Mathematical and Computational Applications

Special Issue Reprint

New Trends in Computational Intelligence and Applications 2025

Edited by
Nancy Pérez-Castro and Aldo Márquez-Grajales

mdpi.com/journal/mca

MDPI

New Trends in Computational Intelligence and Applications 2025

New Trends in Computational Intelligence and Applications 2025

Guest Editors

Nancy Pérez-Castro
Aldo Márquez-Grajales

MDPI

Basel • Beijing • Wuhan • Barcelona • Belgrade • Novi Sad • Cluj • Manchester

Guest Editors

Nancy Pérez-Castro
Artificial Intelligence Research Institute
University of Veracruz
Xalapa
Mexico

Aldo Márquez-Grajales
Instituto de Ciencias Básicas e Ingeniería
Universidad Autónoma del Estado de Hidalgo
Hidalgo
Mexico

Editorial Office
MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of the Special Issue, published open access by the journal *Mathematical and Computational Applications* (ISSN 2297-8747), freely accessible at: https://www.mdpi.com/journal/mca/special_issues/EA155JI4OR.

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

Lastname, A.A.; Lastname, B.B. Article Title. *Journal Name* **Year**, *Volume Number*, Page Range.

ISBN 978-3-7258-9131-3 (Hbk)
ISBN 978-3-7258-9132-0 (PDF)
https://doi.org/10.3390/books978-3-7258-9132-0

© 2026 by the authors. Articles in this reprint are Open Access and distributed under the Creative Commons Attribution (CC BY) license. The reprint as a whole is distributed by MDPI under the terms and conditions of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) license (https://creativecommons.org/licenses/by-nc-nd/4.0/).

Contents

About the Editors . . . . . **vii**

Preface . . . . . **ix**

Jeeangh Jennessi Reyes-Montiel, Ericka Janet Rechy-Ramirez and Antonio Marin-Hernandez
Techniques Applied to Autonomous Liquid Pouring: A Scoping Review
Reprinted from: *Math. Comput. Appl.* **2026**, *31*, 30, https://doi.org/10.3390/mca31010030 . . . . **1**

Luis Gabriel Macias Santillán, Jonathan A. Zea, Lorena Isabel Barona López, Ángel Leonardo Valdivieso Caraguay and Marco E. Benalcázar
HGREncoder: Enhancing Real-Time Hand Gesture Recognition with Transformer Encoder—A Comparative Study
Reprinted from: *Math. Comput. Appl.* **2025**, *30*, 101, https://doi.org/10.3390/mca30050101 . . . **32**

Hao Hu, Jinshun Cai and Chenke Xu
A Mathematical Framework for E-Commerce Sales Prediction Using Attention-Enhanced BiLSTM and Bayesian Optimization
Reprinted from: *Math. Comput. Appl.* **2026**, *31*, 17, https://doi.org/10.3390/mca31010017 . . . . **52**

Carlos-Alberto López-Herrera, Héctor-Gabriel Acosta-Mesa, Efrén Mezura-Montes and Jesús-Arnulfo Barradas-Palmeros
Neuroevolution of Liquid State Machine Based on Neural Configurations and Positions
Reprinted from: *Math. Comput. Appl.* **2026**, *31*, 65, https://doi.org/10.3390/mca31020065 . . . . **73**

José Luis Llaguno-Roque, Adriana Laura López-Lobato, Juan Carlos Pérez-Arriaga, Héctor Gabriel Acosta-Mesa, Ángel J. Sánchez-García, Gabriel Gutiérrez-Ospina, et al.
AI-Driven Biopsychosocial Screening for Breast Cancer: Enhancing Risk Prediction via Differential Evolutionary Linear Discriminant Analysis for Feature Extraction
Reprinted from: *Math. Comput. Appl.* **2026**, *31*, 66, https://doi.org/10.3390/mca31030066 . . . . **98**

Emmanuel Isaac Juárez Caballero, Horacio Tapia-McClung and Efrén Mezura Montes
A Comparison Between Heuristic andAutomatic Design in Variational QuantumCircuits for the MaxCut Problem Under Noise Effects
Reprinted from: *Math. Comput. Appl.* **2026**, *31*, 78, https://doi.org/10.3390/mca31030078 . . . . **124**

Adriana-Laura López-Lobato, Héctor-Gabriel Acosta-Mesa, Efrén Mezura-Montes, Jimena-Esther Alba-Jiménez, Amalia-Guadalupe Rodríguez-Gómez, Elia-Nora Aquino-Bolaños and Rosa-Hayde Alfaro-Rodríguez
Evolutionary Linear Discriminant Projection for Sensory Analysis of Tortillas Fortified with Chilacayote Powder
Reprinted from: *Math. Comput. Appl.* **2026**, *31*, 82, https://doi.org/10.3390/mca31030082 . . . . **142**

Verónica Neri-Mendoza, Yulia Ledeneva, Jonathan Rojas-Simón, Muhammad Tayyab Zamir, René Arnulfo García-Hernández and Ángel Hernández-Castañeda
Benchmarking Ethnic Hate Speech Detection in México
Reprinted from: *Math. Comput. Appl.* **2026**, *31*, 88, https://doi.org/10.3390/mca31030088 . . . . **177**

Vicente-Josué Aguilera-Rueda, Nicandro Cruz-Ramírez, Efrén Mezura-Montes and Ricardo Vilalta
Learning Bi-Objective Bayesian Network Structure from Data Using Particle Swarm Optimization
Reprinted from: *Math. Comput. Appl.* **2026**, *31*, 94, https://doi.org/10.3390/mca31030094 . . . . **193**

About the Editors

Nancy Pérez-Castro

Nancy Pérez-Castro is currently a postdoctoral researcher at the Artificial Intelligence Research Institute of the Universidad Veracruzana. From 2018 to 2025, she worked as a Full-time Research Professor at the Universidad del Papaloapan. She has published scientific articles as first author and co-author in indexed journals, as well as outreach texts aimed at bringing artificial intelligence and data science closer to broad audiences. She has served as a reviewer for multiple national and international conferences, international journals, and outreach publications. She is a regular member of the Mexican Academy of Computing A.C. since 2025 and member of the Mexican National Researchers System Level 1. Her research focuses on computational intelligence, metaheuristic optimization, and machine learning, with recent applications in precision agriculture, the environment, population ecology, and green artificial intelligence.

Aldo Márquez-Grajales

Aldo Márquez-Grajales is a full-time professor and researcher in the Computation and Electronics Academic Area of the Autonomous University of Hidalgo State. His research interests include machine learning, bio-inspired computation, time series analysis, and artificial intelligence applied to climate science. He has been a professor in higher education since 2016. From 2021 to 2024, he made two postdoctoral stays at the Artificial Intelligence Research Institute of the University of Veracruz. Moreover, he has several publications in high-impact scientific journals and participates in national and international conferences and workshops. Finally, he is a member of the Mexican National Researchers System Level 1 and a Regular Member of the Mexican Computing Academy.

Preface

Nowadays, the field of computational intelligence has attracted the attention of both emerging and established researchers, who are developing new tools and methods capable of performing tasks that were unthinkable just a few years ago. Furthermore, researchers from other fields have turned to these techniques to solve the research problems they face, opening up possibilities for multidisciplinary work.

Consequently, workshops such as the 7th Workshop on New Trends in Computational Intelligence and Applications (CIAPP) 2025 are organized to present the latest and most innovative findings and applications related to evolutionary computation, machine learning, natural language processing, and quantum computing, among others.

This Special Issue serves as a vehicle for disseminating the work presented at CIAPP 2025 and for presenting, in greater detail, the methods introduced to the international scientific community. Accordingly, it bears the same title, "New Trends in Computational Intelligence and Applications 2025", and its scope and objectives are fully aligned with those of the conference.

This issue comprises nine articles, eight research papers and one state-of-the-art review article. These works present novel methods related to the aforementioned fields, with applications to problems of general interest. The list of papers included in this issue is as follows:

1. **Learning Bi-Objective Bayesian Network Structure from Data Using Particle Swarm Optimization.** This article describes the implementation of MOPSO, a multi-objective optimization algorithm based on Particle Swarm Optimization (PSO), to optimize Bayesian network structure using test datasets.
2. **Benchmarking Ethnic Hate Speech Detection in Mexico.** This article presents a methodology for detecting ethnic hate speech directed at indigenous groups in Mexico.
3. **Evolutionary Linear Discriminant Projection for Sensory Analysis of Tortillas Fortified with Chilacayote Powder.** The authors compared PCA, LDA, PCA+LDA, and the evolutionary method DE-LDAFE (Differential Evolutionary Linear Discriminant Analysis for Feature Extraction and Visualization) to reduce dimensionality and identify sensory profiles associated with fortifying corn tortillas with chilacayote powder.
4. **A Comparison Between Heuristic and Automatic Design in Variational Quantum Circuits for the MaxCut Problem Under Noise Effects.** This study compares two methods for finding efficient variational quantum circuits (VQCs) using a variational quantum algorithm (VQA). These methods were the Quantum Approximate Optimization Algorithm (QAOA) and Quantum Neural Architecture Search (QNAS).
5. **AI-Driven Biopsychosocial Screening for Breast Cancer: Enhancing Risk Prediction via Differential Evolutionary Linear Discriminant Analysis for Feature Extraction.** This article proposes a comprehensive methodology, based on a biopsychosocial model, to predict breast cancer risk. This approach posits that emotional suppression and repression act as key neuroendocrine disruptors and predisposing factors among the Mexican female population.
6. **Neuroevolution of Liquid State Machine Based on Neural Configurations and Positions.** This article presents a neuroevolution algorithm, based on a genetic algorithm, to optimize both the parameters and the architecture of a type of recurrent neural network: Liquid State Machines (LSMs).
7. **A Mathematical Framework for E-Commerce Sales Prediction Using Attention-Enhanced BiLSTM and Bayesian Optimization.** In this study, the authors propose a bidirectional long

short-term memory (BiLSTM) neural network to identify complex patterns related to inventory management and marketing.

8. **HGREncoder: Enhancing Real-Time Hand Gesture Recognition with Transformer Encoder—A Comparative Study.** The authors used a generalizable deep learning model based on transformers and convolutional neural networks (CNNs) for real-time hand gesture recognition.
9. **Techniques Applied to Autonomous Liquid Pouring: A Scoping Review.** This scoping review explores the various techniques employed to control autonomous liquid-dispensing systems.

Each of the articles published in this issue reflects current research trends in the field of computational intelligence. Finally, as Guest Editors of this Special Issue, we wish to thank each author for publishing their valuable work. We also thank the *MCA* editorial team, especially the Editor-in-Chief, for the opportunity to launch this edition and disseminate such innovative work, as well as for the support received throughout the editorial process.

Nancy Pérez-Castro and Aldo Márquez-Grajales
Guest Editors

Mathematical and Computational Applications

MDPI

Systematic Review

Techniques Applied to Autonomous Liquid Pouring: A Scoping Review

Jeeangh Jennessi Reyes-Montiel, Ericka Janet Rechy-Ramirez and Antonio Marin-Hernandez *

Artificial Intelligence Research Institute, Universidad Veracruzana, Campus Sur, Col. Nueva Xalapa, Xalapa C.P. 91097, Mexico; zs23000648@estudiantes.uv.mx (J.J.R.-M.); erechy@uv.mx (E.J.R.-R.)
* Correspondence: anmarin@uv.mx

Abstract

In recent years, autonomous liquid pouring systems have gained more relevance, with applications from daily service tasks to complex industrial operations. While seemingly simple for humans, this task poses major challenges for automated systems, as it requires precise control and adaptation to varying container geometries, liquid properties, and environmental conditions. This review examines the state-of-the-art on liquid pouring through five research questions: (1) What are the characteristics of the liquids used in the experiments? (2) What are the characteristics of the containers used in the experiments and how do they affect the performance of the pouring tasks? (3) What techniques are used to control liquid pouring (i.e., to control the robotic arm or device)? (4) What metrics are used to assess the methods for pouring liquid? (5) What devices are used to measure poured volume? This scoping review follows the Arksey and O'Malley framework, and uses the PRISMA-ScR protocol to filter the articles. A total of 285 studies published between 2018 and 2025 were screened from IEEE Xplore, SpringerLink, ScienceDirect, Web of Science, and EBSCOhost, of which 23 met the inclusion criteria. Results showed that the most widely used methods for autonomous liquid pouring were classical control methods—PID, PD (30.4% of the studies). Conversely, the least widely used methods for autonomous liquid pouring were learning, imitation learning, and probabilistic models (15% of the studies).

Keywords: pouring liquid; autonomous control techniques; robotic manipulation; sensor-based control; liquid characteristics

1. Introduction

Liquids pouring is a very important task in different sectors, such as: chemical engineering, metallurgical processes, and beverage production. Although pouring liquids seems to be a very simple and natural task for humans, this action involves multiple challenges for autonomous systems, including pouring several types of liquids, using various container shapes, and achieving specific volume targets.

The development of automated systems to pour liquids accurately is making significant changes in industry and services. For example, in beverage preparation, such systems improve efficiency and reduce manual effort [1–3]. In industrial contexts such as chemical synthesis and metal casting, advanced robotic pouring could increase precision, safety, and process reliability [4–10].

The challenge of pouring liquids involves a variety of physical factors that affect the accuracy and stability of the process, such as the viscosity, density, and transparency of the liquids. These properties vary significantly according to the type of liquid [11,12].

In terms of the liquid's viscosity, the control strategy for pouring the liquid should analyze whether the liquid has high-viscosity (e.g., syrup, yogurt, and honey) or low-viscosity (e.g., water, milk, and juice) to adjust the velocity—to perform the pouring. Consequently, excessive buildup in the container or risk of spills might be preventable [13]. Furthermore, liquid transparency introduces challenges for vision-based approaches. Specifically, detecting flow or measuring fill levels requires an accurate real-time computation of the pouring angle and the liquid surface [14].

The pouring of liquids requires a set of actions performed with precision to ensure a successful and safe execution. Although humans often perform this task intuitively [15], for machines and robots, each step must be carefully specified to ensure proper coordination and avoid spills or accidents [11]. Finally, the pouring control process requires the incorporation of sensors to provide feedback.

Research on liquid pouring has incorporated a variety of sensing modalities (e.g., vision, audio, and haptics) to implement efficient solutions for robotic pouring. Additionally, studies have tested their solutions for robotic pouring in real-world scenarios using opaque containers, as well as variations in liquid color, lighting conditions, and camera positioning [16,17]. Audio modalities have also been used to estimate fill levels by analyzing sounds produced when liquid falls onto the container, offering a complementary strategy in scenarios where visual feedback is limited [18,19]. In the case of haptics, this modality can allow the robot to modulate pouring force and speed through tactile feedback, increasing accuracy when other sensory inputs might be less accurate [20,21].

It is important to note that traditional control approaches such as PID [22] and PD [23] are also used for controlling the pouring process. Moreover, computational fluid modeling has been employed to simulate how liquids behave under diverse conditions [24]. Container modeling [25,26], meanwhile, emphasizes the role of geometry and physical properties in optimizing pouring accuracy and minimizing spillage. Together, these complementary techniques contribute to the development of pouring systems capable of performing across diverse environments and tasks.

To select works related to liquid pouring techniques, the following three keywords were defined: pouring, liquid, and control. Those keywords are related to: the action, the element, and the way the task is carried out, respectively. Pouring is the process of transferring liquid from one container to another. Liquid refers to intrinsic properties, such as viscosity, density, and transparency, which influence the flow behavior. And finally, control encompasses the algorithms, sensory feedback, and adaptive strategies that regulate the process.

To include the latest and more relevant advances in techniques and technologies applied to pouring liquid, the period from 2018 to 2025 was selected. Throughout the search, no previous reviews were found that focused specifically on techniques used to pour liquids. Although there are reviews associated with object manipulation, these often focus on the grasping [27–34], transporting [35], and positioning of solid objects [36,37] (see Table 1), thus leaving a significant gap in the literature about the challenges and approaches of pouring liquid. The gap highlights the need for a review that analyzes and compares existing techniques, identifies current limitations, and proposes future lines for research in this field.

Despite recent advances in robotic manipulation, liquid pouring remains a complex task due to the nonlinear behavior of fluids and the interplay between perception and control. As robots are increasingly deployed in real-world applications, such as service robotics and industrial environments, interest in autonomous liquid pouring continues to grow. Therefore, it is essential to systematically map the different methodologies, clarify how control and perception are integrated, and identify critical gaps in real-world implementation.

https://doi.org/10.3390/mca31010030

Table 1. Summary of reviews on methods for liquid pouring.

