol. 23. World Bank Publications (2001)
2. Marsden, G., Anable, J., Shires, J., Docherty, I.: Travel behaviour response to major transport system disruptions: implications for smarter resilience planning. In: International Transport Forum Discussion Paper (2016)
3. Schuckmann, S.W., Gnatzy, T., Darkow, I.L., von der Gracht, H.A.: Analysis of factors influencing the development of transport infrastructure until the year 2030—a Delphi based scenario study. *Technol. Forecast. Soc. Chang.* **79**(8), 1373–1387 (2012)
4. Narayanaswami, S.: Urban transportation: innovations in infrastructure planning and development. *Int. J. Logistics Manag.* **28**(1), 150–171 (2017)
5. Zhang, Y., Zhang, Y., Haghani, A.: A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model. *Transp. Res. Part C Emerg. Technol.* **43**, 65–78 (2014)
6. Botzoris, G.N., Papadopoulos, B.K., Sfiris, D.S.: Modeling queueing systems using fuzzy estimators. *Fuzzy Econ. Rev.* **18**(2), 3 (2013)
7. Profillidis, V.A., Botzoris, G.N.: Modeling of Transport Demand: Analyzing, Calculating, and Forecasting Transport Demand. Elsevier (2018)
8. Hassani, H.: Singular spectrum analysis: methodology and comparison. *J. Data Sci.* **5**(2), 239–257 (2007)
9. Sitohang, Y.O., Darmawan, G.: The accuracy comparison between ARFIMA and singular spectrum analysis for forecasting the sales volume of motorcycle in Indonesia. *AIP Conf. Proc.* **1868**(1), 040011 (2017)
10. Marques, C.A.F., et al.: Singular spectrum analysis and forecasting of hydrological time series. *Phys. Chem. Earth Parts A/B/C* **31**(18), 1172–1179 (2006)
11. Lima, S., Gonçalves, A.M., Costa, M.: Predictive accuracy of time series models applied to economic data: the European countries retail trade. *J. Appl. Stat.* **51**(9), 1818–1841 (2024)
12. Engle, R.F.: Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica J. Econ. Soc.*, 987–1007 (1982)
13. Bollerslev, T.: Generalized autoregressive conditional heteroskedasticity. *J. Econ.* **31**(3), 307–327 (1986)
14. Siami-Namini, S., Tavakoli, N., Namin, A.S.: A comparison of ARIMA and LSTM in forecasting time series. In: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), pp. 1394–1401. IEEE, December 2018

15. Lara-Benítez, P., Carranza-García, M., Riquelme, J.C.: An experimental review on deep learning architectures for time series forecasting. *Int. J. Neural Syst.* **31**(03), 2130001 (2021)
16. Tang, Q., Shi, R., Fan, T., Ma, Y., Huang, J.: Prediction of financial time series based on LSTM using wavelet transform and singular spectrum analysis. *Math. Probl. Eng.* **2021**(1), 9942410 (2021)
17. Israeli, O.: A Shapley-based decomposition of the R-square of a linear regression. *J. Econ. Inequality* **5**(2), 199–212 (2007)
18. Hodson, T.O.: Root mean square error (RMSE) or mean absolute error (MAE): when to use them or not. *Geosci. Model Dev. Discuss.* **2022**, 1–10 (2022)
19. Vespier, U., Nijssen, S., Knobbe, A.: Mining characteristic multi-scale motifs in sensor-based time series. In: *International Conference on Information and Knowledge Management, Proceedings*, pp. 2393–2398 (2013). <https://doi.org/10.1145/2505515.2505620>
20. Dickey, D.A., Fuller, W.A.: Distribution of the estimators for autoregressive time series with a unit root. *J. Am. Stat. Assoc.* **74**, 427–431 (1979)
21. Schwert, G.W.: Tests for unit roots: a Monte Carlo investigation. *J. Bus. Econ. Stat.* **20**(1), 5–17 (2002)
22. Hassani, H.: A brief introduction to singular spectrum analysis. In: *Optimal Decisions in Statistics and Data Analysis* (2010). https://ssa.cf.ac.uk/ssa2010/a_brief_introduction_to_ssa.pdf
23. Shuai, C., Pan, Z., Gao, L., Zuo, H.: Short-term traffic flow prediction of express-way: a hybrid method based on singular spectrum analysis decomposition. *Adv. Civil Eng.* **2021**(1), 4313970 (2021)
24. Haykin, S.: *Neural Networks and Learning Machines*, 3rd edn. Pearson Education (2009)
25. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **5**(4), 115–133 (1943)
26. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. *Nature* **323**(6088), 533–536 (1986)
27. Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press (2016)



Assessing Mobility Policies by Traffic Simulation and Change Detection

Felipe Bagni¹ , Edoardo Peretti¹ , Danilo Macció² ,
Cristiano Cervellera² , and Giacomo Boracchi¹

¹ Politecnico di Milano, Milan, Italy

`felipe.bagni@mail.polimi.it,`

`{edoardo.peretti,giacomo.boracchi}@polimi.it`

² National Research Council, Institute of Marine Engineering, Genova, Italy

`{danilo.maccio,cristiano.cervellera}@cnr.it`

Abstract. Defining effective policies for managing traffic in large cities, particularly near major logistics hubs such as ports, is a challenging problem due to the critical interaction between mobility and freight flows. In this work, we combine a traffic simulator with a change-detection test to assess a priori whether a specific policy would have an impact on city mobility. More specifically, we propose a general methodology to identify the expected number of days/monitoring samples before gaining evidence that the policy has introduced a detectable change in the traffic data acquired after enforcing the policy. Our experiments, conducted on simulated traffic, focused on the port-city context of Genova, showcase that our proposed methodology can provide outcomes that are consistent with the kind and expected effectiveness of policies under evaluation.

1 Introduction

Traffic management is a critical problem in cities, especially in areas where the urban infrastructure supports both mobility flows and traffic from large logistics hubs. A port located near the city center is a paradigmatic example in which this kind of disruptive interaction occurs, requiring city authorities to define suitable mobility policies to reduce externalities like pollution and congestion [16,24,27]. Indeed, policy definition is crucial and involves many stakeholders: citizens, logistics operators, public transport companies, and port authorities.

Policies that can be implemented include diverse strategies like opening or closing roads, changing routing plans and, most commonly, changing traffic light timing [6]. Yet, for any potential new policy, assessing a priori whether it would be effective is crucial to reduce the risk of making wrong or ineffective choices. To this purpose, sophisticated analytics are typically employed, including traffic simulators that can generate realistic traffic conditions over multiple days [4].

The work was supported by the Project “AI for Sustainable Port-city logistics” funded by the European Union - Next Generation EU Program through the Italian Ministry for University and Research PRIN 2022 PNNR Grant P2022FLLPY.

While the ultimate validation of a policy needs to be conducted by analyzing data acquired on the field, it is important for policy makers to understand a priori the effectiveness of a new policy. In this paper, we propose a general methodology that leverages traffic simulator and change-detection tests for assessing whether a specific policy would be effective, namely whether it has a statistically significant impact on the simulated traffic at specific locations, and how long policymakers should wait before having a confirmation from the field that the policy is effective.

Adopting change detection in urban traffic monitoring is not unheard of, and there is a broad literature in this regard, primarily addressing the detection of accidents or road impairments (see, e.g., [7]). However, using statistical tools to assess how long it will take to conclude that a policy would be effective is challenging for several reasons. First, the input to be analyzed easily becomes high dimensional, as traffic is conveniently monitored at different city locations and selectively for different categories, like the flow and the speed for each type of vehicle (e.g., cars, trucks, etc.). Second, urban networks are complex systems characterized by chaotic and non-linear dynamics, making the distributions of measured quantities unknown. Therefore, running a statistical test to determine whether a high-dimensional stream of mobility data (following an unknown distribution) has changed after policy enforcement is far from trivial, in particular when it comes to granting control over the false alarm rate [11].

Furthermore, the analysis needs to be conducted in a sequential manner, as the final policy assessment is to be conducted on online real data. However, acquiring real data before and after policy enforcement is not trivial, since uncontrolled system dynamics might invalidate the stationarity assumption, and collecting real data requires complex logistics and a waiting time longer than simulated data. Our proposal focuses on assisting policymakers in their planning (the relevant literature is summarized in Sect. 2). We aim at understanding whether a policy is expected to have an influential impact on traffic. In particular, we combine change-detection with traffic simulation, to make policymakers informed whether the policy would be perceivable in simulated data, and to estimate the delay required to observe an impact on the traffic.

Our methodology (illustrated in Sect. 4) combines a traffic simulator and a nonparametric change-detection test to assess the expected latency for the policy to induce a detectable change. Traffic simulators can nowadays achieve realistic outcomes, simulating multiple vehicle categories and modeling complex dynamics [8]. Thus, we simulate traffic before and after policy enforcement to replicate mobility data over multiple days. The second component, the change-detection test, is a sequential statistical tool that analyzes multivariate streams. More specifically, we adopt QT-EWMA [11], which does not underpin any parametric model (e.g., a Gaussian distribution) to describe the multivariate distribution of pre-policy enforcement data, and it allows monitoring while controlling the expected number of samples before having a false positive detection.

In our experiments (Sect. 5), we consider as a case study the port-city context of Genova, one of the main ports in the Mediterranean Sea. In particular, we investigate three different policies and show that our method provides consistent

findings with the expected traffic behavior. More specifically, our test can control the expected delay for false positives, and confirms that changes are more perceivable in specific streets. This is an encouraging result, which paves the way for extending the use of QT-EWMA to address real-time traffic monitoring, combining the monitoring scheme with time-series regression methods.

2 Related Work

Most of the literature on policy evaluation in the context of urban traffic and mobility is focused on ex-post assessment, relying on the analysis of real-world data collected after policy enforcement. By this approach, the effectiveness of mobility policies is assessed only after they have been implemented, while policy makers would need to assess the policy effectiveness beforehand. Typically, the assessment involves the definition of ad-hoc indices of performance of the network, often based on user experience surveys [12, 15, 17], rather than directly relying on traffic flow.

A popular approach to evaluate ex-post the impact of policies relies on a regression model trained on the observed traffic data. This translates into analyzing the impact of the model variables affected by the policy on the predicted outcome. For instance, in [13] a simple nonlinear regression model is employed to evaluate the impact of urban transportation policies from various real world datasets, while in [14] a similar approach has been devised to evaluate the impact of driving restrictions in Beijing. In [20] a parametric time series modeling approach based on the so-called basic structural model has been implemented to evaluate the effect of a new mobility policy in Milan, comparing the pollution levels before and after the policy enforcement. In contrast, our approach leverages simulated scenarios and a nonparametric methodology, to investigate a priori whether a policy is effective with statistical guarantees.

A popular and convenient choice to evaluate the impact of policies a priori is to resort to simulation. By simulating the traffic dynamics of urban networks, it is possible to evaluate lines of intervention in a “what-if fashion” before they are actually deployed. Some approaches in the literature adopt a modeling methodology to describe policies at a strategic level, involving several actors and stakeholders of the urban environment (see, for instance, [25, 26] based on a system dynamics approach). However, these papers do not address the problem of determining whether a policy has significantly impacted urban traffic in general, but they limit themselves to particular cases such as strategies to reduce carbon emissions or urban traffic restriction policies.

When the analysis is focused on the actual performance of the urban network, as in our settings, traffic models of the micro-simulation kind are the most suited to capture the effects of changes induced by policies. Micro-simulation [8, 9] is based on the “car following” principle and can represent the behavior of individual vehicles, providing a dynamic and detailed representation of all the relevant features of the network (e.g., traffic lights, junctions, priorities, etc.). Thus, traffic simulation models can be used to provide a realistic assessment of the effect of

a wide range of policies, both at the strategic and tactical level. Examples of policy evaluation approaches based on traffic micro-simulation can be found in [19, 22] where the Eclipse/SUMO open source simulator is used, in [10] where the commercial VISSIM software is used, and in [1] using the commercial software AIMSUN. In all these approaches, the simulation tool generates scenarios with different policies, and then suitable performance indices are defined to evaluate the outcomes. Yet, there is no principled statistical methodology to determine whether a new policy would actually affect urban traffic, and how long it takes to confirm it from real measurements. In contrast, our methodology utilizes traffic simulation combined with a sequential change-detection test to estimate the expected time required for a policy to take effect.

Urban traffic contexts characterized by the interaction of mobility and logistics flows are probably among the most critical for policy changes. A port located within a city is the most typical example of these kinds of settings, thus in this paper we take this as the reference scenario for our proposed methodology. The presence of systematic flows from heavy vehicles on top of the general mobility traffic is a source for negative externalities such as pollution and congestion [5, 16, 28]. These externalities increased the need for policies to efficiently regulate the urban road network where port and mobility traffic interact. In the literature, there are quite a few papers focused on policy evaluation and assessment in port-city environments, but the analysis is generally performed at a very strategic level with a focus on economic and social impact, and is based on ex-post real data analysis (see, e.g., [21, 23]). The present paper introduces for the first time a methodology to evaluate the impact of policies in port-city scenarios a priori, focusing on actual changes detectable in the network performance.

3 Problem Formulation

We address the problem of assessing whether a mobility policy significantly impacts an urban network. To this end, we measure traffic data, such as the number and speed of vehicles, at one or more network's points of interest. We denote by $x_t \in \mathbb{R}^d$ a sample of traffic data measured on day t . We assume that the measurements on different days are independent and identically distributed (i.i.d.), and denote by ϕ_0 the unknown distribution of traffic data under a basic scenario.

We define a mobility policy p as a set of actions an authority can enforce on urban roads to modify mobility. For example, a policy may involve altering the timing of traffic lights or regulating access to a street depending on the vehicle class. Each policy p generates a change in the traffic distribution, namely $\phi_0 \xrightarrow{p} \phi_1$, where ϕ_1 is the distribution of the data after the enforcement of the policy. We say that a policy p is *effective* when the distribution ϕ_1 differs from the distribution of traffic data ϕ_0 before policy enforcement.

We assume a training set of N stationary samples $\tilde{x}_1, \dots, \tilde{x}_N \stackrel{\text{i.i.d.}}{\sim} \phi_0$ from the initial distribution ϕ_0 , where each sample corresponds to a distinct day. Then, we enforce a policy p and start collecting samples $x_1, x_2, \dots \stackrel{\text{i.i.d.}}{\sim} \phi_1$ from the traffic scenario generated by p . Our goal is to determine whether the impact of a policy

p on the network has a statistically significant effect by testing for $\phi_1 \neq \phi_0$. Also, we aim to perform the testing sequentially, meaning that a decision should be taken for every incoming sample x_t , and to measure how many samples after policy enforcement are needed to conclude that the policy is effective. To avoid long data collections, we select candidate policies on simulated data.

4 Proposed Solution

In this section, we describe our solution to assess mobility policies in a road network. As anticipated in Sect. 1, our solution combines: (i) a micro-traffic simulator, and (ii) a nonparametric change-detection algorithm. To avoid long acquisition campaigns, we generate realistic traffic data using a traffic simulator. In this way, we can simulate large datasets of traffic data under arbitrary mobility policies, which would otherwise require months of measurements on the real world network. Using simulated data, we model the behavior of the network in a basic scenario with a change-detection algorithm, obtaining an estimate $\hat{\phi}_0$ of the statistical distribution ϕ_0 of the traffic data in the basic scenario. Each mobility policy p gives rise to a different traffic scenario with data distribution ϕ_1 . We monitor a stream of traffic data in search of statistical evidence to assert that $\phi_1 \neq \phi_0$, meaning that p significantly impacts the network's traffic.

4.1 Traffic Simulator

We select candidates policies employing a stochastic simulator to generate traffic data according to a user-defined mobility policy. Thus, we implement the city road network in the Eclipse/SUMO traffic micro-simulator [18], a popular open-source tool that enables a detailed representation of traffic dynamics. Since we are mostly interested in scenarios where mobility and freight traffic interact we define standard vehicle classes (i.e., general mobility and logistic), each characterized by specific size, speed, acceleration, etc. In the simulation, each vehicle represents an equivalent unit of the corresponding class (truck or car).

The simulator models the city road network with several *edges*, which are the basic components of the road system (e.g., a portion of a street or a roundabout). We denote by $\mathcal{S}$ the set of all the network edges. For each edge $s \in \mathcal{S}$, the simulator generates the average values, over intervals of arbitrary length, for the following six variables: (i) number of cars (`num_cars`), (ii) speed of cars (`speed_cars`), (iii) flow (i.e., vehicles that left the edge) of cars (`flow_cars`), (iv) number of trucks (`num_trucks`), (v) speed of trucks (`speed_trucks`), and (vi) flow of trucks (`flow_trucks`). We generate simulations for several days by independent runs of the simulator. Thus, we obtain a sequence of day-indexed traffic data $x_1, x_2, \dots \in \mathbb{R}^d$, where d is the total number of variables to monitor.

4.2 Input Preparation

For each simulated day, we generate 6 traffic variables (`num_cars`, `speed_cars`, `flow_cars`, `num_trucks`, `speed_trucks` and `flow_trucks`) for each edge in $\mathcal{S}$,

where the cardinality $|\mathcal{S}|$ is typically in the order of hundreds. Monitoring streams of variables with hundreds of dimensions is challenging. Moreover, most of the edges are strongly correlated due to the simulation dynamics, thus not very statistically informative. Therefore, we select a small subset $\mathcal{S}_m \subset \mathcal{S}$ of edges of interest, namely the monitoring points of the network. We define $\mathcal{S}_m$ in agreement with traffic experts, including edges that are crucial for assessing mobility policies in the road network under study. To facilitate interpretability, we monitor both traffic at all edges and separately at each edge $s \in \mathcal{S}_m$ individually, resulting in $|\mathcal{S}_m|$ distinct monitoring problems with 6-dimensional data.

The raw data provided by the simulator needs to be preprocessed before it can be used in a change-detection algorithm. A mobility policy or the current traffic regulations may restrict the access to certain streets to only cars, resulting in simulated trucks related variables (`num_trucks`, `speed_trucks` and `flow_trucks`) that are constant equal to zero. Since these variables are useless for the monitoring problem, we remove them from the dataset. Moreover, we note that 2 variables, namely `flow_cars` and `flow_trucks`, take on discrete values in any scenario. Since histogram-based methods (including QT-EWMA [11]) struggle with discrete variables, we add a small Gaussian noise to these two variables, exclusively for change-detection purposes.

4.3 Change-Detection Method

We frame the policy assessment problem as a special change-detection problem where the change point is known and corresponds to the enforcement of a mobility policy, where the stationary distribution ϕ_0 corresponds to the current scenario in the real-world network. We use the training set $\{\tilde{x}_1, \dots, \tilde{x}_N\}$ to fit an estimate $\hat{\phi}_0$ of the distribution ϕ_0 of the basic scenario. Then, given a policy p , we configure the simulator according to p and generate observations $x_1, x_2, \dots \in \mathbb{R}^d$, which we model as i.i.d. samples from the unknown distribution ϕ_1 of the traffic scenario generated by the policy p .

For each new incoming sample x_t , an online change-detection algorithm assesses whether the sequence $\{x_1, \dots, x_t\}$ provides sufficient statistical evidence of a change in the data-generating process, namely whether the policy is effective. Specifically, at each time instant t , the algorithm computes a statistic T_t , based on the estimated initial distribution $\hat{\phi}_0$, and applies a decision rule comparing it to a time-dependent threshold h_t . The detection time t^* is defined as the first instant of time when the statistic T_t exceeds the threshold h_t , that is:

$$t^* := \min \{t \in \mathbb{N} : T_t > h_t\}. \quad (1)$$

We recall that a policy is always enforced at the beginning of the data simulation process. Thus, when p induces a real change $\phi_0 \rightarrow \phi_1$, such that $\phi_0 \neq \phi_1$, the detection time t^* corresponds to the *detection delay*, since the policy is enforced at the beginning of the monitored stream.

A fundamental metric of any online change-detection algorithm is the expected number of time steps before a false detection, the Average Run Length (ARL_0) [2] is defined as:

$$\text{ARL}_0 := \mathbb{E}_{x \sim \phi_0}[t^*],$$

where the expectation is with respect to the training distribution ϕ_0 , namely the data stream comes from the basic traffic scenario. Our goal is to estimate the average detection delay t^* , controlling the ARL_0 , namely obtaining an *empirical* ARL_0 close to a user-specified target ARL_0 . We note that controlling the ARL_0 is of utmost importance, since it is the sequential counterpart of the Type I error probability in offline hypothesis testing. An ARL_0 that is too small would result in numerous false alarms, causing us to incorrectly conclude that a policy is effective when it is not, making the detection delay meaningless.

We detect changes in traffic data using the QuantTree Exponentially Weighted Moving Average (QT-EWMA) algorithm [11]. During training, the algorithm uses the training set $\tilde{x}_1, \dots, \tilde{x}_N \stackrel{\text{i.i.d.}}{\sim} \phi_0$ to approximate the unknown distribution ϕ_0 of the basic scenario, constructing a histogram with K bins, which are created by partitioning $\mathbb{R}^d$ along random directions (a QuantTree [3]). The splitting points are defined so that each bin contains a user-specified quantile of the training set, resulting in a uniform partition of the data into bins. During monitoring, for each new sample x_t , QT-EWMA updates a statistic T_t in an incremental manner and compares it against a time-dependent threshold h_t . The details of the QT-EWMA algorithm can be found in [11].

We choose QT-EWMA since it is a state-of-the-art online nonparametric change-detection method suitable for high-dimensional data with arbitrary distribution. Remarkably, QT-EWMA provides theoretical guarantees on the ARL_0 , allowing one to control the false alarm performance of the algorithm. The thresholds h_t do not depend on the stationary distribution ϕ_0 or the number d of dimensions. They depend exclusively on the size of the training set and the number of histogram bins, and need to be precomputed using Monte Carlo simulations. In our experiments, we use the publicly available thresholds that can be found in the official repository¹.

5 Experiments

5.1 The Genova Port-City Scenario

We choose as case study the port-city context of Genova, a town in north-west Italy hosting one of the most important container ports in the Mediterranean Sea. The area considered is depicted in Fig. 1, and is characterized by the presence of port gates and highway toll booths in close proximity, thus it is an area of intense interaction between logistics and mobility traffic. In the figure, the ‘T’ signs denote a traffic entry/exit point. In particular, T1 corresponds to mobility vehicles to/from western parts of the town through Lungomare Canepa, and trucks to/from the Aeroporto highway toll booth. T2 is the direction to/from the Genova Ovest highway toll booth. T3 is a direction to/from western parts

¹ <https://github.com/diegocarrera89/quantTree>.

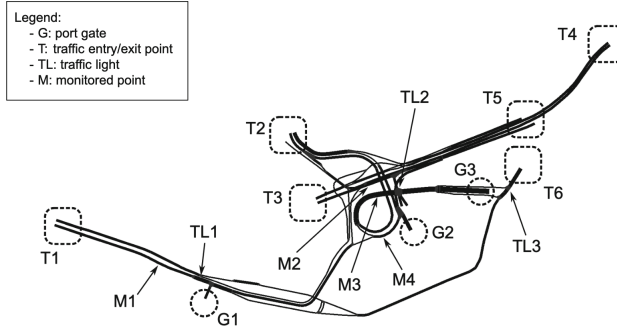


Fig. 1. The Genova's port-city network under study.

of the town interested mainly by mobility traffic and ferry passengers. T4, T5 and T6 are various entry/exit points for eastern parts of the town, interested only by mobility traffic. The points denoted by 'G' are port gates. Specifically, G1 and G2 are the Etiopia and San Benigno gates, respectively, devoted mainly to freight traffic, while G3 is the Alberatzzi gate, mainly devoted to ferry passengers. The 'TL' signs mark the position of traffic lights that will be used to define changes of policies as will be detailed below. The points at which the variables of interest are monitored are indicated by 'M', and are respectively called **Lungomare Canepa**, **Via di Francia**, **Elicoidale Upstream** and **Elicoidale Downstream**. These four edges of interest represent edges in which general mobility traffic mixes with logistics (M1, M3 and M4) and ferry traffic, giving rise to possible congestion.

In order to define the traffic demand between origins and destinations, we assume different behavior for trucks and mobility vehicles. In particular, we assume that trucks follow fixed routes between gates and highway toll booths. We considered 6 truck routes, namely (G2 $\rightarrow$ T2), (T2 $\rightarrow$ G2), (T1 $\rightarrow$ G1), (T1 $\rightarrow$ G2), (G1 $\rightarrow$ T1) and (G1 $\rightarrow$ T2). General vehicles and ferry passengers instead are assumed to move freely among the origins and destinations, adapting the route depending on current traffic conditions. For this kind of traffic we considered a total of 19 origin/destination pairs, i.e., (T1 $\rightarrow$ T4), (T1 $\rightarrow$ T5), (T1 $\rightarrow$ T6), (T1 $\rightarrow$ T2), (T3 $\rightarrow$ T2), (T3 $\rightarrow$ T4), (T3 $\rightarrow$ T5), (T3 $\rightarrow$ G3), (T2 $\rightarrow$ T1), (T2 $\rightarrow$ G3), (T2 $\rightarrow$ T4), (T2 $\rightarrow$ T5), (T5 $\rightarrow$ T3), (T4 $\rightarrow$ T3), (T4 $\rightarrow$ T2), (T4 $\rightarrow$ T1), (G3 $\rightarrow$ T2), (T6 $\rightarrow$ T1) and (T6 $\rightarrow$ T2). Overall, these correspond to a 25-dimensional vector of inflow rates F used to drive the simulation scenario.

5.2 Simulator Configuration and Datasets Generation

To test our methodology, we define two basic scenarios by setting Poissonian distributions as input inflows of the simulator and we then record the 25-dimensional vector of hourly inflow rates F . Adopting Poissonian inflows is a standard in queueing and traffic systems, and yield realistic discrete values. The first basic scenario (denoted by A) is intended to simulate the traffic during working days, while the second (denoted by B) is for weekends, which are character-

Table 1. Empirical ARL_0 with respect to training set size, target ARL_0 and basic scenario distribution. The empirical ARL_0 is close to the target ARL_0 in all cases.

N	ARL_0	Average Empirical ARL_0		
		ϕ_0^A	ϕ_0^B	ϕ_0^{AB}
64	500	490.4	482.6	487.3
	1000	998.0	984.4	984.5
128	500	502.8	489.5	509.0
	1000	1017.6	1003.1	1006.5
256	500	495.0	510.9	496.0
	1000	986.4	1006.8	1017.5

ized by less traffic. We additionally consider a mixture basic scenario (denoted by AB) by sampling traffic data from scenario A and scenario B, with a 70-30 proportion. We denote the data distributions under these three basic scenarios by ϕ_0^A , ϕ_0^B and ϕ_0^{AB} , respectively. For each basic scenario, we consider the following three policies with increasing impact:

- *Policy 1* (p_1): little shifting of a single traffic light’s phase (TL2).
- *Policy 2* (p_2): shifting the phase of three traffic lights (TL1, TL2, TL3).
- *Policy 3* (p_3): two traffic lights (TL1, TL2) are now dynamic, adjusting the phases based on current traffic instead of using fixed phase programs.

To generate the datasets, for each scenario, we simulate 1000 days with 1 h of simulation time. More specifically, we considered a simulated time of 4200 s per day, the first 600 s are considered as warm up to populate the empty network, and then discarded. Then, to generate the six traffic data variables of interest described in Sect. 4.1, we aggregate the values from the simulator. Notice that, given the short time interval we consider every day, the traffic at the gates and the highway booths contain all the required information for the logistic flows, while other port-related factors such as incoming ships/trains or yard storages would act on longer time frames.

5.3 Results

In this section, we present the results obtained from the experiments described above. We start by measuring empirical ARL_0 , then show the ability of our methodology to assess the effectiveness of policies.

Table 1 shows that the empirical ARL_0 consistently aligns with the target across all training set sizes N , target ARL_0 values, and data distributions, confirming the reliability of our method. The agreement improves when N increases, with $N = 256$ showing the best match. Slight deviations, particularly in the ϕ_0^B setting at $ARL_0 = 1000$, suggest some sensitivity to data characteristics. Overall, the method controls ARL_0 under all conditions, even small N yielding reliable

Table 2. Average detection delays (in days) during weekdays. Policy p_1 is weakly effective since delays are large in all segments, policy p_2 is more effective in few edges, while policy p_3 is effective at network level and in all edges except **Elicoidale Upstream**.

		Average Detection Delay						
Policy	N	ARL ₀	Elicoidale Upstream	Elicoidale Downstream	Lungomare Canepa	Via di Francia	All 4 edges	
p_1	64	500	478.9	238.9	556.9	49.3	251.9	
		1000	935.3	497.2	1050.1	73.7	405.1	
	128	500	488.2	202.6	615.3	35.5	170.2	
		1000	876.5	358.3	1068.3	46.4	255.4	
	256	500	471.8	158.8	701.9	30.6	130.7	
		1000	922.0	268.6	1207.5	35.8	186.1	
p_2	64	500	489.8	35.7	132.5	16.0	42.1	
		1000	982.9	50.1	223.5	19.2	55.4	
	128	500	531.8	26.5	79.8	13.9	26.5	
		1000	926.5	32.7	112.7	16.0	38.7	
	256	500	520.3	22.4	59.7	12.2	22.7	
		1000	950.0	27.6	74.4	13.7	27.7	
p_3	64	500	323.4	5.0	22.2	8.7	7.7	
		1000	622.6	5.0	26.0	9.8	8.2	
	128	500	266.1	5.0	16.4	7.8	6.6	
		1000	455.5	5.0	18.7	8.6	7.3	
	256	500	194.9	5.0	14.4	7.5	6.4	
		1000	335.0	5.0	16.1	8.2	6.9	

results. Tables 2, 3, and 4 report the detection delays for the three basic scenarios, namely weekdays (ϕ_0^A), weekends (ϕ_0^B) and mixture (ϕ_0^{AB}). We present the detection delays for each of the three different policies (p_1 , p_2 , and p_3), as a function of the training set size N and target ARL₀. In particular, we report in each column the detection delays for each monitored edge individually $s \in \mathcal{S}_m$, as well as for their aggregation $\mathcal{S}_m$. The results reveal a strong dependency between the type of policy change and the timeliness of the detection of alterations in the traffic flow.