Reviews on Methods for Liquid Pouring					
Review	**Year**	**Type**	**Keywords Used for the Search**	**Aim**	**Comment**
Khan et al. [27]	2023	Literature	"degrees-of-freedom", "end-effector", "kinematics", "manipulator", "pick-and-place", "robotic arm", "trajectory planning".	It focuses on robotic arm manipulation and mobile navigation techniques.	Does not address liquid manipulation or pouring tasks.
Meattini et al. [28]	2023	Literature	"human-hand", "robot-hand", "motion-mapping".	It proposes a classification of human-to-robot hand motion-mapping methods for teleoperation and learning by demonstration.	Does not address liquid manipulation or pouring tasks.
Freschi et al. [29]	2012	Literature	"da Vinci", "surgical robotics", "laparoscopy".	It describes a technical analysis of the da Vinci surgical telemanipulator.	Does not address liquid manipulation or pouring tasks.
Gentile et al. [30]	2023	Literature	"prostheses", "prosthetic hand"; "control strategy", "human hand", "human brain".	It addresses the neuroscientific and engineering principles underlying hand and grasp control.	Does not address liquid manipulation or pouring.
Yamanobe et al. [31]	2017	Survey	"affordance", "grasping", "manipulation."	This study examines affordance-based approaches for robotic object recognition, grasping, and manipulation.	Does not address liquid manipulation or pouring.
Elguea-Aguinaco et al. [32]	2023	Scoping	"reinforcement learning", "contact-rich manipulation", "industrial manipulators", "rigid object manipulation", "deformable object manipulation".	This work analyzes recent advances of reinforcement learning applied to contact-rich robotic manipulation tasks, including rigid and deformable objects.	Does not address liquid manipulation or pouring.
Paulius and Sun [33]	2019	Survey	—	It revises how knowledge is represented, acquired, and reused in service robotics, focusing on its role in perception, learning, and manipulation tasks.	Does not address liquid manipulation or pouring.
Nahavandi et al. [34]	2024	Survey	"machine-learning", "deep-learning", "reinforcement-learning", "manipulator".	It presents recent trends of machine learning methods applied to real-world robotic manipulation tasks.	Does not address liquid manipulation or pouring.
Nate et al. [35]	2024	Literature	—	It proposes an origami-based robotic gripper capable of transporting solids together with liquids using a deformable cylindrical structure.	It focuses on grasping and transporting liquids rather than controlled liquid pouring; pouring is discussed only as a related challenge.
Nguyen and La [36]	2019	Literature	"deep-reinforcement-learning", "robot-manipulation", "reinforcement-learning".	It analyzes the state-of-the-art of reinforcement learning methods applied to robotic manipulation.	It focuses on general robotic manipulation learned from pixels.
Paulius et al. [37]	2016	Literature	—	It proposes the Functional Object-Oriented Network (FOON) as a knowledge representation for learning and generating robotic manipulation actions.	It focuses on general manipulation learning; liquid pouring is not considered a specific manipulation task.

https://doi.org/10.3390/mca31010030

Contribution

The main contribution of this review is the identification and synthesis of methods and techniques employed for autonomous liquid pouring. Furthermore, it considers complementary aspects such as: liquids' characteristics, devices used for volume measurement, containers' features, and the metrics applied to evaluate the pouring performance. A further relevant aspect is the variety of container types used across studies, including cups, mugs, bottles, bowls, and industrial ladle robots, as container geometry directly impacts on pouring dynamics and control design. Analyzing these elements is crucial for identifying research gaps and ensuring comparability across studies. The insights obtained provide information for advancing robotic pouring systems. Moreover, this review helps establish standardized benchmarks and supports the design of reliable and generalizable solutions.

The areas identified for future research are diverse. It is crucial to develop algorithms that improve the ability of robotic systems to detect and adjust to changes in the liquid properties. Moreover, it is necessary to have innovation in dynamic control algorithms that can manage with high precision the pouring angle and speed in response to real-time conditions. In addition, applications of robotic pouring to new environments, such as automated plant care or handling of dangerous substances, will expand the scope of robotic applications.

2. Methodology

This review follows the methodological framework proposed by Arksey and O'Malley [38], consisting of the following five stages: (1) identification of the research question; (2) identification of relevant studies; (3) selection of studies; (4) plotting of data; and (5) collation, summarizing, and communication of results. Being a structured approach, it improves the transparency and reproducibility of the review process, providing a mapping of the literature related to the pouring liquid control process.

The initial search was conducted from January 2018 to April 2025. An update was performed in August 2025 to include the most recent studies. No additional papers were identified during the update.

2.1. Protocol and Registration

This scoping review was conducted in accordance with the PRISMA-ScR guidelines [39].

2.2. Identification of Research Questions

This work was conducted to map the research done in this field and identify any existing gaps in knowledge. For that purpose, the following research questions were formulated: (1) What are the liquid characteristics considered in the experiments? (2) What are the container characteristics considered in the experiments, and how do their properties affect the performance of the pouring tasks? (3) What techniques are used to control liquid pouring (i.e., to control the robotic arm or device)? (4) What metrics have been used to assess the methods for pouring liquid? (5) What devices are used to measure the volume poured?

Those questions are fundamental, as they address key issues of pouring liquid control. The first question, focused on the characteristics of liquids used in the experiments, provides an insight into how the physical properties of different liquids, such as viscosity, density and transparency, influence the results of pouring tasks. The second question addresses the characteristics of the container used in the experiments, examining how their geometry and material properties influence the pouring tasks. The third question examines the techniques employed to control liquid pouring, enabling the comparison and evaluation of

https://doi.org/10.3390/mca31010030

methodologies in terms of their adaptation and robustness in different scenarios. The fourth question explores the accuracy of various pouring methods, enabling the identification of the most effective approaches for pouring control. Finally, the fifth question, regarding the methods for measuring the volume poured, is fundamental to understanding how the results are measured and evaluated.

2.3. Identification of Relevant Works

The quest for relevant studies was conducted using the following keywords: pouring, liquid, control. Five major academic databases were used, namely IEEE, SpringerLink, Web of Science (WoS), ScienceDirect and EBCOhost. These databases were selected due to a broad coverage of engineering, technology and applied sciences literature, which ensures access to a wide range of studies relevant to the control of pouring liquid processes.

2.4. Study Selection

2.4.1. Inclusion Criteria

The eligibility criteria were designed to focus on studies associated with the objectives of controlling liquid pouring processes. Specifically, articles were included if they satisfied the following criteria: (1) use of an open-mouth container instead of dispensers—the study must involve transferring liquid from an open-top source container (e.g., beakers, bottles, or cups), in which the flow is dependent on the tilting angle; (2) consideration of geometric and dynamic variables of the container to model the pouring process; (3) written in English; and (4) published between 2018 and 2025.

2.4.2. Exclusion Criteria

Following the application of the inclusion criteria, the final set of studies was refined by excluding those that: (1) did not involve a rigid container, (2) focused primarily on fluid modeling and not in the pouring process, and (3) were based on teleoperation to control pouring.

For example, study [35] was excluded because it did not focus mainly on the autonomous control of the pouring phase, as defined in Research Question 3. Instead, it focused on the design of a deformable robotic gripper for simultaneously transporting liquids and solids.

2.5. Chart the Data

There are three key items involved in the pouring liquid task, i.e., (i) the liquid, (ii) the container, and (iii) the robotic arm or device control. The latter specifically relates to the method used for pouring liquid. Based on these aspects, this scoping review aims to answer the following five questions:

- RQ1: What are the liquid characteristics considered in the experiments?
- RQ2: What are the container characteristics considered in the experiments and how do they affect the performance of the pouring tasks?
- RQ3: What techniques are used to control liquid pouring (i.e., to control the robotic arm or device)?
- RQ4: What metrics have been used to assess the methods for pouring liquid?
- RQ5: What devices are used to measure the volume poured?

Parameters proposed to evaluate each of the research questions are given below.

2.5.1. RQ1: What Are the Liquid Characteristics Considered in the Experiments?

As noted in [14,40], previous research has emphasized the importance of considering liquid properties when selecting appropriate pouring strategies.

https://doi.org/10.3390/mca31010030

- Density;
- Color;
- Transparency;
- Viscosity.

2.5.2. RQ2: What Are the Container Characteristics Considered in the Experiments and How Do They Affect the Performance of the Pouring Tasks?

Containers have a central role in the pouring process, as their geometry and material properties directly influence the liquid's flow and task performance. As described in [11,25,41,42], key parameters include shape, height, spout diameter, and transparency. These factors, essential for both modeling and control, determine how liquids accumulate, flow, and are perceived by vision-systems.

- Container shape;
- Container transparency;
- Opening diameter of the container's spout;
- Container height.

2.5.3. RQ3: What Techniques Are Used to Control Liquid Pouring (i.e., to Control the Robotic Arm or Device)?

Successful pouring depends on control strategies that guide the robotic arm and regulate the liquid flow. As highlighted in [43,44], these techniques range from classical controllers to adaptive and learning-based approaches, each with its own advantages and limitations. Based on this, the following parameters were selected to analyze control strategies:

- Algorithm type.
- System robustness: Is the approach able to dynamically adjust the pouring process in response to unexpected changes?
- Experimental adaptability: Was the methodology tested under different initial volumes or filling levels? (Yes/No).
- Computational requirements: What type of hardware or software is necessary to implement the methodology?

2.5.4. RQ4: What Metrics Are Used to Assess the Methods for Pouring Liquid?

As described by [45,46], standardized measures are essential to evaluate and compare the accuracy, efficiency, and robustness of liquid pouring systems. In line with this view, the following parameters were selected to address RQ4.

- Average pouring error;
- Success rate: The percentage of attempts in which the spilled volume falls within a margin of error of $\pm 5\%$;
- Spill: Volume of liquid spilled out of the target container during the process (in milliliters or as a percentage of the total volume);
- Pouring run time: The time required to complete the pouring process.

2.5.5. RQ5: What Devices Are Used to Measure the Volume Poured?

Reliable performance assessment relies on appropriate measurement instruments [26]. Therefore, the following parameter was included to address RQ5:

- Measurement instrument: Type of instrument used to measure the poured volume.

https://doi.org/10.3390/mca31010030

2.6. Information Sources

To identify relevant studies, the following bibliographic databases were searched: IEEE Xplore, WoS, SpringerLink, ScienceDirect and EBCOhost. To ensure the inclusion of recent advances, the search strategy was limited to publications from 2018 to 2025.

2.7. Search

The authors designed and executed the search strategy. For instance, in IEEE Xplore, an advanced search was conducted using the keywords Pouring, Liquid, and Control, restricted to metadata fields. The search was limited to articles published between 2018 and 2025. Filters were applied by publication year and content type, limiting the results to journal articles and conference proceedings. The same strategy and filters were consistently applied across all databases.

As an example, the following is the detailed search query used for IEEE Xplore: ("All metadata": Pouring) AND ("All metadata": Liquid) AND ("All metadata": Control). This query returned 45 results: 36 from conference proceedings, 7 from journals, 1 from a book, and 1 from a magazine article.

The search results across selected databases yielded to 285 publications before applying the inclusion and exclusion criteria. Gray literature was not included, and studies not focused on open-mouth containers or involving teleoperated pouring systems were excluded, as these were not relevant to the scope of this review.

2.8. Synthesis of Results

To ensure consistency, authors of this work screened the same set of 285 publications, discussed the results, and reviewed the data extraction protocol before beginning the full review. The team of reviewers sequentially evaluated the titles, abstracts, and then the full texts of all articles identified in our searches as potentially relevant. Disagreements about study selection and data extraction were resolved by consensus. This process ensured a standardized approach and minimized bias during the selection and extraction steps.

3. Results

3.1. Selection of the Sources of Evidence

Collected articles were distributed as follows: *IEEE Xplore: 45*, *SpringerLink: 45*, *Web of Science (WoS): 18*, *ScienceDirect: 153*, and *EBSCOhost: 24*. After applying inclusion and exclusion criteria, and removing duplicates, articles selection per database remained as follows: *IEEE Xplore: 12/45*, *SpringerLink: 2/45*, *Web of Science (WoS): 7/18*, *ScienceDirect: 2/153 (with 1 duplicate)*, and *EBSCOhost: 0/24 (with 4 duplicates)*. A total of 23 articles were included for further analysis, as depicted in Figure 1.

Articles excluded from the review were removed at different stages of the screening process ($n = 285$). At the title and abstract level, 181 records were excluded for being unrelated to liquid pouring, and five duplicate entries were identified. During the full-text eligibility assessment, 6 articles did not address the modeling of liquid pouring, 19 were based on another subject, 4 relied on dispensers rather than open-mouth containers, 37 were not written in English, and 10 focused exclusively on general object manipulation issues.

The temporal distribution of the included studies is shown in Figure 2. Research on liquid pouring has gradually expanded since 2018, with variations across years and a significant increase in 2024, reflecting the growing interest in this topic.

https://doi.org/10.3390/mca31010030

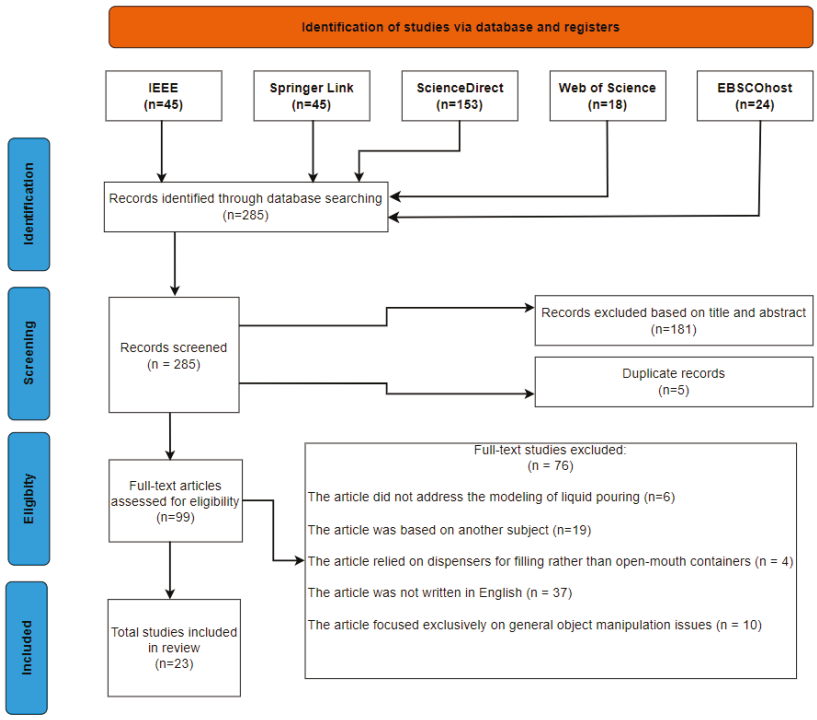


Figure 1. Flow chart of the PRISMA-ScR protocol.

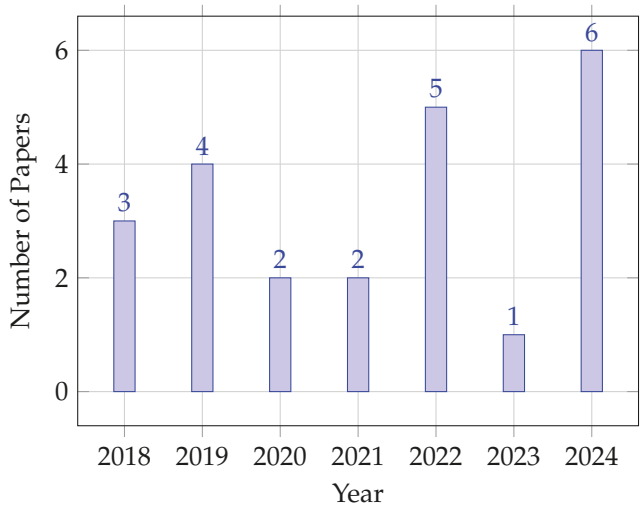


Figure 2. Temporal distribution of studies on autonomous liquid pouring published between 2018 and 2024.

3.1.1. RQ1: What Are the Liquid Characteristics Considered in the Experiments?

Based on the analysis of the articles included in this review, five liquid properties were considered for pouring.

- **Density:**
  Most of the studies (i.e., 87%: [9,14,16,26,45–60]) explicitly named the liquid density used in their experiments. Several density water values have been used in the experiments: 997.6 kg/m^3 in [60], 1.000 g/cm^3 in [45], and 1.000 kg/m^3 in [45]. Furthermore, milk tea with a density of 1.0276 g/cm^3 was used in the study [26]. Another work [47] used pancake batter in their experiments, with density values between 1.1 and 1.3 g/cm^3. Finally, Wang et al. [49] used three liquids in their experiments with the following densities: mineral water (1000 kg/m^3), cola (1030 kg/m^3), and milk tea (1040 kg/m^3).

In contrast, three articles (13%: [18,42,61]) did not explicitly report liquid density values used in the experiments. These studies only indicated the liquids used in their experiments, such as: water, milk, oil, and ketchup [18]. Nevertheless, density values can be assumed based on the liquid type.