In the weekday-only scenario (Table 2), detection delays are longest under p_1 , which involves a minor phase shift at a single traffic light. This is most evident in **Lungomare Canepa**, where delays consistently exceed the target ARL₀, indicating that p_1 only produces subtle traffic changes and is weakly effective. Increasing the training set size N yields limited improvement, as the underlying change remains minimal. In contrast, p_2 , which modifies three signals, shows shorter delays, especially in **Elicoidale Downstream**, where detection occurs in less than 50 days, confirming that broader interventions improve responsiveness. p_3 , which implements dynamic signal control, further reduces delays: in **Elicoidale Downstream**, detection occurs in just five days, the minimum for QT-EWMA. Even **Lungomare Canepa**, previously slow to respond, shows a significant improvement under p_3 , confirming the effectiveness of system-wide adaptive changes, whereas **Via di Francia** demonstrates consistently short detection delays across all policies, indicating a strong and immediate sensitivity to

Table 3. Average detection delays (in days) during weekends. All policies appear less effective because of reduced traffic on weekends.

		Average Detection Delay							
Policy	N	ARL ₀	Elicoidale Upstream	Elicoidale Downstream	Lungomare Canepa	Via di Francia	All 4 edges		
p_1	64	500	515.1	353.0	558.1	537.7	441.3		
		1000	936.7	671.0	1027.7	1034.6	906.7		
	128	500	555.5	301.9	623.2	594.3	486.0		
		1000	1016.6	608.8	1079.7	1079.3	923.0		
	256	500	604.3	288.7	712.4	664.9	512.0		
		1000	1079.2	517.9	1130.6	1191.5	923.6		
p_2	64	500	511.2	194.8	409.4	530.7	328.0		
		1000	983.7	366.9	749.3	1053.4	646.4		
	128	500	525.4	144.4	408.8	590.6	323.5		
		1000	1024.8	268.3	736.6	1020.5	563.6		
	256	500	577.7	121.1	425.7	676.6	290.2		
		1000	1089.9	187.3	685.2	1146.1	535.0		
p_3	64	500	505.9	14.1	198.0	555.9	131.7		
		1000	926.2	16.5	310.9	1022.6	228.8		
	128	500	539.7	12.5	149.4	594.1	99.7		
		1000	1026.5	14.4	228.8	1075.5	147.8		
	256	500	565.7	12.0	112.5	687.6	83.0		
		1000	1038.5	13.6	153.2	1199.7	126.8		

all 3 policies. However, detection remains slow in **Elicoidale Upstream**, suggesting limited local impact. Across all the policies, higher ARL₀ values increase the detection delay, reflecting the expected trade-off between false alarms and responsiveness. This is most pronounced in p_1 due to its weak signals, while p_3 maintains rapid detection even at higher ARL₀, confirming the distribution change is very apparent.

The weekend-only (Table 3) and mixed scenarios (Table 4) show similar behaviors, but with added complexity. Detection is less timely during weekends, probably due to lower traffic volumes. Without congestion, variations in vehicle counts and speeds are minimal, limiting the effectiveness of change detection. In near free-flow conditions, signal timing changes have little influence on overall dynamics, so even structural interventions like p_3 may go largely undetected. This highlights a key dependency: while high congestion amplifies the effects of policy changes, low traffic volumes reduce their detectability.

The location of affected road edges highlights the localized impact of policy changes. In the weekday-only scenario, **Elicoidale Downstream** is most impacted, followed by **Via di Francia**, **Lungomare Canepa**, and lastly **Elicoidale Upstream**. This suggests that an edge's response depends not only on detection timing but also on its proximity to the modified signals and local traffic patterns. Edges closer to the changes react more clearly and quickly, while distant ones like **Elicoidale Upstream** show weaker effects. This spatial variation underscores the need to consider localized traffic dynamics when evaluating mobility policies.

Table 4. Average detection delays (in days) on the mixed scenario. All policies appear less effective because data are influenced by weekend samples.

		Average Detection Delay					
Policy	N	ARL ₀	Ellicoidale Upstream	Ellicoidale Downstream	Lungomare Canepa	Via di Francia	All 4 edges
p_1	64	500	502.4	295.2	555.1	109.6	314.5
		1000	959.3	541.1	1078.1	169.0	538.1
	128	500	520.4	255.6	623.7	76.6	293.0
		1000	967.5	454.2	1114.2	109.7	486.5
	256	500	558.1	222.5	691.0	59.2	256.1
		1000	1025.5	377.9	1215.9	75.4	436.1
p_2	64	500	523.3	79.2	225.2	37.5	123.7
		1000	983.1	120.6	383.7	48.4	187.8
	128	500	523.4	53.0	146.3	29.2	87.4
		1000	971.2	73.1	232.1	35.5	127.7
	256	500	570.8	42.7	106.8	24.2	61.0
		1000	1054.9	52.6	145.9	28.8	87.4
p_3	64	500	381.5	5.5	38.8	27.3	19.2
		1000	756.8	5.9	51.3	33.5	24.7
	128	500	536.1	5.4	26.0	19.8	14.1
		1000	619.8	5.6	31.0	23.7	16.3
	256	500	323.3	5.3	21.3	17.7	12.1
		1000	520.1	5.6	23.7	20.1	14.0

Our experiments confirm that the magnitude of policy interventions significantly influences detection performance, thus the time needed to confirm a policy is effective. Small traffic light phase shifts in p_1 are hard to detect, while broader interventions—such as modifying multiple intersections (p_2) or implementing adaptive signals (p_3)—substantially reduce detection delays. Detection is also affected by the choice of target ARL₀, with higher values increasing delay, and by traffic flow conditions, with weekday congestion enhancing detectability and weekend free-flow hindering it. Our analysis demonstrates that our methodology can effectively differentiate between weak, moderate, and strong interventions based on detection latency, offering a statistically rigorous improvement over traditional ex-post evaluations, which often miss short-term or spatially localized effects [13, 20]. Notably, p_3 consistently yielded rapid detections, even under strict false alarm thresholds, supporting prior findings that dynamic, coordinated policies outperform isolated changes [6]. In contrast, the limited impact of p_1 reflects the ineffectiveness of marginal measures, echoing real-world challenges like those observed in Milan [20]. Effectiveness also varied by time and location—detections were slower during low-demand periods and differed across road segments—aligning with insights from [19, 21].

6 Conclusions

We presented a general methodology for assessing mobility policies in an urban road network, with a focus on areas characterized by the competition of mobility and logistics flows, such as port-adjacent city centers. Using a change-detection algorithm in a simulated environment, our approach assesses a priori the impact of candidate policies without relying on extensive real-world data collection. Through online change-detection, the algorithm can promptly identify changes in the data-generating process, allowing for timely and data-driven assessment of policies. Applied to the port-city context of Genova, our experiments showed how the methodology is able to promptly identify effective policies, namely those requiring few days of monitoring to confirm the change. In future work, we will apply QT-EWMA to the residuals of surrogate models (e.g., neural networks) predicting traffic at each location. We believe that this approach may enhance the sensitivity to subtle deviations while filtering out expected variability.

References

1. Aditjandra, P.T., Galatioto, F., Bell, M.C., Zunder, T.H.: Evaluating the impacts of urban freight traffic: application of micro-simulation at a large establishment. *Eur. J. Transp. Infrastruct. Res.* **16**(1) (2016)
2. Basseville, M.: Detection of Abrupt Changes: Theory and Application, vol. 2, pp. 3–11 (1993)
3. Boracchi, G., Carrera, D., Cervellera, C., Macciò, D.: QuantTree: histograms for change detection in multivariate data streams. In: International Conference on Machine Learning, pp. 639–648. PMLR (2018)
4. Cervellera, C., Macciò, D., Rebora, F.: Copula-based scenario generation for urban traffic models. *Expert Syst. Appl.* **210**, 118389 (2022)
5. Cervellera, C., Macciò, D., Rebora, F.: Model predictive control of port-city traffic interactions over shared urban infrastructure. *IEEE Trans. Control Syst. Technol.* **32**, 688–695 (2024)
6. de Souza, F., Minatto Saucedo, R., Mousavizadeh, O., Castelan Carlson, R., Keyvan-Ekbatani, M.: On the evaluation and selection of network-level traffic control policies: perimeter control, TUC, and their combination. *Transp. Res. Part A Policy Pract.* **186**, 104161 (2024)
7. Djenouri, Y., Belhadi, A., Lin, J., Djenouri, D., Cano, A.: A survey on urban traffic anomalies detection algorithms. *IEEE Access* **7**, 12192–12205 (2019)
8. Dorokhin, S., Artemov, A., Likhachev, D., Novikov, A., Starkov, E.: Traffic simulation: an analytical review. *IOP Conf. Ser. Mater. Sci. Eng.* **918**, 012058 (2020)
9. Ejercito, P.M., Nebrija, K.G.E., Feria, R.P., Lara-Figueroa, L.L.: Traffic simulation software review. In: 2017 8th International Conference on Information, Intelligence, Systems & Applications (IISA), pp. 1–4. IEEE (2017)
10. Fontes, T., et al.: Are HOV/eco-lanes a sustainable option to reducing emissions in a medium-sized European city? *Transp. Res. Part A Policy Pract.* **63**, 93–106 (2014)
11. Frittoli, L., Carrera, D., Boracchi, G.: Nonparametric and online change detection in multivariate datastreams using QuantTree. *IEEE Trans. Knowl. Data Eng.* **35**(8), 8328–8342 (2022)

12. Gao, S., Ran, Q., Su, Z., Wang, L., Ma, W., Hao, R.: Evaluation system for urban traffic intelligence based on travel experiences: a sentiment analysis approach. *Transp. Res. Part A Policy Pract.* **187**, 104170 (2024)
13. de Grange, L., Troncoso, R., González, F.: An empirical evaluation of the impact of three urban transportation policies on transit use. *Transp. Policy* **22**, 11–19 (2012)
14. Gu, Y., Deakin, E., Long, Y.: The effects of driving restrictions on travel behavior evidence from Beijing. *J. Urban Econ.* **102**, 106–122 (2017)
15. Karjalainen, L.E., Juhola, S.: Urban transportation sustainability assessments: a systematic review of literature. *Transp. Rev.* **41**(5), 659–684 (2021)
16. Kong, Y., Liu, J.: Sustainable port cities with coupling coordination and environmental efficiency. *Ocean Coast. Manag.* **205**, 105534 (2021)
17. Liu, X., Dijk, M.: The role of data in sustainability assessment of urban mobility policies. *Data Policy* **4**, e2 (2022)
18. Lopez, P.A., et al.: Microscopic traffic simulation using sumo. In: 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 2575–2582. IEEE (2018)
19. Lu, Q.L., Qurashi, M., Antoniou, C.: Simulation-based policy analysis: the case of urban speed limits. *Transp. Res. Part A Policy Pract.* **175**, 103754 (2023)
20. Maranzano, P., Fassò, A., Pelagatti, M., Mudelsee, M.: Statistical modeling of the early-stage impact of a new traffic policy in Milan, Italy. *Int. J. Environ. Res. Public Health* **17**(3), 1088 (2020)
21. Merk, O., Dang, T.T.: The effectiveness of port-city policies: a comparative approach (2013)
22. Pais, F.V., Nogueira, B., Pinheiro, R.G.: Performance evaluation of urban traffic using simulation: a case study in Brazil. *IEEE Lat. Am. Trans.* **21**(12), 1275–1281 (2023)
23. Schipper, C., Vreugdenhil, H., De Jong, M.: A sustainability assessment of ports and port-city plans: comparing ambitions with achievements. *Transp. Res. Part D: Transp. Environ.* **57**, 84–111 (2017)
24. Shankar, R., Choudhary, D., Jharkharia, S.: An integrated risk assessment model: a case of sustainable freight transportation systems. *Transp. Res. Part D* **63**, 662–676 (2018)
25. Shi, J., Guo, X., Hu, X.: Engaging stakeholders in urban traffic restriction policy assessment using system dynamics: the case study of Xi'an city, China. *Sustainability* **11**(14), 3930 (2019)
26. Wen, L., Bai, L.: System dynamics modeling and policy simulation for urban traffic: a case study in Beijing. *Environ. Model. Assess.* **22**, 363–378 (2017)
27. Zheng, Y., Zhao, J., Shao, G.: Port city sustainability: a review of its research trends. *Sustainability* **12**(20), 8355 (2020)
28. Zhou, Y., et al.: Port-related emissions, environmental impacts and their implication on green traffic policy in Shanghai. *Sustainability* **12**(4162), 1–17 (2020)



Automatic Long-Term Forecasting of Mortality Rates with Generalized Regression Neural Networks

Giovanna Apicella¹ , Michele La Rocca² , Cira Perna²  ,
and Marilena Sibillo² 

¹ Department of Economics and Statistics, University of Udine, Udine, Italy
giovanna.apicella@uniud.it

² Department of Economics and Statistics, University of Salerno, Salerno, Italy
{larocca,perna,msibillo}@unisa.it

Abstract. This paper focuses on the problem of automatically forecasting mortality rates over long time horizons. In the context of the Lee-Carter model, an approach based on general regression neural networks is presented and discussed. Our proposal preserves the LC defining parameters and structure, adds flexibility at reduced costs in terms of complexity, and requires a weak human intervention for the identification of the optimal parameters. Moreover, GRNN models need relatively few data to train, an advantage useful in actuarial data. An application to real data shows that an additive GRNN model has, in general, better forecasting performances than both the multiplicative GRNN model and the KNN model, taken as a benchmark. Furthermore, between two different long term forecasting strategies, the analysis highlights how, in general, MIMO is preferable to the classic recursive procedure.

Keywords: Lee-Carter model · Generalized Regression Neural Networks · Long-term forecasting strategies

1 Introduction

Generally, in the actuarial practice, it is crucial to obtain mortality forecasts over long time horizons. The Lee-Carter (henceforth LC) model is widely recognised in the literature as being a keystone for stochastic mortality modeling. Along with its extensions, the LC model is deemed a useful baseline to improve understanding of longevity risk in a variety of scientific fields and beyond academia [1], mainly thanks to the simplicity of the modeling framework, the plain interpretation of the involved parameters and the quite good accuracy of mortality forecasts over rather long horizons [2], being of crucial relevance for policy decisions.

The LC model was introduced by [3] as a parsimonious model for long-run projections of both the level and the age pattern of mortality [4]. In particular, in the LC parameterization, a single stochastic time-varying index k_t , expressing

the level of mortality, drives the dynamics of mortality over time. Accordingly, the modeling of such a time index plays a fundamental role to prevent forecasts and prediction intervals for mortality rates from being biased, and thus less robust and less reliable. As explained in [4], long-term extrapolation can be an hazardous task, but it is less risky if the time-varying index is approximately linear over the fitting period and thus if the underlying mortality data show a regular change and small variability throughout the period. However, in practice, the linearity of change may encounter violations, for instance when analysing data relative to the twentieth century in developed countries due to an observed changing pace of longevity improvements over time [5]. If the linearity of mortality decline does not hold systematically, the drift term may be misaligned with respect to the recent observed trend in the time-varying index and the volatility of the $\hat{k}_t$ may be inflated, thus affecting the validity of the point forecasts and of the probabilistic prediction intervals provided by the LC model (for instance, too wide prediction intervals).

Given that, in many cases, the LC time index presents a nonlinear behaviour [6,7] that is difficult to trace back to known functional forms, the use of neural network models is increasingly frequent, since they can be seen as flexible nonlinear functional tools suitable for approximating arbitrary mappings.

Among the many alternatives proposed in this context, recent advances in mortality modeling and prediction have exploited the neural network models (see [6] and the references therein). However, these methods require iterative approaches for their training and the optimal selection of a large number of hyperparameters. In the context of actuarial data, mortality rates are typically collected annually and reliable data often refer to very short periods. The limited sample size, useful for building a reliable predictive model, of a few dozen observations, could make difficult, if not impossible, to use neural networks with complex structures. Furthermore, these models require a strong human intervention both for the specification of the network topology and for the identification of the optimal parameters on which each architecture depends, making them difficult to automate.

In this paper an efficient and fast tool to automatically forecast the LC time series of observed mortality rates is presented and discussed. It is based on general regression neural networks (GRNN), a kind of radial basis function networks developed in [8]. GRNN is a powerful regression tool with a dynamic network structure which, despite the simplicity of the structure and of its implementation, has several clear advantages that make it valuable and attractive in time series prediction. Firstly, GRNN models are characterized by a fast single-pass learning, regardless of the sample size. Therefore, no backpropagation is required, and the models are always able to converge globally, avoiding the problem of possible local solutions. Secondly, GRNN models can achieve high accuracy in the estimation since they use Gaussian functions in estimation and in prediction [9]. Thirdly, their structures are surprisingly simple, with only one hidden layer and the number of neurons equal to the number of training samples. They have only one tuning parameter to control the overall generalization. Finally, as shown in

[10], GRNNs generally achieve slightly better forecasting performance than both multilayer perceptrons and feed-forward neural networks.

The GRNN models can be appropriately used to derive the forecasts of mortality rates at a specific forecast horizon. However, when the approach is used for large future horizons, some peculiar aspects need to be considered, to obtain reliable forecasts [11]. In particular, in this paper, we have implemented two different strategies for long-term forecasting. The first is the classical recursive multi-step strategy which involves the use of the predicted values from previous time steps as input to forecast the values for the subsequent time steps. The second is a Multi-input Multi-output strategy, proposed in [12], in which forecasts are derived as a one-step by using a multi-output model.

The paper is organized as follows. In Sect. 2, we introduce the predictor structure of the LC model. In Sect. 3, we discuss our method and assess its forecast accuracy. In Sect. 4, we discuss an application to real mortality data. Section 5 concludes the paper.

2 Lee Carter Model

The LC model was introduced in [3] as a parsimonious model for long-run projections of both the level and the age pattern of mortality, based on the stochastic description of the log of a time series of death rates $m_{x,t}$ [4].

The death rate $m_{x,t}$ is an age- and time-dependent mortality measure being traditionally referred to as *central death rate*, in that it expresses the expected number of deaths between ages x and $x + 1$, in the time interval $[t, t + 1]$, out of the central number of people in the same age- and time- range.

Lee and Carter [3] proposed to model the dynamics of mortality change over time, by means of the following parameterization, where a single stochastic time-varying index k_t , expressing the level of mortality, plays a crucial role:

$$\log m_{x,t} = a_x + b_x k_t + e_{x,t}, \quad (1)$$

where a_x denotes the general shape of mortality across ages, while b_x defines the sensitivity of the logged death rate at age x to changes in k_t . $e_{x,t}$, with mean 0 and variance σ_e^2 , expresses age-specific historical effects being not captured by the model and thus the residual variance that is not explained by the k_t parameter. The model assumes that all the age-specific death rates are driven by the same time index k_t ; accordingly, all age-specific death rates move upward or downward together, even though the magnitude of such changes may vary across ages. The age-specific rates of mortality improvement are given by $b_x (dk/dt)$, where dk/dt is the derivative of k_t with respect to time. If such a derivative is constant, k_t declines linearly over time, thus implying that each age-specific rate of mortality improvement declines at the own age-specific exponential rate, proportional to b_x , conditionally on the speed of the decline in k_t [2].

Lee and Carter normalized the b_x 's to sum to unity and the k_t 's to sum to 0. Consequently, the a_x 's are the averages over time of the logged death rates and, by means of the singular value decomposition (SVD) method, one least squares

solution to Eq. 1 can be identified, after subtracting the averages over time of the logged death rates. [3] incorporated a second-step adjustment of the fitted time-varying index $\hat{k}_t$, to ensure the match between the fitted and the observed death rates in each year of the calibration window.

The estimates of $\hat{k}_t$ over the fitting period are modeled and forecast through standard statistical methods; point and interval forecasts of mortality rates are then obtained by combining the forecasts of the general level of mortality with the estimated age effects. For instance, [3] adopted a random walk with drift to model the time-varying index for U.S. mortality data spanning calendar years from 1933 to 1989. As shown in Eq. 1, in the LC model, the stochastic process for $m_{x,t}$ is driven by the model chosen to describe $\hat{k}_t$. Life tables can be derived from the point forecasts of age-specific death rates, based on the forecasts of $\hat{k}_t$. Analogously, the probability intervals on the forecasts of the time index can be used to derive intervals for the forecasts of age-specific death rates.

3 Mortality Forecasting with Generalized Regression Neural Networks

Given the observed path of the time series $\hat{k}_t$, $t = 1, \dots, T$, one of the main goals is to derive its forecasts at a specific horizon h , conditioned on the information set available up to time T ; that is $\hat{k}_{T+h|T}$.

The traditional approach assumes that $\hat{k}_t$ is a linear time series and that the distribution of the error process is known. However, the derived forecasts could be adversely affected by departures from these assumptions. For example, the standard Gaussian Box Jenkins method may perform poorly, both in the case of a skewed bimodal error distribution and in the case of the presence of nonlinearities in the observed time series. As shown in [6], the estimated mortality time series index may have a complex and potentially non-linear behavior, which is difficult to trace back to a predefined linear functional form.

In the following, we assume that $\hat{k}_t$, $t = 1, \dots, T$ can be decomposed in

$$\hat{k}_t = TR_t + g(\mathbf{x}_t) + \varepsilon_t \quad (2)$$

where TR_t is the trend, which allows to capture the long-run pattern observed in mortality rates time series, $g(\cdot)$ is an unknown (possibly nonlinear) function assumed to be continuously differentiable and defined on a compact subset of $\mathbb{R}^m$. The error terms ε_t are assumed to be *i.i.d.* innovations with zero mean and finite variance.

Note that, since $\hat{k}_t$ generally tends to show a trend magnitude which does not vary over time, the additive model (2) seems to be the best specification. However, if the trend behaves differently, also a multiplicative model could be considered:

$$\hat{k}_t = TR_t \cdot g(\mathbf{x}_t) \cdot \varepsilon_t \quad (3)$$

We also assume that $\mathbf{x}_t$ is a vector of m lagged values of $\hat{k}_t$; that is

$$\mathbf{x}'_t = (\hat{k}_{t-1}, \dots, \hat{k}_{t-m}) \quad (4)$$

Once the detrended time series of mortality rates is obtained, the function $g(\cdot)$ can be approximated by using a GRNN. This neural network, proposed in [8], is a variant of a radial basis neural network based on kernel regression networks and is widely used in classification and regression problems.

GRNN is an artificial neural network made up of three layers: the input, hidden and output. The input layer consists of the explanatory variables of the model which, in this context, represent the lagged values of the mortality rates; that is the vector $\mathbf{x}_t$ defined in (4). The hidden layer uses radial basis functions (RBF) to perform nonlinear transformation of the input data and has so many neurons (called pattern neurons) as the input layer. Generally, the radial basis function used in GRNN models is the multivariate Gaussian function:

$$G(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2}\right) \quad (5)$$

where $\mathbf{x}$ is a generic input vector, $\mathbf{x}_i$ is the center of a neuron, that is its associated training example (so that its output gives a measure of the closeness of the input vector to the training example), σ^2 is a smoothing parameter and $\|\cdot\|$ is a norm, typically taken to be the Euclidean distance. The output of a hidden layer neuron is related to the closeness of the input vector to the center, scaled by the smoothing parameter.

Given a training set consisting of T training patterns, the vectors $(\mathbf{x}_1, \dots, \mathbf{x}_T)$, defined in 4, and their associated T targets $\hat{k}_1, \dots, \hat{k}_T$, the GRNN output for an input pattern $\mathbf{x}$ is computed in two steps. In the first step, the hidden layer creates a set of weights associated with the closeness of $\mathbf{x}$ to the training patterns. The weights are scaled to sum to one, representing in this way the contribution of every training pattern to the final result. The weights are defined as:

$$w_t = \frac{G(\mathbf{x}, \mathbf{x}_t)}{\sum_{t=1}^T G(\mathbf{x}, \mathbf{x}_t)} \quad (6)$$

In the second step, the output is computed using a weighted average of the training targets where the weights are related to the closeness of the input to the training pattern.

Formally, the function $g(\cdot)$ can be approximated with a GRNN

$$f(\mathbf{x}) = \sum_{t=1}^T w_t \hat{k}_t \quad (7)$$

The GRNN model we propose is one of the simplest types of artificial neural network and, despite its unsophisticated structure, it has several clear advantages over alternative neural network models that make them valuable and attractive in forecasting mortality rates. They are characterized by a fast single-pass learning, regardless of the sample size, requiring no backpropagation and avoiding the

problem of possible local solutions. Moreover, they can achieve high accuracy in the estimation [9] and slightly better forecasting performance than both multi-layer perceptrons and feed-forward neural networks [10]. More importantly, they depend on only one tuning parameter which controls the overall generalization and then require a weak human intervention for its optimization.

3.1 Automatic Model Selection

In order to automatically select a GRNN forecasting model for the time series of observed mortality rates, a model selection procedure, which does not require a human intervention, is needed. It involves the selection of the input variables, useful for model identification and interpretation, and of the smoothing parameter which accounts for the trade-off between estimation bias and variability.

Since the proposed GRNN model is an autoregression, the latter problem concerns the specification of the autoregressive lags of the mortality rates. In the literature, the automatic criterion is to insert in the GRNN model as many lags as there are significant lags in the partial autocorrelation function [10]. However, it only tests for linear relationship and, as a consequence, is not able to identify significant lags in the time series of mortality rates, which may have a non linear structure.

In this context, we propose the use of the estimated Auto Distance Correlation Function $ADCF(j)$, $j = 1, \dots, J$, which measures temporal dependence structure, at lag j , of a non linear time series [13]. Since this function captures any kind of dependence, including nonlinear or non monotone dependencies, it appears natural to apply it for variable selection in non linear models [14].

A suitable value for smoothing parameter can be selected through an optimization robust tool, based on the rolling origin technique. According with this method, the forecasting origin is continually moved forward by a fixed number of observations, and forecasts are produced from each origin [15]. This technique allows obtaining several forecast errors, which gives a better understanding of how the models perform.

3.2 Long-Term Forecasting Strategies

Given the observed path of the time series $\hat{k}_t$, $t = 1, \dots, T$, one of the main goals is to derive its forecasts at a specific horizon h , conditioned on the information set available up to time T ; that is $\hat{k}_{T+h|T}$.

When the goal of the analysis is the forecasting of the time series over large future horizons, some important aspects need to be considered to obtain reliable and accurate forecasts. For example, unlike one-step ahead forecasting, accumulation of errors, reduced accuracy, and increased uncertainty could negatively affect the performance of the forecasters.

To address this complex problem, several strategies have been proposed in the literature (for a review, see [11]). They can significantly affect the performance of multi-step ahead forecasts and therefore its choice is of crucial importance.

In the following, we will refer to two different forecasting strategies which can be easily implemented to automatically derive forecasts.

The first forecasting strategy is the recursive strategy, also called iterated or multi-stage strategy. According to this approach, to perform the h -steps ahead forecast, a single model is firstly applied to obtain the one step ahead forecast. Subsequently, the value just forecasted is used as part of the input variables for forecasting the next step, using the same model. The procedure continues until the entire horizon has been considered.

The recursive strategy is the most used and natural forecasting tool and has been successfully used to forecast many real-world time series by using different machine learning models. However, it may have poor performance in multi-step ahead forecasting tasks, especially if the forecasting horizon h is long. In fact, as the time horizon increases, the procedure could be sensitive to the accumulation of errors, due to the use of intermediate forecasting values in the model to determine subsequent forecasts. Moreover, the recursive strategy may neglect possible stochastic dependencies among future values with serious consequences on the accuracy of the forecasts.

Alternatively, to solve this last problem, a Multi-Input Multi-Output (MIMO) strategy, proposed in [12], could be used for long term time series forecasting. It is a direct method according to which all the forecasts are derived as one-step by using a Multiple-Output model, so that the forecasted value is no more a scalar quantity but a vector of future values. The logic of the MIMO strategy is to preserve, in the forecasting values, the stochastic dependency which characterizes the observed time series. The MIMO strategy has been successfully used to forecast many real-world time series. One of the main advantages of this strategy is that it avoids the accumulation of errors which instead afflicts the recursive method. However, it has a reduced flexibility due to the use of a single model for all the forecast horizons [16].

4 An Application to Real Data

In this section, we introduce GRNNs in the classical scheme of the LC model. In particular, our aim is to evaluate the advantages of their use to improve the predictive ability of the LC model.

The analysis is carried out on the unisex population of five countries: Australia, Austria, Italy, Japan and Sweden. The mortality data has been collected from the website of Human Mortality Database (<https://www.mortality.org/>).

The five countries have different availability of reliable data; in particular, the Australian data cover the period 1921–2021, the Austrian and Japanese data the period 1947–2023, the Italian data the period 1872–2021 and, finally, the Swedish data the period 1751–2023. Note that the different sample sizes of the mortality rates for the considered countries allow to assess whether the length of the time series could influence the validity of the proposed approach.

The first step of our analysis is to estimate the LC model parameters a_x , b_x and k_t , using a singular value decomposition. The extracted time series $\hat{k}_t$,

appropriately adjusted to ensure that the fitted total number of deaths corresponds to the observed number of deaths in each year of the fitting period, is the starting point of our investigation.

For all the five data sets, we have initially checked whether it is reasonable to assume linearity in $\hat{k}_t$. To this aim two different statistical tests have been implemented. The first is the Wang and Van Keilegom test (WVK), proposed in [17], which verifies whether a linear trend can be assumed in mortality time series. The second is the Teräsvirta test (TE), proposed in [18], which, verifies whether a linear autoregressive dynamics is present in the data. The results, reported in Table (1), show a clear rejection of the linearity of both the trend and the autoregressive dynamics for 3 out of 5 countries (Australia, Japan and Sweden). For Austria, the autoregressive dynamics appears linear whereas Italy seems to have a linear trend (Table 1).

Table 1. Test statistics and p-values of the Wang and Van Keilegom (WVK) test for linearity trend and of the Teräsvirta (TE) test for linear dynamics

		WVK Test	TE Test
AUSTRALIA	Statistic	3.334	1.796
	pvalue	0.02	0.41
AUSTRIA	Statistic	11.282	5.665
	pvalue	0.00	0.06
ITALY	Statistic	0.741	54.859
	pvalue	0.38	0.00
JAPAN	Statistic	3.223	18.179
	pvalue	0.03	0.00
SWEDEN	Statistic	10.262	25.187
	pvalue	0.00	0.00

In bold the p-values less than 0.05.

For all the five countries, two different time series decompositions have been examined for the adjusted series $\hat{k}_t$. In particular, the additive trend as in model (2) has been considered but, for sake of comparison, a multiplicative trend, which allows the trend magnitude to vary over time, has also been implemented. The mortality rates time series has been detrended by using a transformation defined in [10]. According to this approach, in the additive decomposition, the target vector is transformed by subtracting the mean of the training pattern vector whereas in the multiplicative decomposition the value of a training example is divided by the mean of its training pattern.

The parameters of the GRNN model have been fixed by using the strategy illustrated in Sect. 3.1. In particular, the autoregression order of the input variables has been fixed by using the auto distance correlation function whereas the smoothing parameter by using the rolling origin technique. In this latter

method, the forecast origin is repeatedly moved forward by a fixed number of observations (one, in our case) and forecasts are produced from each origin [15].

To assess the forecasting accuracy of the GRNN model, we have considered three different values of the forecast horizon h , namely $h = 5, 10, 15, 20, 25$. This choice allows to evaluate both the short and the long term forecasting.

The GRNN model has been compared with a benchmark model, the K-Nearest Neighbors (KNN) regression which also relies on combining the targets of the patterns that are similar to the input pattern [19]. It is a simple method for regression problems, easy to implement and to interpret. Moreover, it is an effective method for small datasets and it is able to handle non-linear regressions with no assumptions about data distribution. Additionally, as shown in [10], KNN is the best competitor of GRNN since alternative statistical and computational intelligence models, such as exponential smoothing, ARIMA, multilayer perceptron and feed-forward neural networks with a single hidden layer and lagged inputs, have worse performances.

For both the two implemented models (GRNN and KNN), the two forecasting strategies, MIMO and Recursive, discussed in Sect. 3.2, have been compared, to assess their impact on the performance of multi-step ahead forecasts as the time horizon increases.

The forecast accuracy has been evaluated using the Mean Absolute Error (MAE), the Mean Percentage Absolute Error (MAPE) and the Root Mean Square Error (RMSE). Given the time series of the mortality rates $\hat{k}_t$ whose next h future values are predicted, these accuracy measures are defined, respectively, as:

$$\text{MAE} = \frac{1}{h} \sum_{t=1}^h |\hat{k}_t - f_t| \quad (8)$$

$$\text{MAPE} = \frac{1}{h} \sum_{t=1}^h \frac{|\hat{k}_t - f_t|}{|\hat{k}_t|} \quad (9)$$

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^h (\hat{k}_t - f_t)^2}{h}} \quad (10)$$

where f_t is the forecast value for horizon t .

In the following, we report, for all the five considered countries, the three forecasting accuracy measures for additive and multiplicative QRNN and KNN models, using both the MIMO and the recursive forecasting strategies.

For the Australian population (Fig. 1), it is evident that multiplicative models perform worse than additive ones, especially for longer forecast horizons, both for GRNN and KNN models and for all three accuracy measures. With respect to the two forecast strategies, the recursive strategy still turns out to provide more accurate forecasts than MIMO for all forecast horizons. As for additive models, both GRNN and KNN show good forecasting accuracy for all forecast horizons, with GRNN performing better for long term forecasting. In this case,

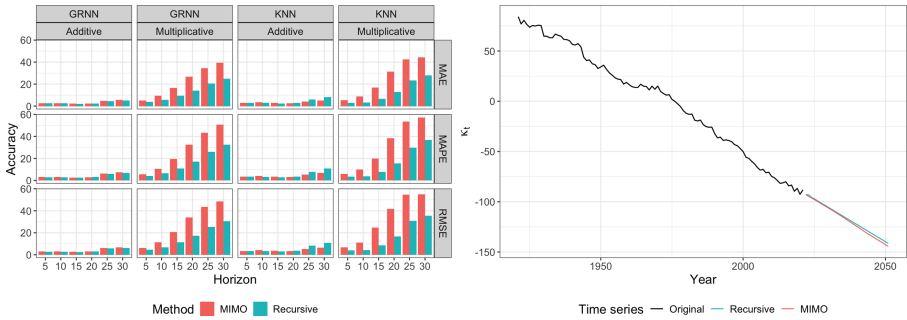


Fig. 1. Australia. On the left: accuracy measures (MAE, MAPE and RMSE) for GRNN with additive and multiplicative trend and for additive e multiplicative KNN; in red the forecasts accuracy obtained with MIMO strategy and in green with the recursive strategy. On the right: observed and forecasted values with additive GRNN model with MIMO and recursive strategies. (Color figure online)

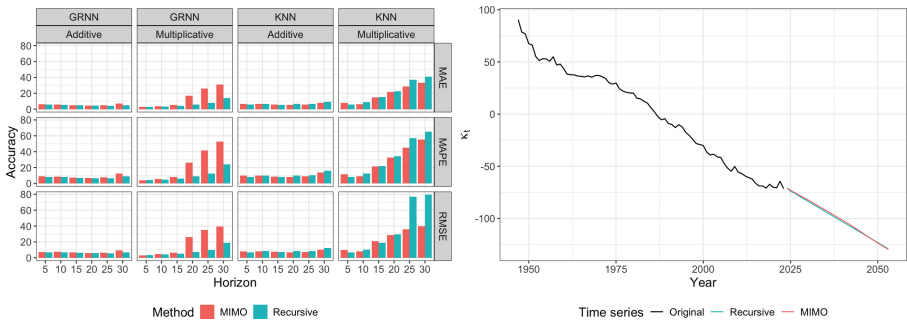


Fig. 2. Austria. On the left: accuracy measures (MAE, MAPE and RMSE) for GRNN with additive and multiplicative trend and for additive e multiplicative KNN; in red the forecasts obtained with MIMO strategy and in green with the recursive strategy. On the right: observed and forecasted values with additive GRNN model with MIMO and recursive strategies. (Color figure online)

the MIMO strategy seems slightly preferable to the recursive one. The plot of the observed and forecasted values obtained by additive GRNN model with MIMO and recursive strategies, in Fig. 1, confirms the equivalence of the two strategies for moderate values of the forecast horizon h and the best performance of MIMO for large values of h .

For the Austrian population (Fig. 2), the same situation occurs as we found for Australia. In this case, however, the multiplicative GRNN model seems to perform as well as the additive model for short-term forecasting, both with the MIMO and recursive strategies. Also for the Austrian population, the additive GRNN model seems to provide more accurate forecasts, and the two strategies, MIMO and recursive, seem equivalent. The plot of the observed and forecasted values obtained by additive GRNN model with MIMO and recursive strategies, in

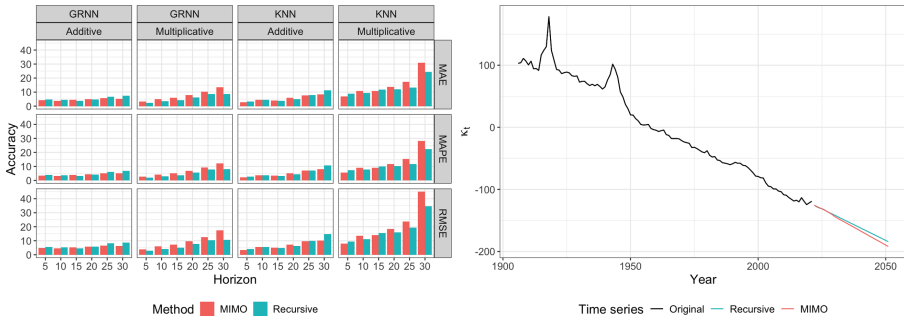


Fig. 3. Italy. On the left: accuracy measures (MAE, MAPE and RMSE) for GRNN with additive and multiplicative trend and for additive e multiplicative KNN; in red the forecasts obtained with MIMO strategy and in green with the recursive strategy. On the right: observed and forecasted values with additive GRNN model with MIMO and recursive strategies. (Color figure online)

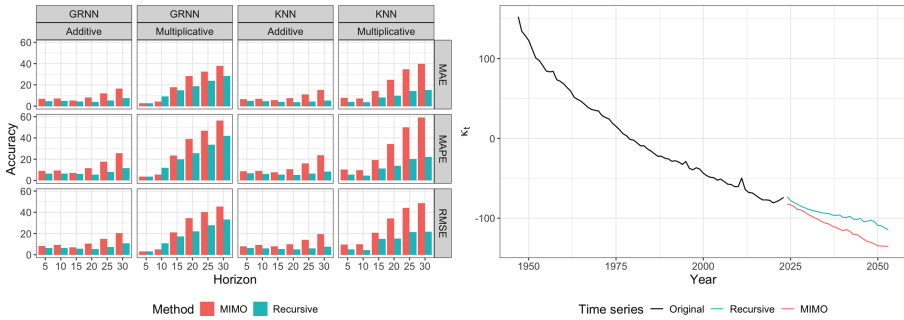


Fig. 4. Japan. On the left: accuracy measures (MAE, MAPE and RMSE) for GRNN with additive and multiplicative trend and for additive e multiplicative KNN; in red the forecasts obtained with MIMO strategy and in green with the recursive strategy. On the right: observed and forecasted values with additive GRNN model with MIMO and recursive strategies. (Color figure online)

Fig. 2, confirms the equivalence of the two strategies for all values of the forecast horizon h .

For Italy (Fig. 3), the best forecasting model seems to be the additive GRNN model with MIMO strategy. Again, all three accuracy measures have high values in both the multiplicative GRNN and KNN models. Furthermore, the additive GRNN and KNN models seem equivalent for short-term forecasts, but the former model is preferable for long-term forecasts. Also in this case, the plot of the observed and forecasted values obtained by additive GRNN model with the MIMO and recursive strategies, in Fig. 3, validates the similarity of the two strategies for all the values of the forecast horizon h .

Even for the Japanese population (Fig. 4), multiplicative models do not provide reliable forecasts, especially when using the MIMO strategy. Again, the

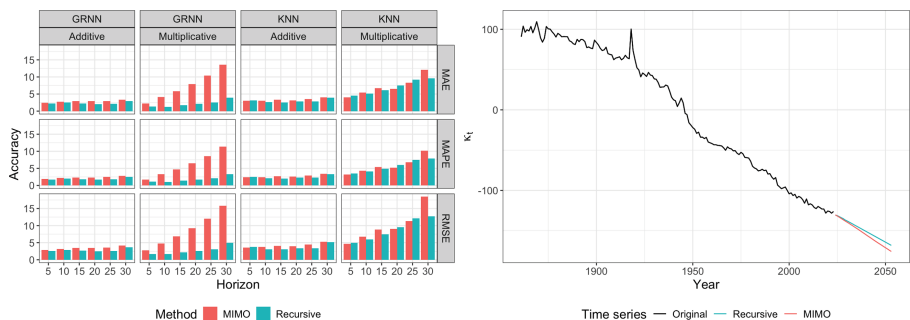


Fig. 5. Sweden. On the left: accuracy measures (MAE, MAPE and RMSE) for GRNN with additive and multiplicative trend and for additive e multiplicative KNN; in red the forecasts obtained with MIMO strategy and in green with the recursive strategy. On the right: observed and forecasted values with additive GRNN model with MIMO and recursive strategies. (Color figure online)

additive GRNN and KNN models seem to perform equally well, for all three forecast accuracy measures. However, in this case, the recursive strategy seems preferable to MIMO for long-term forecasting. In this case, the plot of the observed and forecasted values obtained by additive GRNN model with the MIMO and recursive strategies, in Fig. 4, shows a fairly substantial difference between the two strategies for all the values of the forecast horizon h .

Finally, also for Sweden (Fig. 5), the better performance of the additive GRNN model is evident, for all three measures of forecast accuracy. In this case, the two forecasting strategies, MIMO and Recursive, seem to be equivalent, with a slight preference for the latter in the long-term forecasting. Note that, once again, the multiplicative models fail even if the GRNN model with recursive strategy can be considered equivalent to the additive model.

Regarding the plot of observed and predicted values obtained by the additive GRNN model with the MIMO and recursive strategies, Fig. 5 highlights equivalent performances of the two strategies for moderate values of the forecast horizon h and the best performance of MIMO for high values of h .

5 Concluding Remarks

In this paper, we proposed an approach to automatically forecast mortality rates, within the predictive framework of the Lee-Carter model. It is based on the use of generalized regression neural networks, a type of radial basis function networks, which are particularly appropriate in the case of actuarial data. In fact, while maintaining the property of more complex neural networks, GRNNs allow the analysis of time series with a reduced number of available data. Furthermore, they can be easily automated, providing a simple forecasting tool that does not require a strong human intervention.

An application to real data has been performed to validate the empirical performance of the proposed method and to compare it with KNN, an alternative forecasting model. The analysis highlighted that an additive GRNN model has, in general, better forecasting performances than both the multiplicative GRNN model and the KNN model, taken as a benchmark. Furthermore, we also evaluated two different long-term forecasting strategies, highlighting how, in general, MIMO is preferable to the classic recursive procedure.

Several lines of research, still under investigation, concern the use of GRNN models in resampling schemes and Monte Carlo simulation techniques, as it happens when it is of interest to accurately estimate the forecast distributions. The simplicity of the model and the possibility of automating the choice of the optimal topology is in fact even more justified when neural networks are used in this context.

References

1. Basellini, U., Camarda, C.G., Booth, H.: Thirty years on: a review of the Lee-Carter method for forecasting mortality. *Int. J. Forecast.* **39**, 1033–1049 (2023)
2. Lee, R., Miller, T.: Evaluating the performance of the lee-carter method for forecasting mortality. *Demography* **38**(4), 537–649 (2001)
3. Lee, R.D., Carter, L.R.: Modeling and forecasting US mortality. *J. Am. Stat. Assoc.* **87**(419), 659–671 (1992)
4. Lee, R.D.: The Lee-Carter method for forecasting mortality, with various extensions and applications. *North Am. Actuarial J.* **4**(1), 80–91 (2000)
5. Bergeron-Boucher, M.-P., Kjærgaard, S.: Mortality forecasting at age 65 and above: an age-specific evaluation of the Lee-Carter model. *Scand. Actuar. J.* **2022**(1), 64–79 (2022)
6. Apicella, G., La Rocca, M., Perna, C., Sibillo, M.: Neural Network Lee-Carter model and the actuarial relevance of longevity risk assessment. *Scand. Actuarial J.*, 1–25 (2024)
7. La Rocca, M., Perna, C., Sibillo, M.: Evaluating forecast distributions in neural network Lee-Carter type model for mortality rate. In: *Mathematical and Statistical Methods for Actuarial Sciences and Finance*, pp. 218–223. Springer, Cham (2024)
8. Specht, D.F.: A general regression neural network. *IEEE Trans. Neural Netw.* **2**(6), 568–576 (1991)
9. Park, J., Sandberg, I.W.: Universal approximation using radial-basis-function networks. *Neural Comput.* **3**(2), 246–257 (1991)
10. Martínez, F., Charte, F., Rivera, A.J., Frías, M.P.: Automatic time series forecasting with GRNN: a comparison with other models. In: *International Work-Conference on Artificial Neural Networks*, pp. 198–209. Springer, Cham (2019)
11. Taieb, S.B., Bontempi, G., Atiya, A.F., Sorjamaa, A.: A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert Syst. Appl.* **39**(8), 7067–7083 (2012)
12. Bontempi, G.: Long term time series prediction with multi-input multi-output local learning. In: *Proceedings 2nd ESTSP*, pp. 145–154 (2008)
13. Zhou, Z.: Measuring nonlinear dependence in time-series, a distance correlation approach. *J. Time Ser. Anal.* **33**(3), 438–457 (2012)

14. Edelmann, D., Fokianos, K., Pitsillou, M.: An updated literature review of distance correlation and its applications to time series. *Int. Stat. Rev.* **87**(2), 237–262 (2019)
15. Tashman, L.J.: Out-of-sample tests of forecasting accuracy: an analysis and review. *Int. J. Forecast.* **16**(4), 437–450 (2000)
16. Taieb, S.B., Bontempi, G., Sorjamaa, A., Lendasse, A.: Long-term prediction of time series by combining direct and MIMO strategies. In: 2009 International Joint Conference on Neural Networks, pp. 3054–3061. IEEE (2009)
17. Wang, L., Van Keilegom, I.: Nonparametric test for the form of parametric regression with time series errors. *Stat. Sin.* **17**(1), 369–386 (2007)
18. Teräsvirta, T., Lin, C.F., Granger, C.W.: Power of the neural network linearity test. *J. Time Ser. Anal.* **14**(2), 209–220 (1993)
19. Martínez, F., Frías, M.P., Pérez, M.D., Rivera, A.J.: A methodology for applying k-nearest neighbor to time series forecasting. *Artif. Intell. Rev.* **52**(3), 2019–2037 (2019)



Contextualized Segmentation of Milling Processes Using Discrete Rule-Based Pattern Recognition

Lukas Klehr^{1,2}(✉) , Bastian Engelmann² , Frank-Michael Schleif³ ,
and Daniel Regulin⁴

¹ Siemens AG, Frauenauracher Straße 80, Erlangen 91056, Germany
lukas.klehr@siemens.com

² Technical University of Applied Sciences Würzburg-Schweinfurt,
Ignaz-Schön-Straße 11, Schweinfurt 97421, Germany

³ Technical University of Applied Sciences Würzburg-Schweinfurt, Münzstraße 12,
Würzburg 97070, Germany

⁴ Siemens AG, Otto-Hahn-Ring, Munich 81739, Germany

Abstract. Time series data continuously generated during CNC machining processes contain essential information on their performance and efficiency, but also on produced parts. Using industrial edge devices, among low frequency and event data also high frequency data at 1 kHz can be generated. Within these large, multi-dimensional datasets, inter-relationships and semantics are usually not available without additional processing. Among others, pattern recognition algorithms have been used to find repeated patterns within the data or to find areas with constant technological parameters. However, the approaches often lack reference to parts and part characteristics, leading to individual analytics solutions. To face this issue, a data-driven, multi-level segmentation is developed. The first part of the algorithm extracts production processes of parts of the same product type; the second part segments these production process data by each individual part. For the second method, a discrete rule-based pattern recognition algorithm is proposed to perform the segmentation. Before, the available data are contextualized to gain additional process knowledge, enabling the effective use of the implicitly available information. This part-focused segmentation is verified on milling processes and not only allows for more effective analyses of CNC machining processes, but also enables the analysis of historical data which increases their value significantly.

Keywords: Data Mining · Semantic Segmentation · Pattern Recognition · CNC machining processes

1 Introduction

Today, CNC machining processes are widely tracked by collecting large amounts of various data. The purpose of this is, among others, the analysis of production

efficiency, part quality, process stability, or tool wear [1, p. 7 f.]. An active field of research is the analysis of machining processes with the aim of allowing automatic decision making and fast post-process analyses [2]. This paper will focus on the analysis of milling data, which is widely used for the production of the most complex parts [3]. The data collected containing information about the manufactured part and the underlying production process are as complex as the part itself. Due to the large amounts and high variety of the data, the analysis often leads to process-individual solutions only offering results for single processes. Furthermore, such solutions often lack context to the milling process, making the results hard to interpret and even harder to map them to a specific part. For subsequent analyses, the current machine state should also be known as context to avoid misinterpretation of the data or of occurring events [4].

Thus, one aspect in the analysis of CNC machining data is the semantic segmentation of the time series collected. In the literature, time series segmentation is seen as an enabler for various subsequent data mining tasks such as trend analysis, dimensionality reduction, or other field-dependent analysis tasks [5]. Data collection can be applied without any preliminary constraints or trigger conditions by continuously tracing the data. However, this can lead to a dataset that contains different types of machined products that usually should be analyzed individually. To enable a data-driven analysis of manufacturing processes, it is important to split the data into process-related segments, each of which carries the context of the manufacturing process [6]. In addition to the currently manufactured type of product, the individual parts for each type of product can be of interest and represent the context to the process on a different level.

To achieve a segmentation where the context of the machining process is still given to map the data to a real machined part, a novel two-step algorithm is proposed. Section 2 first provides a brief summary of related work and previous approaches to segmenting machining data. Section 3 explains how data are captured during CNC machining and how tool sequences as context-sensitive data can be extracted to improve the segmentation of milling processes. A two-step data segmentation algorithm will then be explained in Sect. 4 including the key principles of the method and the underlying metrics. Section 5 presents the experimental results that can be achieved using the proposed method and verifies it with triggers set in the NC code. Section 6 then draws a conclusion.

2 Related Work

Due to the increasing amounts of time series data generated in various fields, segmentation and structuring of time series have gained more and more importance [6]. A common technique for analyzing time series data for repeated patterns is motif discovery [7]. A motif is a pair of time series segments which show the smallest difference between all possible segments. The distance between segments often relies on similarity analysis, such as the Euclidean distance, and is usually calculated for given patterns to find similar patterns within a time series [8]. Such similarity search can also be used to compare whole time series with

each other to find a time series most similar to a given one [9]. A prominent implementation of motif discovery is the matrix profile. Given a fixed window size, this method searches for the most similar window of the same size for every possible window within a time series [10]. For every possible window, the matrix profile then contains the index of the most similar. This also means that the window size must be given before searching for a specific pattern [10]. Mueen et al. show another implementation of a motif discovery which optimizes the brute-force algorithm by reducing the number of loop iterations and the number of distance calculations [11]. The brute-force algorithm iterates over the data and, for each window, it again iterates over the data to calculate the similarity between those windows. The aim here is the same as for the matrix profile to find the most similar pattern compared to the currently looked at pattern [11]. Those motif discovery algorithms are limited to finding the most similar pattern for a given window size, whereas segmentation aims at finding all possible similar or equal windows. Furthermore, when there are multiple equal patterns, which can arise when using discrete time series, examining the most similar pattern is not possible anymore since there are multiple equally similar patterns.

Besides motif discovery, there exist approaches utilizing clustering algorithms for time series data. While some approaches assign each data point in a time series to a specific cluster, also called time point clustering [12], Hallac et al. use a small window which they then assign [13]. This should increase the reliability of the results and prevent overfitting during the training of the classifier, but the number of clusters must be known or extracted before [13]. Similarly to such clustering algorithms, Hidden Markov Models can also be used to split machining data into predefined states based on observations in the data, sometimes supported by Gaussian Mixture Models [14]. These models take different hyperparameters as input and the number of states must be known before [14]. To find segments of workpieces of the same type, those methods induce the challenge of finding suitable states or clusters. This issue will lead to a high implementation effort and makes those methods costly to transfer to different use cases. Since the aim is to find a widely applicable method with as few parameters as possible, such methods are not further discussed in this paper.

A third method for segmenting time series data is to extract patterns with constant parameter values which can be based on expert knowledge to examine suitable variables to search for constant conditions [15]. In milling, the speed of the spindle can be used to determine different process phases, since it shows constant phases during cutting, stopped phases during tool changes, and acceleration and deceleration phases [15]. Furthermore, statistical values such as the change in standard deviation can be used to examine the temporal segments of constant values [16]. However, the results of such methods can be difficult to interpret and often require expert knowledge.

In different domains, rule-based reasoning is also used to separate different states or different segments from each other [15, 17, 18]. These methods require some kind of knowledge base and can only be used to a certain extent for the

segmentation of machining data. Since a lot of expert knowledge is necessary to build up the knowledge base, these methods are not further explained here.

Table 1 summarizes the challenges of the described algorithms that arise within the production context. Facing the challenges resulting from the different segmentation methods, a direct transfer of the algorithms to the required use case to retain the production process is not possible. A comparison of the proposed method with the algorithms from the literature is also not possible due to the different segmentation goals. The idea of similarity analysis used by methods such as the matrix profile shows the highest potential to find the repeating patterns of single parts within manufacturing data. Furthermore, these methods require the least parameters and knowledge that must be given to the algorithm. As a metric for such distance measure-based algorithms, the Euclidean distance is often used [19]. Other metrics can be, for example, different statistical values such as mean, median, and standard deviation. Even more complex metrics are possible, such as a generalized L_p norm that tries to include the temporal order of the data which is usually lost by using a regular L_p norm [19].

Table 1. Summary of limitations given by the algorithms.

Algorithms	Related articles	Input	Output	Limitations
Motif Discovery	[7–11]	–Time Series –Window size or defined pattern	Most similar pattern	–Window size must be given –Only most similar window is found –Usually loss of production context
Time series clustering	[12–14]	–Time Series –Clusters/states	Clusters for data points or areas	–Multiple thresholds –Clusters/states must be known –Training of a classifier required –Costly to transfer
Expert-based	[15, 16]	–Time Series –Expert knowledge	Technological values, constant phases, statistical evaluation	–Deep knowledge of the process required –Hard to interpret –Usually loss of production context
Rule-based reasoning	[15, 17, 18]	–Time Series –Knowledge base	As defined, could retain context	–Knowledge base required –High effort and deep expert knowledge required

Dimensionality reduction is an important step in further enhancing the performance of the algorithm used. Time series tend to gain high complexity especially by using a high sampling frequency and having high-dimensional multivariate time series. Therefore, dimensionality reduction is often used to generate a lower-dimensional representation of a time series that can be analyzed and visualized faster than the original time series [15]. It is possible to reduce the

dimension numerically by reducing the number of points and approximating a curve similar to the original data [9]. Furthermore, it is possible to generate a symbolic representation of the data where each symbol can be a real-valued approximation or a discretized value [9]. An aspect of dimensionality reduction is that it often causes information loss that can affect the accuracy of a subsequent analysis [9]. The use of low-dimensional data is still preferred for the proposed method to reduce computational costs and computation time, but the chosen data used for segmentation must be evaluated to see whether they affect the segmentation results due to information loss.

3 Data Preparation

During machining processes, the data produced by several internal sensors, as well as control variables, control differences, and axis positions, can be captured using edge devices. Those edge devices can also be used to capture external sensor values throughout machining processes, such as temperature sensors, accelerometers, or acoustic sensors. Today, edge devices are capable of recording up to 100 variables with sample frequencies of up to 1 kHz [20], also called high-frequency data. In addition to those, several low-frequency data can be collected with sample frequencies of up to 10 Hz. Furthermore, data coming from irregular events, such as tool changes or specific NC code lines, are recorded as well.

During data recording, temporal events can lead to data variations. In human-operated machines, such temporal variations can be caused by the regular loading process with new raw parts, breaks, or after machine warnings and can influence segmentation or pattern recognition algorithms, as the temporal difference between parts is not equal. Furthermore, small variations within the process can also influence the machining time of each part to some extent, also resulting in temporal differences between parts. Thus, an automated segmentation of the data must be independent of these temporal differences.

In addition to temporal differences, the data can show further variations that can be caused by tool breakage or by manually stopping the milling process during manufacturing. This will lead to an unfinished tool sequence for the last part manufactured. Another irregular event that can occur quite often is an initial tool change when the last tool of the previous process is still mounted in the spindle and needs to be changed at the beginning of the current process. It is necessary to take those irregular events into account, since they can highly influence the segmentation results and can lead to wrong segmentation borders or not detected parts.

Some derived challenges of milling process time series data are a high dimensional complexity, temporal differences, unknown segment borders, unpredictable events, and interpretation challenges without context. Another soft constraint is to use as few data as possible to increase the algorithm's performance and to keep the capture resources of the edge device used by the segmentation low. Subsequently, a dedicated time series segmentation algorithm is derived taking these constraints and challenges into account. As the choice of input data can

affect the performance and accuracy of the algorithm, choosing the right data used for segmentation and properly preprocessing these data is a key challenge.