- **Color:**
  A total of 74% of the articles explicitly reported the color of the liquids used in their experiments. The most widely used color was the color of water [26,42,45,51,53], followed by carbonated water [14,49], and milk tea (light brown) [26,49]. Other studies have used colors for the following liquids in their experiments: olive oil (golden) [14], pancake batter of a light cream to yellow color [47], coffee, tea, milk, grape juice, melon juice [51], cola (dark brown) [49], black liquid used for visibility [45], tea (black, green), juice [50], and liquids of varying color depending on the type of beverage [61].
  It is important to note that one study [18] simulated the following liquids for their experiments: water, ink, oil, glycerin, honey, and ketchup.
  In contrast, six articles (26%: [9,16,52,55,57,58]) did not explicitly report the color of the liquid. These studies focused on system performance or technical aspects rather than providing details on the appearance of the liquid.
- **Transparency:**
  A significant proportion of the reviewed literature, 22 articles (i.e., 96%), provided explicit information regarding the transparency of the liquids used. For example, ref. [47] described pancake batter as opaque, whereas ref. [51] included liquids such as water (transparent), coffee (opaque), tea (translucent), milk (opaque), and juices (opaque to translucent). In addition, ref. [18] simulated liquids with different transparencies, including water (transparent), ketchup (opaque), and honey (translucent). Similarly, ref. [26] specified the transparency of liquids like milk tea (opaque) and water (transparent), while ref. [49] reported mineral water (transparent), cola (opaque), and milk tea (opaque).
  Moreover, one paper (i.e., 4%) did not explicitly report on the transparency of the liquids [16].
- **Viscosity:**
  From the total number of records, 15 papers (65%) provided explicit reports of the viscosity of the liquids utilized in their experiments. For instance, ref. [18] reported simulated liquids with varying viscosities: water (0.01 mPa·s), ink (0.05 mPa·s), oil (6.5 mPa·s), glycerin (50 mPa·s), honey (65 mPa·s), and ketchup (80 mPa·s). Similarly, ref. [26] provided indirect viscosity information by mentioning liquids such as milk tea (moderate viscosity) and water (low viscosity). Also, ref. [60] reported the viscosity of water as approximately 0.889 mPa·s. Other papers like [14,45] referred to viscosity indirectly through liquid types such as water, oil, milk, and juice. Industrial-scale applications also included molten metals like aluminum and iron, characterized by high viscosity [55,57,58].
  In contrast, eight papers (35%) did not explicitly report the viscosity of the liquids used. These include [9,16,18,46,51,54,56,59]. Instead of focusing on the physical properties of the liquids, these studies prioritized aspects such as control performance, learning algorithms, or perception models.

The reporting of liquid properties across the 23 studies is summarized in Figure 3, while the distribution of the most frequently used liquids is presented in Figure 4.

https://doi.org/10.3390/mca31010030

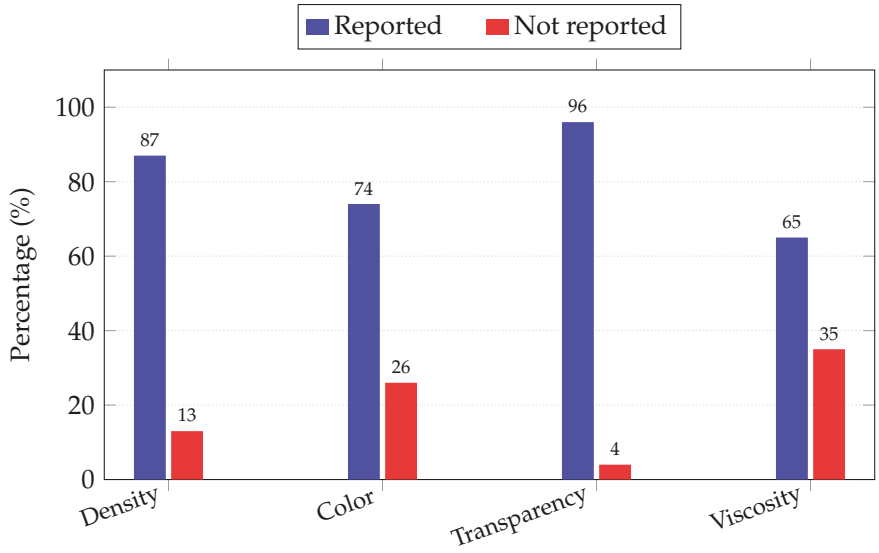


Figure 3. Summary of reported liquid properties across the 23 studies analyzed.

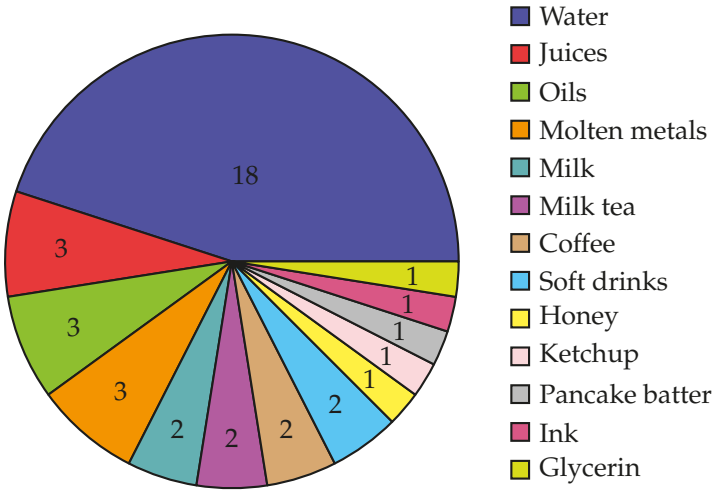


Figure 4. Types of liquids used in pouring experiments across the 23 studies.

3.1.2. RQ2: What Are the Container Characteristics Considered in the Experimentation and How Do Their Properties Affect the Performance of the Pouring Tasks?

As a result of the analysis of 23 studies, four characteristics were extracted: container shape, container transparency, opening diameter of the container's spout, and container height.

- **Container shape:**
  All studies described at least one container type, either as the *source* (pouring) or the *target* (receiving) container. Considering that some experiments involved multiple configurations, the classification is described below:
  - **Source containers.**
    The most frequently used source containers were *cups or mugs* (82%), typically with cylindrical glass cups as the standard cup. These containers were widely adopted due to their simple geometry and predictable pouring behavior [9,14,16, 26,45–51,53–60].
    In industrial studies, *ladles* were used in 17% of the studies as source containers [55–58]. These were generally metallic and operated through tilting mechanisms to control molten materials or heavy liquids.

https://doi.org/10.3390/mca31010030

Bottles were employed in 13% of the studies as source containers. These included wine and water bottles [59], pancake batter bottles [47], and cola bottles [49]. Finally, *cuboid or box-shaped containers* were the least common (4.3%) and appeared exclusively in simulation-based studies using Smoothed Particle Hydrodynamics (SPH) [46], where a rotating cuboid was employed as the source container.

– **Target containers.**
Only three studies explicitly described the target container geometry, which mainly included bowls and cuboid containers. *Bowls* were used in two works (9%) as target containers, mainly for viscous materials such as pancake batter or for visual–tactile experiments [45,47]. *Cuboids* (4.3%) were reported only in simulation-based work using the Smoothed Particle Hydrodynamics (SPH) method [46].

The distribution of container shaped documented in the literature is presented in Figure 5.

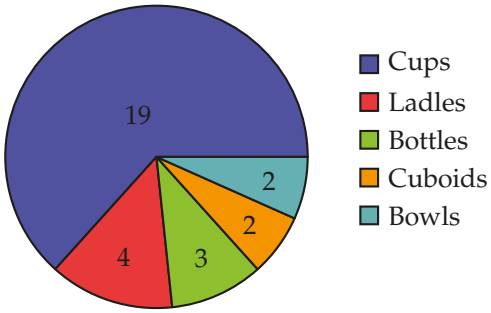


Figure 5. Distribution of container types reported across the studies.

- **Container transparency:**
Based on the results, the container transparency can be classified into transparent containers and opaque containers.
Most studies have used transparent containers (i.e., 17 studies: 74%) such as glass cups [18,26,45,47,49–53,59,61] and a beer mug [9]. Conversely, few studies employed opaque containers, including opaque plastic cups [16] and an opaque aluminum beverage [60]. In industrial contexts, opaque metallic ladles were reported in [56,57], while a transparent ladle was used in [55]. The remaining studies (26%: [42,46,48,50, 53,54]) did not explicitly specify the transparency of the containers.
- **Opening diameter of the container's spout:**
A total of 10 studies (43%) reported opening diameter measurements. For instance, diameters of 64 mm (standard cup), 56 mm (thin glass), and 79 mm (beer mug) were reported in [9]. Another study [16] indicated three measures for the diameters of the containers' spout (50–80 mm, 55–73 mm, and 70 mm). Similarly, Nishio et al. [60] described the beverage can with a 1.3 mm rim. Industrial ladles were modeled with upper and lower radii of r_1 = 195 mm and r_2 = 212 mm [58]. Additional examples include cylindrical or frustum cups with openings from 80 to 85 mm [52], as well as measuring cups or beakers with calibrated spouts [45,47,49,59].
In contrast, 13 studies (57%: [18,26,46,48,50,51,53,55–57,61]) did not provide explicit diameter measurements.
- **Container height:**
Twelve articles (52%) explicitly reported container's height. For instance, cups and glasses measured 110, 150, and 167 mm in [9], and 85, 82, and 100 mm in [16]. Other examples include a 110 mm beverage can [60], an acrylic ladle of 200 mm [55], and a cuboid container of 159 mm [46].

https://doi.org/10.3390/mca31010030

Additional studies incorporated container height as a geometric or learning feature rather than a measured parameter. For example, ref. [59] used container height as an input feature in a neural network to generalize pouring behavior across containers of different sizes, while [52] implicitly modeled container height to calibrate the source–target trajectory. Bottles with variable fill levels ranging from 20 to 70 mm were reported in [47]. Moreover, industrial ladles ranged from 350 to 490 mm in height were used in [56,58].

In contrast, almost half of the studies (48%: [16,18,26,47–51,53,57,61]) did not report the container's height. In these cases, container geometry was described qualitatively or mentioned implicitly; nevertheless, specific height values were not provided.

Table 2 shows a summary of the container properties documented in the literature, including shape, transparency, opening diameter, and height.

Table 2. Summary of container properties documented in the literature on robotic pouring.

Container Property	Reported	Not Reported	% Reported
Shape	23/23	0/23	100%
Transparency	17/23	6/23	74%
Opening diameter	10/23	13/23	43%
Height	12/23	11/23	52%

3.1.3. RQ3: What Are the Techniques Applied to Control Liquid Pouring?

The analysis of the selected papers based on RQ3 is summarized below, describing the type of algorithm, system robustness, experimental adaptability, and computational requirements.

In this review, control techniques are categorized according to the key control approach used to regulate the pouring process. In other words, this classification focuses on the control layer responsible for modulating the pouring behavior, rather than low-level motor or joint controllers. In this sense, autonomy processes or controls outside this scope are not considered, i.e., navigation, grasping nor transporting task control, are outside.

- **Algorithm type:**
  Out of the 23 reviewed articles, 18 explicitly reported a computational algorithm designed to control or regulate the pouring process, representing approximately 78% of the studies. The remaining five studies (22%) did not implement autonomous pouring controllers. However, they examined complementary aspects such as augmented reality for human-in-the-loop adjustment [48], multimodal monitoring of pouring success and failure [54], computational fluid dynamics (CFD) simulations of beverage pouring [60], kinematic and dynamic modeling of heavy-load ladle robots [58], and human motor control experiments involving both visual and tactile feedback [45].
  The algorithms reported in the studies of this review can be classified into nine categories: (1) Classical control (PID/PD/P/on–off), (2) Predictive control (MPC/NMPC), (3) Reinforcement learning (RL), (4) Imitation and end-to-end learning, (5) Probabilistic models (Gaussian Mixture Models), (6) Multimodal fusion (Vision and Audio), (7) Mechatronic pipelines with MLP and haptic/visual feedback, (8) State observers and estimation (Kalman Filters), and (9) Feedforward model-based control with online identification.

  (1) Classical control (PID/PD/P/on–off)
  Almost a third of the studies (30.4%) applied closed-loop control using simple regulators, occasionally integrated with vision-based perception. In the

context of PID controllers, Do and Burgard [14] implemented PID controllers on a PR2 robot using an RGB-D perception system, while Wang et al. [56] implemented a three-loop PID scheme for foundry pouring processes. Similarly, Camporredondo et al. [46] applied PID control for slosh stabilization through SPH simulations.

In terms of PD controllers, studies have implemented them in visual-based pouring scenarios as follows: (i) combining high-speed vision with PD control [9]; (ii) implementing P/PD strategies grounded on geometric container models [26]; (iii) introducing an on–off (bang–bang) control approach supported by self-supervised segmentation of transparent liquids [53]; and (iv) integrating weakly supervised perception with PID control in the PourIt! framework [51].

(2) Predictive control (MPC/NMPC)
A few studies (8.7%) have used predictive control to pour the liquid. Specifically, a study [42] employed Model Predictive Control (MPC) enabled by RNN predictions. A study [18] combined NMPC and reinforcement learning framework; however, the authors focused mainly on NMPC.

(3) Reinforcement learning (RL)
Reinforcement learning was applied in only two studies (8.7%). Zhang et al. [50] developed an offline RL method using a TD3+BC variant, while Babaians et al. [18] combined deep RL with PPO and an Intrinsic Curiosity Module (ICM), in conjunction with NMPC.

(4) Imitation and end-to-end learning.
Only three studies (13.0%) explored imitation and end-to-end learning. Zhang et al. [61] developed Explainable Hierarchical Imitation Learning (EHIL), Huang et al. [53] applied self-supervised segmentation using an LSTM, and Saito et al. [52] proposed end-to-end sensorimotor coordination using a Convolutional Autoencoder and LSTM.

(5) Probabilistic models (Gaussian Mixture Models)
Probabilistic motion planning was explored in a single study (4.3%). In this work, Wang et al. [49] combined a Gaussian Mixture Model (GMM) with a multimodal fusion network (MMFNet) to perform trajectory planning.

(6) Multimodal fusion (Vision and Audio)
Only one article (4.3%) employed multimodal fusion of visual and auditory inputs. Wang et al. [16] integrated ResNet-based visual features with LSTM-based audio features to design a three-stage pouring controller.

(7) Mechatronic pipelines with MLP and haptic/visual feedback
One study (4.3%) introduced a hybrid perception–control pipeline. Luo et al. [47] applied multilayer perceptrons (MLPs) to estimate pouring speed and time, while combining haptic and visual perception modules.

(8) State observers and estimation (Kalman Filters)
Another study (4.3%) employed decentralized Kalman Filters (DKFs) to estimate flow rate in real time for an automatic tilting-ladle pouring machine. Sueki and Noda [55] implemented both SSKF and EKF variants. It is worth noting that Do and Burgard [14] also applied a Kalman filter for liquid-level tracking; however, their primary control strategy was PID, so this work was categorized as classical controllers.

(9) Feedforward model-based control with online identification
Finally, one article (4.3%) implemented feedforward control with online parameter identification. Kabasawa and Noda [57] proposed inverse motor and process

https://doi.org/10.3390/mca31010030

models combined with optimization to adaptively tune parameters such as angle, density, and flow coefficient.

The techniques applied to control liquid pouring in the reviewed studies are summarized in Table 3.

Table 3. Techniques employed to control liquid pouring across the reviewed literature.

Technique	Frequency of Studies	%	References
Classical control (PID, PD, P, on–off)	7	30.4	Do and Burgard [14]; Camporredondo et al. [46]; Wang et al. [56]; Zhu and Yamakawa [9]; Dong et al. [26]; Narasimhan et al. [53]; Lin et al. [51]
Predictive control (MPC/NMPC)	2	8.7	Chen et al. [42]; Babaians et al. [18] (hybrid)
Reinforcement learning (RL)	2	8.7	Zhang et al. [50]; Babaians et al. [18]
Imitation and end-to-end learning	3	13.0	Zhang et al. [61]; Huang et al. [53]; Saito et al. [52]
Probabilistic models (GMM)	1	4.3	Wang et al. [49]
Multimodal fusion (vision + audio)	1	4.3	Wang et al. [16]
Hybrid pipeline (MLP + haptics/vision)	1	4.3	Luo et al. [47]
State observers (Kalman filters)	1	4.3	Sueki and Noda [55]
Feedforward model-based control	1	4.3	Kabasawa and Noda [57]
No autonomous controller (complementary methods)	5	22.0	Cleaver et al. [48]; Wu et al. [54]; Nishio et al. [60]; Li et al. [58]; Lin et al. [45]

Percentages are relative to the total number of reviewed studies (n = 23). Techniques are classified based on their primary control approach for pouring.

While Table 3 summarizes the distribution of control approaches in the literature, Tables 4 and 5 provides a complementary analysis explaining why each technique is employed, its limitations, and how performance is evaluated.