Algorithms usually depend on thresholds that allow some variations within data to be seen as equal or similar. Those thresholds are usually hard to define and often cannot be transferred to different use cases. Furthermore, some set thresholds are hard to describe when, for example, they are set on already processed data. It is possible to reduce or eliminate the use of thresholds by using discrete values. Next to variables that only show discrete values, the discretization of continuous or close-to-continuous variables such as high-frequency axis positions eliminates small variations in the data. As described in Sect. 2, discretization usually induces loss of information that can affect the results of segmentation. Thus, variables that already contain discrete values only are preferred for use in segmentation. In addition, the use of high-frequency data greatly increases execution time and memory usage, which is why the use of data reduced in dimension or low-frequency data is preferred for segmentation.

In milling, the currently active tool can be recorded as low-frequency data using an industrial edge device. Furthermore, tool-change events are also captured as event data, making the currently active tool available for every time step. Even though the time every tool is active can differ between different parts of the same product type caused by, for example, lunch breaks, the tool sequence carries the information about the single workpieces. The tool sequence is defined as the sequence of tools that is used to manufacture one or more parts of the same product type. The time each tool is in use is not considered as only parts of the same type are analyzed for part segmentation.

The tool sequence can be retrieved by looking at the tool changes of the active tool variable, which reduces the number of observations. The upper part of Fig. 1 shows the raw event variable of the active tool where values are generated at specific lines in the NC code, such as lines to perform a tool change. In the time from just before 04:58:00 until around 05:00:30, no further events are detected,

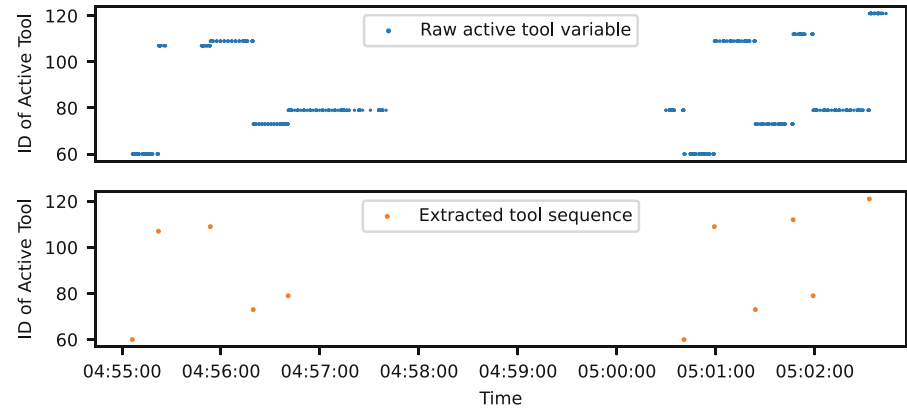


Fig. 1. Comparison of raw active tool variable and extracted tool sequence.

resulting in a gap in the raw data. The lower part of Fig. 1 shows the extracted tool sequence based on the tool change positions of the raw active tool variable, resulting in values only at the tool change positions.

During series production, different product types can be produced multiple times on the same machine, with each product type having its own NC program. For each of those product types, multiple parts are manufactured. Upon continuous capture of data, the resulting dataset contains data for multiple parts of different product types. Figure 2 shows a schematic overview of a production process that results in the dataset containing all specified high- and low-frequency variables and event variables, as described at the beginning of the chapter.

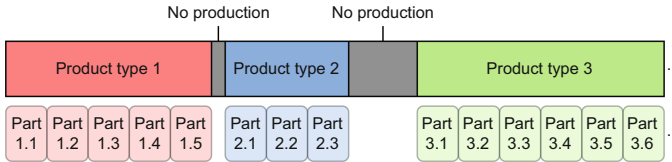


Fig. 2. Schematic description of a production process.

For the proposed segmentation method, only the active tool variable based on relevant NC code events, the program state, and the high-frequency axis positions for the x, y, and z axes of the machine need to be traced, thus resulting in a raw dataset containing five variables. The remaining capture resources of the edge device can then be freely chosen by the applicant.

4 Multi-level Data Segmentation

This section introduces a novel method to perform a segmentation on series production data. First, Sect. 4.1 describes a simple segmentation by product types using specific NC variables, if available. Second, Sect. 4.2 explains the method for automatically finding segments representing individual parts in series production data for single product types.

4.1 Segmentation by Product Types

To be able to segment the data into single parts, it is necessary to first split the continuously generated data into segments for each product type. Each of those segments represents a dataset that contains then data of the manufacturing of multiple parts. The execution of the NC program can be used to perform this first segmentation using an internal NC variable that represents the program state. This variable can enter five different states that are 1: interrupted, 2: stopped, 3: running, 4: waiting, and 5: aborted [21]. If a program is finished after execution with the NC command *M30*, the program state variable enters state 5. If the NC program is programmed to complete with every single part and the whole

NC program needs to be executed from the beginning, this variable can be used to extract the segments for each individual part. If the parts are programmed as an infinite loop in the program, the data for each individual part cannot be extracted using the program state. However, it is possible to still use this program state variable to find data segments for the same product types, which simplifies subsequent analyses for individual parts. Figure 3 shows the program state variable during a regular shift producing parts of one single product type. In the beginning and in the end, there are a few changes between states 2–4 and state 5, whereas in the time from around 05:40 until around 12:50 there seems to be continuous production without changes into state 5.

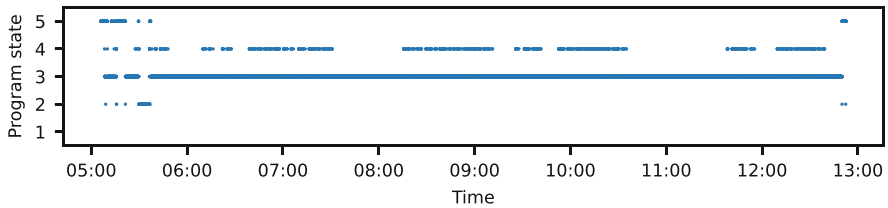


Fig. 3. Changes in the program state variable during a regular working shift.

Thus, the first segmentation follows a rule-based method that splits the data at locations where a change in the program state variable occurs using a common state change detection. While during production states 1–4 can appear, state 5 only appears when the program is completely aborted. It is now possible to detect a change in the variable from state 5 to any other state or from any other state to state 5. At these changes, it is assumed that the production of parts changes, either caused by some error or by changing to a different product type.

The resulting temporal segments now contain either data at states 1 to 4 or data at state 5. The segments with data at state 5 can be seen as meaningless, since no program is executed that produces parts. There can still be some manual action on the machine, as can be seen in Fig. 3 in the beginning and end, but these are not of interest for a context-sensitive analysis of a production process. Therefore, these data can be omitted, and the remaining segments then contain data of the production of one or multiple parts of the same product type each.

This kind of segmentation can be performed offline or online, since such a rule-based state change can easily be detected. This allows to further build up real-time algorithms for segmenting data on top of this first step.

4.2 Segmentation by Parts

Having those datasets containing only data of parts of the same product type, the data is split again with the aim of extracting the data for each individually milled part. After a method is shown for handling an initial tool change, the proposed method to find the segments for each part is described. Afterwards, it is discussed how to handle processes that were aborted during manufacturing.

Handling of Initial Tool Change.

Before segmentation is performed, it is necessary to handle a possible initial tool change. Since the data are already segmented by the program state, an initial tool change is the only case that can occur at the beginning of the data. Without a segmentation based on the program state, the data recording could randomly start anywhere in the data, meaning it can also start in the middle of a process. Finding subsequent patterns will still be possible, but they will start and end somewhere within the production of one part. Thus, it would not be possible to semantically segment the data by single parts but only by repeated patterns not necessarily representing whole parts.

The detection of an initial tool change can be performed by looking at the spindle movement until the first tool change occurs. In general, if the distance between the tool change position and the actual spindle position D_{S-CP} continuously decreases until the first tool change, an initial tool change has been performed. However, it can happen that the spindle first moves towards positive z-direction to avoid collisions. This can lead to increasing distances during movement, although the spindle is moving away from the workpiece. Thus, the times the spindle performs a movement in positive z-direction are not considered while looking for a decrease in D_{S-CP} . A threshold is also necessary to allow aligning movements in the area of the tool change position. This threshold depends on the dimensions of the machine and is set to 200 mm by default.

Figure 4 visualizes the spindle movement and the distance D_{S-CP} for one production process as an example. On the left side of Fig. 4, the path of the tool center point before (blue) and after (black) the initial tool change is shown. The position of the tool change is shown by the orange star. The right side of Fig. 4 shows D_{S-CP} over time until (blue) and after (black) the initial tool change. The horizontal line (red) shows the threshold of 200 mm, the vertical zone (red) past 04:55:00 shows the excluded area where the spindle performs a movement in

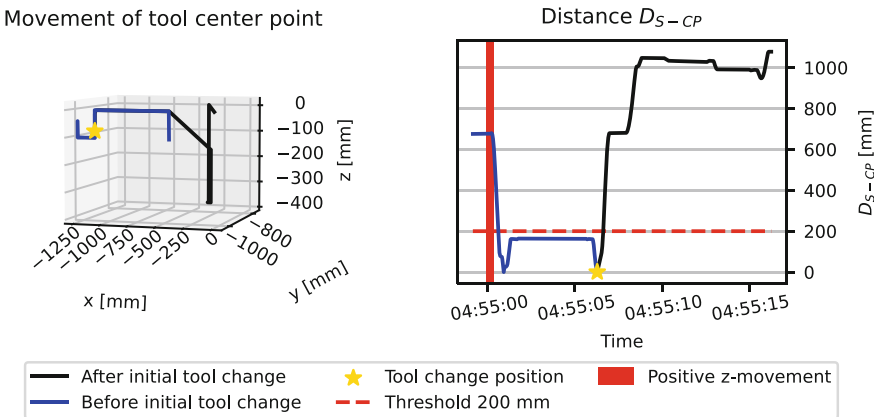


Fig. 4. Spindle movement before and short after the initial tool change (left) and distance D_{S-CP} between spindle position and tool change position before and short after the initial tool change (right).

positive z-direction. The time of the tool change is again marked with an orange star.

Segmentation. After handling a possible initial tool change, segmentation is performed on a deeper level. The proposed method is based on a distance matrix as used to compute the matrix profile, but has the aim to find any similar pattern rather than finding only the most similar one. As a distance measure, the Euclidean distance is used, showing small values for equal or similar time windows, while time windows with different characteristics result in higher values.

As mentioned in Sect. 2, for pattern recognition or similarity search algorithms, the window size for the pattern must be known beforehand. Since the size of the pattern for every product type depends on the workpiece to be produced, on the NC program, on the tools used, and on further parameters, the pattern size shall be found automatically. As data used by the segmentation to find patterns, the tool sequence is chosen as described in Sect. 3.

To automatically find the pattern size of the tool pattern, the locations of the minima in the distance matrix are examined, as they represent the positions where a tool pattern with the respective window size repeats itself. The discrete distances between these minima D_{min} for each row in the distance matrix are then compared to the window size that is used to compute the distance matrix, as D_{min} represent possible pattern sizes. For this, the distance matrix is initially computed with window size $s_w = 2$. If $\min(D_{min}) = s_w$, the tool pattern has been found and the process is stopped; otherwise s_w is iteratively increased to $\min(D_{min})$ and the distance matrix is computed with the new window size until $\min(D_{min}) = s_w$. Since parts of the tool sequence can be used multiple times while manufacturing one part, D_{min} can contain multiple possible window sizes. If the tool pattern is only of size 2, this procedure is completed after the first iteration with $s_w = 2$. The algorithm returns the size of the underlying tool pattern s_{pat} . The principle of the algorithm is shown below as pseudo-code.

Discrete rule-based time series pattern recognition algorithm

Require: $X \leftarrow [x_1, x_2, \dots, x_n]$ ▷ Tool sequence
 $s_w = 2$
while True **do**
 $M \leftarrow \text{computeDistanceMatrix}(X, s_w)$
 while $i \leq \text{length}(M)$ **do**
 $\text{Index}_{min} \leftarrow \text{findMinimaLocations}(M[i])$
 $D_{min} \leftarrow \text{findDiscreteDistances}(\text{Index}_{min})$
 end while
 if $s_w = D_{min}$ **then**
 Stop
 end if
 $s_w \leftarrow \text{getMinimum}(D_{min})$
end while
 $s_{pat} \leftarrow s_w$

Since the distance matrix is symmetric, only the upper right triangle of the matrix is computed to reduce computational costs. The main diagonal of the matrix is 0 as it represents the comparison of any window with itself, also called trivial similarity or trivial match [11].

Figure 5 visualizes the process with $s_{pat} = 11$ as example. The upper plots show the Euclidean distance matrices for the window sizes $s_w = 2$, $s_w = 4$, and $s_w = 11$, respectively. The further s_w approaches $s_{pat} = 11$, the sharper the edges in the Euclidean distances, visible in the color differences. In the lower plots, a zoom of the respective upper plots is shown. For $s_w = 2$, the minima with $D_{min} = \{4, 7\}$ are found. As described above, since $D_{min} \neq s_w$, the algorithm continues with the next higher window size $s_w = 4$. With this, the distances result in $D_{min} = \{11\}$. Again, $D_{min} \neq s_w$, so the algorithm continues with the next larger window size $s_w = 11$. In this iteration, we again find $D_{min} = \{11\}$ resulting in $D_{min} = s_w$, which stops the algorithm and returns the found pattern size $s_{pat} = 11$. It must also be mentioned that the matrix is initiated with infinity, resulting in the white areas with only the upper triangle being computed.

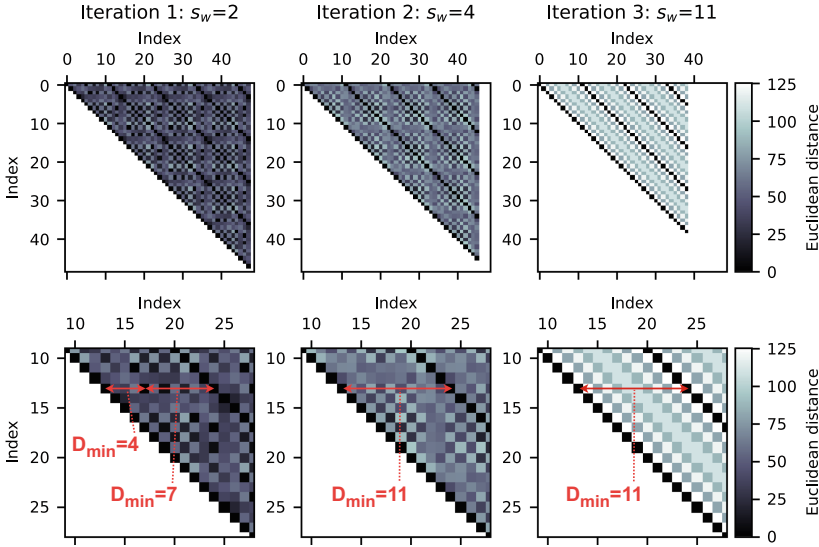


Fig. 5. Visualization of the iterations of the algorithm with $s_w = 2$ (left), $s_w = 4$ (center) and $s_w = 11$ (right) for finding the tool pattern size $s_{pat} = 11$.

Using s_{pat} and considering that an initial tool change has been eliminated, the actual tool pattern pat containing the tool IDs can be found at the beginning of the tool sequence. pat is then matched to the complete tool sequence, where the matching indices represent the times the first tool of pat was substituted, thus the starting indices of each part. These matching indices, either containing the time or the NC cycle, are returned by the complete algorithm and can be used to link the raw data containing also high-frequency data to each part.

A big advantage of using the tool sequence is that it contains only discrete values, eliminating the need for thresholds. The Euclidean distance then equals 0 if the windows show the same tools in the same order, otherwise it shows values higher than 0. Furthermore, using the tool sequence makes the analysis independent from temporal variations, making the method more robust.

Handling of Interrupted Processes. Due to tool breakage, machine errors, or other unpredictable events, the manufacturing process can be aborted during machining. The tool sequence then contains only a fraction of *pat* for the last part in which the abortion occurred. This pattern will not be found by searching for *pat* as described in the previous segmentation section. To identify aborted parts, it is checked if there is a sequence of tools left after *pat* was last found. If there is no sequence of tools left, the process has not been aborted or at least been aborted within the last tool used, and hence all parts have been found anyway. If some sequence is left, this sequence is seen as additional machined part and is compared for validation if it is contained in *pat*. This sequence is labeled as aborted, while this label can also contain the time the process ended or tool number that caused the process to stop for further evaluation.

5 Experimental Results

The proposed algorithm was tested on real-world data from a production plant. The data is collected at a 5-axis CNC machine producing various different product types, of which the results of two different types are shown as examples in this paper. The tool sequences and sequence patterns mentioned below contain the tool IDs used within the machine.

The dataset of the first product type contains 5,439,888 entries for 133 variables, of which 100 variables contain high-frequency data captured at 1 kHz. The tool sequence of the whole production process contains 50 entries in total. The underlying tool pattern of length 11 contains the following tool IDs:

$$pat = \{60, 107, 109, 73, 79, 60, 109, 73, 112, 79, 121\}.$$

The dataset of the second product type contains 4,079,837 entries for 133 variables, of which 100 variables contain high-frequency data captured at 1 kHz. The tool sequence of the whole production process contains 107 entries in total. The underlying tool pattern of length 31 contains the following tool IDs:

$$pat = \{60, 77, 75, 81, 72, 121, 119, 74, 93, 76, 90, 88, 82, 79, 120, 75, \\ 86, 109, 72, 121, 71, 119, 74, 88, 87, 85, 83, 84, 92, 82, 91\}.$$

Both products showed an initial tool change at the beginning and were aborted during the process, resulting in unfinished last parts. To be able to verify the proposed method, start and end triggers have been implemented in the NC code, indicating the start and end of a single part. Those triggers are set only for verification and have not been used by the algorithm.

Figure 6 shows the IDs of the active tool versus time for product type 1. The upper part of the figure shows the whole sequence, with the detected parts having different colors. The lower part of the figure shows the areas of the part changes in more detail. All detected change points are positioned between the end and start triggers, showing a correct segmentation of the parts. In the leftmost plot, the initial tool change can be seen, with an initial tool ID of 121. Afterwards, four regular parts are produced that all show slightly different production times, with Part 2 showing the shortest production time. Part 5 has not been completed since the process was aborted during the manufacturing process. In this case, the remaining data sequence is assigned to the last machined part, here Part 5.

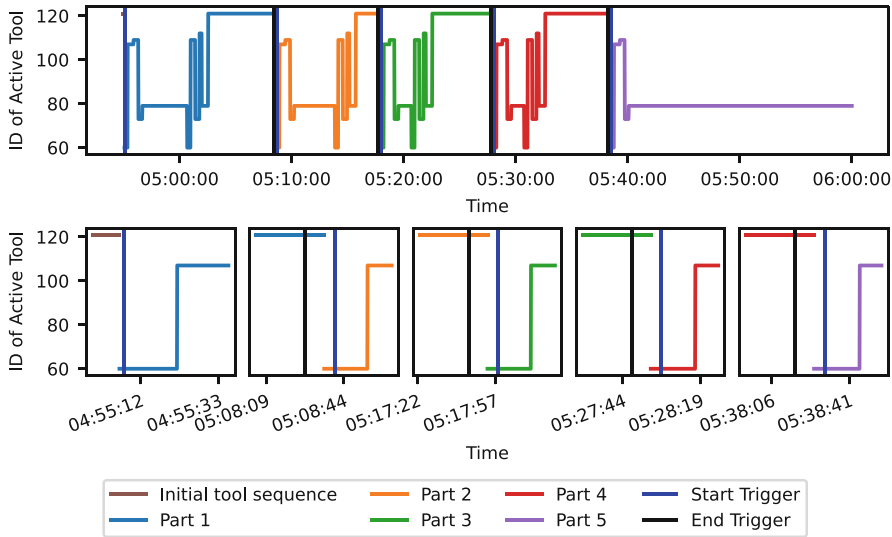


Fig. 6. Tool sequence segmented by single parts for product type 1.

For product type 2, shown in Fig. 7, the upper part of the figure shows the whole tool sequence with the tool IDs of the active tool versus time, with each detected part having a different color. The additional triggers to verify the algorithm can again be seen in the areas of the part changes; however, end trigger of Part 3 and start trigger of Part 4 are about 25 min apart. Still, the detected part change is positioned between end and start trigger as can be seen in the rightmost plot in the lower part of the figure and thus shows a correct segmentation. In the leftmost plot, the initial tool change can be seen with an initial tool ID of 0. The other detailed plots also show the part change in between the end and start triggers, demonstrating a correct segmentation. Looking at the tool sequence of Part 4, it can be seen that the machining process was aborted again during production. As for product type 1, the remaining data that have been captured are assigned to the last machined part, here Part 4.

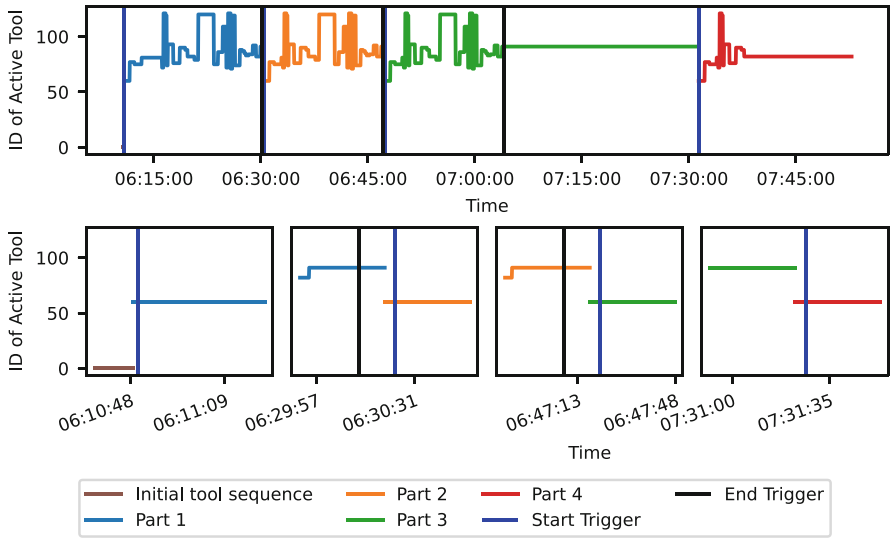


Fig. 7. Tool sequence segmented by single parts for product type 2.

The efficiency of the method can be seen in the execution time of the segmentation. For product types 1 and 2, the segmentation took 0.08 s and 0.20 s, respectively. Within this time, the complete datasets containing 5,439,888 and 4,079,837 entries are segmented according to the indices returned by the algorithm. The two examples further show that temporal differences between parts can still be handled using the proposed method, resulting in correctly segmented data even though there are temporal variations between single parts. The results also show that the method is able to handle an initial tool change since a first segmentation based on the program state has been performed to separate single product types. As the method uses the minima in the Euclidean distances between the windows, the method could be extended by a threshold around the minimum to allow variations in the tool sequences, which probably will be hard to set. The tool sequences for different parts of one product type are usually equal and therefore do not show a need for such a threshold. However, occurring variations within the tool sequences could lead to a wrong segmentation. On the other hand, such variations in the sequence usually occur during machine problems or process interruptions, which are detected by segmentation using the program state. After a process is interrupted due to an unexpected event or problem, the dataset is completed by the program state segmentation. The part produced last before the interruption can either be a scrap part or can be continued afterward. A continuation of the process currently cannot be integrated in the data automatically and has to be matched manually since the detection based on the program state will complete the dataset. As described in Sect. 2, due to the different segmentation goals of the algorithms described in the literature, a benchmark against other methods was not possible.

6 Summary and Conclusion

To perform a segmentation of machining data considering the context of machined workpieces, a novel two-level algorithm was proposed that relies only on five variables. The algorithm segments the raw data based on the machine state utilizing a regular state change detection on the first level. On the second level, it uses a discrete rule-based pattern recognition algorithm based on a distance matrix to find any similar or equal repeating pattern, each of which represents a single machined part. Some random cases such as an initial tool change or an aborted process are also taken into account to eliminate their influence on the segmentation results. An initial tool change is detected when the distance between the actual tool position and the position of the tool change continuously decreases until the first tool change and the tool moves away from the workpiece. For this, the high-frequency axis positions before the first tool change are used. The segmentation itself only relies on the sequence of tools that are used for manufacturing, resulting in a very efficient segmentation of high-frequency data based on the low-dimensional tool sequence. The efficiency of the method can be seen in the low execution times of the segmentation of the whole datasets. Additionally, the results have shown that the proposed method can correctly segment the data into segments representing real machined parts, validated using additional trigger points set in the NC program.

The segmentation itself does not rely on any threshold, making the method widely applicable to milling processes. However, to detect an initial tool change, one threshold must be set to allow spindle alignment movements for the tool change. Furthermore, the method can only be applied to machining processes that use more than one tool for the production of one part, since the window size has a lower limit of $s_w = 2$ to detect patterns. Another limitation is given when continuing a part after the process has been interrupted unexpectedly. In this case, the matching of the data has to be done manually since the program state-based segmentation will separate those connected datasets due to the change in the machine state. The proposed method has been tested on real-world CNC milling processes and needs to be tested on other machining operations, such as drilling or turning. Thus, the generalization ability to other machining operations than milling processes remains unexplored up to now. Future research will also deal with the elimination of the threshold currently used to detect an initial tool change. Furthermore, a more granular segmentation of the data will be part of subsequent research to increase the value of the available data.

Disclosure of Interests. L. Klehr has received research support from Siemens AG.

References

1. Schlick, J., Stephan, P., Loskyll, M., et al.: Industrie 4.0 in der praktischen Anwendung. In: Vogel-Heuser, B., Bauernhansl, T., ten Hompel M. (eds.) *Handbuch Industrie 4.0 Bd. 2. Automatisierung*. 2nd edn., pp. 3–29. Springer Vieweg, Berlin (2017). https://doi.org/10.1007/978-3-658-04682-8_3
2. Trauth, D.: Monetization of data at the example of manufacturing machines. In: Trauth, D., Bergs, T., Prinz, W., (eds.) *The Monetization of Technical Data. Innovations from Industry and Research*. Springer-Verlag GmbH, pp. 1–14 (2023). https://doi.org/10.1007/978-3-662-66509-1_1
3. Kief, H.B., Roschiwal, H.A., Schwarz, K.: *CNC-Handbuch*, 31st edn. Carl Hanser Verlag, Munich (2020)
4. Godreau, V., Ritou, M., Chové, E., et al.: Continuous improvement of HSM process by data mining. *J. Intell. Manuf.* **30**, 2781–2788 (2019)
5. Lovrić, M., Milanović, M., Stamenković, M.: Algorithmic methods for segmentation of time series: an overview. *J. Contemp. Econ. Bus. Issues.* **1**(1), 31–53 (2014)
6. Ochel, J., Fey, M., Brecher, C.: Semantically meaningful segmentation of milling process data. In: Behrens, B.-A., Brosius, A., Drossel, W.-G., Hintze, W., Ihlenfeldt, S., Nyhuis, P. (eds.) *WGP 2021. LNPE*, pp. 319–327. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-78424-9_36
7. Hennig, M., Grafinger, M., Gerhard, D., et al.: Comparison of Time Series Clustering Algorithms for Machine State Detection. In: *CIRP Conference on Manufacturing Systems*. Chicago, IL, USA, pp. 1352–1357 (2020)
8. Janka, D., Lenders, F., Wang, S., et al.: Detecting and locating patterns in time series using machine learning. *Control. Eng. Pract.* **93**, 104169 (2019)
9. Schäfer, P., Höggqvist, M.: SFA: a symbolic Fourier approximation and index for similarity search in high dimensional datasets. In: *International Conference on Extending Database Technology*, pp. 516–527. Berlin, Germany (2012)
10. Yeh, C.-C.M., Zhu, Y., Ulanova, L., et al.: Matrix profile i: all pairs similarity joins for time series: a unifying view that includes motifs, discords and shapelets. In: *IEEE 16th International Conference on Data Mining (ICDM)*, pp. 1317–1322. Barcelona, Spain (2016)
11. Mueen, A., Keogh, E., Zhu, Q., et al.: Exact discovery of time series motifs. In: *SIAM International Conference on Data Mining (SDM)*, pp. 473–484. Sparks, NV, USA (2009)
12. Zolhavarieh, S., Aghabozorgi, S., Teh, Y.W.: A review of subsequence time series clustering. *Sci. World J.* **2014**, 312521 (2014)
13. Hallac, D., Vare, S., Boyd, S., et al.: Toeplitz inverse covariance-based clustering of multivariate time series data. In: *23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2017)*, pp. 215–223. Halifax, NS, Canada (2017)
14. Pantazis, D., Goodall, P., Conway, P.P., et al.: An automated feature extraction method with application to empirical model development from machining power data. *J. Mech. Syst. Signal Proc.* **124**, 21–35 (2019)
15. Ochel, J.: Datengetriebene Strukturierung von NC-Zerspanprozessen. In: Bergs, T., Schuh, G., Brecher, C., et al.: *Ergebnisse aus der Produktionstechnik*. Aachen: Apprimus Verlag (2024)
16. Liao, L., Honda, T., Pavel, R.: Device health estimation by combining contextual control information with sensor data and device health prognostics utilizing restricted Boltzmann machine. In: *Annual Conference of the PHM Society*, vol. 7, No. 1, Coronado, CA, USA (2015)

17. Triboan, D., Chen, L., Chen, F., Wang, Z.: Semantic segmentation of real-time sensor data stream for complex activity recognition. *Pers. Ubiquit. Comput.* **21**(3), 411–425 (2017). <https://doi.org/10.1007/s00779-017-1005-5>
18. Hong, X., Nugent, C.D.: Segmenting sensor data for activity monitoring in smart environments. *Pers. Ubiquit. Comput.* **17**, 545–559 (2013)
19. Lee, J., Verleysen, M.: Generalization of the LP norm for time series and its application to self-organizing maps. In: 5th Workshop on Self-Organizing Maps, pp. 733–740. Paris, France (2005)
20. Industrial Edge für Werkzeugmaschinen. Die Vorteile der Edge Plattform von Siemens. <https://tinyurl.com/4uud3smc>. Accessed 19 Jan 2025
21. Siemens AG: Sinumerik 840D SL NC-Variable und Nahtstellensignale. Listenhandbuch, December 2018. <https://tinyurl.com/bdvfxmvr>. Accessed 10 Jan 2025



Estimation of River Ebro Streamflow Using Fuzzy Inference Systems and FCM

Anastasios Panagiotis Psathas¹ , Stamatios Petkopoulos¹,
Antonios Papaleonidas¹ , Luis Garrote² , Mike Spiliotis¹ , and Lazaros Iliadis¹ 

¹ School of Engineering, Department of Civil Engineering, Democritus University of Thrace,
67100 Xanthi, Greece

{anspatha, papaleon, mspiliot, liliadis}@civil.duth.gr

² Department of Hydraulics, Energy and Environment, Universidad Politécnica de Madrid
(UPM), Madrid, Spain
l.garrote@upm.es

Abstract. Floods rank among the most destructive natural disasters, inflicting severe harm on human lives, societies, and ecosystems. A flood can be characterized as a complex nonlinear phenomenon, thus the use of Machine Learning (ML) techniques and of Artificial Intelligence (AI) tools tend to be of vital importance. The authors of this paper propose an innovative hybrid approach that employs the *Fuzzy C-Means* (FCM) Algorithm and a novel *Fuzzy Inference System* (FIS) in order to determine the relation between the inflow and the outflow. Overall, the whole effort is based on a fuzzy rule-based methodology. It is a novel and hybrid research effort, that employs both FCM (for the determination of clusters) and a novel FIS for the outflow prediction. In fact, it incorporates fuzziness blurring the crisp margins of the clusters, offering a rational model. The model has proved to be quite robust in the testing phase, achieving an overall R^2 of 0.9104. Furthermore, the proposed hybrid approach seems capable of assessing the beginning of flood events.

Keywords: Flood · River Routing · Prediction · Fuzzy C-Means · Fuzzy Inference System

1 Introduction

Floods rank among the most destructive natural disasters, inflicting severe harm on human lives, societies, and ecosystems. The simulation of flood flows is crucial for effective flood management. Existing flood routing models are generally classified into two main types: physically based models (traditional engineering approach) and data-driven models [10]. Physically based models are further categorized into hydraulic and hydrologic flood routing models. The hydrological models are based on parametric (also called conceptual) models which consist of the exploitation of the mass balance and a physical analogue. The structure of a physical analogue is based on several simplifications, while a calibration procedure is required [8]. The physical analogue is composed of a set of interconnected systems designed to represent hydrological processes across various spatial and temporal scales [10].

The hydraulic models (also called white-box models) are based on the precise implementation of the three conservation equations (mass, energy, momentum). In the case of flood routing, hydraulic models are derived from the Saint-Venant equations, which are typically solved through various numerical methods, while hydrological models rely on the Muskingum method or its modifications [8, 9].

Among data-driven methods, nowadays machine learning (ML) methods have gained applicability in hydrological problems since they can be used in more complex nonlinear phenomena without an explicit knowledge of the physical system. As a flood wave travels along a channel, its shape is influenced by several factors, such as channel resistance (which can change because of bedload transport phenomena) and lateral flow. Hence, since the flood can be characterized as a complex nonlinear phenomenon, the use of ML techniques can be very useful. Sometimes the absolute distinction among the aforementioned methods of routing is not so obvious [8]. Many methods which exploit ML techniques are based on Artificial Neural Networks (ANNs) [11] and other techniques exist.

Hybrid approaches have also been developed, such as using fuzzy numbers to expand the hydrological Muskingum method [8] or improving the calibration procedure by using advanced intelligent ML-based optimization techniques [9]. Another hybrid approach could involve using fuzzy linear regression to calibrate the linear Muskingum method. However, Tanaka's fuzzy linear regression (1987) is highly sensitive to outliers, which can significantly increase the model's spread. In addition, at times, the resulting fuzzy coefficients fail to align with the physical interpretation of the storage equation [8].

In this article, the proposed methodology deals with a natural river with significant lateral flow along the reach. The main aim is to provide a relation between the inflow and outflow based on fuzzy rule-based methodology. The use of fuzzy rules incorporates fuzziness, which aligns better with reality. However, the scope of the proposed method is not to provide the most accurate and efficient model, but to preserve some physical interpretation as much as possible. There is a plethora of applications of Sugeno Systems to water systems based on the MATLAB toolbox, and even if the errors remain within an acceptable range, it is not rare for the rational and logical basis of these applications to be rather ambiguous [12]. In this article, the creation of the rules is based on fuzzy clustering, and consequently, the fuzzy inference system was modulated to preserve the physical meaning of the proposed model.

The rest of the manuscript is organized as follows; Sect. 2 describes the dataset of the flood events and the pre-processing of the source data. Section 4 presents the theoretical foundations of the FCM algorithm and the split of the dataset into clusters. Section 5 provides an insight into the FIS process and the proposed model is deployed. Section 6 analyzes the experimental results of the research and Sect. 7 concludes the paper.

2 Dataset Description and Pre-Processing

For the purpose of this scientific effort, data of inflow and outflow from a point A to a point B, of the Spanish river Ebro, were obtained and used [8, 14]. Ebro River Basin (85,500 km² in NE Spain) is characterized by a marked spatial heterogeneity of its geology, topography, climatology and land use. Many reservoirs (138, total water storage

capacity of 6,837 hm³) and canals were constructed during this century for agricultural irrigation, electricity production and domestic use. There are also 40 aquaculture systems producing 35% of the total Spanish trout production. The mouth of the Ebro River (a 320 km² delta area) is very productive for agriculture and fisheries. Industrial activities are localized in the upper NW and central parts of the Basin, causing low water quality in the fluvial system. Salinization of land and river water occurs in the central part, because of salt dissolution, run-off from irrigated fields and groundwater abstraction [8, 14].

The dataset considered in this manuscript, comprises four different flood events, observed in a part of river Ebro basin in Spain, during the years 1997, 1998 and 1999, including lateral inflow (Fig 1).

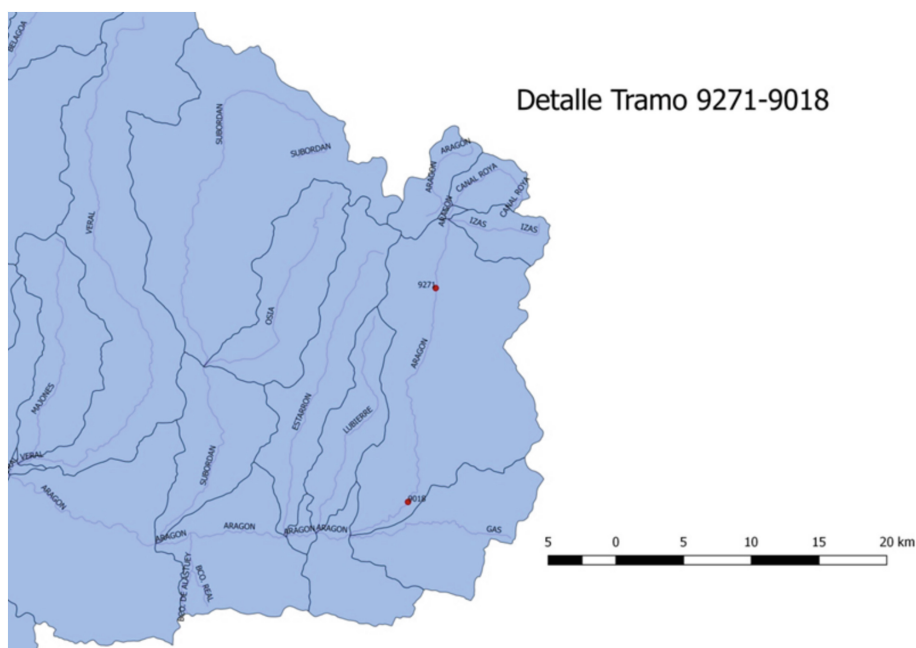


Fig. 1. Map of river Ebro basin

The considered dataset, (before processing) contains values that were measured (Fig 2) every 15 minutes. Specifically, each measurement comprises a pair (X_i, Y_i) where X_i stands for inflow and Y_i stands for outflow volume. The recent flood events of river EBRO are briefly described below:

- 1st flood event: 287 price pairs from July 1, 1997 to July 4, 1997
- 2nd flood event: 476 price pairs from May 29, 1998 to June 3, 1998
- 3rd flood event: 292 price pairs from May 6, 1999 to May 9, 1999
- 4th flood event: 434 price pairs from September 19, 1999 to September 23, 1999

Cuenca Hidrográfica río Ebro

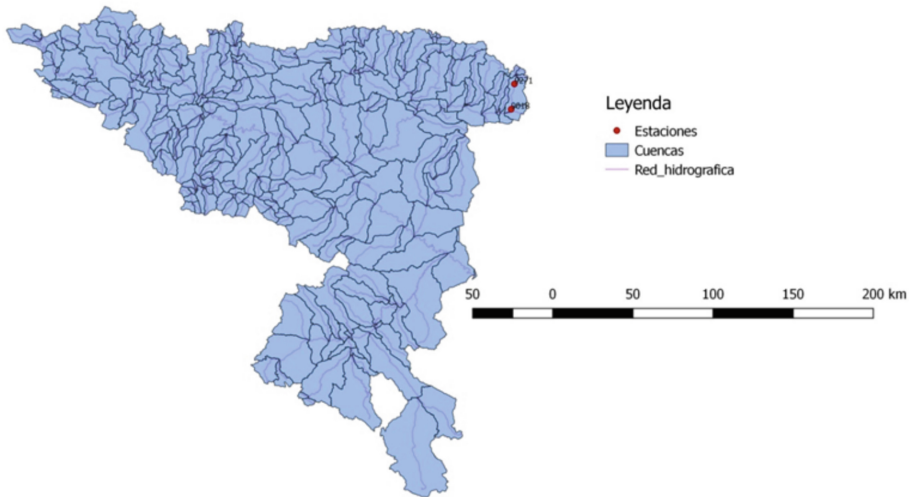


Fig. 2. Hydrographic map of river Ebro.

The measured data vectors comprise the following records:

- Inflow measurement point: sub-cathcement #9,271 with a basin area of 101 km²
- Discharge measurement point: sub-cathement #9018 with a basin area of 238 km²
- Total area: 136.7 km²
- Reach length between the two points: 18.09 km
- Elevavtion difference between the two points: 247 m
- Slope: 1.3650%

Special code was developed in Matlab platform by our research team, aiming to preprocess the available data. More specifically, records with missing or extreme values were removed.

As already mentioned, the source data set comprises a table with dimensions 1,489 × 2, where the first feature is the inflow measurement at point A (ranges between 0.68 and 28.8 m³/s, independent variable). The second feature, dependent variable, is the outflow at point B ranges between 0.37 and 72.25 m³/s.

It is a common practice in cases of hydrological tolling to consider real time values and not average ones. Furthermore, we have avoided considering values corresponding to very close time intervals, due to the fact that there are issues of instability in purely hydrological methods. Additionally, another issue that arises from such measurements is the possibility of a dependence on the outflow supply of the previous minutes, whereas the goal is the development of a model that depends on the inflow (Fig 5).

As a matter of fact, we have considered only hourly values (e.g., pairs of the form (value_{21:00 hours}, value_{22:00 hours})). The values recorded at the starting point of the flood phenomenon, or at its peak are considered as important. A check for important values was performed in order not to delete any of them during data pre-processing.

After data preprocessing, the final dataset forms a table of 368×2 dimensions, where the first column is the inflow measurement at point A (independent variable), and the second column is the outflow measurement at point B (dependent variable). The measurements from the 1st to the 71st row correspond to the first flood event, the ones from the 72nd to 189th row correspond to the second flood, whereas the ones from the 190th to 261st row correspond to the third flood. The rest of them are related to the fourth and last flood.

3 Fuzzy C-Means

3.1 FCM Theoretical Foundation.

Clustering numerical data forms the basis of many classification and system-modeling algorithms [1]. Fuzzy clustering (a well-known soft computing method) is a very flexible methodology in which each data point can be assigned to more than one cluster with different degrees of membership [2] where the respective membership values add to the value of 1. One of the most widely used fuzzy clustering algorithms is the *Fuzzy C-means clustering* (FCM) which was developed by Dunn in 1973 [3] and improved by Bezdek [4] FCM is frequently used in pattern recognition.

The FCM algorithm starts with an initial guess for the cluster centers, which represent the mean location of each cluster. The initial guess for these cluster centers is most likely incorrect. Additionally, FCM assigns every data point to a degree of membership for each cluster. By iteratively updating the cluster centers and the membership values for each point, the algorithm iteratively moves the cluster centers to the optimal location within a data set. This iteration is based on minimization of the following objective function (eq.1) that represents the distance from any given data point to a cluster center weighted by the respective membership value.

$$J_m = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m \|x_i - c_j\|^2, \quad 1 \leq m < \infty \quad (1)$$

N is the number of data points, C is the number of clusters, m is the fuzzifier (the fuzzifier m determines the level of cluster fuzziness and controls the degree of overlapping, $m > 1$), u_{ij} is the degree of membership of x_i in the cluster j , x_i is the i th of d -dimensional measured data, c_j is the d -dimension center of the cluster, and $\|\cdot\|$ is any norm expressing the similarity between any measured data and the center. The Euclidean distance [2], the Mahalanobis distance [5] or the Gath-Geva one [6] can be used in this respect. The Euclidean distance metric was employed in this research to compute the distance between each data point and the cluster center. The Euclidean distance is calculated as follows in Eq. 2:

$$\|x_i - c_j\| = \sqrt{(x_i - c_j)^T (x_i - c_j)}, \quad 1 \leq i \leq N, \quad 1 \leq j \leq C \quad (2)$$

Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above. The following functions were used to update the membership values u_{ij} and the respective cluster centers c_j :

$$u_{ij} = \frac{1}{\sum_{k=1}^C \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}}, \quad c_j = \frac{\sum_{i=1}^N u_{ij}^m \cdot x_i}{\sum_{i=1}^N u_{ij}^m} \quad (3)$$

This iteration process stops when $\max_{ij} \left\{ \left| u_{ij}^{(k+1)} - u_{ij}^k \right| \right\} < \varepsilon$, where ε is a termination criterion between 0 and 1, and k stands for the number of the iteration steps. This procedure converges to a local minimum or a saddle point of J_m .

3.2 FCM Hyperparameters and Optimized Values.

The following Table 1 presents the Hyperparameters employed by the FCM algorithm. The inflow at point A and the outflow at point B, were chosen as input parameters. Exponent (m fuzzier) controls the degree of fuzzy overlapping between clusters. Max_Iterations is the maximum number of optimization iterations and Min_Improvement is the minimum improvement of the objective function between successive iterations. When the objective function improves by a value below this threshold, the optimization stops. A smaller value produces more accurate clustering results, but the clustering can take longer to converge. The hyperparameters were assigned the most widely used values in the literature [7].

Table 1. FCM's Hyperparameters with their corresponding value

Hyperparamter	Value
Exponent (m fuzzier)	2
Max_Iterations	100
Min_Improvement	0.00001

3.3 FCM on Flood Data.

Due to the complexity of the model as a result of the storage capacity present in the stream which varies from section to section, the number of chosen clusters could not be very small. Thus, for this research effort, the authors decided to employ 12 clusters, due to the fact that it results in 2 similar outflow clusters for the same inflow. This clearly shows a different response to the behavior of the water stream. Increasing the number of clusters did not repeat the above phenomenon and therefore it was decided to stop at 12 clusters. Figure 3 presents the clusters, with their corresponding centers, derived from the FCM algorithm. Table 2 indicates the center of each cluster and the number of respective flood instances.

Table 2. Name, Center and Number of Instances for each Cluster

Cluster	Lowest_1	Lowest_2	Lowest_3	Low_1
Center	(2.26, 2.99)	(2.96, 6.99)	(3.61, 15.54)	(7.07, 18.97)
Number	46	17	25	22
Cluster	Low_2	Medium_1	Medium_2	High
Center	(7.86, 14.59)	(11.58, 21.77)	(13.55, 27.59)	(16.54, 32.76)
Number	46	22	28	54
Cluster	Very_High	Dangerous_1	Dangerous_2	Very_Dangerous
Center	(16.54, 32.76)	(19.76, 48.85)	(24.01, 59.06)	(24.74, 68.78)
Number	16	13	44	35

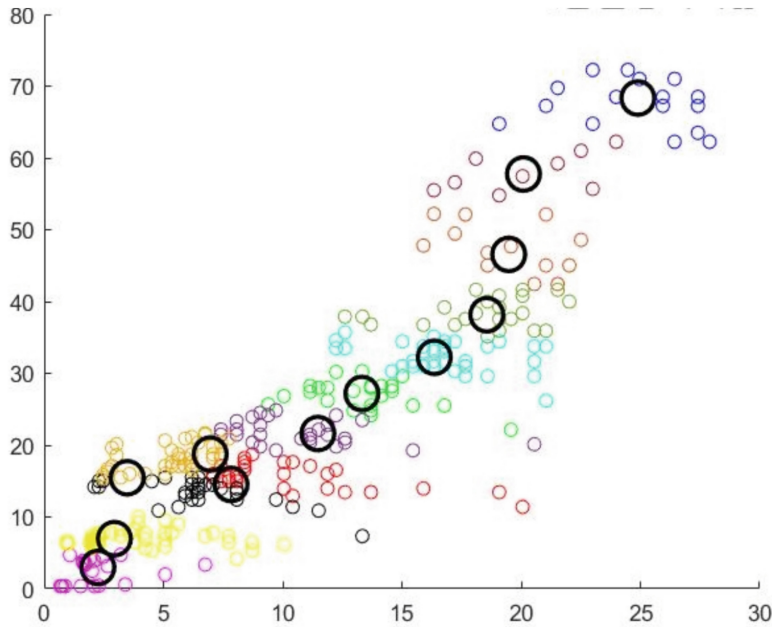


Fig. 3. The 12 clusters for the flood events

4 Fuzzy Inference System

4.1 FIS Theoretical Foundation.

Fuzzy inference is the process of formulating the mapping from a given input to an output, using fuzzy logic. The mapping provides a basis from which decisions can be made, or patterns discerned. This process exploits the potential of fuzzy inference Membership Functions, Logical Operators , If-Then Rules, based on Logical “OR” ,“AND” for drawing essential decision rules [15].

The output of the model is always a fuzzy set irrespective of its input which can be fuzzy or crisp. Additionally, it is necessary to have fuzzy output when it is used as a controller. Furthermore, a defuzzification unit would be there with FIS to convert fuzzy variables into crisp ones [16].

There are five functional blocks that determine the core of this architecture. The “IF-THEN” Rule Base. The existing Database on which the membership functions to the respective fuzzy sets are defined. The Decision-making Unit, which performs the operation on rules. The Fuzzification Interface that supports the application of numerous fuzzification methods and the conversion of the crisp input into fuzzy. Last but not least, the Defuzzification Interface Unit, that converts the fuzzy quantities into crisp ones. The following figure 4 presents a block diagram of fuzzy Inference system [17]. Figure 4 presents the aforementioned Units for the FIS process.

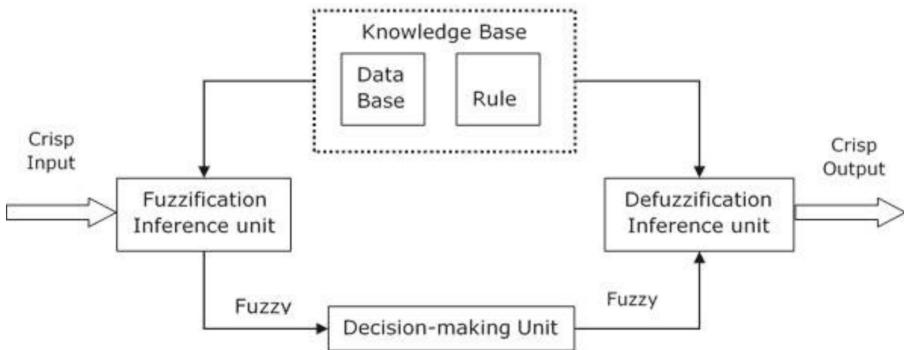


Fig. 4. The 5 core units of FIS

The overall algorithm is presented below [15]:

- Setting – determination fuzzy rules
- Convert the input to fuzzy values by using proper membership function,
- Establish the rules strength by combining the fuzzified inputs according to fuzzy rules.
- Determine the consequence of a rule by combining the rule strength and the output membership function.
- All consequences are combined in order for the output distribution to be derived.
- Application of a defuzzified.

4.2 FIS Application in the Flood Dataset.

In this research effort, a FIS was developed and applied in Matlab using the Fuzzy Logic Toolbox [18]. Fuzzy Logic Toolbox provides MATLAB functions, apps, and a Simulink block for analyzing, designing, and simulating fuzzy logic systems. It allows the user to specify and configure inputs, outputs, membership functions, and rules of type-1 and type-2 fuzzy inference systems. The toolbox allows the user to automatically tune

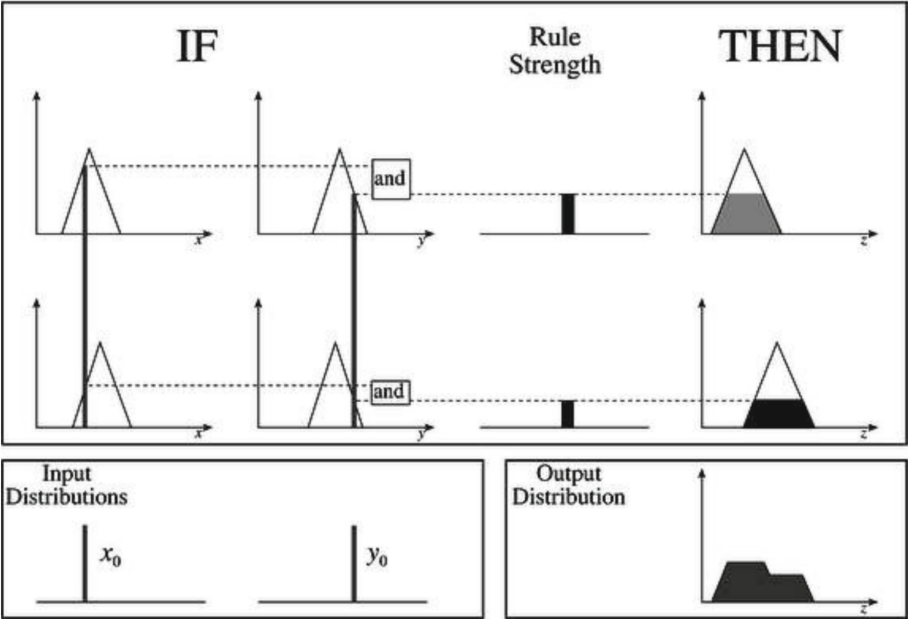


Fig. 5. Example of FIS

membership functions and rules of a fuzzy inference system from data. For the problem of the flood events, the hyperparameters that were set, for both input and output by the authors are presented in Table 3.

Table 3. FIS’s Hyperparameters with their corresponding value

Hyperparamter	Value
FIS Type	mamdani
“And” method	min
“Or” method	max
Implication	min
Aggregation	max
Defuzzification	centroid

Input and Output of the FIS model had the same treatment as far as the type of membership functions are concerned. They both start with a trapezoid membership function. Then 10 triangle membership functions follow. Lastly, another trapezoid membership function completes the model. The clusters cluster from table 2 were used to set the parameters of the trapezoid and the triangle membership functions. In the input functions the abscissa of the centers was used, while in the output the ordinate was utilized.

For the first trapezoid function of both Input and Output the following parameters were used $[0, 0, 1^{\text{st}} \text{ cluster center}, 2^{\text{nd}} \text{ cluster center}]$. For the last trapezoidal function, the employed parameters are $[\text{center of } 11^{\text{th}} \text{ cluster}, \text{center of the } 12^{\text{th}} \text{ cluster}, 35, 35]$ for input and $[\text{center of the } 11^{\text{th}} \text{ cluster}, \text{center of the } 12^{\text{th}} \text{ cluster}, 75, 75]$.

The parameters of all triangular functions have the following format: $[\text{center of the previous cluster}, \text{center of the current cluster}, \text{center of the following cluster}]$.

The limits and the peaks of all membership functions were chosen to be the coordinates of the centers, in order to reduce the interference of very distant classes, but at the same time to maintain the level of fuzziness as high as possible. In Figures 6 and 7, the Input and Output of the FIS model are displayed.

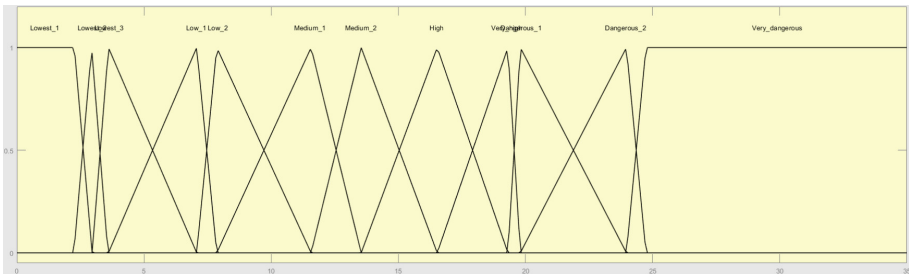


Fig. 6. Input of FIS

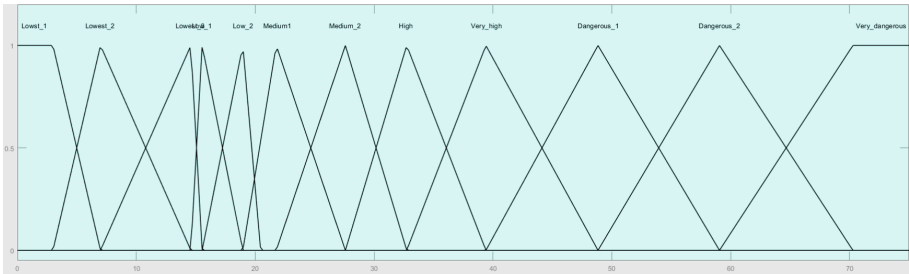


Fig. 7. Output of FIS

5 Experimental Results

In this Section of the paper, the results of the proposed approach are presented. As already mentioned, the input of the FIS model is the inflow measurement at point A and the output is the outflow measurement at point B. After applying the FIS model in the inflow, the model generated the predicted Outflow. In Figure 8, the X Axis displays the Predicted Outflow, while Y Axis displays the actual Outflow of the river.

R Squared (R^2) Index was used for the evaluation of the model. R-Squared (R^2 or the coefficient of determination) is a statistical measure in a regression model that determines the proportion of variance in the dependent variable that can be explained by the independent variable [19] see equation 4.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \quad (4)$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, $\bar{y}$ is the mean value, and N is the total number of the instances.

The value of R^2 was estimated as high as 0.9104, which leads to the conclusion that it is an efficient and robust model which can predict with a high degree of accuracy, the future outflow at point B, given the corresponding inflow at point A.

There are 14 instances for which the model predicts much higher value than the actual one. For example, the point (72.43, 24, 17), has a real outflow equal to 24.17, but the predicted value was equal to 72.43,

In order to understand the reason behind this problem, the authors identified all cases for which this phenomenon is observed. After a detailed analysis of the dataset, it has been realized that these points are related to the beginning of flood events. More specifically, in the moments before the flood phenomenon both inflow at starting point A and the outflow at ending point B are considerably low. However, during the duration of the flood event, both inflow and outflow are evolving. During the beginning of the flood, the flow value in point A is high, but the respective value t in point B is still low. In addition, the model overestimates the peak flow regarding the second event and this discrepancy is caused by a reduced lateral flow. Figure 9 represents all the examined

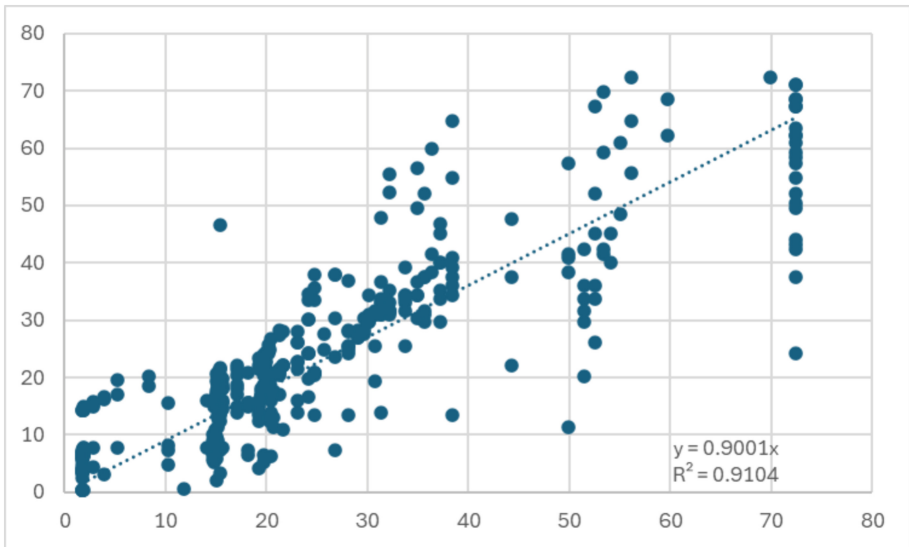


Fig. 8. Predicted Outflow (X Axis) VS Real Outflow (Y Axis)

events within a unique figure (for simplicity) where the two reasons for overestimation can be distinguished.