Table 4. Comparative analysis of control strategies for robotic liquid pouring: technical foundations, constraints, and performance assessment (Part I).

Control Technique	Representative Studies	Key Advantages	Technical Constraints	Assessment Metrics Reported
Classical control (PID, PD, P, on–off)	[9,14,26,46,51,53,56]	Selected for their computational efficiency and straightforward implementation. Enables real-time regulation of pouring velocity without requiring complex fluid dynamics models.	Limited adaptability to dynamic changes in liquid viscosity or container geometry. Performance often relies on fixed parameters and manual tuning.	Final volume error, flow stability, spill occurrence, and success rate. Statistical validation is frequently omitted.
Predictive control (MPC/NMPC)	[18,42]	Predicts pouring dynamics through approximate models or learned predictors. Optimizes trajectories under dynamic constraints while compensating for flow-response latency.	Highly dependent on model fidelity and prediction accuracy. Requires significant computational resources and is sensitive to modeling errors.	Volumetric error, task success rate, and comparison against baseline controllers.

Table 4. *Cont.*

Control Technique	Representative Studies	Key Advantages	Technical Constraints	Assessment Metrics Reported
Reinforcement learning (RL)	[18,50]	Used to model nonlinear fluid dynamics by learning control policies directly from interaction data, without the need for explicit physical models.	High data dependency and computational overhead. Discrepancies between the simulated and actual conditions are often encountered, and formal guarantees of stability are lacking.	Success rate, average pouring error, and generalization to novel containers or volumes.
Imitation and end-to-end learning	[52,53,61]	Leverages human demonstrations to map sensor inputs directly to control actions, eliminating the need for complex fluid–structure interaction models.	Lacks formal interpretability and stability guarantees. Performance is constrained by the quality and diversity of training demonstrations.	Pouring accuracy, qualitative success analysis, and cross-trial consistency.

Table 5. Comparative analysis of control strategies for robotic liquid pouring (Part II): Technical foundations, constraints, and metrics.

Strategy	Studies	Key Advantages	Technical Constraints	Assessment Metrics
Probabilistic (GMM)	[49]	Captures trajectory variability and learns patterns under uncertainty. Enables adaptive planning via sensory feedback.	Requires extensive demonstration data; sensitive to container/liquid diversity.	Volumetric error, height estimation, and task completion rate.
Multimodal Fusion	[16]	Augments vision in opaque/low-light scenarios. Provides redundant flow-state cues via acoustic feedback.	High complexity; sensitive to environmental noise and FFT-induced latencies.	State classification accuracy, robustness, and height estimation.
Hybrid Pipelines	[47]	Integrates perception and learning for viscous or non-Newtonian fluids. Enables adaptive speed and duration time.	High computational cost due to multi-sensor integration and learning overhead.	Pouring accuracy, distribution consistency, and qualitative quality.
State Observers	[55]	Used when direct flow measurement is not possible (e.g., molten metal). Kalman filters allow for real-time estimation by industrial sensors with low computational cost.	Focused on estimation rather than control; limited disturbance rejection.	Flow estimation error, noise reduction, and parameter robustness.
Feedforward Model-based	[57]	Leverages physical models for real-time regulation. Suitable for resource-constrained industrial platforms.	Constrained by model fidelity; lacks adaptability to unmodeled fluid dynamics.	Volumetric MAE, trial adaptation, and sensitivity analysis.
Complementary Methods	[45,48,54,58,60]	Simplifies the analysis of sensing and fluid dynamics by evaluating them independently, avoiding the complexities of full-system integration.	Open-loop execution only; lacks real-time flow regulation or adaptive control.	Perception accuracy, simulation results, or human-performance metrics.

- **System robustness:**
  System robustness was assessed in terms of whether the proposed approaches demonstrated the capacity to dynamically adjust the pouring process in response to unex-

pected changes. Of the 23 studies, eight papers (34.8%) explicitly reported adaptive mechanisms (i.e., adaptive approaches), while the remaining 15 papers (65.2%) relied on fixed parameters without real-time adjustment (i.e., non-adaptive approaches).

Adaptive approaches (34.8%)

The following eight studies have implemented approaches that adapted to changes:

- Zhu and Yamakawa [9] integrated high-speed vision with PD control to modify pouring in real time based on liquid dynamics;
- Narasimhan et al. [53] applied self-supervised segmentation to detect transparent liquid flow, enabling on–off control adjustments;
- Lin et al. [51] combined weakly supervised perception with PID in the *PourIt!* framework to adapt trajectories under variable conditions.;
- Zhang et al. [50] proposed an offline reinforcement learning approach that transferred liquid under previously unseen pouring conditions;
- Babaians et al. [18] integrated deep RL with NMPC, explicitly addressing changes in liquid behavior;
- Wang et al. [16] combined ResNet-based vision with LSTM-based audio features, where multimodal fusion supported robust detection of flow states;
- Wang et al. [49] incorporated probabilistic motion planning with GMM and MMFNet to adapt trajectories under multimodal uncertainties;
- Luo et al. [47] proposed a hybrid perception–control framework (MLPs with haptic and vision), adapting the pour speed and duration to external feedback.

Non-adaptive approaches (65.2%)

The remaining studies employed fixed controllers or modeling techniques without explicit mechanisms to handle unexpected variability:

- Do and Burgard [54], Camporredondo et al. [46], and Wang et al. [56] applied PID schemes with no adaptive mechanisms.
- Dong et al. [26] and Li et al. [58] relied on geometric or dynamic models with pre-set parameters;
- Sueki and Noda [55] implemented Kalman filters for flow estimation but not for adaptive control;
- Nishio et al. [60], Cleaver et al. [48], and Lin et al. [45] explored CFD, augmented reality, and motor control experiments, respectively, without autonomous adjustments;
- Huang et al. [53] and Saito et al. [52] focused on imitation and end-to-end learning, but did not demonstrate real-time adaptation;
- Wu et al. [54] concentrated on multimodal monitoring without adaptation mechanisms.

Figure 6 summarizes the distribution between adaptive and non-adaptive control strategies identified across the reviewed studies. As shown, only 34.8% of the studies employed adaptive mechanisms, whereas most (65.2%) were based on fixed, non-adaptive control environments, showing a dependence on predefined parameters rather than real-time adaptation.

https://doi.org/10.3390/mca31010030

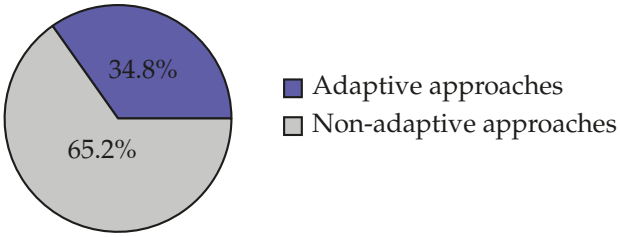


Figure 6. Proportion of adaptive and non-adaptive control approaches in the reviewed studies.

- **Experimental adaptability:** Experimental adaptability was evaluated according to whether the methodology was tested under different initial volumes or filling levels. Out of the 23 studies, seven works (30.4%) reported explicit validation under multiple starting conditions, while the remaining 16 papers (69.6%) did not include such variations.

  Tested under different volumes/levels (30.4%)

  The following seven studies conducted experiments using several volumes/levels of liquid:

  - Zhu and Yamakawa [9] evaluated PD control with high-speed vision under varying liquid amounts;
  - Narasimhan et al. [53] tested on–off control with transparent liquids at different filling levels;
  - Lin et al. [51] validated the *PourIt!* system using PID across multiple initial volumes;
  - Babaians et al. [18] applied reinforcement learning to different container fill levels;
  - Zhang et al. [50] assessed offline RL with TD3 and BC for generalization to new starting volumes;
  - Wang et al. [16] examined multimodal fusion robustness under variable initial fills;
  - Luo et al. [47] tested the hybrid perception–control pipeline across different liquid amounts.

  Not tested under different volumes/levels (69.6%)

  The remaining 16 studies operated with fixed experimental setups and did not explicitly assess performance across varying filling levels:

  - Do and Burgard [54] applied PID schemes with constant initial volumes;
  - Camporredondo et al. [46] and Wang et al. [56] applied PID schemes with constant initial volumes;
  - Dong et al. [26] and Li et al. [58] conducted experiments using predefined static volumes only;
  - Sueki and Noda [55] estimated flow rate with Kalman filters but did not test different liquid levels;
  - Nishio et al. [60], Cleaver et al. [48], and Lin et al. [45] explored CFD, augmented reality, and motor control experiments without varying initial conditions;
  - Huang et al. [53] and Saito et al. [52] applied imitation and end-to-end learning at single fixed levels;
  - Wu et al. [54] analyzed multimodal monitoring of pour success and failure under constant setups;
  - Wang et al. [49] evaluated trajectory generation under fixed initial states.

As shown in Figure 7, only 30.4% of the reviewed studies tested their methods under varying initial liquid volumes, while 69.6% relied on fixed setups. This highlights a limited

experimental adaptability across most works, where pouring control is typically evaluated under static rather than dynamic conditions.

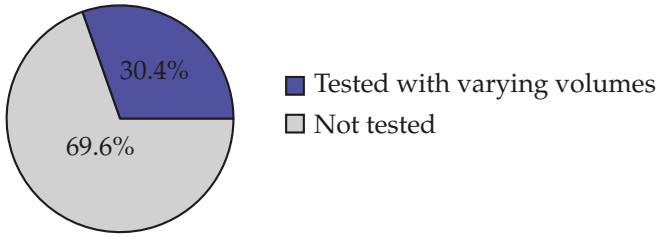


Figure 7. Experimental adaptability across studies: proportion tested under multiple starting volumes vs. fixed setups.

- **Computational Requirements:**
  Computational requirements were analyzed in terms of the hardware and software resources necessary to implement each methodology. Among the 23 studies, eleven articles (47.8%) required GPU-based or high-performance computing (HPC) environments, nine articles (39.1%) operated on standard robotic or PC hardware, and the remaining three papers (13%) were conducted in simulation without explicit hardware deployment.

  GPU/HPC

  Eleven studies (47.8%) required GPU or HPC resources for deep learning, reinforcement learning, or high-fidelity simulation. These include end-to-end CAE + LSTM control [52], self-supervised transparent liquid segmentation [53], self-supervised LSTM training for accurate pouring [59], explainable imitation learning [61], offline RL with TD3 + BC [50], multimodal audio–vision fusion [16], multimodal GMM/MMFNet planning [49], RL with PPO + ICM and NMPC integration [18], SPH-based simulations [46], CFD with Volume of Fluid Method (VOF) [60], and hybrid haptic–vision MLP frameworks [47].

  Standard PC/robot hardware.

  Approximately 39.1% of the studies operated on conventional computing platforms such as PCs, PLCs, or industrial robot controllers, typically using ROS, MATLAB/Simulink (https://www.mathworks.com/products/simulink.html (accessed on 19 December 2025)), OpenCV, or vendor-specific toolchains, without dedicated GPU or HPC requirements. Representative cases include RGB-D pouring with PID on a PR2 robot [14], P/PD control on industrial manipulators [26], on–off control with self-supervised segmentation on a UR5 arm [53], decentralized Kalman filtering and feedforward model-based control in tilting-ladle machines [55,57], hybrid foundry pouring with a three-loop PID implemented on PLC hardware [56], high-speed vision with PD control [9], and kinematic/dynamic modeling of heavy-load ladle robots in MATLAB/Simulink [58].

  Not reported.

  The remaining three studies (13%) implemented simulations; nevertheless, the authors did not report the hardware deployment used in the research. These included augmented reality guidance with HoloLens/Unity [48], human sensorimotor experiments without robotic computation [45], and multimodal monitoring on PR2 that reported only a basic ROS/Kinect setup [54].

https://doi.org/10.3390/mca31010030

As shown in Figure 8, nearly half of the reviewed studies relied on GPU/HPC resources, while the remainder operated on standard PC/robotic hardware or did not specify computational requirements.

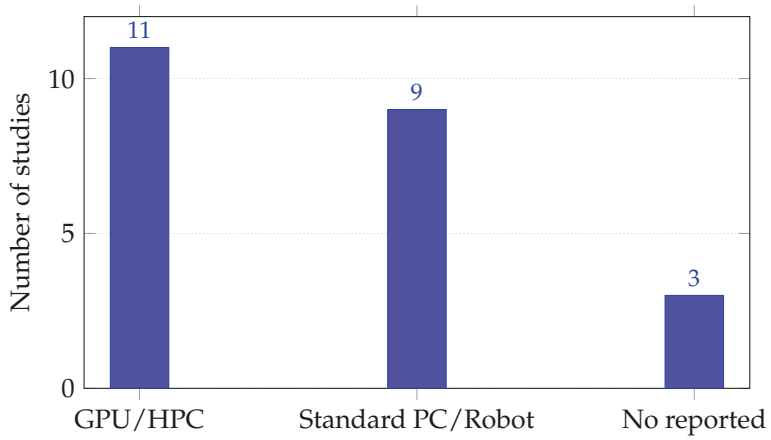


Figure 8. Computational requirements across the 23 studies: GPU/HPC (11), standard PC/robotic hardware (9), and unspecified/not reported (3). Each paper is assigned to its *primary* requirement to avoid double counting.

3.1.4. RQ4: What Metrics Have Been Used to Assess the Methods for Pouring Liquid?

It is essential to note that the studies employed different metrics to evaluate their proposals. Specifically, they employed the following five metrics:

- **Average pour error:** Over half of the studies (i.e., 61%: [9,14,16,18,26,42,49,52–55,57,59,60]) used the average pour error as a metric to evaluate the performance of robotic pouring systems. Specifically, three types of units have been used to report this error: percentage, milliliters, and grams.
  At least six studies reported the error in percentage terms, including ref. [52], who reported deviations between 6.0% and 7.8%, ref. [53], with a 0.94% relative error in fluid level estimation, and ref. [9], who achieved a steady-state error of less than 5% in beer foam control. Similarly, refs. [57,60] reported relative deviations under 5%, while ref. [54] defined successful pours as those within a $\pm 5\%$ tolerance.
  Regarding the error measured in milliliters, at least six studies, including [14,16,26,42,49,59], reported values ranging from approximately 3.7 mL to 30.5 mL, depending on the fluid, container, and level of generalization. These studies often used volumetric error to compare learning-based and classical controllers.
  Lastly, two studies reported the error in grams [18] and achieved errors between 3.8 and 8.66 g across different liquids, and ref. [57] showed improvement from 0.1346 kg to 0.0498 kg after parameter tuning, reinforcing the robustness of their model-based control strategy.
  Of these, 14 papers are distributed as shown in Figure 9.
- **Success rate:**
  Among the 23 reviewed papers, only six explicitly report the success rate of the poured liquid in terms of percentage, representing 26% of the total number of articles. These include: [61] with 92%, [50] with 95.25%, [51] with 93.3%, [54] with a classification success rate within $\pm 5\%$ error, [48] with 53.8%, and [16] with a completion rate of 73–85%, depending on container and liquid.
  In addition to the studies that explicitly reported success rates as percentages, three papers (13%) evaluated pouring success using qualitative or binary criteria rather than numerical values. For example, ref. [59] assessed success using binary labels (success/failure), based on tasks completed and the minimal trajectory deviation,

https://doi.org/10.3390/mca31010030

without providing quantitative accuracy. Another case is ref. [56], who evaluated pouring performance qualitatively, comparing poured and target masses without reporting explicit percentages. Similarly, ref. [14] reported stable performance with low error margins over 290 trials but did not specify a numerical success rate. These studies provide indirect insights into system performance, but lack standardized metrics for quantitative comparison.

A considerable proportion of the reviewed literature, comprising over half of the papers (61%), did not explicitly report or assess the success rate of the poured liquid (i.e., [9,18,26,42,45–47,49,52,53,55,57,58,60]). The overall distribution of papers according to success rate reporting is illustrated in Table 6.

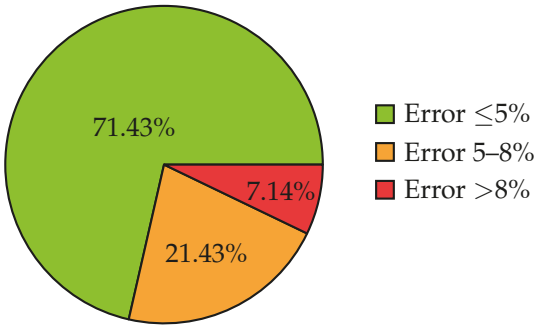


Figure 9. Distribution of average pour error (APE) ranges among the 14 studies that reported this metric.

Table 6. Success rate reporting across the reviewed literature, aligned with Figure 10 (1–6).