Thus, during this temporal delay, the model predicts higher outflow than the actual one. This fact enhances the efficiency of the model as it seems that the model starts giving large values to the outflow, at the moment that the volume starts to increase in the inflow, as a result of which it can essentially "predict" the flood phenomenon.

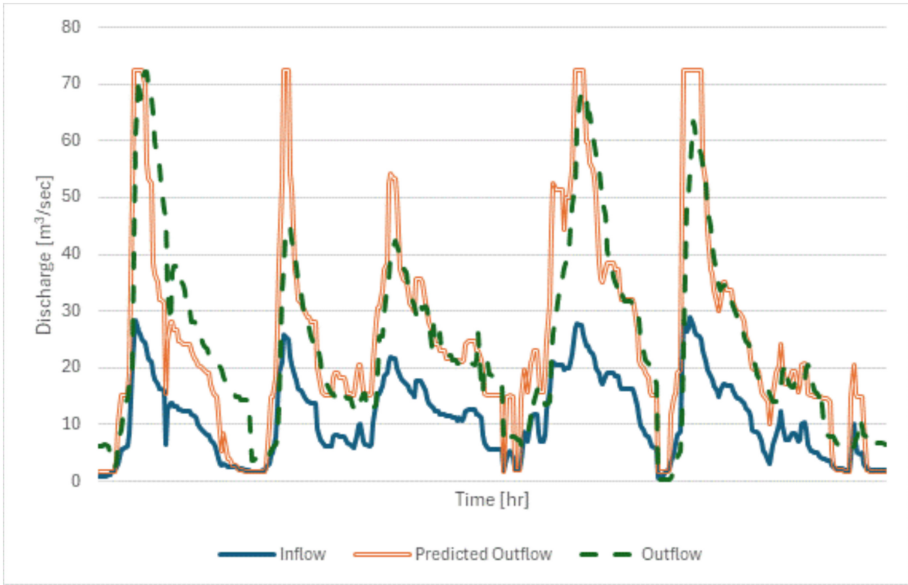


Fig. 9. The flood events: inflow, observed and the predicted outflow

6 Conclusion

In this research effort, an innovative approach for prediction of the outflow of the Ebro River in Spain has been being presented. The source dataset comprises of 1,489 data records with pairs (x_i, y_i) of values where X_i is the inflow at point A and y_i is the corresponding outflow at point B. Data preprocessing was performed using Matlab interface, with scripts written from scratch. After data pre-processing the final dataset is a 368x2 table where the first column (inflow) is the independent variable, and the second column (outflow) is the dependent variable. The authors exploited the potential of the FCM algorithm for developing 12 clusters. The centers of the clusters were used to form the parameters of the Trapezoidal and Triangular memberships functions to be used in the FIS model. Hence, the FCM is used instead of a conventional calibration procedure to modulate the rules of the Mamdani system. The model achieved satisfying results as it accurately predicted the outflow at the ending point B, given the inflow at the starting point A. R^2 was estimated to be as high as 0.9104. Although the results look very

promising, there is always room for further improvement. The robustness of the proposed approach is paving the way for further research in this scientific field.

Testing this approach on different datasets should determine if the model can be applied in different flood areas. Additionally, the authors should experiment more with the limits of the membership functions of the FIS model. Last but not least more ML approaches should be exploited towards a hybrid modeling effort that could have potentially higher accuracy.

References

1. Nayak, J., Naik, B., Behera, H.: Fuzzy C-means (FCM) clustering algorithm: a decade review from 2000 to 2014. In: *Computational Intelligence in Data Mining*, vol. 2: *Proceedings of the International Conference on CIDM, 20-21 December 2014*, pp. 133-149. Springer India (2015).
2. Bezdek, J. C.: Pattern recognition with fuzzy objective function algorithms. In: Springer Science and Business Media (2013).
3. Dunn, J. C.: A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters (1973)
4. Bezdek, J.C., Ehrlich, R., Full, W.: FCM: The fuzzy c-means clustering algorithm. *Comput. Geosci.* **10**(2–3), 191–203 (1984)
5. Gustafson, D. E., & Kessel, W. C.: Fuzzy clustering with a fuzzy covariance matrix. In: *1978 IEEE Conference on Decision and Control Including the 17th Symposium On Adaptive Processes*, pp. 761-766. IEEE. (1979)
6. Gath, I., Geva, A.B.: Unsupervised optimal fuzzy clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **11**(7), 773–780 (1989)
7. Khalilia, M.A., Bezdek, J., Popescu, M., Keller, J.M.: Improvements to the relational fuzzy c-means clustering algorithm. *Pattern Recogn.* **47**(12), 3920–3930 (2014)
8. Spiliotis, M., Sordo-Ward, A., Garrote, L.: Estimation of fuzzy parameters in the linear Muskingum model with the aid of particle swarm optimization. *Sustainability* **13**(13), 7152 (2021)
9. Li, L., Jun, K.S.: Review of machine learning methods for river flood routing. *Water* **16**(2), 364 (2024)
10. Kwon, M., Kwon, H.H., Han, D.: A hybrid approach combining conceptual hydrological models, support vector machines and remote sensing data for rainfall-runoff modeling. *Remote Sensing* **12**(11), 1801 (2020)
11. Hassanvand, M.R., Karami, H., Mousavi, S.F.: Investigation of neural network and fuzzy inference neural network and their optimization using meta-algorithms in river flood routing. *Nat. Hazards* **94**, 1057–1080 (2018)
12. Sen, Z.: Fuzzy logic and hydrological modeling. CRC Press (2009)
13. Spiliotis, M., Garrote, L.: Applying a Flexible Fuzzy Adaptive Regression to Runoff Estimation. *Envir. Sci. Proc.* **25**(1), 85 (2023)
14. Comín, F.A.: Management of the Ebro River Basin: Past, present and future. *Water Sci. Technol.* **40**(10), 161–168 (1999)
15. Sabri, N., Aljunid, S.A., Salim, M.S., Badlishah, R.B., Kamaruddin, R., Malek, M.A.: Fuzzy inference system: short review and design. *Int. Rev. Autom. Control* **6**(4), 441–449 (2013)
16. Azeem, M. F. (Ed.): *Fuzzy inference system: theory and applications*. BoD–Books on Demand. (2012)

17. Cao, J., et al.: Fuzzy inference system with interpretable fuzzy rules: advancing explainable artificial intelligence for disease diagnosis—a comprehensive review. *Inf. Sci.* **662**, 120212 (2024)
18. MathWorks-Fuzzy Logic Toolbox: <https://www.mathworks.com/products/fuzzy-logic.html>
19. Psathas, A. P., Iliadis, L., Papaleonidas, A.: Strain prediction of a bridge deploying autoregressive models with Arima and machine learning algorithms. In *International Conference on Engineering Applications of Neural Networks* (pp. 403-419). Cham: Springer Nature Switzerland. (2023)



Forecasting vehicle crossing volumes by using Nonlinear Autoregressive Neural Networks sets

Ioannis Skopelitis^(✉), Antonios Papaleonidas^{ib}, Anastasios Panagiotis Psathas^(✉)^{ib}, Lazaros Iliadis^{ib}, and George N. Botzoris^{ib}

Department of Civil Engineering-Lab of Mathematics and Informatics (ISCE), Democritus University of Thrace, 67100 Xanthi, Greece

johnskope2000@gmail.com, {papaleon, anpsatha, liliadis, gbotzoris}@civil.duth.gr

Abstract. Traffic congestion remains one of the most significant issues affecting highways worldwide. This problem is directly associated with productivity loss (due to time wasted in traffic jams), environmental pollution, increased fuel consumption, and adverse effects on human health. Toll booths constitute a major contributor to congestion, especially during peak hours and periods of high travel demand, such as holidays and vacation seasons. In this research, the authors examine the case of the Egnatia Odos Motorway, focusing on the Mesti-Komotini toll station over a three-year period. Using a large set of Nonlinear Autoregressive Neural Networks (NARNNs), the study forecasts vehicle crossing volumes based on past traffic values. The models were trained using the Hold-Out Validation Method, and their performance was evaluated using the R-squared (R^2) index, achieving values above 0.94. The results are highly promising, indicating that the more historical data is available, the higher the prediction accuracy becomes. The proposed models can serve as valuable tools for traffic management authorities to alleviate motorway congestion when necessary.

Keywords: Machine Learning · Neural Networks · Univariate · Autoregression · toll · traffic management

1 Introduction

The increase in traffic due to the growth of global tourism and trade puts considerable pressure on transportation infrastructure, which can lead to negative consequences [1]. In recent years, the analysis and forecasting of travel and trade demand have gained increasing importance, as transportation operators recognize their value in improving both infrastructure and services. Long-term forecasting is crucial for the effective planning of new transportation infrastructure, while short-term forecasting contributes to the optimal management of operational needs [27]. Incorrect demand estimation can lead to serious consequences. Overestimation results in overinvestment and underutilization of resources, thereby wasting valuable assets. Conversely, underestimation can cause shortages in transport services, reduce service quality, and create mobility challenges

[28]. Forecasting travel demand is directly linked to the challenges facing urban areas and has wider implications for the economy. Transport is a sector that depends on the needs of other sectors, as travel occurs for reasons such as work, education, leisure and social activities [29]. Accordingly, the transport of goods depends on demand in the production and trade sectors. This "derivative" nature of transport makes it a key factor in economic development at local and national levels [30]. Accurate forecasting of transport demand is crucial in the context of transport planning, as it provides valuable information for:

- The assessment of traffic congestion in the study area.
- The potential levels of environmental pollution.
- The risk of traffic accidents due to high traffic.
- The need to upgrade existing infrastructure or build new ones.
- The estimation of revenues from future transport projects.

Based on these forecasts, the competent bodies can design appropriate interventions, determine the specifications of the projects, assess their impacts and proceed with economic analyses for their implementation. Systematic demand forecasting, therefore, constitutes a key tool for strategic transport planning and effective infrastructure management [2].

The main consequence of failure to predict travel speeds and trip duration on highways is none other than traffic congestion. Traffic congestion is one of the most important problems on highways today. This problem is directly linked to lost productivity due to time spent in traffic jams [3], environmental pollution [4], increased fuel consumption [5], and risks to human health [6]. Manned toll booths are one of the main factors contributing to congestion, particularly during peak hours or periods of increased travel due to holidays and vacations. The costs associated with driver waiting time (due to the need to stop and pay tolls), as well as the operating costs of staffed toll stations, can amount to up to a third of total toll revenue. This highlights the need for more efficient solutions, such as electronic toll payment, to reduce delays and improve traffic flow. In order to address the problem of congestion on motorways, it is observed that one of the main factors causing it is the manned toll booths. The intensity of congestion depends on the number of active toll booths (staffed booths). The more booths are open, the less congestion there is. However, operating more toll booths also entails higher operating costs. Consequently, there is a trade-off between traffic congestion (i.e. increased waiting times for drivers) and the cost of operating toll booths. Determining the optimal number of open booths is a critical practical issue for highway management [7].

2 Literature review

In 2003, Williams and Hoel [12] demonstrated that Autoregressive Integrated Moving Average (ARIMA) models can effectively capture short-term traffic variations, particularly for highway traffic data. Their study reported a Mean Absolute Percentage Error (MAPE) of 9.8% for short-term traffic forecasting but noted performance degradation for nonlinear traffic patterns. In 1984, Okutani and Stephanedes [13] introduced the use of Kalman filters for real-time traffic estimation, reporting an accuracy of 85% based on

Root Mean Square Error (RMSE) evaluation. Later, in 2007, Xie et al. [14] found that Kalman filters struggle with abrupt traffic changes, with RMSE increasing by 25% during peak congestion periods. In 2007, Su et al. [15] applied multiple linear regression models to predict traffic volume, achieving an R-squared value of 0.72 and a MAPE of 12.3%. Their study highlighted the model's limitations in capturing nonlinear dependencies in toll traffic flow.

Recent advances in machine learning (ML) have led to the development of robust predictive models capable of handling complex traffic patterns. The emergence of ML techniques has improved the accuracy and scalability of traffic forecasting. In 1997, Dougherty and Cobbett [16] developed Artificial Neural Network (ANN) models for traffic flow prediction, reporting a Mean Squared Error (MSE) of 0.0025 and a MAPE of 7.4%. However, their study highlighted long training times and sensitivity to hyperparameters. In 2006, Zheng et al. [17] proposed an Support Vector Machines (SVMs)-based model for urban traffic flow prediction, achieving an RMSE of 3.5 and an R-squared value of 0.81. However, their study found that model performance degraded with increasing dataset size due to computational overhead. In 2001, Breiman [18] introduced Random Forests, which later studies (Liu et al., 2018 [19]) found to outperform standalone decision trees for toll traffic prediction. In 2018, Liu et al. [19] reported a 92% classification accuracy for toll congestion prediction using Random Forests, compared to 85% with single decision trees.

Deep learning has revolutionized traffic prediction by enabling spatiotemporal feature extraction from large datasets. In 2015, Ma et al. [20] demonstrated the effectiveness of Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) in handling sequential traffic data, achieving a MAPE of 5.8% and an RMSE of 2.7, significantly outperforming ARIMA (MAPE = 9.8%). In 2015, Lv et al. [21] applied Convolutional Neural Networks (CNNs) to traffic heatmaps, reporting an accuracy of 89.3% for congestion level prediction, outperforming traditional statistical methods by 12%. In 2020, Wu et al. [22] explored Graph Neural Networks (GNNs) for road network modeling, achieving a prediction accuracy of 94.1% and an RMSE of 1.9, marking a substantial improvement over RNN-based methods. To enhance predictive performance, researchers have explored hybrid models combining multiple ML techniques. In 2019, Xu et al. [23] developed a hybrid LSTM-ARIMA model, achieving an RMSE of 1.8 and a MAPE of 4.5%, improving prediction accuracy over standalone models. In 2017, Vaswani et al. [24] introduced the Transformer architecture, which has been adapted for dynamic toll traffic prediction. In 2021, Sun et al. [25] reported a 96% prediction accuracy using attention-based models, significantly improving over LSTM (92%). In 2016, Chen and Guestrin [26] demonstrated the power of Gradient Boosting, with more recent applications in toll traffic forecasting [25] reporting a 95.2% accuracy and a reduction in RMSE by 30% compared to standalone deep learning models.

In this paper, the authors use a publicly available dataset provided by Egnatia Motorway S.A. Specifically, these are the hourly vehicle crossings of all categories from the Mesti front toll station of Egnatia Motorway, in both directions, namely towards Komotini (Kipoi - Igoumenitsa) and towards Alexandroupoli (Igoumenitsa - Kipi), from 01/01/2022 to 31/12/2024. That is, three years of data are used and a total of 26,225 instances in both directions, i.e. towards Komotini (Kipoi – Igoumenitsa) and towards

Alexandroupoli (Igoumenitsa – Kipoi). The aim of this research effort is the prediction of the future value of vehicle crossings of all categories. The goal is to perform a comprehensive analysis of the dataset, deploying a vast amount of Autoregressive (AR) models, using Artificial Neural Networks (ANNs), in order to exploit which, feature (dataset split, hidden neurons, lags) affects forecasting more, achieving very promising results and paving the way for future research.

The rest of the paper is organized as follows. Section 3 describes the area of research. Section 4 describes the dataset, its features and the pre-processing process. Section 5 provides the architecture of the proposed models, and the evaluation index used for this research effort. Section 6 presents the experimental results and the evaluation of the model. Finally, Sect. 7 concludes the research.

3 Area of Research

Motorway 2 (A2), also known as Egnatia Odos, is recognized as the longest motorway in Greece, with a total length of 670 kilometers, and is part of the European Route 90. Its route extends from Igoumenitsa in Epirus to the Greek - Turkish border at Kipi in Thrace. Its main construction phase was completed between 1994 and 2009, while its last sections were delivered in 2014. The modern Egnatia Odos was developed near National Road 2, which it replaced for the most part, in the Thessaloniki–Kipi section. Its name comes from the Ancient Egnatia Odos, which crossed approximately the same geographical area. In addition, sections of various national and provincial roads were replaced, such as National Road 4 (Alexandria–Kozani), National Road 20 (Kozani–Siatista), National Road 15 (Siatista–Grevena), the Grevena–Metsovo provincial road, and National Road 6 (Panagia–Igoumenitsa) (Fig. 1).

The motorway has two carriageways, one towards Igoumenitsa and one towards Kipi in Evros, which are separated by a safety island. Each carriageway includes two traffic lanes and one emergency lane (LEA - hard shoulder), while in the Kleidi–Lagkadas section there are three traffic lanes in each direction. The maximum permitted speed has been set at 130 km/h. Due to the mountainous terrain of the Pindos mountain range and the Vermio massif, numerous tunnels and bridges have been constructed. Specifically, there are 73 twin road tunnels along the motorway, with a total length of 100 km (50 km in each direction) and a construction cost of €1.8 billion. The total travel time from Igoumenitsa to Kipi is estimated at approximately 6 hours and 10 minutes. The total construction cost of the project estimated at €6 billion. The motorway includes:

- 63 junctions with the road network.
- 177 major bridges, with a total length of 42 km.
- 350 overpasses and underpasses.
- 73 tunnels, with a maximum length of 4.8 km and a total length of 50 km.
- 43 river crossings.
- 11 railway crossings.

The highway's area of influence accounts for 36% of the country's total population and produces 33% of the gross domestic product.

It is recorded that 54% of the country's total agricultural land and 65% of the irrigated area are located in the region served by Egnatia motorway. It is also noted that 41% of

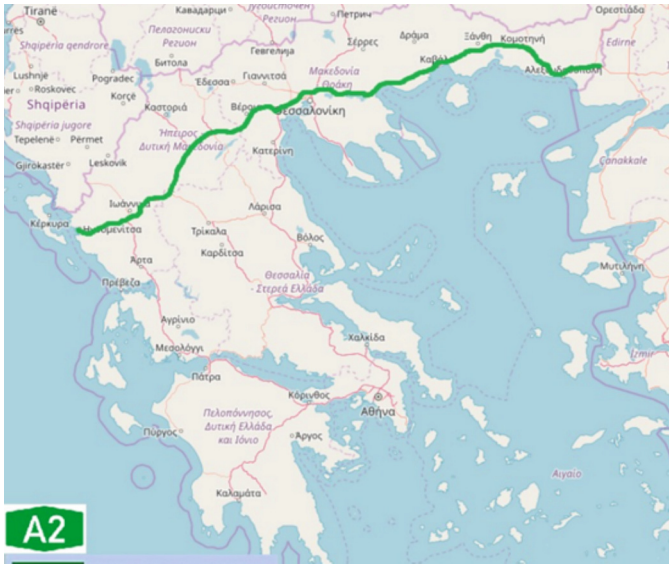


Fig. 1. The map of the Egnatia Motorway.

total employment in industry and crafts comes from the serving region, while 51% of the country's mining activity also takes place within this region.

There are fourteen front toll stations operating on the Egnatia Motorway, four front stations on the vertical axes, one off-axis station, and seventeen side toll stations. Specifically, the front toll stations on the main motorway axis are the following:

1. Tyria toll station (Ioannina Prefecture)
2. Pamvotida toll station (Ioannina Prefecture)
3. Malakassi toll station (Trikali Prefecture)
4. Siatista toll station (Grevena Prefecture)
5. Polymylos toll station (Kozani Prefecture)
6. Malgara toll station (Thessaloniki Prefecture)
7. Thessaloniki toll station (Thessaloniki Prefecture)
8. Analipsi Toll Station (Thessaloniki Prefecture)
9. Asprovalta Toll Station (Thessaloniki Prefecture)
10. Moustheni Toll Station (Kavala Prefecture)
11. Kavala Toll Station (Kavala Prefecture)
12. Iasmos Toll Station (Rhodopi Prefecture)
13. Mestis Toll Station (Evros Prefecture)
14. Ardanios Toll Station (Evros Prefecture)

At the Egnatia Motorway toll stations, vehicles are served either through manned stations or, starting from 1 September 2020, through automatic electronic stations using the EgnatiaPass service (Fig. 2). In the latter case, the toll is charged automatically via a NFC transceiver installed in the vehicle, which is connected to the corresponding user account.

Depending on the type of vehicle passing through each toll station, the fee that the driver must pay changes (the higher the category, the higher the fee). The four vehicle categories are as follows:

- Category 1: this category includes two-wheelers and three-wheelers.
- Category 2: this category includes light vehicles, such as vehicles with or without a trailer and a height of up to 2.70m.
- Category 3: this category includes trucks, buses and other vehicles with fewer than four axles, i.e. vehicles with or without a trailer with two or three axles and a height greater than 2.70m.
- Category 4: this category includes trucks and other vehicles with four or more axles, i.e. vehicles with or without a trailer with four or more axles and a height greater than 2.70m.

Between Komotini and Alexandroupoli lies the Mesti toll station, located on the main axis of the motorway, in the Prefecture of Evros. It serves vehicles through three in each direction crossing positions (Fig. 2). Of these positions, two are manned and one is an automatic service point using EgnatiaPass as mentioned above.



Fig. 2. Mesti front toll station, direction from Kipoi to Igoumenitsa

4 Dataset

The dataset deals with finite sequences of numerical measurements (samples), collected automatically by the toll management system and represented as time series [8]. A time series of length n is an ordered sequence of values $x = x[1], \dots, x[n]$ of finite precision. A subsequence $x[a : b]$ of x is defined as follows

$$x[a : b] = (x[a], x[a + 1], \dots, x[b]), \quad 1 \leq a < b \leq n \quad (1)$$

The data that will be used at this research are provided by Egnatia Motorway S.A. Specifically, these are the hourly vehicle crossings of all categories from the Mesti front toll station of Egnatia Motorway, in both directions, namely towards Komotini (Kipoi – Igoumenitsa) and towards Alexandroupoli (Igoumenitsa – Kipi), from 01/01/2022 to 31/12/2024. That is, three years of data are used, and a total of 26,225 instances in both directions, i.e., towards Komotini (Kipoi – Igoumenitsa) and towards Alexandroupoli (Igoumenitsa – Kipi). However, the range of these crossings is large, so it is advisable to apply a normalization process to this data.

According to the literature review [9, 10], normalizing time series data usually leads to better results. Normalization assumes that the observations fit a Gaussian distribution (bell curve) with a well-behaved mean and standard deviation. A common and effective normalization method is mean normalization. Mean normalization is defined as a process used in machine learning in which the average of each feature is calculated and subtracted. It is used to implement feature scaling, which brings all features of a model into a similar range. Feature scaling is a method used to normalize the range of independent variables or features of data, that is, the process of converting all features of a problem to a similar scale or range. Feature scaling can significantly affect the training efficiency of a machine learning model and improve the time required to train it. For the standardization process, Eq. 2 was used:

$$y = \frac{x - \hat{x}}{x_{\max} - x_{\min}} \quad (2)$$

where, x is the value of the time series, $\hat{x}$ is the mean value of all x , $x_{\max}$ is the maximum value of the time series, $x_{\min}$ is minimum value of the time series and y is the value of the normalized Dataset. Thus, Figs. 3 and 4 present the normalized values for vehicle crossings of all categories in both directions, i.e., towards Komotini (Kipoi – Igoumenitsa) and towards Alexandroupoli (Igoumenitsa – Kipi), respectively.

Normalization to the mean, as mentioned above, is an important procedure in the efficient training and operation of a neural network. When gradient descent is applied, the method converges to the minimum faster if the data is normalized. If the data are not normalized, gradient descent requires more steps before converging to the minimum, which can slow down the model's learning process. Data handling was performed in MATLAB with code written from scratch.

5 Algorithms and Evaluation Indices

5.1 Autoregressive Models

Autoregressive (AR) models are based on the idea that the current value of a time series, X_t , can be explained as a linear combination of the p previous values, X_{t-1} , X_{t-2} , ..., X_{t-p} , along with a random error in the series itself. An autoregressive model of order p , abbreviated AR(p), has the form:

$$X_t = \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \dots + \varphi_p X_{t-p} + w_t = \sum_{i=1}^p \varphi_i X_{t-i} + w_t \quad (3)$$

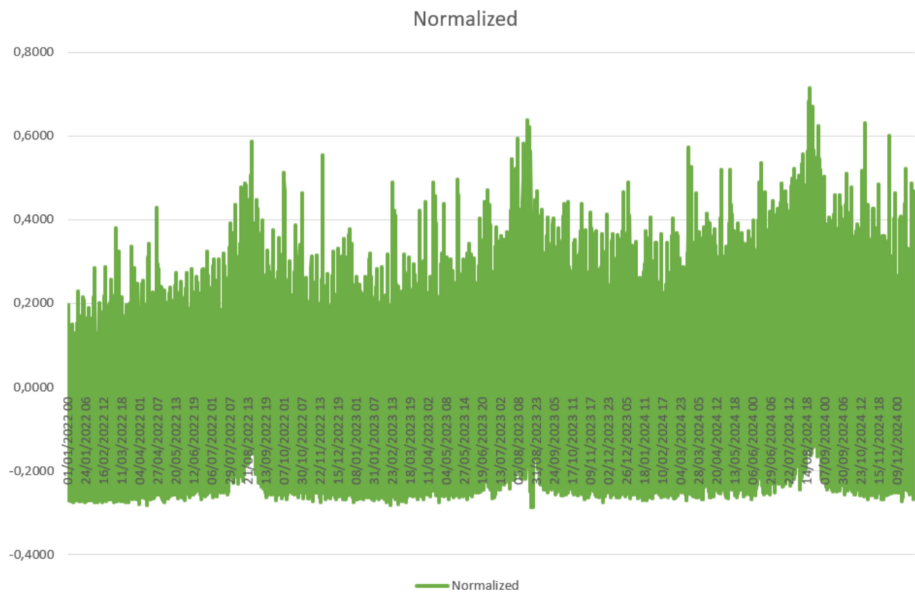


Fig. 3. Normalized hourly vehicle crossings at the Mestis front toll station, direction Komotini – Alexandroupoli (Kipoi – Igoumenitsa)

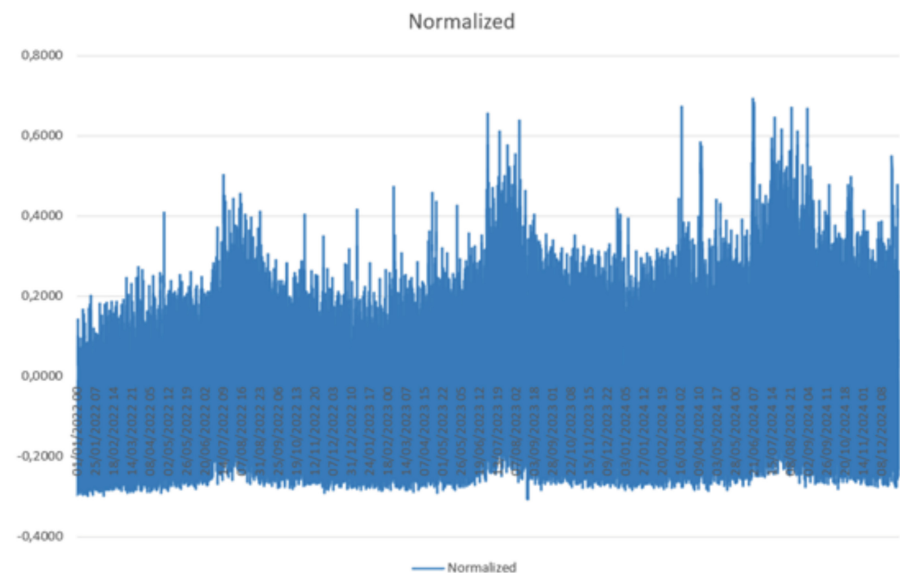


Fig. 4. Normalized hourly vehicle crossings at the Mestis front toll station, direction Alexandroupoli – Komotini (Igoumenitsa – Kipi)

where X_t is stationary, $w_t \sim w_n(0, \sigma^2 w)$, and $\phi_1, \phi_2, \dots, \phi_p$ (with $\phi_p \neq 0$) are the parameters of the model. The hyperparameter p represents the length of the "direct lookback" in the series.

To further explore the AR process, we need to introduce the backshift operator. The backshift operator is defined as $BX_t = X_{t-1}$. It can be expanded, $B^2X_t = B(BX_t) = B(X_{t-1}) = X_{t-2}$, and so on. Thus, $B^kX_t = X_{t-k}$. We can also define an inverse operator (forward shift operator) by imposing $B^{-1}B = 1$, such that $X_t = B^{-1}BX_t = B^{-1}X_{t-1}$. Using the folding operator, Eq. 3 can be rewritten as:

$$X_t - \phi_1X_{t-1} - \phi_2X_{t-2} - \dots - \phi_pX_{t-p} = w_t \quad (4)$$

$$(1 - \phi_1B - \phi_2B^2 - \dots - \phi_pB^p)X_t = w_t \quad (5)$$

The autoregressive operator is defined as:

$$\varphi(B) = 1 - \phi_1B - \phi_2B^2 - \dots - \phi_pB^p = 1 - \sum_{j=1}^p \phi_jB^j \quad (6)$$

Then, the AR(p) process can be rewritten more concisely as:

$$\varphi(B)X_t = w_t \quad (7)$$

The simplest AR process is AR(0), which has no dependence between terms. In fact, AR(0) is essentially white noise. AR(1) can be given as $X_t = \phi_1X_{t-1} + w_t$.

- Only the previous term in the process and the noise term contribute to the output.
- If $|\phi_1|$ is close to 0, then the process still resembles white noise.
- If $\phi_1 < 0$, then X_t tends to oscillate between positive and negative values.
- If $\phi_1 = 1$, then the process is equivalent to a random walk, which is not stationary since the variance depends on t (and is infinite) [31]

5.2 Non-linear Autoregression Neural Network (NARNN)

In most cases, time series applications are characterized by high fluctuations and transient transition periods. This fact makes it difficult to process the time series with a linear model, so a nonlinear approach should be proposed. A nonlinear autoregressive neural network, which is applied to time series forecasting, describes a discrete, nonlinear, autoregressive model that can be written as follows:

$$y(t) = h(y(t-1), y(t-2), \dots, y(t-p)) + \varepsilon(t) \quad (8)$$

This equation describes how a NARNN is used to predict the value of a time series y at time t , $y(t)$, using the p previous values of the series. The function $h(\cdot)$ is unknown a priori, and training the neural network aims to approximate it by optimizing the network weights and shifting the neurons. Finally, the term $\varepsilon(t)$ represents the error of the approximation of the time series y at time t . The topology of a NARNN network is shown in Fig. 5. The features p , $y(t-1)$, $y(t-2)$, ..., $y(t-p)$, are called feedback delays. The number of

hidden layers and neurons per layer is fully flexible and is optimized through a trial and error process to determine the network topology that provides the best performance. However, it is important to consider that increasing the number of neurons makes the system more complex, while a low number of neurons can limit the generalization abilities and computational power of the network.

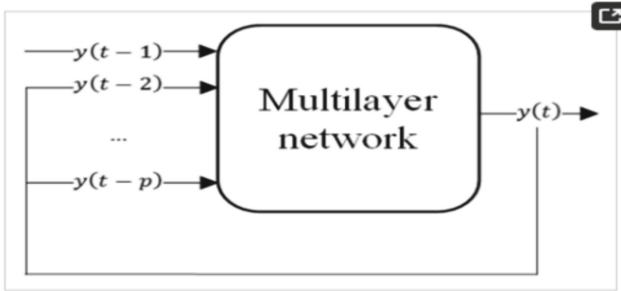


Fig. 5. NARNN architecture

The most common learning rule for the NAPNN is the Levenberg-Marquardt backpropagation process (LMBP). This training function is often considered the fastest backpropagation-type algorithm. The LMBP algorithm is designed to approximate the second derivative without requiring the calculation of the Hessian matrix, which leads to increased training speed. When the performance function takes the form of a sum of squares (as is often the case in training feedforward neural networks), then the Hessian matrix can be approximated, as shown in the equation $\mathbf{H} = \mathbf{J}^T \mathbf{J}$. The slope can be calculated from the equation $\mathbf{g} = \mathbf{J}^T \mathbf{e}$.

In the above two equations, $\mathbf{J}$ is the Jacobian matrix that includes the first derivatives of the network errors with respect to the weights and biases, and $\mathbf{e}$ is a vector of the network errors over all training samples. To estimate the Jacobian matrix, the study in uses a standard backpropagation algorithm to approximate the Hessian matrix. This approach is simpler than computing the Hessian matrix. The LMBP algorithm uses this approach in the Newton-type update described in the equation $\mathbf{x}_{k+1} = \mathbf{x}_k - [\mathbf{J}^T \mathbf{J} + \mu \mathbf{I}]^{-1} \mathbf{J}^T \mathbf{e}$.

It should be noted that this method uses the Jacobian matrix for the calculations, assuming that the performance function is the average of the sum of squared errors.

In this study, NAR neural networks are used to model the normalized time series of traffic flow as follows: the network structure includes an input (corresponding to the traffic flow at time $t-1$, $y(t-1)$), and an output (the next value of the time series, $y(t)$, to be predicted). The number of delays (or lag) is adjusted experimentally after a stage of data preprocessing and analysis. [32]

5.3 Evaluation of Deep Learning Algorithms.

For the evaluation of the aforementioned NARNN algorithms the R^2 evaluation index was used. The index is calculated by the following equation:

$$R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2} \quad (9)$$

where y_i is the actual value, $\hat{y}_i$ is the predicted value, $\bar{y}$ is the mean value, and N is the total number of the instances. R^2 measures how much variability in dependent variable can be explained by the model. R^2 is valued in closed interval $[0, 1]$ and a bigger value indicates a better fit between prediction and actual value R Square is a good measure to determine how well the model fits the dependent variables [11].

6 Evaluation and Experimental Results

All algorithms, their evaluation and the training-testing process were made in Matlab platform. The parameters and characteristics of the neurons used are as follows:

- For the training, validation and testing percentages, the combinations 65-15-20 (65% training, 15% validation, 20% testing), 65-20-15 (65% training, 20% validation, 15% testing) and 70-15-15 (70% training, 15% validation, 15% testing) were chosen
- For the hidden neurons, the values 14, 18, 22 were chosen
- For the delays / (lags) d the values 2, 3, 4, 12, 24 were used

In Tables 1 and 2, the Training-Validation-Testing split, lags, Hidden Neurons, Best Validation Performance and R^2 (Training, Validation, Testing, Overall) are being presented in both directions, i.e. towards Komotini (Kipoi – Igoumenitsa) and towards Alexandroupoli (Igoumenitsa – Kipi), respectively.

From the two tables it is observed that the NARNNs can predict with high accuracy the total volume of crossing vehicles from the tolls. More specifically, the achieving R^2 for the first case is 0.954 for training, 0.94 for validation and testing and 0.942 overall, while in the second case R^2 is 0.951 for training, 0.941 for validation, 0.942 for testing and 0.943 overall. In both cases R^2 indicates that the volume of crossing vehicles in both directions follows a similar pattern. Additionally, Training, Validation and Testing indices are almost identical, a fact that states that the NARNNs can generalize with high accuracy. Furthermore, it can be noted that the split ratio and hidden neurons do not play a significant role in increasing performance. The parameter that seems to affect the optimization of the forecast is the number of delays / lags. The more past values we take into account, the more accurate the forecast becomes.

Table 1. Results for vehicle crossings towards Komotini (Kipoi – Igoumenitsa)

Training-Validation-Testing Split (%)	Lags	Hidden Neurons	Best Validation Performance	Training R ²	Validation R ²	Testing R ²	All R ²
65-20-15	2	14	0.0039376	0.878	0.878	0.877	0.878
65-20-15	2	18	0.0041891	0.882	0.874	0.879	0.880
65-20-15	2	22	0.0039274	0.881	0.883	0.875	0.881
65-20-15	3	14	0.0034293	0.893	0.898	0.895	0.894
65-20-15	3	18	0.0036585	0.893	0.888	0.895	0.892
65-20-15	3	22	0.0035466	0.896	0.892	0.889	0.894
65-20-15	4	14	0.0032197	0.902	0.901	0.890	0.900
65-20-15	4	18	0.0032755	0.900	0.902	0.898	0.900
65-20-15	4	22	0.0032949	0.902	0.900	0.901	0.901
65-20-15	12	14	0.0022730	0.931	0.929	0.925	0.930
65-20-15	12	18	0.0022873	0.928	0.929	0.930	0.929
65-20-15	12	22	0.0025647	0.934	0.923	0.923	0.930
65-20-15	24	14	0.0020109	0.940	0.938	0.938	0.940
65-20-15	24	18	0.0019743	0.942	0.940	0.935	0.941
65-20-15	24	22	0.0020794	0.944	0.938	0.937	0.942
65-15-20	2	14	0.0039708	0.881	0.879	0.876	0.880
65-15-20	2	18	0.0038815	0.877	0.882	0.882	0.879
65-15-20	2	22	0.0038612	0.880	0.883	0.877	0.880
65-15-20	3	14	0.0034547	0.954	0.892	0.870	0.891
65-15-20	3	18	0.0033841	0.896	0.896	0.886	0.894
65-15-20	3	22	0.0034316	0.895	0.897	0.884	0.893
65-15-20	4	14	0.0033640	0.900	0.899	0.900	0.900
65-15-20	4	18	0.0031090	0.900	0.906	0.900	0.901
65-15-20	4	22	0.0034526	0.902	0.894	0.898	0.900
65-15-20	12	14	0.0024623	0.929	0.924	0.931	0.929
65-15-20	12	18	0.0022206	0.929	0.933	0.930	0.930
65-15-20	12	22	0.0022471	0.931	0.931	0.924	0.930
65-15-20	24	14	0.0019921	0.939	0.938	0.937	0.938
65-15-20	24	18	0.0022208	0.941	0.932	0.934	0.938
65-15-20	24	22	0.0021825	0.941	0.934	0.937	0.939
70-15-15	2	14	0.0038741	0.880	0.881	0.878	0.880
70-15-15	2	18	0.0040365	0.881	0.878	0.880	0.880

(continued)

Table 1. (continued)

Training-Validation-Testing Split (%)	Lags	Hidden Neurons	Best Validation Performance	Training R^2	Validation R^2	Testing R^2	All R^2
70-15-15	2	22	0.0040424	0.880	0.873	0.883	0.879
70-15-15	3	14	0.0035897	0.894	0.889	0.888	0.892
70-15-15	3	18	0.0035494	0.895	0.890	0.893	0.894
70-15-15	3	22	0.0036902	0.896	0.884	0.888	0.893
70-15-15	4	14	0.0032416	0.900	0.906	0.905	0.902
70-15-15	4	18	0.0034230	0.902	0.898	0.898	0.901
70-15-15	4	22	0.0032004	0.904	0.902	0.898	0.903
70-15-15	12	14	0.0024179	0.927	0.929	0.930	0.928
70-15-15	12	18	0.0024018	0.931	0.928	0.930	0.930
70-15-15	12	22	0.0024834	0.933	0.925	0.920	0.930
70-15-15	24	14	0.0020210	0.943	0.937	0.938	0.942
70-15-15	24	18	0.0020994	0.945	0.935	0.933	0.941
70-15-15	24	22	0.0022166	0.942	0.932	0.940	0.940

Table 2. Results for vehicle crossings towards Alexandroupoli (Igoumenitsa – Kipi)

Training-Validation-Testing Split (%)	Lags	Hidden Neurons	Best Validation Performance	Training R^2	Validation R^2	Testing R^2	All R^2
65-20-15	2	14	0.0042350	0.887	0.884	0.888	0.887
65-20-15	2	18	0.0040369	0.887	0.882	0.888	0.886
65-20-15	2	22	0.0038102	0.885	0.886	0.891	0.886
65-20-15	3	14	0.0035406	0.896	0.898	0.894	0.896
65-20-15	3	18	0.0035602	0.897	0.899	0.893	0.897
65-20-15	3	22	0.0036592	0.894	0.892	0.891	0.896
65-20-15	4	14	0.0035619	0.901	0.893	0.897	0.899
65-20-15	4	18	0.0033146	0.899	0.901	0.894	0.899
65-20-15	4	22	0.0034087	0.902	0.895	0.895	0.900
65-20-15	12	14	0.0025384	0.930	0.927	0.930	0.929
65-20-15	12	18	0.0025130	0.933	0.930	0.927	0.932
65-20-15	12	22	0.0023908	0.931	0.929	0.930	0.931
65-20-15	24	14	0.0021052	0.943	0.939	0.937	0.942
65-20-15	24	18	0.0020837	0.945	0.936	0.937	0.942
65-20-15	24	22	0.0021384	0.951	0.937	0.938	0.943

(continued)

Table 2. (continued)

Training-Validation-Testing Split (%)	Lags	Hidden Neurons	Best Validation Performance	Training R ²	Validation R ²	Testing R ²	All R ²
65-15-20	2	14	0.0042350	0.887	0.879	0.890	0.886
65-15-20	2	18	0.0040369	0.885	0.885	0.881	0.884
65-15-20	2	22	0.0038102	0.884	0.890	0.883	0.885
65-15-20	3	14	0.0035406	0.897	0.897	0.892	0.896
65-15-20	3	18	0.0035602	0.898	0.895	0.897	0.897
65-15-20	3	22	0.0036592	0.896	0.892	0.897	0.896
65-15-20	4	14	0.0035619	0.901	0.897	0.898	0.900
65-15-20	4	18	0.0033146	0.899	0.902	0.901	0.900
65-15-20	4	22	0.0034087	0.902	0.904	0.899	0.901
65-15-20	12	14	0.0025384	0.930	0.928	0.929	0.930
65-15-20	12	18	0.0025130	0.932	0.929	0.924	0.929
65-15-20	12	22	0.0023908	0.932	0.933	0.927	0.931
65-15-20	24	14	0.0021052	0.943	0.941	0.937	0.941
65-15-20	24	18	0.0020837	0.945	0.941	0.937	0.943
65-15-20	24	22	0.0021384	0.946	0.937	0.939	0.943
70-15-15	2	14	0.0037886	0.887	0.890	0.884	0.887
70-15-15	2	18	0.0041487	0.886	0.877	0.885	0.884
70-15-15	2	22	0.0039524	0.885	0.886	0.892	0.887
70-15-15	3	14	0.0035636	0.896	0.898	0.899	0.897
70-15-15	3	18	0.0036163	0.898	0.894	0.896	0.897
70-15-15	3	22	0.0036281	0.898	0.895	0.893	0.897
70-15-15	4	14	0.0033536	0.900	0.900	0.901	0.900
70-15-15	4	18	0.0035630	0.903	0.899	0.899	0.902
70-15-15	4	22	0.0035763	0.901	0.896	0.893	0.899
70-15-15	12	14	0.0025187	0.930	0.927	0.929	0.929
70-15-15	12	18	0.0025037	0.932	0.927	0.929	0.931
70-15-15	12	22	0.0024217	0.933	0.931	0.928	0.932
70-15-15	24	14	0.0020081	0.943	0.941	0.941	0.943
70-15-15	24	18	0.0021770	0.944	0.936	0.942	0.943
70-15-15	24	22	0.0020893	0.944	0.940	0.938	0.943

7 Conclusions and Future Work

In this paper, authors address a univariate autoregression problem, aiming to predict the future values of vehicles toll crossings using past crossings. A large number of NARNNs were developed, exploiting different data splitting ratios, numbers of hidden neurons, and

time lags. The research effort yielded promising results, as all models demonstrated the ability to capture the fluctuations in vehicle crossing volumes. The R2 index was chosen as the evaluation metric, and its values are satisfactory for the majority of the models, especially in the case of 24 lags. Additionally, this paper confirms that normalizing the dataset can lead to great model performance.

Although the results were very good, there is always room for improvement. Several directions for future work are already under consideration. One scenario involves identifying the optimal number of time lags to further enhance model accuracy. Moreover, authors could explore alternative approaches beyond NARNNs, such as Recurrent Neural Networks (RNNs) with LSTM layers, or even classical statistical methods like ARIMA, to gain a better understanding of the dataset's behavior. Lastly, additional variables could be incorporated into the time series, such as temperature conditions or traffic congestion data, which are expected to improve model performance. After all, no model is perfect — a model is considered effective when it proves useful in practice.

References

1. Goeldner, C.R., Ritchie, J.B.: *Tourism: Principles, Practices, Philosophies*. John Wiley & Sons, Hoboken (2011)
2. Tyrinopoulos, I., Milioti, C.: *Methods and Applications for Forecasting Demand for Transportation* (2023)
3. De, U.K., Rajbongshi, G.: Statistical application for the analysis of traffic congestion and its impact in a Hill city. *Int. J. Stat. Sci.* **20**, 19–48 (2020)
4. Chowdhury, S., Dey, S., Tripathi, S.N., Beig, G., Mishra, A.K., Sharma, S.: “Traffic intervention” policy fails to mitigate air pollution in megacity Delhi. *Environ. Sci. Policy* **74**, 8–13 (2017)
5. Bharadwaj, P., Gibson, M., Zivin, J.G., Neilson, C.: Gray matters: fetal pollution exposure and human capital formation. *J. Assoc. Environ. Resour. Econom.* **4**(2), 505–542 (2017)
6. Requia, W.J., Adams, M.D., Arain, A., Papatheodorou, S., Koutrakis, P., Mahmoud, M.: Global association of air pollution and cardiorespiratory diseases: a systematic review, meta-analysis, and investigation of modifier variables. *Am. J. Publ. Health* **108**(S2), S123–S130 (2018)
7. Petrović, A., Nikolić, M., Bugarić, U., Delibašić, B., Lio, P.: Controlling highway toll stations using deep learning, queuing theory, and differential evolution. *Eng. App. Artif. Intell.* **119**, 105683 (2023)
8. Vespier, U., Nijssen, S., Knobbe, A.: Mining characteristic multi-scale motifs in sensor-based time series. In: *Proceedings of International Conference on Information and Knowledge Management*, pp. 2393–2398 (2013). <https://doi.org/10.1145/2505515.2505620>
9. Boeva, V., Tsiporkova, E.: A multi-purpose time series data standardization method. In: Sgurev, V., Hadjiski, M., Kacprzyk, J. (eds.) *Intelligent Systems: From Theory to Practice. Studies in Computational Intelligence*, vol. 299, pp. 445–460. Springer, Berlin, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13428-9_22
10. Batyrshin, I.: Constructing time series shape association measures: Minkowski distance and data standardization. In: *2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence*, pp. 204–212. IEEE, September 2013
11. Israeli, O.: A Shapley-based decomposition of the R-square of a linear regression. *J. Econom. Inequal.* **5**(2), 199–212 (2007)

12. Williams, B.M., Hoel, L.A.: Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results. *J. Transport. Eng.* **129**(6), 664–672 (2003)
13. Okutani, I., Stephanedes, Y.J.: Dynamic prediction of traffic volume through Kalman filtering theory. *Transport. Res. Part B: Methodol.* **18**(1), 1–11 (1984)
14. Xie, Y., Zhang, Y., Ye, Z.: Short-term traffic volume forecasting using Kalman filter with discrete wavelet decomposition. *Comput.-Aid. Civil Infrastruct. Eng.* **22**(5), 326–334 (2007)
15. Su, H., Zhang, L., Yu, S.: Short-term traffic flow prediction based on incremental support vector regression. In: *Third international conference on natural computation (ICNC 2007)*, vol. 1, pp. 640–645. IEEE, August 2007
16. Dougherty, M., Cobbett, M.: Short-term inter-urban traffic forecasts using neural networks. *Int. J. Forecast.* **13**(1), 21–31 (1997)
17. Zheng, X., Liu, X., Chen, Y., Wang, H.: Support vector machine-based urban traffic flow prediction and its performance evaluation. In: *Proceedings of the IEEE Intelligent Transportation Systems Conference*, pp. 326–331 (2006)
18. Breiman, L.: Random forests. *Mach. Learn.* **45**(1), 5–32 (2001). <https://doi.org/10.1023/A:1010933404324>
19. Liu, Y., Wang, L., Wang, X.: Random forest algorithm for traffic flow prediction. *IEEE Access* **6**, 40771–40780 (2018). <https://doi.org/10.1109/ACCESS.2018.2852829>
20. Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., Wang, Y.: Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transport. Res. Part C: Emerg. Technol.* **54**, 187–197 (2015). <https://doi.org/10.1016/j.trc.2015.03.014>
21. Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.: Traffic flow prediction with big data: a deep learning approach. *IEEE Trans. Intell. Transport. Syst.* **16**(2), 865–873 (2015). <https://doi.org/10.1109/TITS.2014.2345663>
22. Wu, Z., Pan, S., Long, G., Jiang, J., Chang, X., Zhang, C.: Graph WaveNet for deep spatial-temporal graph modeling. In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1907–1913 (2020)
23. Xu, L., Chen, Z., Li, K.: Hybrid LSTM-ARIMA model for short-term urban traffic prediction. *J. Adv. Transport.* **2019**, 1–10 (2019). <https://doi.org/10.1155/2019/9872506>
24. Vaswani, A., et al.: Attention is all you need. *Adv. Neural Inf. Process. Syst. (NeurIPS)* **30**, 5998–6008 (2017)
25. Sun, X., Wang, Y., Liu, P.: Attention-based deep learning models for real-time traffic prediction. *IEEE Trans. Neural Netw. Learn. Syst.* **32**(8), 3245–3256 (2021). <https://doi.org/10.1109/TNNLS.2021.3058487>
26. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794 (2016). <https://doi.org/10.1145/2939672.2939785>
27. de Dios Ortúzar, J., Willumsen, L.G.: *Modelling Transport*. John Wiley & Sons, Hoboken (2024)
28. Keleperas, D.: Exploring the importance of demand forecasting in B2B: Enhancing the overall customer experience and operational efficiency by utilizing forecasting and collaboration at a manufacturing company (2023)
29. Rasouli, S., Timmermans, H.: Uncertainty in travel demand forecasting models: literature review and research agenda. *Transport. Lett.* **4**(1), 55–73 (2012)
30. Roson, R., Small, K.A.: *Environment and Transport in Economic Modelling*, vol. 10. Springer Science & Business Media, Dordrecht (2013). <https://doi.org/10.1007/978-94-015-9109-6>
31. Xiong, J., et al.: Autoregressive Models in Vision: A Survey (2024). arXiv preprint [arXiv:2411.05902](https://arxiv.org/abs/2411.05902)
32. Hsieh, W.W.: Nonlinear multivariate and time series analysis by neural network methods. *Rev. Geophys.* **42**(1), 1–25 (2004). <https://doi.org/10.1029/2002RG000112>



Retrospective Clustering of COVID-19 Mortality Time Series Using Dynamic Time Warping

Murat Razi and Manuel Graña^(✉)

Computational Intelligence Group, University of the Basque Country, UPV/EHU,
San Sebastian, Spain
`manuel.grana@ehu.es`

Abstract. The COVID-19 pandemic has been a shacking experience for the entire world. Retrospective analysis of the data gathered during the pandemic can be used for the preparation for future pandemics. It is now possible to ascertain if the pandemic response has been the same across the world. Detecting differences in responses over time can be useful for preparation for future pandemics by researching on the causes for different responses. In this direction, this paper contributes evidence that the pandemic response, as measured by the death time series of each country, was not the same everywhere. Clusters of countries with similar death time series can be detected, such that countries in different clusters have rather different patterns of deaths. Dynamic Time Warping (DTW) allows the elastic matching of time series. The cost of DTW matching provides a measure of similarity between the time series. Applying Hierarchical Clustering it is possible to find these clusters. Our findings confirm the existence of a robust cluster of western Europe countries previously reported following rather diverse approaches. Furthermore, we examine in detail the different patterns of several representative countries relative to Spain.

1 Introduction

A quick query in PUBMED with the keyword “COVID-19” provides as of the time of writing almost half a million references. This is a raw measure of the publication effort ignited by the pandemic declaration by the World Health Organization in January 2020. In this panorama of publications, the application of Machine Learning and statistical approaches was mostly devoted to the prediction of patient risks for severe disease [3]. The number of cases was the relevant variable in epidemiological studies and predictive models [1, 6]. On the contrary, we think that the number of deaths is a more relevant indicator of the severity of the pandemic and the pandemic response of each country, because coding a death as COVID-19 has a single source of confusion which is if the death was due to COVID-19 or it has been incidentally diagnosed but it was not the main cause of death [4]. Although the differences in coding criteria may influence the

final death toll attributed to the pandemic, we think that any retrospective analysis should be grounded on actual measurements on the population, contrary to estimates derived from regression models on the basis of subjective reasoning [2]. Specifically, these works [2] assume that the so-called underdeveloped countries do not know how to count deaths, so they extrapolate arbitrary regression models which, of course, provide much higher death tolls for some countries.

Recent works [5, 8–10] are aimed to the retrospective analysis of the collected data. Specifically, these works study the clustering of the COVID-19 death time series by different means and approaches. One work [5] applies Latent Dirichlet Allocation (LDA) in order to decompose the mortality time series into time patterns (topic) of mortality evolution finding that there are some latent geopolitical aggregation of countries with the same dominant topic. Another work [10] carries out a comprehensive exploration of feature extraction and clustering of the mortality time series, finding that some clusters of countries appear recurrently as the results of different approaches, specifically the western Europe countries together with the United States are a robust cluster of mortality responses.

In this paper, first we carry out a selection of countries under study on the basis of the accumulated number of deaths, restricting the study to the countries with a death toll higher than a defined percentile of the accumulated number of deaths. This selection avoids computational artifacts introduced by small size countries. Then, we compute the pairwise distance induced by the Dynamic Time Warp (DTW) between selected countries, which is used for the Hierarchical Clustering of the countries. We discuss the clustering findings, and some specific relations between the mortality patterns of some countries relative to that of Spain. This discussion illustrates some aspects of the evolution of the pandemic mortality. This work converges with previous ones [5, 10] in the detection of a specific cluster of western Europe countries with strong geopolitical ties, and similar socio-cultural response to the pandemic. Also, this work provides additional evidence that the pandemic has been a rather spatially heterogeneous event, contrary to the assumption of an homogeneous response due to a single cause associated to a specific pathogen. In other words, cultural and geopolitical factors appear to have played some significant role in the pandemic response.

2 Materials and Methods

2.1 Our World in Data

“Our world in data” (OWID) is a web site devoted to the publication of relevant data for the pressing geopolitical and economic issues. It is hosted by the University of Oxford. OWID has been publishing relevant data for the follow up of COVID-19 pandemic¹, aggregating it from several sources. We have used “new_deaths_smoothed” data for the analysis. The OWID site was capturing mortality time series from the John Hopkins University COVID-19 site

¹ <https://ourworldindata.org/explorers/coronavirus-data-explorer>.

until march 2023, when the latter stopped updating data after the WHO published the end of the sanitary emergency. After this date most countries stopped reporting. For this work, we have taken the dataset as published on the first day of 2023, which we think covers the most significant events of the pandemic evolution. We consider the seven day moving average of the population normalized mortality time series as published in the OWID site. The raw mortality time series has some artifacts such as zero deaths in weekends, or even some negative reports due to corrections made by the local agencies, hence the need for the time series smoothing.

2.2 Mathematical Methods

First we select a subset of countries in order to avoid the effect of small countries, such as Andorra and small islands. We concentrate on the countries that have the highest death toll. Thus we select countries that have an accumulated number of deaths over the 80% percentile of the distribution of accumulated deaths. That way, countries with small death toll, such as Australia or New Zealand are ignored, despite their protagonism in some of the political decision and effects during the pandemic time.

We have computed the dynamic time warping (DTW) [7, 11] among the smoothed death signals of the selected countries. The DTW is a dynamic programming algorithm that looks for the optimal alignment of two signals. To this end, DTW first computes the matrix of all distances between pairs of values of the signals, and then searches the path of minimal cost traversing this matrix from the origin (given by the initial time points) to the final point (given by the last time points of the signals). Formally, given two time series $X = \{x_1, \dots, x_N\}$ and $Y = \{y_1, \dots, y_N\}$, with $x_i, y_j \in \mathbb{R}^n$, the cost matrix is defined as follows:

$$C_{X,Y}(i, j) = \|x_i - y_j\|, \quad (1)$$

where $\|\cdot\|$ denotes the Euclidean distance. The DTW finds the path that minimizes the cost of alignment between X and Y by traversing the cost matrix starting from $C_{X,Y}(1, 1)$ and ending in $C_{X,Y}(N, N)$. The algorithm performs a greedy search, where at each step finds the direction of minimal cost for the next matching forward, never going back. The DTW distance can be computed by the following recursive equation:

$$d_{X,Y}(i, j) = C_{X,Y}(i, j) + \min \{d_{X,Y}(i-1, j-1), d_{X,Y}(i-1, j), d_{X,Y}(i, j-1)\}$$

Search has been limited to a specific time window, in order to avoid excessive distortion of the signals. Hence, the effective area for the search of matching points is a diagonal area of the cost matrix. Signals can have different lengths, but in our case the death time series have the same length. The distance used to build up the cost matrix can be the Euclidean distance, most often, or other, such as the Kullback-Leibler, or the Manhattan distance. The values of the

signals can be scalar or vectorial. The search is a greedy one, so there is no guarantee that the algorithm finds the global matching optima.

A salient advantage of DTW analysis is that the signal morphing provides insights on the potential time dependencies between signals. In other words, the examination of the displacement of the signals required for the matching can provide some information about the temporal relation between signals. In the case of COVID-19 mortality time series, we can assess the synchronization of the signals in some periods of time, or we can discuss about the potential diffusion from one country to another if the alignment is always in one direction, i.e. the peaks in one country precede those in another country.

In order to aggregate countries with similar response to the pandemic, we carry out Hierarchical Clustering (applying the Ward method) of the DTW distance computed for each pair of selected countries. Ward's method minimizes the total within-cluster variance. To implement this method, at each step find the pair of clusters that leads to minimum increase in total within-cluster variance after merging. This increase is a weighted squared distance between cluster centers. At the initial step, all clusters are singletons (clusters containing a single point). To apply a recursive algorithm under this objective function, the initial distance between individual objects must be (proportional to) squared Euclidean distance. In our case, the DTW distance behaves like the squared Euclidean distance, although there is no proportionality. The initial cluster distances in Ward's minimum variance method are therefore defined to be the DTW distance between points.