ID	Reference	Reporting Type	Success Metric/Notes
Explicit quantitative reporting (6 papers; 26%)			
1	Zhang et al. [61]	Percentage (%)	92%
2	Zhang et al. [50]	Percentage (%)	95.25%
3	Lin et al. [51]	Percentage (%)	93.3%
4	Wu et al. [54]	Percentage (%)	Classification success within ±5% error (shown as ∼95% in Figure 9)
5	Cleaver et al. [48]	Percentage (%)	53.8%
6	Wang et al. [16]	Percentage (%)	73–85% (depends on container/liquid)
Qualitative/binary reporting (3 papers; 13%)			
–	Huang et al. [59]	Qualitative/binary	Comparable to human performance; consistent trajectories and reduced errors (no explicit %)
–	Wang et al. [56]	Qualitative/binary	Success assessed via qualitative observations and mass transfer; no explicit percentage
–	Do and Burgard [14]	Qualitative/binary	Stable/low error across 290 trials; no explicit success rate
Not reported (14 papers; 61%)			
–	Chen et al. [42]; Luo et al. [47]; Wang et al. [49]; Babaians et al. [18]; Dong et al. [26]; Saito et al. [52]; Narasimhan et al. [53]; Lin et al. [45]; Zhu and Yamakawa [9]; Camporredondo et al. [46]; Sueki and Noda [55]; Li et al. [58]; Kabasawa and Noda [57]; Nishio et al. [60]	Not reported	No explicit success rate provided.

Percentages are relative to the total number of reviewed studies (n = 23).

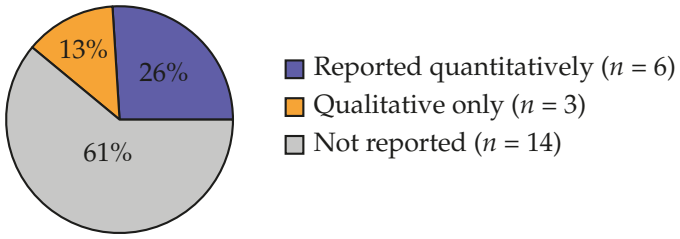


Figure 10. Success rate reporting across the 23 reviewed studies.

- **Spill volume:**
  Only five articles explicitly reported spill volume measurements or provided indirect estimations. For instance, ref. [18] reported deviations of approximately 2.3 g for soap and 5.5 g for water, indirectly indicating spill volumes. Similarly, ref. [60] quantified spill through overflow measurements, reporting average errors of approximately 4.6 mL and 2.6 mL. Ref. [56] provided measurements of external dripping in cm^3 as a spill metric, while ref. [14] reported pour errors between 11.88 ± 4.63 mL and 4.96 ± 2.57 mL, which can be interpreted as spill volume under certain conditions. Only one article [54] provided binary or qualitative evaluations of spill volume without explicit values, while the remaining 18 papers, comprising approximately 78% of the total (i.e., [9,16,26,42,45–53,55,57–59,61]), did not quantify spill volume in their experiments.
- **Pouring run time:**
  A total of 11 papers (48%) reported explicit measurements of the pouring run time, that is, the duration required to complete the liquid pouring process. For instance, ref. [49] reported an average pouring time of 8 s; ref. [52] reported a typical duration of 38 s; ref. [45] measured times through phase-based segmentation without providing a total duration; ref. [16] reported a duration of 10 s; ref. [9] reported around 11 s; ref. [55] detailed extremely short durations of less than $0.02s$, reflecting high-speed industrial processes; ref. [56] derived the time indirectly from control parameters; ref. [58] provided modeled durations using mathematical formulas; ref. [57] reported pouring durations of less than 4 s; [59] presented a range of 2.8 to 7.6 s; and ref. [60] provided dynamic measurements across trials, reporting durations ranging from 3.2 to 8.7 s depending on container and fluid properties. These reports illustrate the diversity of approaches to measuring pouring performance, highlighting variability in completion times between different robotic systems, experimental setups, and evaluation conditions.
  Conversely, 12 papers (52%) did not report explicit measurements of pouring run time, nor did they provide indirect or qualitative evaluations of this variable, thereby limiting the assessment of temporal performance in their studies. These papers include [14,18,26,42,47–51,53,59,61], which focused on other performance metrics such as accuracy, trajectory planning, or liquid detection, without explicitly quantifying the duration of the pouring process.

3.1.5. RQ5: What Devices Are Used to Measure the Volume Poured?

Across the 23 reviewed studies, different strategies were adopted to measure or estimate the poured liquid volume. These can be grouped into four main categories: *weight/force sensors*, *vision-based methods*, *multimodal fusion*, and *simulation-based estimation*. A minority of works did not include any explicit measurement device.

- **Weight and force sensors**
  A subset of studies (21.7%) employed load cells or force/torque (F/T) sensors as their primary modality. These works used six-axis F/T sensors and integrated load

cells in automatic pouring systems [42,52,55,57,59]. These devices directly measure transferred mass, enabling closed-loop regulation.

- **Vision-based estimation.**
  A total of 30.4% of the studies relied primarily on vision-based approaches to estimate the poured volume, using RGB cameras and their depth-extended variant (RGB-D). For instance, ref. [14] applied an RGB-D sensor with refraction correction, ref. [26] used depth data from a calibrated camera, and ref. [51] integrated depth and monocular RGB information. In addition, vision techniques were employed for transparent liquid segmentation [53], foam/liquid differentiation [9], RGB-based state inference [61], and hardware-in-the-loop feedback [49].
- **Multimodal Fusion.**
  Two papers (8.7%) integrated multimodal inputs, mainly combining vision and audio. refs. [16,49] integrated depth vision with a microphone and validated results using an external electronic scale. In both cases, multimodal signals drove the controller, while the scale served only as a ground-truth reference.
- **Simulation-based estimation.**
  Three studies (13.0%) relied solely on computational modeling without physical devices. ref. [46] simulated pouring with SPH, ref. [60] used CFD with the VOF method, and ref. [58] derived kinematic/dynamic models of ladle pouring.
- **No explicit measurement.**
  Finally, six studies (26.1%) did not employ explicit devices to quantify poured volume. These include work on augmented reality [48], multimodal monitoring [54], human motor control [45], and reinforcement learning approaches that relied on fixed targets or classifiers rather than in-loop measurement [18,50].

Table 7 summarizes the devices and strategies used to measure poured volume, showing a predominance of vision- and sensor-based approaches over simulation or qualitative methods.

Table 7. Categories of devices and strategies used to measure poured liquid volume across the reviewed studies.

Category	Frequency of Studies	Proportion	References
Sensor-based approaches			
Weight/Force sensors	5	21.7%	[42,52,55,57,59]
Vision-based estimation (RGB-D)	7	30.4%	[9,14,26,49,51,53,61]
Multimodal fusion (vision + audio)	2	8.7%	[16,49]
Non-sensor approaches			
Simulation-based estimation	3	13.0%	[46,58,60]
No explicit measurement	6	26.1%	[18,45,48,50,54]

4. Discussion

The following section presents a detailed analysis of the results for each of the five research questions (RQs).

4.1. Discussion for Research Question 1

As shown in Figure 3, transparency was the most frequently reported liquid property across the reviewed works, followed by density, color, and viscosity. The following sections analyze the relationships between density and viscosity, as well as between color and transparency, to better understand these differences.

4.2. Density vs. Viscosity

Although both density and viscosity are fundamental to fluid dynamics, studies indicate that they are utilized differently during the pouring process. Only a subset of articles quantified both properties simultaneously. The majority reported either density or viscosity, depending on their experimental objective.

Studies using common low-viscosity liquids, such as water, milk, or tea, typically report only density, as this directly influences mass-based control and simplifies model parameterization for feedback systems (e.g., those using load cells or F/T sensors). Density is therefore used in models that approximate fluid behavior through simplified physical priors, where flow dynamics are not explicitly simulated.

In contrast, research on non-Newtonian or high-viscosity substances, such as honey, syrup, ketchup, and molten metals, tends to emphasize viscosity, as it regulates the flow resistance, strain rate, and time constant of the pouring flow. Viscosity becomes particularly relevant for adaptive or trajectory-based control, where precise flow modulation depends on the fluid's response to motion and container geometry.

Despite their close physical relationship, few studies have examined density and viscosity together. However, their combined influence determines flow regime transitions, such as laminar or turbulent flow.

4.3. Color vs. Transparency

Even though color and transparency are both optical properties of liquids, they are reported with different purposes across the studies. Most studies (74%) explicitly described color, whereas transparency was reported in 96% of the studies. This contrast reflects that color is typically treated as a perceptual or dataset attribute, while transparency directly influences how robotic systems perceive and measure liquids through vision-based sensors.

Most studies using common beverages such as water, milk, tea, or juice reported color mainly to distinguish experimental conditions or support visual learning tasks (e.g., water or milk-tea segmentation). In contrast, works focused on control performance or industrial applications rarely mentioned color, as visual appearance was irrelevant to their objectives. As shown in Figure 4, water was the most frequently used liquid due to its stability and well-known physical properties. However, this strong dependence on water limits the exploration of more complex optical and flow behaviors.

Transparency was reported in almost all studies because it determines the optical interaction between the liquid surface and the sensing modality. Transparent and translucent liquids present refraction and depth estimation problems in RGB-D or stereo systems. Conversely, opaque liquids facilitate segmentation because their boundaries are visually defined. However, they interfere with real-time surface monitoring, since light cannot pass through to indicate flow or level variations.

4.4. Discussion for Research Question 2

As discussed in Section 4.1, the physical and optical properties of liquids have a significant influence on pouring performance. However, the container that holds or receives the liquid is equally important, as its geometry, size, and material determine how the liquid flows, is sensed by sensors, and can be controlled during the pouring process. As shown in Table 2, container shape was reported in all reviewed studies, highlighting its central role in both experimental setup and task performance.

4.4.1. Source vs. Target Containers

There is a distinction between the source and target containers, which perform complementary functions during the pouring process. Its symmetrical and regular geometry

https://doi.org/10.3390/mca31010030

allows for liquid flow to be predicted using simple equations, without the need to model turbulence or irregular shapes. This symmetry also facilitates the estimation of liquid transfer and the change in the container's center of gravity. These characteristics, in combination with the physical simplifications described in Research Question 1 (RQ1), enable dynamic modeling and control.

Conversely, target containers were mentioned in just three studies, usually as bowls or cuboid containers. This limited consideration suggests that target containers are often regarded as elements that simply receive the liquid, rather than influencing the pouring dynamics.

4.4.2. Material Transparency vs. Geometric Constraints

Transparent glass containers are used in most laboratory experiments because they allow vision-based systems to monitor liquid levels and flow continuity. In contrast, opaque or metallic containers are common in industrial settings. However, opaque materials hinder visual tracking and require force or torque sensors to compensate for the lack of visual feedback.

Additional constraints on pouring dynamics are caused by geometric parameters such as spout width and container height. A wider spout enables faster flow but reduces precision, while narrower openings improve control. Some studies did not directly measure the height of the container, but instead used it as an input variable in their models. This suggests a trend toward data-driven approaches, where the system learns how geometry affects the pouring process rather than relying solely on predefined geometric measurements.

4.5. Discussion for Research Question 3

As previously shown in Table 3, the majority of studies implemented algorithm-based approaches to regulate the pouring process, while a smaller portion focused on aspects such as simulation or experiments with human intervention.

Classical controllers (PID, PD, and on/off) remain the most widely used for liquid pouring. As reported by authors, the prevalence is based on their computational efficiency, ease of implementation, and stable behavior in real-time robotic systems. These controllers can regulate the pouring angle or rate without explicitly modeling nonlinear fluid dynamics. Furthermore, their control parameters (e.g., proportional and derivative gains, threshold values, or fixed pouring angles) can be manually modified, allowing the system's behavior to be tuned based on empirical observations. This quality makes classical control particularly suitable for scenarios with limited training data, limited hardware resources, or critical safety requirements.

In contrast, learning-based methods, such as reinforcement learning, imitation learning, and probabilistic models, are used less frequently. While these approaches can be adapted to complex nonlinear fluid behaviors, they typically require large datasets, extensive training, or accurate fluid dynamics simulators, which are often unavailable or computationally expensive. Furthermore, issues related to stability guarantees, convergence, and transfer from simulation to reality remain significant challenges that limit their adoption in real-world pouring systems.

A critical factor in the efficacy of a control system is robustness. The majority of conventional controllers operated with fixed parameters, constraining their adaptability to unexpected changes in liquid behavior. Adaptive approaches, particularly those incorporating reinforcement learning, multimodal fusion, or hybrid perception and control frameworks, demonstrated greater flexibility, allowing for real-time adjustments under varying conditions. These findings underscore the importance of prioritizing robustness as a central criterion for future evaluations.

https://doi.org/10.3390/mca31010030

Although multimodal fusion offers robustness advantages, technical implementation introduces significant challenges regarding fusion architectures and control loop latency. The reviewed literature reveals three primary strategies for integrating sensory inputs.

1. Deep Learning-based Fusion: For example, in [16], the authors employ a "late fusion" architecture, where visual features (from attention-based ResNets) and audio features (processed via FFT and LSTMs) are concatenated. While effective for opaque liquids, this approach relies on high GPUs and introduces latency due to the signal processing window required for FFT and the inference time of deep models.
2. State estimation and Kalman filtering: In contrast, industry-oriented studies such as [55] use decentralized Kalman filters (DKFs) to fuse data from motor encoders and flow sensors. This architecture is computationally lightweight and is optimized for real-time execution, although it is less adaptable to complex dynamics.
3. Haptic integration: ref. [47] describes a specialized architecture, in which haptic feedback from a force/torque sensor is used to estimate the properties of the fluid, during the agitation phase. In this work, the authors propose a parameter-level fusion in which the output of a multilayer perceptron (MLP) is used to adjust the pour controller's speed. Although this approach improves robustness for variable viscosities (e.g., non-Newtonian fluids, such as pancake batter), the system requires calibration before execution. While using haptics reduces reliance on visual feedback, sequential task execution is required to maintain state estimation accuracy.

There is a significant gap concerning experimental adaptability and computational requirements. Few studies validated their methods using different initial volumes or fill levels, even though real-world applications rarely involve identical fill levels. Approximately half of the studies relied on GPU or HPC resources, particularly to support computationally intensive components such as learning-based controllers, perception modules, or simulation environments. However, real-world robotic platforms operate on standard PCs or embedded hardware.

4.6. Discussion for Research Question 4

As established in Section 3.1.4 and Figure 9, the most frequently used metric was the average pour error (APE), which was reported in 61% of the studies. Most errors were under 5%, indicating high precision in controlled settings. However, some studies reported higher deviations, reflecting variability across tasks and liquids.

Success rates were reported less frequently. While a minority of works quantified it explicitly (Table 6), others relied on qualitative or binary labels. More than half of the literature did not assess this metric (Figure 10), which limits comparability. A standardized definition of "success" is needed to assess robustness across different systems.

Spill volume was the least considered metric, mentioned only in five studies. Its omission reduces its real-world applicability because it directly reflects efficiency and safety. The pouring run time was documented in about half of the papers. Reported values ranged from fractions of a second in industrial contexts to over 30 s in laboratory experiments, thus demonstrating differences across the diverse scenarios.

The diversity of these metrics highlights a significant lack of standardization in this field. To better understand these findings, the identified metrics can be organized into a taxonomy based on four dimensions: accuracy (pour error), safety (spills), efficiency (runtime), and reliability (success rate). Currently, the lack of the parameters makes it difficult to compare the effectiveness of different control techniques. For example, measuring only the final volume does not capture for flow quality, such as whether splashing occurs. Since this is a new area of research, it is important to establish standardized evaluation protocols.

https://doi.org/10.3390/mca31010030

4.7. Discussion for Research Question 5

Based on the data detailed in Section 3.1.5 and Table 7, four main strategies were used to measure poured volume: vision-based estimation (30.4%), weight or force sensors (21.7%), multimodal fusion (8.7%), and simulation approaches (13.0%). More than a quarter of the studies (26.1%) did not employ explicit measurement devices.

There are two types of volume measurement approaches: direct and indirect. Direct measurement using weight and force sensors provides an accurate value of mass transfer, facilitating closed-loop control systems. However, its use was limited, as not all containers, particularly scoops, can be placed on instrumented stands. Indirect approaches rely on perceptual inference, primarily using vision-based systems that estimate liquid levels from visual signals. Although camera-based solutions are easier to integrate into robotic platforms, they are highly sensitive to transparency and lighting conditions.

Multimodal fusion attempted to address these limitations by combining complementary signals, such as visual and audio, to estimate flow. However, environmental noise limits its real-world applicability. Simulation-based methods provide useful data for algorithm validation. Nevertheless, they could not compare physical evaluation under varying conditions.

4.8. Limitations

This scoping review presents the following limitation:

Terminology-and-scope boundary. The search terms were selected based on the most commonly used and recognized terminology in is field. Based on this, “pouring”, “liquid”, and “control” were included as main search terms. However, it is important to note that studies using related or synonymous terms, such as “dispensing” or “decanting”, are excluded. These terms were excluded in accordance with our inclusion criteria, which prioritize liquid transfer via tilting open-mouthed containers over internal mechanical valves or pressurized systems, which are typically described as “dispensers”. Similarly, while the review focuses on pouring control, studies referring to “liquid transport” or “spill control” during locomotion were excluded.

5. Future Directions

Based on the results of question 1, future research in terms of this RQ should focus on indicating the liquid properties. Most studies used qualitative labels to report density, viscosity, transparency, and color. Consequently, the experiments could not be reproduced. In order to ensure the reproducibility of the research, it is necessary to at least provide quantitative values for the following liquid characteristics: density, viscosity, transparency, and color. Moreover, the majority of the studies have employed water in the experiments. Nevertheless, several types of liquid are manipulated in daily human activities and industry applications. Consequently, experiments should expand the range of liquids analyzed, including viscous, opaque, carbonated, or non-Newtonian fluids that might provide conditions closed to real-world situations. Additionally, the algorithms for controlling pouring liquid might be more robust and fault-tolerant when they are tested using more types of liquid.

Focusing on RQ2, opportunity areas for further research could expand the range of container shapes beyond cups, jugs, and cuboids to incorporate bottles, ladles, and irregular geometries. Additionally, transparency should be addressed in greater detail. Transparency should be reported with degrees of opacity rather than as a binary property (transparency or opaque), since this directly influences detection algorithms. Studies must report precise numerical values for container characteristics, such as spout height and diameter, as these parameters determine the flow rate and pouring accuracy. Therefore, combining a variety

https://doi.org/10.3390/mca31010030

of containers could improve adaptability, enabling systems to handle unconventional containers common in industrial and service environments.

Regarding RQ3, and the prevalence of classical controllers in the current literature, future research should emphasize adaptive and learning-based approaches. Reinforcement learning, imitation learning, and multimodal fusion are promising for handling nonlinear fluid dynamics and diverse container geometries. Effective validation of these approaches requires testing across varied experimental conditions encompassing multiple liquid quantities, container fill levels, and pouring rates to demonstrate generalizability and robustness. Moreover, the balance between system accuracy and latency is rarely addressed in the literature. For example, audio sensors are very useful in low-light environments, but processing these signals in real time requires significant computational effort. At the same time, algorithms must be optimized for deployment on resource-constrained hardware. Developing lightweight architectures that minimize dependence on GPUs or HPCs will prove essential for practical implementation on robotic platforms operating in real-world environments.

According to RQ4, the average pour error was the most frequently reported metric, but it remains inconsistent across studies. Some studies expressed this measurement as a percentage, others in milliliters or grams, without a standardized definition. Conversely, researchers rarely quantified spill volume, despite its direct relevance to precision assessment. Pouring run time was largely ignored in most studies, even though it is crucial for assessing performance in service environments. Future research should evaluate the trade-off between speed and accuracy, exploring how adaptive controllers can optimize both speed and accuracy without compromising safety.

Finally, based on RQ5, future research should focus on combining the advantages of direct and indirect measurement methodologies. For instance, combining vision with weight or haptic sensors could provide both the accuracy of physical measurements and the adaptability of perception-based control. Additionally, multimodal fusion should extend beyond vision and audio setups to incorporate more robust combinations capable of processing noise and variable lighting. Furthermore, simulation studies should always be supported by physical validation to ensure reproducible and transferable results.

6. Conclusions

This work highlights several methods and techniques from the literature on robotic liquid pouring. Most studies rely on qualitative descriptors of liquid properties and use water as the default test substance, which limits the ability to generalize. Container shape was consistently reported, but other characteristics such as transparency, height, and spout diameter were less frequently addressed, even though they directly affect pouring dynamics.

Classical control strategies, such as PID and PD, remain predominant, while adaptive and learning-based methods, despite their potential to handle nonlinear dynamics and diverse container conditions, remain unexplored. The average pour error is the most common metric for evaluating pouring performance; however, other relevant measures, such as spill volume, have not been explored. Devices used to measure poured volume reveal a gap between direct methods, such as force sensors, and indirect perception-based techniques. Furthermore, many studies do not report any measurements.

The state of research on robotic liquid pouring is limited by the use of simplified liquids, containers, and metrics. This results in limited robustness and comparability across studies. Advancing the state of the art will require broader experimental conditions and standardized reporting. Integrating adaptive and multimodal approaches is also essential for improving accuracy and real-world applicability.

https://doi.org/10.3390/mca31010030

Author Contributions: Conceptualization, J.J.R.-M. and A.M.-H.; data curation, J.J.R.-M.; methodology, J.J.R.-M. and E.J.R.-R.; formal analysis, J.J.R.-M.; writing—original draft preparation, J.J.R.-M.; writing—review and editing, J.J.R.-M., A.M.-H. and E.J.R.-R.; supervision, A.M.-H. and E.J.R.-R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding, and the APC was funded by Universidad Veracruzana.

Data Availability Statement: No new data were created or analyzed in this study.

Acknowledgments: J.J.R.-M. thanks Mexican National Secretariat of Science, Humanities, Technology, and Innovation Secihti (Secretaría de Ciencia, Humanidades, Tecnología e Innovación) for funding his PhD studies (CVU number: 1142861).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Si, W.; Guan, Y.; Wang, N. Adaptive Compliant Skill Learning for Contact-Rich Manipulation With Human in the Loop. *IEEE Robot. Autom. Lett.* **2022**, *7*, 5834–5841. [CrossRef]
2. Paulius, D.; Huang, Y.; Meloncon, J.; Sun, Y. Manipulation Motion Taxonomy and Coding for Robots. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 5596–5601.
3. Sakib, M.S.; Paulius, D.; Sun, Y. Approximate Task Tree Retrieval in a Knowledge Network for Robotic Cooking. *IEEE Robot. Autom. Lett.* **2022**, *7*, 11492–11499. [CrossRef]
4. Castilla, R.; Gamez-Montero, P.; Raush, G.; Khamashta, M.; Codina, E. Numerical Study of Impingement Location of Liquid Jet Poured from a Tilting Ladle with Lip Spout. *Metall. Mater. Trans. B* **2017**, *48*, 1390–1399. [CrossRef]
5. Noda, Y.; Terashima, K. Nonlinear Modeling with Hydrodynamics and Flow Control Using Inverse Pouring Dynamics of Tilting-Ladle-Type Automatic Pouring Process. In Proceedings of the 67th World Foundry Congress, Harrogate, UK, 5–7 June 2006.
6. Sueki, Y.; Noda, Y. Development of Flow Rate Feedback Control in Tilting-ladle-type Pouring Robot with Direct Manipulation of Pouring Flow Rate. In Proceedings of the International Conference on Informatics in Control, Automation and Robotics, Prague, Czech Republic, 29–31 July 2019.
7. Tsuji, T.; Noda, Y. High-precision pouring control using online model parameters identification in automatic pouring robot with cylindrical ladle. In Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), San Diego, CA, USA, 5–8 October 2014; pp. 2563–2568. [CrossRef]
8. Noda, Y.; Terashima, K. Falling Position Control of Outflow Liquid for Automatic Pouring System with Tilting-Type Ladle. *IFAC Proc.* **2007**, *40*, 53–58. [CrossRef]
9. Zhu, H.; Yamakawa, Y. Robotic Pouring Based on Real-Time Observation and Visual Feedback by a High-Speed Vision System. *J. Robot. Mechatronics* **2022**, *34*, 965–974. [CrossRef]
10. Zhu, Y.; Fusano, K.; Aoyama, T.; Hasegawa, Y. Intention-reflected predictive display for operability improvement of time-delayed teleoperation system. *ROBOMECH J.* **2023**, *10*, 17. [CrossRef]
11. Pan, Z.; Manocha, D. Motion planning for fluid manipulation using simplified dynamics. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 7–14 October 2016; pp. 4224–4231.
12. Pan, Z.; Park, C.; Manocha, D. Robot Motion Planning for Pouring Liquids. In Proceedings of the International Conference on Automated Planning and Scheduling, London, UK, 12–17 June 2016.
13. Zhu, F.; Hu, S.; Leng, L.; Bartsch, A.; George, A.; Farimani, A.B. Pour me a drink: Robotic Precision Pouring Carbonated Beverages into Transparent Containers. *arXiv* **2023**, arXiv:2309.08892. [CrossRef]
14. Do, C.; Burgard, W. Accurate Pouring with an Autonomous Robot Using an RGB-D Camera. In *Intelligent Autonomous Systems 15*; Strand, M., Dillmann, R., Menegatti, E., Ghidoni, S., Eds.; Springer: Cham, Switzerland, 2019; pp. 210–221.
15. Cleaver, A.; Aronson, R.M.; Sinapov, J. Helping People Predict the Outcome of Robotic Pouring Behaviors with Augmented Reality. In Proceedings of the 7th International Workshop on Virtual, Augmented, and Mixed-Reality for Human-Robot Interactions, Boulder, CO, USA, 11–14 March 2024.
16. Wang, Z.; Tian, G.; Pan, H. Robot gaining robust pouring skills through fusing vision and audio. *ISA Trans.* **2023**, *135*, 428–437. [CrossRef]
17. Do, C.; Schubert, T.; Burgard, W. A probabilistic approach to liquid level detection in cups using an RGB-D camera. In *Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2016, Daejeon, Republic of Korea, 9–14 October 2016*; IEEE: Piscataway, NJ, USA, 2016; pp. 2075–2080. [CrossRef]

18. Babaians, E.; Sharma, T.; Karimi, M.; Sharifzadeh, S.; Steinbach, E. PourNet: Robust Robotic Pouring Through Curriculum and Curiosity-based Reinforcement Learning. In Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 23–27 October 2022; pp. 9332–9339. [CrossRef]
19. Wilson, J.; Sterling, A.; Lin, M.C. Analyzing Liquid Pouring Sequences via Audio-Visual Neural Networks. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 7702–7709. [CrossRef]
20. Matl, C.; Matthew, R.; Bajcsy, R. Haptic Perception of Liquids Enclosed in Containers. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 7142–7149. [CrossRef]
21. Xu, Y.; Ma, Y.; Lin, W.; Sun, Z.; Zhang, T.; Wang, Z. Robot Embodied Dynamic Tactile Perception of Liquid in Containers. In Proceedings of the 2024 IEEE-RAS 23rd International Conference on Humanoid Robots (Humanoids), Nancy, France, 22–24 November 2024; pp. 1034–1039. [CrossRef]
22. Piacenza, P.; Lee, D.; Isler, V. Pouring by Feel: An Analysis of Tactile and Proprioceptive Sensing for Accurate Pouring. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 10248–10254. [CrossRef]
23. Kouvaritakis, B.; Cannon, M. *Model Predictive Control: Classical, Robust and Stochastic*; Advanced Textbooks in Control and Signal Processing; Springer: Cham, Switzerland, 2016. [CrossRef]
24. Lopez-Guevara, T.; Taylor, N.K.; Gutmann, M.U.; Ramamoorthy, S.; Subr, K. Adaptable Pouring: Teaching Robots Not to Spill using Fast but Approximate Fluid Simulation. In *Proceedings of the 1st Annual Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017*; Levine, S., Vanhoucke, V., Goldberg, K., Eds.; Proceedings of Machine Learning Research (PMLR); PMLR: Mountain View, CA, USA, 2017; Volume 78, pp. 77–86.
25. Reyes-Montiel, J.; Marin-Hernandez, A.; Hernandez Mendez, S. A Geometric Approach for Partial Liquids' Pouring from a Regular Container by a Robotic Manipulator. In Proceedings of the 19th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Lisbon, Portugal, 14–16 July 2022; pp. 688–694. [CrossRef]
26. Dong, C.; Takizawa, M.; Kudoh, S.; Suehiro, T. Precision Pouring into Unknown Containers by Service Robots. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 5875–5882. [CrossRef]
27. Sieusankar, T.; Chandrasekaran, B. A Review of Current Techniques for Robotic Arm Manipulation and Mobile Navigation. In Proceedings of the 2023 IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 8–11 March 2023; pp. 1070–1075. [CrossRef]
28. Meattini, R.; Suárez, R.; Palli, G.; Melchiorri, C. Human to Robot Hand Motion Mapping Methods: Review and Classification. *IEEE Trans. Robot.* **2023**, *39*, 842–861. [CrossRef]
29. Freschi, C.; Ferrari, V.; Melfi, F.; Ferrari, M.; Mosca, F.; Cuschieri, A. Technical review of the da Vinci surgical telemanipulator. *Int. J. Med Robot. Comput. Assist. Surg.* **2013**, *9*, 396–406. [CrossRef]
30. Gentile, C.; Gruppioni, E. A Perspective on Prosthetic Hands Control: From the Brain to the Hand. *Prosthesis* **2023**, *5*, 1184–1205. [CrossRef]
31. Yamanobe, N.; Wan, W.; Ramirez-Alpizar, I.G.; Petit, D.; Tsuji, T.; Akizuki, S.; Hashimoto, M.; Nagata, K.; Harada, K. A brief review of affordance in robotic manipulation research. *Adv. Robot.* **2017**, *31*, 1086–1101. [CrossRef]
32. Elguea-Aguinaco, I.; Serrano-Muñoz, A.; Chrysostomou, D.; Inziarte-Hidalgo, I.; Bøgh, S.; Arana-Arexolaleiba, N. A review on reinforcement learning for contact-rich robotic manipulation tasks. *Robot. Comput.-Integr. Manuf.* **2023**, *81*, 102517. [CrossRef]
33. Paulius, D.; Sun, Y. A Survey of Knowledge Representation in Service Robotics. *Robot. Auton. Syst.* **2019**, *118*, 13–30. [CrossRef]
34. Nahavandi, S.; Alizadehsani, R.; Nahavandi, D.; Lim, C.P.; Kelly, K.; Bello, F. Machine learning meets advanced robotic manipulation. *Inf. Fusion* **2024**, *105*, 102221. [CrossRef]
35. Nate, I.; Wang, Z.; Hirai, S. Origami-Based Robotic Gripper for Transporting Solids with Liquids. In Proceedings of the 2024 IEEE/SICE International Symposium on System Integration (SII), Ha Long, Vietnam, 8–11 January 2024; pp. 251–256. [CrossRef]
36. Nguyen, H.; La, H. Review of Deep Reinforcement Learning for Robot Manipulation. In Proceedings of the 2019 Third IEEE International Conference on Robotic Computing (IRC), Naples, Italy, 25–27 February 2019; pp. 590–595. [CrossRef]
37. Paulius, D.; Huang, Y.; Milton, R.; Buchanan, W.D.; Sam, J.; Sun, Y. Functional object-oriented network for manipulation learning. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October 2016; pp. 2655–2662.
38. Arksey, H.; O'Malley, L. Scoping studies: Towards a methodological framework. *Int. J. Soc. Res. Methodol.* **2005**, *8*, 19–32. [CrossRef]