In order to select the number of clusters we explore maximizing the value of the silhouette coefficient, but also maximizing the entropy of the cluster sizes, that is avoiding the trivial case of one big cluster and several marginally small clusters. All the computations have been carried out in Matlab, using standard functions "dtw", "linkages", "cluster", and others (Fig. 10).

3 Results

The selection of the upper 80% percentile countries in the distribution of accumulated deaths normalized by the population size, i.e. those with the highest relative death toll, leaves us with 40 countries, most of them in the developed western regions of the world. The DTW distance is computed setting a window of 60 days in order to avoid strong distortion of the warped signals. The hierarchical clustering with maximal silhouette coefficient and entropy of the cluster sizes ended up with an optimal number of clusters equal to 8. Figure 1 shows a visualization of the clusters found over the world map. Non-colored countries have been excluded of the clustering process. Some clusters are unitary, like those containing India, United States, or Russia. Other clusters show some geopolitical aggregation. The salient case is that of Spain, France, Italy, Germany, and United Kingdom (orange colored). This cluster has been found systematically by other means, like topic analysis [5], and clustering of the time series by conventional algorithms with or without feature extraction [10]. Another cluster (cyan

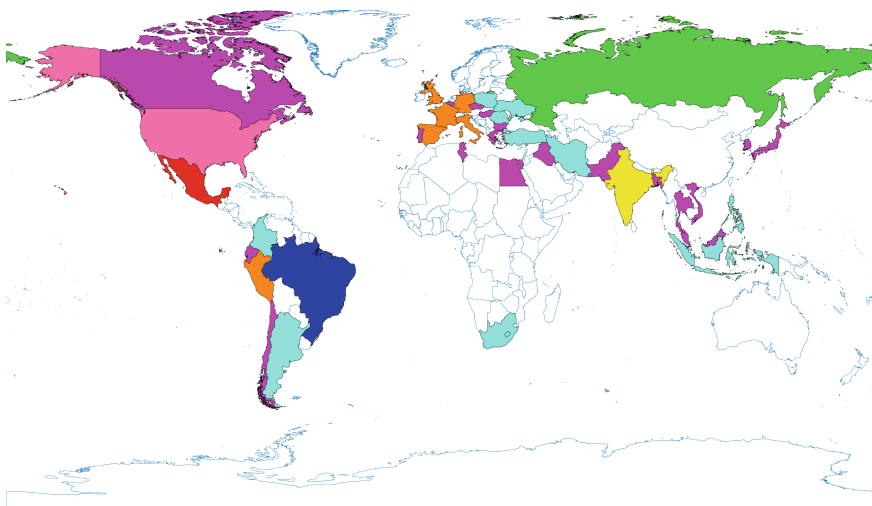


Fig. 1. World map of the detected country clusters by hierarchical clustering of DTW distances among mortality time series

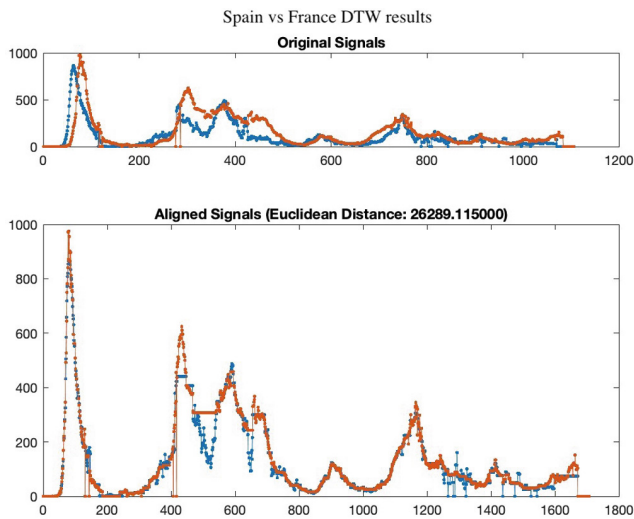


Fig. 2. Spain vs. France results of the DTW correction

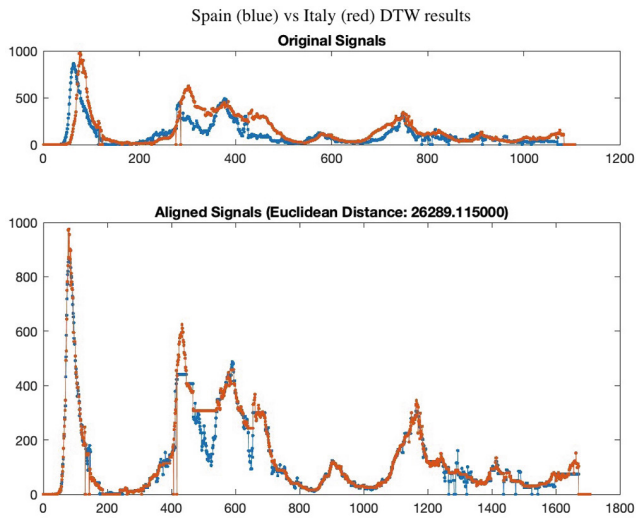


Fig. 3. Spain vs. Italy results of the DTW correction

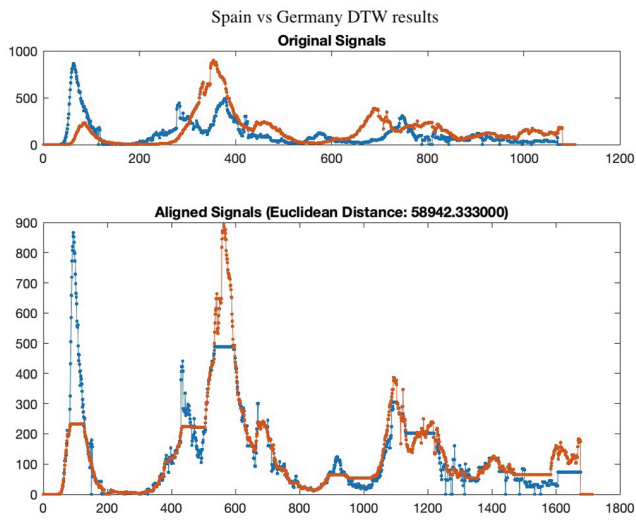


Fig. 4. Spain vs. Germany results of the DTW correction

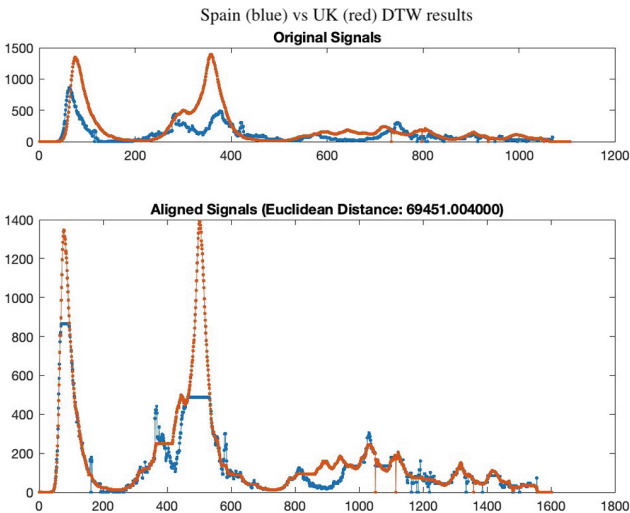


Fig. 5. Spain vs. UK results of the DTW correction

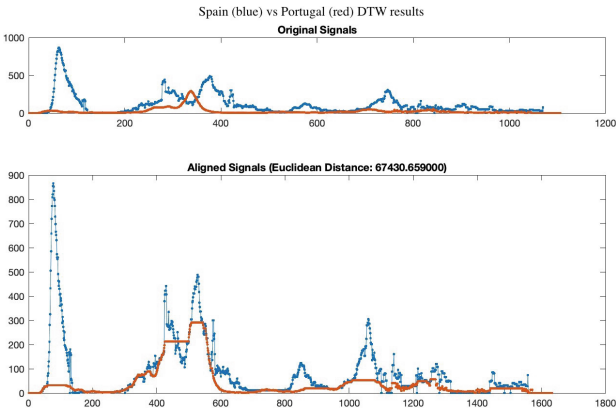


Fig. 6. Spain vs. Portugal results of the DTW correction

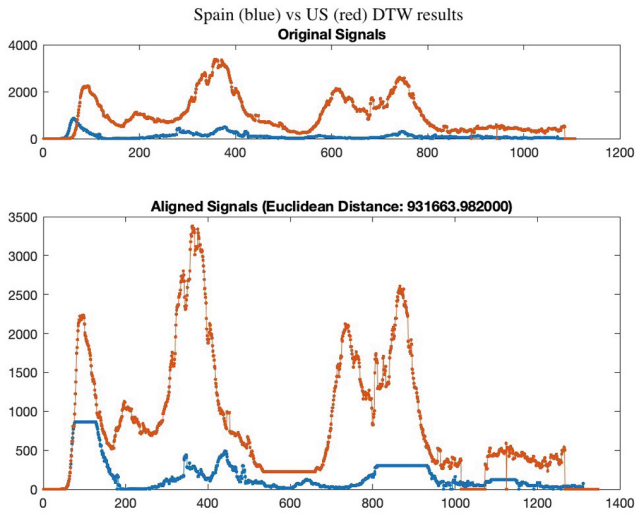


Fig. 7. Spain vs. US results of the DTW correction

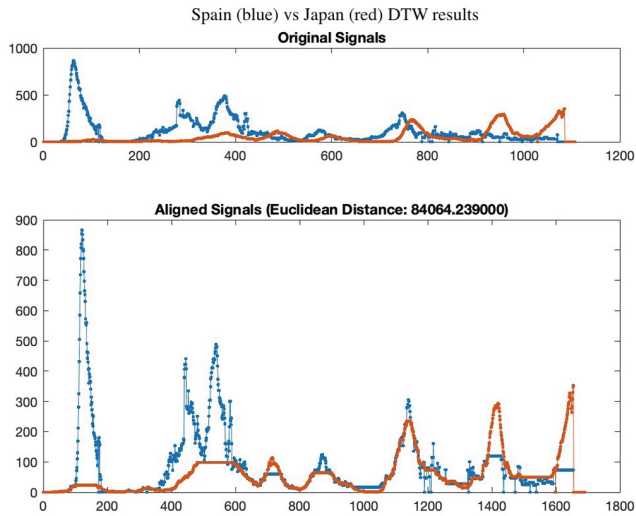


Fig. 8. Spain vs. Japan results of the DTW correction

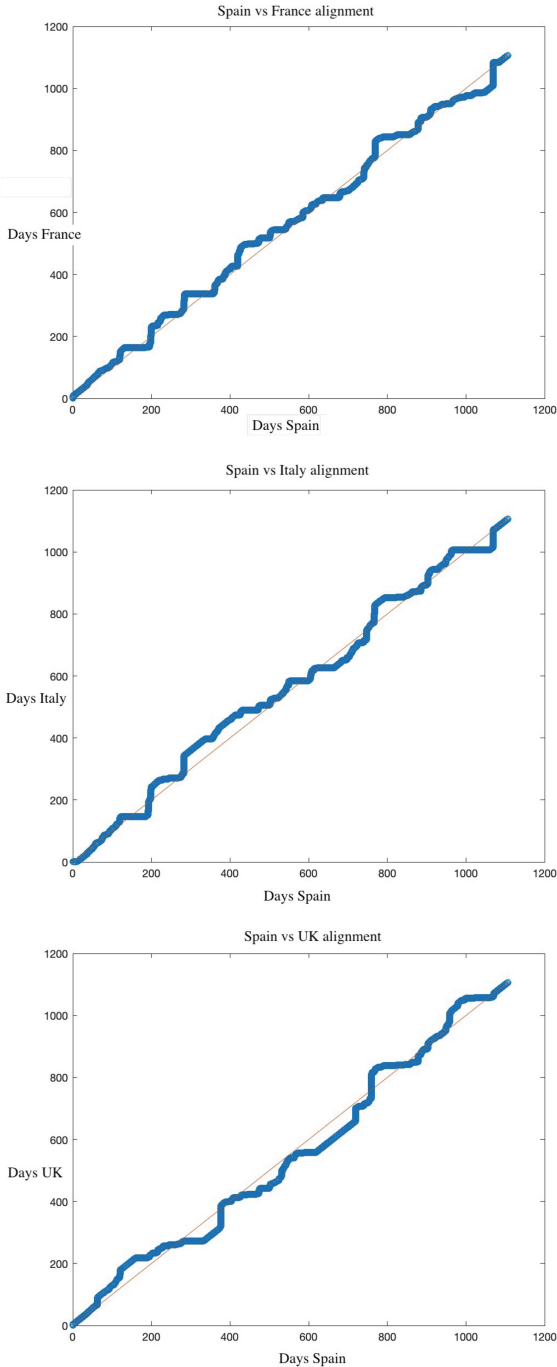


Fig. 9. Alignment of Spain vs. France, Italy, and United Kingdom.

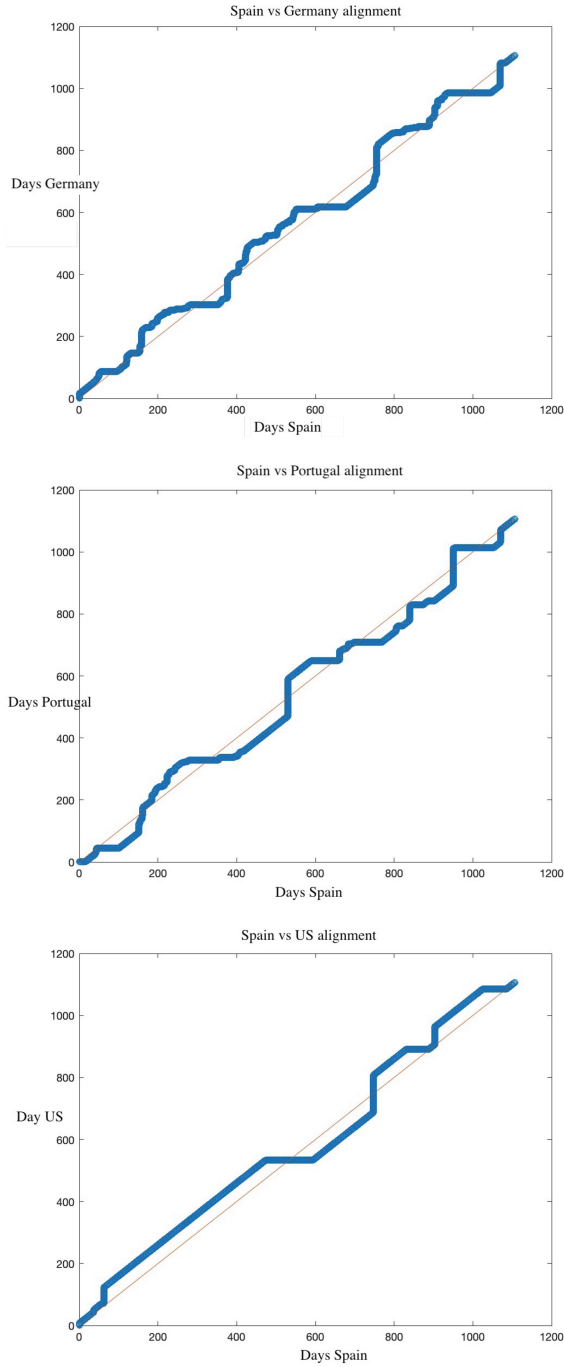


Fig. 10. Alignment of Spain vs. Germany, Portugal, and USA.

colored) includes geographically disperse countries like Canada, Portugal, Chile, or Japan. Hence, disease diffusion processes are very unlikely inside this cluster of countries, which can be characterized by a moderate death toll. Finally, another geographically disperse cluster (colored light blue) appears to have a high concentration of the middle east and eastern Europe countries like Ukraine, Romania, Poland, and Turkey, including remote countries like Argentina, South Africa, and Indonesia.

4 Discussion

One of the most salient characteristics of the COVID-19 death time series is the initial response, i.e. the mortality in the first six months of the pandemic. Some countries show a strong peak of mortality in the spring of the year 2020, while others have little or no mortality response until the autumn of 2020. This characteristic has a definite impact in the formation of the clusters detected by the described procedure. For instance, countries in the western Europe cluster (orange colored in Fig. 1) show an almost synchronized mortality peak in the spring of year 2020, while countries in the eastern Europe cluster (colored light blue) have a very small and smooth response in that period. In light of these observations, it seems necessary to revisit the hypothesis that the COVID-19 pathogen was almost instantaneously traveling the world, with cases appearing simultaneously in Australia and Spain, i.e. at two opposite points of the globe.

In the following we will focus on Spain mortality and its relation to that of other countries that belong in the same cluster or in a different cluster. For countries in the same cluster as Spain, we emphasize the socio-political closeness that may be underlying their strong similarity in terms of DTW distance. For countries with belonging to a cluster that is different from the Spanish cluster, we have two kinds of situations. Some are in the same sociopolitical area as Spain, sharing a lot of cultural elements, like Portugal, but our approach has clustered them into different places. Other countries, like Japan, have rather different cultural background and some sociopolitical divergences. From this detailed comparisons we are seeking to obtain some guide to pose relevant questions on the pandemic measures evolution based on the relation between resulting mortality in different countries.

Figures 2, 3, 4, and 5 show the aligned mortality time series of Spain versus France, Italy, Germany, and United Kingdom, respectively, after computing the DTW correction. Except for Germany, these countries share a large peak in the spring of 2020, with some temporal displacements. We can appreciate that the peak in Spain precedes those in Italy, Germany, and France, but it is almost synchronous with that of the United Kingdom. All these countries have strong mortality after the summer of 2020 until the summer of 2021, with not very different time structure. Mass vaccination in these countries started at the end of the year 2020. However, the COVID-19 mortality did not decrease significantly until the summer of 2021, and did have some rebounds, like in the so-called Omicron wave found during the winter of 2021 and the beginning of the spring of

2022. Figure 9 shows the alignment of the signals, that is, the correspondence of dates in one time series with those in the other. For the mentioned countries, it can be appreciated a quasi-perfect alignment during the first year of the pandemic until the winter of these year, with small deviations above and below the diagonal in the following. In other words the initial synchronization of the mortality time series of these countries seems to weaken after the autumn/winter of 2020, despite the fact that these countries followed very similar synchronized policies.

Figure 6 shows the aligned mortality time series of Spain versus Portugal after DTW. We can appreciate why Portugal belongs to a different cluster despite being the closest spatial neighbour of Spain. For some reason, there was no mortality peak in Portugal in the spring of year 2020. There was no synchronization with Spain, while Spain was perfectly synchronized with the United Kingdom, which is a distant spatial neighbour. The first and only significant mortality peak in Portugal appears in the beginning of year 2020, and, paradoxically, it does match a valley in the Spanish mortality time series. The alignment plot in Fig. 9 shows a bigger oscillation around the diagonal than previous considered countries. Nothing indicates a pattern of diffusion corresponding to an airborne viral pathogen between these spatially close countries.

Figure 7 shows the aligned mortality time series of Spain versus United States after DTW. The United States was the universal reference for guidance in the pandemic response, so it is interesting to examine how the mortality responses were aligned to this reference. The huge difference in the magnitude of the time series should be the main reason for the Spain and United States to be in different clusters. There was some peak synchronization in the spring of year 2020. However, COVID-19 mortality in the United States never came down close to zero until the end of 2021, just before the Omicron wave. It seems that the most influential country for health guidance did suffer the greatest death toll. The alignment between these two countries shown in Fig. 9 is saturated to the 60 days window in many periods. Spain seems to be preceding the United States during most of the pandemic, though it is difficult to appreciate due to the difference in magnitude of the time series.

Finally, Fig. 8 shows the aligned mortality time series of Spain versus Japan after DTW. Again, it is easy to appreciate that Japan did not have a mortality peak in the year 2020, and some small incidence started in 2021. Paradoxically, the stronger mortality peaks in Japan are appearing in the year 2022, when mortality in Spain is disappearing. Therefore, Japan shows a completely different pattern of mortality from Spain.

5 Conclusions

After the end of the pandemic emergency, there are growing works dealing with a retrospective analysis of the collected epidemiological information. There is growing evidence from these works that socio-cultural factors did influence the death toll of the COVID-19 pandemic, maybe more than the designated pathogen. The paradigm of “one pathogen, one disease” seems to be broken in the COVID-19

pandemic. The present paper does add more evident to this departure of the classical medical paradigm. Using DTW induced distance between time series, we found clusters of countries with rather different evolutions of the COVID-19 mortality. Some clusters may support some kind of synchronization or propagation of the disease, while others provide counterfactual evidence. For instance the two spatial neighbours of Spain, Portugal and France, belong to different clusters and show radically different time patterns. The results in this paper invite to look more deeply into the socio-cultural roots of the pandemic response, in order to learn useful lessons for the future.

In the future work, the application of the DTW to study the relation between the different variables describing the pandemic evolution, such as the number of tests, vaccination doses, cases, hospitalizations, and so on. The study of the alignment between these variables may give some insights on their relative influence in a dynamic way. Also, the application of other clustering approaches may provide further robustness to the conclusions of this paper.

References

1. Abbasimehr, H., Paki, R.: Prediction of COVID-19 confirmed cases combining deep learning methods and Bayesian optimization. *Chaos, Solitons Fractals* **142**, 110511 (2021)
2. Wang, H., et al.: Estimating excess mortality due to the COVID-19 pandemic: a systematic analysis of COVID-19-related mortality, 2020–21. *Lancet* **399**(10334), 1513–1536 (2022)
3. Marin, B.G., et al.: Predictors of COVID-19 severity: a literature review. *Rev. Med. Virol.* **31**(1), 1–10 (2021)
4. Goldstein, E.: Mortality associated with omicron and influenza infections in France before and during the COVID-19 pandemic. *Epidemiol. Infect.* **151**, e148 (2023)
5. Graña, M., Badiola-Zabala, G., Cano-Escalera, G.: Topic detection in COVID-19 mortality time series. In: Ferrández Vicente, J.M., Val Calvo, M., Adeli, H. (eds.), *Bioinspired Systems for Translational Applications: From Robotics to Social Engineering*, pp. 358–367. Springer Nature Switzerland, Cham (2024). https://doi.org/10.1007/978-3-031-61137-7_33
6. Iranzo, V., Pérez-González, S.: Epidemiological models and COVID-19: a comparative view. *Hist. Philos. Life Sci.* **43**(3), 104 (2021)
7. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. *Knowl. Inf. Syst.* **7**(3), 358–386 (2005)
8. Landmesser, J.: The use of the dynamic time warping (DTW) method to describe the COVID-19 dynamics in Poland. *Oeconomia Copernicana* **12**(3), 539–556 (2021)
9. Luo, Z., Zhang, L., Liu, N., Ye, W.: Time series clustering of COVID-19 pandemic-related data. *Data Sci. Manage.* **6**(2), 79–87 (2023)
10. Razi, M., Graña, M.: Clustering COVID-19 mortality time series. In: Ferrández Vicente, J.M., Val Calvo, M., Adeli, H., (eds.) *Bioinspired Systems for Translational Applications: From Robotics to Social Engineering*, pp. 410–419. Springer Nature Switzerland, Cham (2024). https://doi.org/10.1007/978-3-031-61137-7_38
11. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process.* **26**(1), 43–49 (1978)

Author Index

A

Abdelrahman, Yahia II-31
Akratos, Chsristos II-123
Alharbi, Abdulrahman I-267, II-94
Ali, Asfak I-128
Aljurbua, Rafaa I-267, II-94, II-165
Almenar, David I-31
Alonso, Serafin II-81
Alvi, Faisal II-43
Anand, Amida I-59
Andersen, Per-Arne I-209
Andjelkovic, Jovan II-94
Apicella, Giovanna II-224
Arya, Arti I-87
Azpiroz, Izar II-137, II-181

B

Baembitov, Rashid II-165
Bagni, Felipe II-210
Banerjee, Jotiraditya I-128
Begum, Nazmin I-87
Belmajdoub, Hanae I-100
Belo, David I-154
Bhatagunaki, Raghavendra S. I-181
Biagiotti, Alessandro II-151
Boancă, Sara I-112
Boracchi, Giacomo II-109, II-210
Botzoris, George N. II-269
Botzoris, Nikiforos II-195

C

Cervellera, Cristiano II-210
Chikhale, Anshuli I-240
Chitnis, Srikrishna R. I-181

D

da Costa Paulo, Breno II-181
Díaz, Ignacio II-81
Dimitriou, Nikolaos II-3
Domínguez, Manuel I-141, II-81
Döring, Nico I-254

Drocourt, Cyril I-17

Durand, David I-17

Dutta, Akash I-128

E

Ebadulla, Farhaan I-87
El Belghiti, Younes I-100
El Omari, Mouad I-100
Engelmann, Bastian II-238

F

Ferreira, João I-154
Florez Tapia, Arantzazu II-137
Florez-Tapia, Ane M. II-137
Fuertes, Juan J. I-141, II-81

G

Gallicchio, Claudio I-141
Garção, Telma I-154
Garrote, Luis II-255
Gaurav, B. V. I-87
Giese, Peter I-59
Gil, Amaia II-137
González-Herbón, Raúl II-81
Graña, Manuel II-285
Grau-Haro, Jordi I-31
Gregory, Jon I-45
Gupta, Shelly I-267, II-94

H

Hai, Ameen Abdel I-267, II-94
Hao, Hong II-68
Harhad, Ahlem I-17
Hewavitharana, Jayani I-59
Hossain, Saddam I-195

I

Iliadis, Lazaros I-225, II-123, II-195, II-255,
II-269
İşlak, Ümit I-168

Islam, Md. Rabiul I-73

J

Jiao, Lei I-209

K

Kakani, Divyam I-240
Karampinis, Ioannis I-225
Keane, Andy I-45
Kezunovic, Mladen II-165
Khan, Faryal II-43
Khatri, Radhika II-43
Kiourt, Nichat II-123
Klehr, Lukas II-238
Korgesaar, Mihkel I-168
Kostinakis, Konstantinos I-225
Kruthik, K. I-87
Kurban, Hasan I-168

L

Lambert, Marian I-254
Landa del Barrio, Iker II-181
La Rocca, Michele II-224
Li, Ling II-68
Li, Qilin II-68
Logofătu, Doina I-195
Logofătu, Doina I-73
Lopez-Ballester, Jesus I-31
Lystbæk, Michael Sahl II-17
Lystbæk, Ulrik Sahl II-17

M

Macció, Danilo II-210
Maharjan, Reshma I-209
Maiza, Mikel II-181
Malchiodi, Dario II-151
Mallesh, Shreyas I-181
Manoj, H. I-87
Margheritti, Riccardo II-109
Mariotto, Samuele II-151
Minaoui, Khalid I-100
Mitsiaki, Andreana II-3
Morán, Antonio I-141
Moretti Ierardi, Carolina I-59
Morfidis, Konstantinos I-225
Muntean, Cristina I-3

N

Naranjo-Alcazar, Javier I-31

Nunez, Marco I-45

O

Obradovic, Zoran I-267, II-54, II-94, II-165
Odriozola, Juan II-181
Olaizola, Igor García II-137
Otudi, Hussain I-267, II-94

P

Papaleonidas, Antonios II-195, II-255, II-269
Papaleonidas, Antonis II-123
Pasta, Qasim II-43
Pathak, Pramod I-3
Peretti, Edoardo II-210
Perna, Cira II-224
Petkopoulos, Stamatios II-255
Petridis, Christos II-54, II-165
Pimenidis, Elias II-31
Pipolo, Carlotta II-109
Prada, Miguel A. II-81
Psathas, Anastasios Panagiotis II-31, II-195, II-255, II-269

Q

Quadrio, Maurizio II-109
Quartulli, Marco II-137

R

Ravindran, Sumedh I-181
Razi, Murat II-285
Regulin, Daniel II-238
Rizwan, Samin II-43
Roca, Lidia I-141
Rodríguez-Ossorio, José Ramón I-141
Romanoff, Jani I-168
Rossi, Lucio II-151

S

Saha, Debam I-128
Saoudi, Samir I-100
Saranovic, Daniel II-165
Sarkar, Ram I-128
Schillaci, Andrea II-109
Schleif, Frank-Michael II-238
Schuster, Thomas I-254
Schwenker, Friedhelm I-128
Shao, Yanda II-68
Sharath, Shylaja S. I-181

Shi, Zifan [II-68](#)
Shrivastava, Abhudaya [II-54](#)
Sibillo, Marilena [II-224](#)
Skopelitis, Ioannis [II-269](#)
Solsona, Roger [II-137](#)
Sousa, Joana [I-154](#)
Spiliotis, Mike [II-255](#)
Steinhofel, Kathleen [I-59](#)
Stuckner, Joshua [I-168](#)
Stynes, Paul [I-3](#)

T

Toal, David [I-45](#)
Trögele, Julius [I-254](#)
Tsita, Christina [II-3](#)
Tzovaras, Dimitrios [II-3](#)

U

Utard, Gil [I-17](#)

V

Vacic, Marijana [II-54](#)
Velásquez, David [II-181](#)
Vidanes, Gerico [I-45](#)

W

Wang, Yixuan [I-3](#)

X

Xu Zhang, Daniel [I-45](#)

Y

Yasin, Sudais [II-43](#)
Yatkin, Muhammed Adil [I-168](#)

Z

Zotova, Elena [II-137](#)
Zuccarello, Pedro [I-31](#)