https://doi.org/10.3390/mca31010030

39. Tricco, A.C.; Lillie, E.; Zarin, W.; O'Brien, K.K.; Colquhoun, H.; Levac, D.; Moher, D.; Peters, M.D.; Horsley, T.; Weeks, L.; et al. PRISMA Extension for Scoping Reviews (PRISMA-ScR): Checklist and Explanation. *Ann. Intern. Med.* **2018**, *169*, 467–473. [CrossRef]
40. Xian, Z.; Zhu, B.; Xu, Z.; Tung, H.Y.F.; Torralba, A.; Fragkiadaki, K.; Gan, C. FluidLab: A Differentiable Environment for Benchmarking Complex Fluid Manipulation. *arXiv* **2023**, arXiv:2303.02346. [CrossRef]
41. Pareigis, S.; Hermosilla-Diaz, J.; Reyes-Montiel, J.; Maaß, F.; Marin-Hernandez, A.; Haase, H.; Mang, M. Offline Feature-Based Reinforcement Learning with Preprocessed Image Inputs for Liquid Pouring Control. In Proceedings of the 20th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Rome, Italy, 13–15 November 2023. [CrossRef]
42. Chen, T.; Huang, Y.; Sun, Y. Accurate Pouring using Model Predictive Control Enabled by Recurrent Neural Network. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 7688–7694. [CrossRef]
43. Kennedy, M.; Queen, K.; Thakur, D.; Daniilidis, K.; Kumar, V. Precise dispensing of liquids using visual feedback. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 1260–1266. [CrossRef]
44. Do, C.; Gordillo, C.; Burgard, W. Learning to Pour using Deep Deterministic Policy Gradients. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 3074–3079.
45. Lin, L.; Li, K.; Yue, S.; Tian, X.; Wei, N. Sensorimotor Control for Manipulating A Glass of Water Effects of Visual and Tactile Feedbacks. In Proceedings of the 2018 14th IEEE International Conference on Signal Processing (ICSP), Beijing, China, 12–16 August 2018; pp. 1015–1018. [CrossRef]
46. Camporredondo, G.; Barber, R.; Legrand, M.; Muñoz, L. A Kinematic Controller for Liquid Pouring between Vessels Modelled with Smoothed Particle Hydrodynamics. *Appl. Sci.* **2019**, *9*, 5007. [CrossRef]
47. Luo, X.; Jin, S.; Huang, H.J.; Yuan, W. An Intelligent Robotic System for Perceptive Pancake Batter Stirring and Precise Pouring. In Proceedings of the 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Abu Dhabi, United Arab Emirates, 14–18 October 2024; pp. 5970–5977. [CrossRef]
48. Cleaver, A.; Aronson, R.M.; Sinapov, J. Enhancing Users' Predictions of Robotic Pouring Behaviors Using Augmented Reality: A Case Study. In Proceedings of the 2024 33rd IEEE International Conference on Robot and Human Interactive Communication (ROMAN), Pasadena, CA, USA, 26–30 August 2024; pp. 1886–1891. [CrossRef]
49. Wang, Z.; Tian, G.; Guo, S. GMM Enabled by Multimodal Information Fusion Network for Detection and Motion Planning of Robotic Liquid Pouring. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *36*, 11943–11952. [CrossRef] [PubMed]
50. Zhang, X.; Liu, J.; Zhang, W. Multi-Level Robotic Pouring Using Offline Reinforcement Learning. In Proceedings of the 2024 IEEE International Conference on Robotics and Biomimetics (ROBIO), Bangkok, Thailand, 10–14 December 2024; pp. 1850–1855. [CrossRef]
51. Lin, H.; Fu, Y.; Xue, X. PourIt!: Weakly-supervised Liquid Perception from a Single Image for Visual Closed-Loop Robotic Pouring. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–3 October 2023; pp. 241–251. [CrossRef]
52. Saito, N.; Dai, N.B.; Ogata, T.; Mori, H.; Sugano, S. Real-time Liquid Pouring Motion Generation: End-to-End Sensorimotor Coordination for Unknown Liquid Dynamics Trained with Deep Neural Networks. In Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 6–8 December 2019; pp. 1077–1082. [CrossRef]
53. Narasimhan, G.; Zhang, K.; Eisner, B.; Lin, X.; Held, D. Self-supervised Transparent Liquid Segmentation for Robotic Pouring. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 4555–4561. [CrossRef]
54. Wu, T.Y.; Lin, J.T.; Wang, T.H.; Hu, C.W.; Niebles, J.C.; Sun, M. Liquid Pouring Monitoring via Rich Sensory Inputs. In Proceedings of the Computer Vision—ECCV 2018, Munich, Germany, 8–14 September 2018; Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., Eds.; Springer: Cham, Switzerland, 2018; pp. 352–369.
55. Sueki, Y.; Noda, Y. Experimental Verification of Real-Time Flow-Rate Estimations in a Tilting-Ladle-Type Automatic Pouring Machine. *Appl. Sci.* **2021**, *11*, 6701. [CrossRef]
56. Wang, C.; Hao, D.; Long, L. Design, simulation, control of a hybrid pouring robot: Enhancing automation level in the foundry industry. *Robotica* **2024**, *42*, 1018–1038. [CrossRef]
57. Kabasawa, N.; Noda, Y. Model-Based Flow Rate Control with Online Model Parameters Identification in Automatic Pouring Machine. *Robotics* **2021**, *10*, 39. [CrossRef]
58. Li, L.; Wang, C.; Wu, H. Research on Kinematics and Pouring Law of a Mobile Heavy Load Pouring Robot. *Math. Probl. Eng.* **2018**, *2018*, 8790575. [CrossRef]
59. Huang, Y.; Wilches, J.; Sun, Y. Robot Gaining Accurate Pouring Skills through Self-Supervised Learning and Generalization. *arXiv* **2020**, arXiv:2011.10150. [CrossRef]

60. Nishio, Y.; Ogawa, T.; Niwa, K.; Chiba, H. Experimental and numerical study on liquid pouring from a beverage can. *J. Food Eng.* **2021**, *291*, 110237. [CrossRef]
61. Zhang, D.; Li, Q.; Zheng, Y.; Wei, L.; Zhang, D.; Zhang, Z. Explainable Hierarchical Imitation Learning for Robotic Drink Pouring. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 3871–3887. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

https://doi.org/10.3390/mca31010030

Mathematical and Computational Applications

MDPI

Article

HGREncoder: Enhancing Real-Time Hand Gesture Recognition with Transformer Encoder—A Comparative Study

Luis Gabriel Macías *, Jonathan A. Zea, Lorena Isabel Barona, Ángel Leonardo Valdivieso and Marco E. Benalcázar

Artificial Intelligence and Computer Vision Research Lab, Departamento de Informática y Ciencias de la Computación, Escuela Politécnica Nacional, Quito 170143, Ecuador; jonathan.zea@epn.edu.ec (J.A.Z.); lorena.barona@epn.edu.ec (L.I.B.); angel.valdivieso@epn.edu.ec (Á.L.V.); marco.benalcazar@epn.edu.ec (M.E.B.)
* Correspondence: luis.macias@epn.edu.ec

Abstract: In the field of Hand Gesture Recognition (HGR), Electromyography (EMG) is used to detect the electrical impulses that muscles emit when a movement is generated. Currently, there are several HGR models that use EMG to predict hand gestures. However, most of these models have limited performance in real-time applications, with the highest recognition rate achieved being 65.78 ± 15.15%, without post-processing steps. Other non-generalizable models, i.e., those trained with a small number of users, achieved a window-based classification accuracy of 93.84%, but not in time-real applications. Therefore, this study addresses these issues by employing transformers to create a generalizable model and enhance recognition accuracy in real-time applications. The architecture of our model is composed of a Convolutional Neural Network (CNN), a positional encoding layer, and the transformer encoder. To obtain a generalizable model, the EMG-EPN-612 dataset was used. This dataset contains records of 612 individuals. Several experiments were conducted with different architectures, and our best results were compared with other previous research that used CNN, LSTM, and transformers. The findings of this research reached a classification accuracy of 95.25 ± 4.9% and a recognition accuracy of 89.7 ± 8.77%. This recognition accuracy is a significant contribution because it encompasses the entire sequence without post-processing steps.

Keywords: EMG; encoder; HGR; self-attention; transformers

1. Introduction

Communication and control of electronic devices through Hand Gesture Recognition (HGR) are fundamental aspects of human–machine interaction. Gestures represent a natural and effective way to convey information and commands in various contexts. Therefore, HGR has a wide range of applications, such as surgery, video gaming, and sign language translation, among others [1,2].

However, implementing HGR systems presents several challenges, particularly for individuals who have experienced limb loss [3]. Hand gestures can be recognized using images, videos, or surface electromyographic (sEMG) signals, among others [4–8]. Although image or video-based approaches are common, they require the physical presence of the hand or limb to capture movement, making them unsuitable for amputees . In contrast, sEMG signals offer a key advantage: they can still be detected in individuals with partial or total hand amputation. This is possible because sEMG measures the electrical

activity generated by muscles in response to neural stimulation, regardless of whether the limb is present or not [9–11].

These signals are used in the HGR to predict hand gestures. Using EMG signals to create an HGR model is a good option in cases of mid- or low-transradial amputations [3]. Despite the existence of many models that predict movement using EMG, a functional model has not yet been achieved. Existing models have difficulty predicting movement, especially in real-time, because EMG signals vary between people and between different people (that is, intrapersonal and interpersonal variations) [9,12,13].

Transformers are a type of Artificial Neural Network (ANN) that has proven successful in Natural Language Processing (NLP) due to its ability to consider context [14]. The context in an EMG signal is useful for predicting a gesture in real-time. The action of performing a gesture is important before, during, and after its execution [15,16]. Furthermore, transformers are based on attention mechanisms. Attention mechanisms can mitigate the problem of variation within a person and between different people.

However, transformer models have been trained on datasets involving only 40 or fewer, which is a relatively small number [6,15,17]. The use of a limited number of participants affects the generalizability of the model [11]. A generalizable and higher accuracy HGR model is crucial to ensure smooth interactions with electronic devices. To address these challenges, transformers are used in this research along with EMG-EPN-612 dataset [18], which contains records of 612 individuals. This research aims to achieve the following objectives:

1.1. Research Objective

Develop a real-time HGR model for five right-hand gestures using a transformer neural network and EMG signals.

1.2. Specific Objectives

1. Analyze the literature on neural network architectures for HGR, focusing on models that have employed transformer neural networks over the past five years, in order to gain a deeper understanding of the existing scientific literature on this topic.
2. Develop a real-time HGR model for five right-hand gestures using a transformer neural network, using the EMG-EPN-612 dataset to address generalization challenges, remove spurious labels, and improve recognition accuracy.
3. Compare the performance of the proposed model with existing models in the HGR field, evaluating their classification and recognition accuracy.

1.3. Hypothesis

A HGR model that uses transformers neural networks can have higher accuracy and better real-time recognition than existing models that use other neural network architectures.

1.4. Problems Addressed by the Study

This study develops a HGR model that uses transformers with attention mechanisms, significantly improving real-time recognition accuracy, eliminating the need for post-processing, and reducing variability and classification errors present in other approaches. Additionally, the use of the EMG-EPN-612 dataset [18], which includes a substantial number of volunteers, helps create a generalizable model, overcoming the limitation of small datasets used in previous studies.

2. Literature Review

Several methods have emerged to capture data related to hand gestures. The methods for capturing hand gesture data can be classified into the following two groups: invasive and non-invasive methods [19].

- Invasive methods stand out for their ability to detect the electromyographic signal of a specific muscle, which allows muscle movements to be monitored with great precision. Invasive methods necessitate the insertion of intramuscular electrodes, which demands specialized skills from the installer [19,20].
- Non-invasive methods are those that do not require penetrating the person's skin. Unlike invasive methods, non-invasive methods are easy to use and apply, making them easy to use even for people with little knowledge [19].

Surface electromyography (sEMG) records muscle activity by placing electrodes on the skin and is a non-invasive method. For this reason, in (HGR), sEMG is commonly used to monitor muscle activity, as shown in [9]. Despite its advantages, sEMG faces a significant challenge related to variability. This variability can be both intrapersonal and interpersonal, as follows [21,22]:

- Interpersonal variability is associated with factors such as fatigue, mood, sensor repositioning, and other aspects [23–26].
- Intrapersonal variability refers to differences in the EMG signal produced by one person compared to another. This variability can be attributed to different factors, such as physical constitution, body mass index, fat level, gender, age, among other aspects [27,28].

2.1. Machine Learning Models for Hand Gesture Recognition

Considering the above, a considerable amount of research has been conducted to develop specific or general models in the field of HGR with sEMG. Several machine learning models have been developed, including Support Vector Machines (SVM), Hidden Markov Models (HMM), Decision Trees, Random Forests, and Artificial Neural Networks (ANN) [9,29]. Specific models are designed by an individual with their or her own data, so they tend to have good accuracy, between 94% and 97%. In other research, a CovGCN architecture was proposed for sEMG-based HGR [30]. This method demonstrated a classification accuracy of 88.02% with a CovGCN model that dynamically refines topologies for each hand gesture. The following two datasets were used in this research: the UWB-Gestures dataset with 8 different subjects and the Ninapro DB2 dataset with 40 subjects.

General models are trained on a multi-user dataset and can be used by several users without additional training [31]. For general models, the results are less favorable, as it is shown in Table 1. According to Chung et al. in their research [32] used ANN to achieve a accuracy recognition of 85.08 ± 15.21%. For their part, Jaramillo et al. in [33] used SVM to achieve a result of 87.5%. Finally, Barona et al. in [16] achieved results of 87.26 ± 11.14% using CNN and 90.55 ± 9.45% using LSTM. It is necessary to note that all of these results stated above use post-processing techniques to calculate their recognition accuracy. Then, machine learning offers improved performance, although the results are still not perfect.

Table 1. Accuracy recognition for HGR general models.

Model	Recognition Accuracy	Variability	Number of Weights
ANN	85.08%	±15.21	70.9 K
SVM	87.5%	–	–
CNN	87.26%	±11.14	219.1 K
LSTM	90.55%	±9.45	11.7 M

2.2. Transformers in Hand Gesture Recognition

In 2017, a machine learning algorithm known as "transformers" was developed. Transformers were initially designed for NLP tasks, as described in the article "Attention is all you need" [14]. This architecture uses attention mechanisms that are part of an encoder and a decoder. In several studies, it has been confirmed that using only the encoder is sufficient to perform classification tasks [34,35]. In the field of HGR with sEMG, research has also been conducted using only the transformer encoder [6,15,36].

In the research conducted by Zabihi et al. in [15]. A model was created with hybrid architecture based on a transformers encoder combined with CNN. This model (TraHGR) was developed using data from 40 individuals with nine gestures, achieving a classification accuracy of 93.84%. On the other side, Montazerin et al. conducted two investigations using transformers in their architecture: ViT-HGR and CT-HGR [6,36]. Montazerin et al. used a variant Vision Transformer (ViT) architecture in both cases (ViT-HGR and CT-HGR) and employed two grids each comprising 64 electrodes to capture sEMG signals while individuals performed 66 distinct hand gestures. These data were collected from a group of 20 individuals. The best model (CT-HGR) achieved a classification accuracy of 91.98 ± 2.22%.

Models using the transformer architecture show better performance than previous machine learning models. This good result is attributed to the transformer's ability to process data in sequential blocks, as noted in [15]. Processing data in sequential blocks reduces spurious labels. Spurious labels are incorrectly assigned values in a classification vector, as explained by Benalcázar et al. in their research [37]. In other words, a classification vector is valid when a user has their hand at relax, performs a gesture for a certain amount of time, and then returns to relax. Therefore, the gesture performed at execution time should not have a different classification label. This different classification label is called a spurious label, as it is shown in Figure 1.

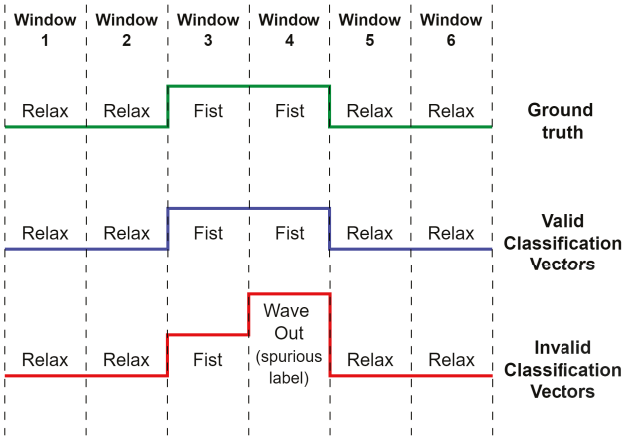


Figure 1. Illustration of valid and invalid classification vectors.

2.3. Research Gap

Transformer models have managed to mitigate the problem of false labels. However, the transformer models mentioned above (TraHGR, ViT-HGR and CT-HGR) are specific models. The results obtained cannot be generalized due to the use of relatively small datasets [31]. For this reason, we used transformers in our research with the [18] EMG-EPN-612 dataset. This architecture uses sequences of blocks that produce fewer spurious labels. The model achieved is generalizable since it has been trained, validated and tested with data from 612 individuals.

Using a larger dataset improves the generalization of the model and reduces spurious labels, as the transformer architecture we employ works with block sequences, which optimizes the model's performance by better handling variability in gestures.

Additionally, our research reduces the need for post-processing techniques, which are necessary in many previous models to correct false labels and inconsistencies in gesture classification. Thanks to the use of CNN for feature extraction and transformers, our model does not rely on these additional steps, significantly improving its real-time performance. This ability to process data quickly and accurately, without extra steps, is essential for its implementation in real-world applications, where efficiency is crucial.

3. Materials and Methods

3.1. Dataset

A common challenge in many HGR models is the use of small datasets. For a model to be generalizable, it needs to be trained with a significant amount of data. Therefore, we have used the "EMG-EPN-612" dataset [18]. This dataset is publicly available for use in similar research.

3.1.1. Origin of the Dataset

The dataset consists of a collection of EMG signals from 612 individuals, of which 66% were male and the remaining 34% were female. This dataset is designed for training, evaluating, and testing HRG models. It was created by the Artificial Intelligence and Computer Vision Research Lab from the Escuela Politécnica Nacional [38].

The EMG-EPN-612 dataset was divided into two groups, each group has 306 individuals. One group was intended for training and validating the model, while the other group was intended for testing the classification and recognition accuracy of the created model. Furthermore, each gesture in each group consists of 50 EMGs, including the hand relaxed state. Consequently, each individual has a total of 300 EMGs, and there are a total of 91,800 EMGs in each group. This is illustrated in Figure 2:

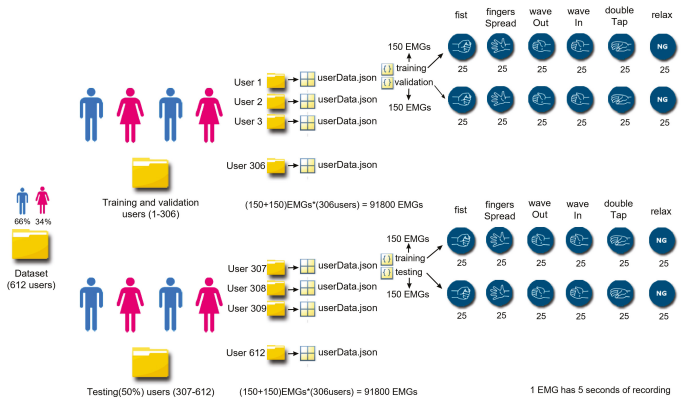


Figure 2. Overview of the EMG-EPN-612 dataset structure [18].

3.1.2. Characteristics of the Dataset

The EMGs were captured using six EMG signal collection device, which was placed on the volunteers' right forearms, as shown in Figure 3a. The volunteers performed five hand gestures: pinch, wave out, wave in, fist, and open, as it is shown in Figure 3b. The recording time for each gesture was 5 s. In addition, during the same time, EMG of the relaxed hand called "noGesture" was recorded.

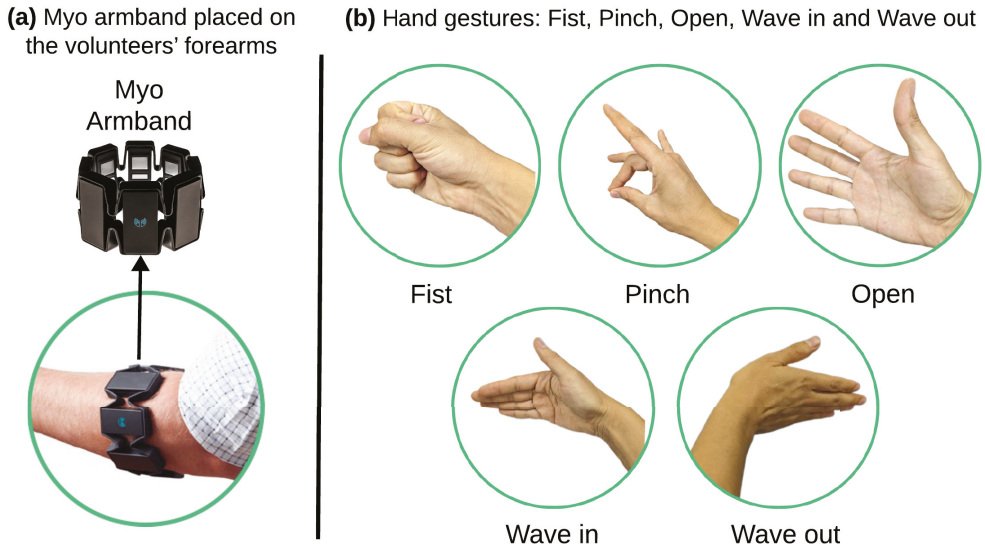


Figure 3. EMG data capture setup and hand gesture performance using an EMG signal collection device.

3.1.3. Comparison with Other Datasets

Compared to other widely used datasets such as NinaPro [39], which is one of the most recognized databases in the HGR domain, EMG-EPN-612 offers features that support the development of generalizable models [18]. Although NinaPro includes a greater variety of gestures, even its most comprehensive versions include no more than 78 participants, which limits its utility for training and evaluating models with high generalization capacity [39]. In contrast, EMG-EPN-612 contains recordings from 612 individuals, making it the largest known public dataset in terms of user count.

Moreover, the dataset was carefully designed with a balanced division: 306 users with ground truth labels are used for training and validation, while the remaining 306 users—whose ground truth is not provided—are reserved for final testing [12]. This allows for a clear and unbiased evaluation of the model's generalization ability and potential overfitting. It is worth noting that this same division strategy has been adopted in previous studies, as indicated in references [12,16,40,41].

3.2. Data Preprocessing

The raw data for all 612 individuals include the EMG signal in 8 channels for each of the hand gestures. In data preprocessing, the entire dataset containing the emg signals is discretized and normalized to facilitate its conversion into spectrograms. The preprocessing process of an EMG signal is detailed below, represented by the following:

$$\mathbf{E}(t) = \begin{bmatrix} E_1(t) \\ E_2(t) \\ \vdots \\ E_8(t) \end{bmatrix}^T \tag{1}$$

The EMG signal is divided into overlapping windows with a selected window size, $|W| = 300$ sample points (1500 ms), as a design criterion, and a stride of 45 (225 ms), as it is shown in Figure 4a. The formal representation of W at time instant n of the signal for 8 channels is given by the following:

$$\mathbf{S}_t = \begin{bmatrix} \mathbf{E}(n-299) \\ \vdots \\ \mathbf{E}(n) \end{bmatrix} \in [-1,1]^{300\times 8} \tag{2}$$

Following the segmentation process, we rectify the signal S_t by taking its absolute value $|S_t|$. This rectification step is crucial to prevent the mean in each channel of S_t becomes zero [42]. Subsequently, we apply a digital low-pass Butterworth filter, denoted as Ψ, to the rectified signal $|S_t|$ in order to smooth the signal and reduce noise. The Butterworth filter has a fifth order and a cutoff frequency of 10 Hz, this is a design criterion. The resulting filtered signal is denoted as $\Psi(|S_t|)$, referred to as S_Ψ, as you can see in Figure 4b. Then, we utilize an internal sliding window, denoted as L, within a window W with a width of 24 sample points , denoted as $|L| = 24$, and an overlapping region of 12 sample points, denoted as $|d| = 12$, as it is shown in Figure 4c.

Once the signal has been rectified and filtered, the spectrograms are calculated using the Short-time Fourier transform (STFT). According to Tsai et al. in [43], the STFT is used to transform the EMG signal from the time domain to the frequency-time domain. This process is necessary to later perform feature extraction. In our case, using STFT, we obtain as a result a matrix denoted as follows:

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{|L|} \end{bmatrix} \in \mathbb{C}^{13\times 24} \tag{3}$$

The value $|L|$ corresponds to the number of columns in $\mathbf{X}$. It is important to note that $\mathbf{X}$ can also be expressed as follows:

$$\mathbf{X} = A + Bi \tag{4}$$

Encompassing the real and imaginary components of the STFT. This entire process can be visualized in Figure 4d. We generate the spectrogram, denoted as follows:

$$M_i = |\mathbf{X}|^2 \tag{5}$$

Calculating the squared magnitude of the signal, as depicted in Figure 4e. We calculate the spectrogram for each of the 8 channels individually, as shown in Figure 4f. Finally, we combine the spectrograms of all channels to create a tensor, as follows:

$$\mathbf{\Lambda} = (\mathbf{M}_1, \mathbf{M}_2, \ldots, \mathbf{M}_8) \tag{6}$$

This tensor serves as input for the feature extraction process, as it is shown in Figure 4g.

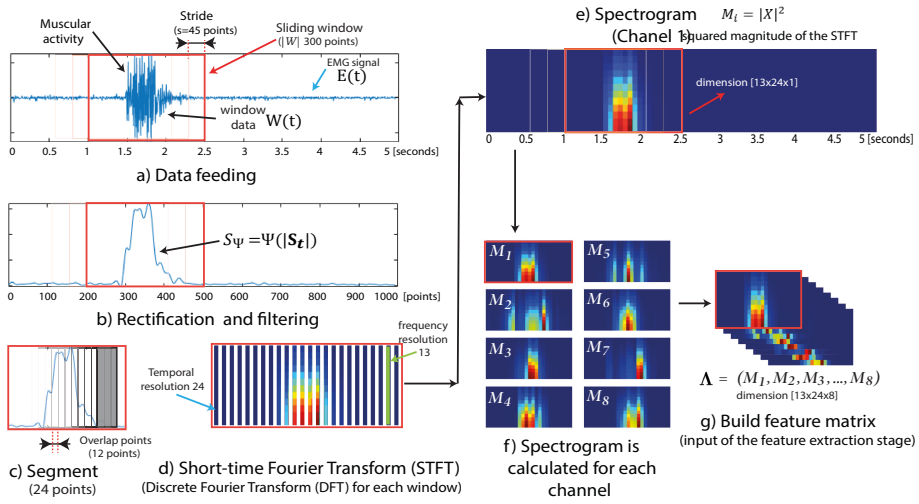


Figure 4. Preprocessing of EMG signals: Application of STF to convert raw data into spectrograms. This figure is a modified version of [16].

3.3. Feature Extraction

As mentioned in the data preprocessing Section, the signal in the time domain was transformed into a time-frequency domain signal using STFT. This resulting signal was processed with a Convolutional Neural Network (CNN). CNNs automatically extract and learn relevant features from the input data [44]. In our study, we employ the CNN architecture for feature extraction as part of preprocessing as it has shown considerable potential to improve overall performance compared to models that do not use CNN for preprocessing [15].

The feature extraction was performed based on the previous experiences and empirical experimentation of Szegedy et al. in their article defining "Inception modules" [45]. The structure that will be used for feature extraction is as follows: 6 convolutional layers, each layer followed by a RELU activation function and with residual networks, as it is shown in Figure 5a. Within each CNN layer, the following settings were used: 4 blocks of 2D convolution of 1×1, 1 block of 3×3 convolution, and one block of 5×5 convolution, and finally a concatenation layer, as it is shown in Figure 5b.

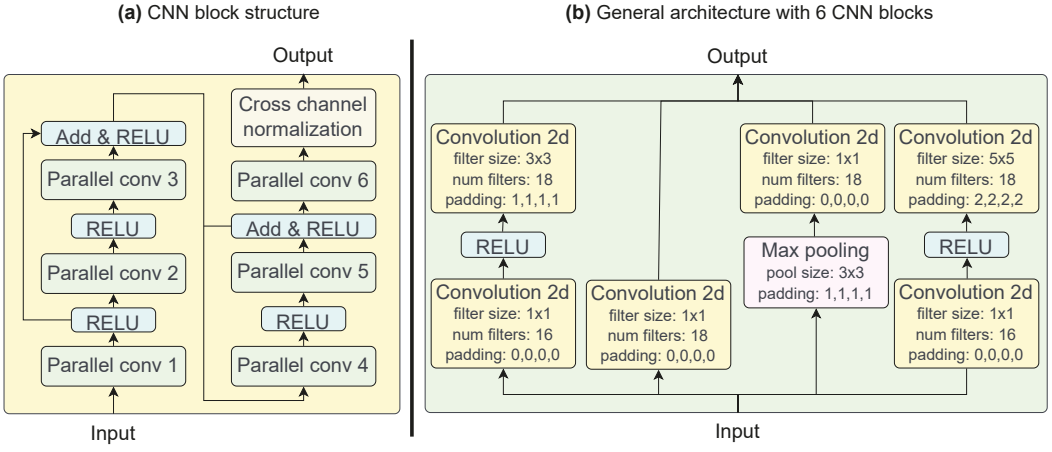


Figure 5. Preprocessing of EMG signals: Application of STF to convert raw data into spectrograms.

3.4. Architecture

3.4.1. Overview of Transformer Architecture

Transformers is a type of ANN effective in NLP, computer vision (CV), and speech recognition [34,35]. Transformer was designed by Vaswani et al. 2017 in the paper "Attention is All You Need" [14]. Transformer architecture has 4 main components, as illustrated in Figure 6a. These main components are:

- **Input Embedding:** Involves representing input tokens as vectors in a high-dimensional space.
- **Positional Encoding:** Adds positional information to the input sequence.
- **The Encoder:** Processes the signal and transforms it into an abstract representation.
- **The Decoder:** Uses this representation to generate an output sequence.

Both the encoder and the decoder employ attention mechanisms. It is necessary to emphasize that in our research we only use the encoder because the output is not a sequence. Our output is a value between 0 and 5, including the "noGesture" class, where each number represents a gesture. Furthermore, we do not use input embedding because our input is already a numeric vector.

3.4.2. Positional Encoding

Positional encoding is a sinusoidal function that is added to each sequence to provide positional information. Positional encoding is necessary because the transformer receives the entire sequence to process it comprehensively, without distinction [14]. To preserve the relative positions of the sequence, a geometric progression is employed. The wavelengths of these sinusoidal signals range from 2π to $10,000 \times 2\pi$ expressed as follows:

$$\mathrm{PE}_{(pos,2i)} = \sin\left(\frac{pos}{10,000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{7}$$

$$\mathrm{PE}_{(pos,2i+1)} = \cos\left(\frac{pos}{10,000^{\frac{2i}{d_{\text{model}}}}}\right) \tag{8}$$

The choice of a geometric progression for the wavelengths is based on [14]. It allows the model to capture different scales of positional information effectively. The lower frequencies, represented by smaller wavelengths near 2π, contribute to encoding broader positional relationships, while the higher frequencies, with larger wavelengths around $10,000 \times 2\pi$, capture finer-grained positional details.

3.4.3. Encoder

The encoder consists of a series (n) identical layers. Each layer has two parts: a multi-head attention layer and a fully connected layer, as can be seen in Figure 6b.

The multi-head attention layer helps the encoder to focus on different parts of the input, capturing long-term relationships between words. The fully connected layer is a connected network applied independently to each input position. After each layer, "layer normalization" is applied to stabilize the training. Additionally, residual connections are used to avoid the vanishing gradient problem [14].

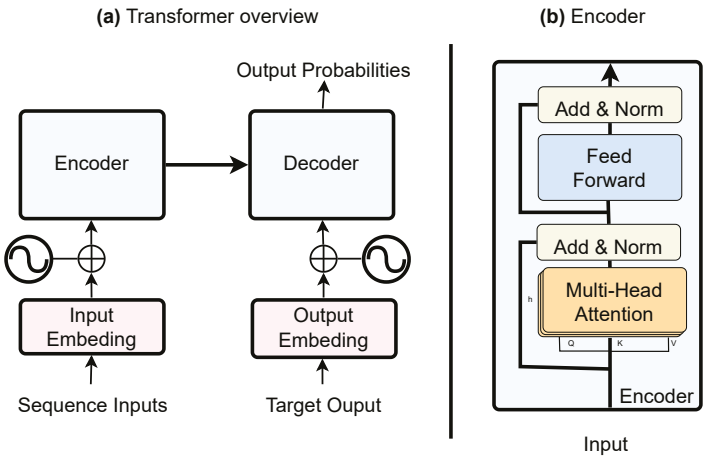


Figure 6. Transformer and encoder architecture.

3.4.4. Multi-Head Attention

Multi-head attention allows the model to pay attention to different parts of the input simultaneously. It performs multiple attention calculations simultaneously and then combines them into a single output. The expression for multi-head attention is as follows:

$$\mathbf{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Concat}(\mathbf{head}_1, \ldots, \mathbf{head}_h)\mathbf{W}^O \tag{9}$$

In this formula, $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ represent the query, key, and value matrices. These matrices are transformed into h different sub-spaces using the projection matrices, denoted as $\mathbf{QW}_i^Q$, $\mathbf{KW}_i^K$ and $\mathbf{VW}_i^V$. Then, a weighted attention calculation is performed for each subspace using the following formula:

$$\mathbf{head}_i = \text{Attention}(\mathbf{QW}_i^Q, \mathbf{KW}_i^K, \mathbf{VW}_i^V) \tag{10}$$

$\mathbf{W}$ is a parameter matrix projected from the matrices $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ for the i-th attention head. The weighted attention function (also called in this particular case "Scaled Dot-Product Attention") is defined as follows:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{QK}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{11}$$

where d_k represents the dimension of the key projection, and "$softmax$" is the function that normalizes the attention weights. The division by $\sqrt{d_k}$ is used to maintain stability during model training. Finally, the results of the weighted attention are combined and transformed again using a projection matrix $\mathbf{W}^O$, producing the final output.

3.5. Customization for Hand Gesture Recognition

To meet the specific demands of this paper, we tailored the transformer architecture with several adjustments. This study focused exclusively on the use of the transformer encoder. Encoder has proven sufficient for classification tasks, as demonstrated in previous research [34,35]. Therefore, after obtaining the output from the CNN feature extraction process, the signal was flattened and used as input for the Transformer encoder. The developed Transformer encoder consists of the following stages:

1. A positional encoding layer that assigns an additional value to each position in the sequence.

2. The processed sequence then passes through 8 heads (multi-head attention). The different details of the sequence will be attended simultaneously by each attention head.
3. After the attention stage, the sequence moves through a fully connected layer.
4. Between the layers of multi-head attention and fully connected layer, normalization is applied to both the inputs and outputs of these layers, mitigating issues such as overflow and gradient vanishing, ensuring more efficient convergence.
5. Finally, a Softmax function is applied to the 6 possible gesture categories, including the "noGesture" class.

This process culminates in the determination of the resulting gesture class, as it is shown in Figure 7.

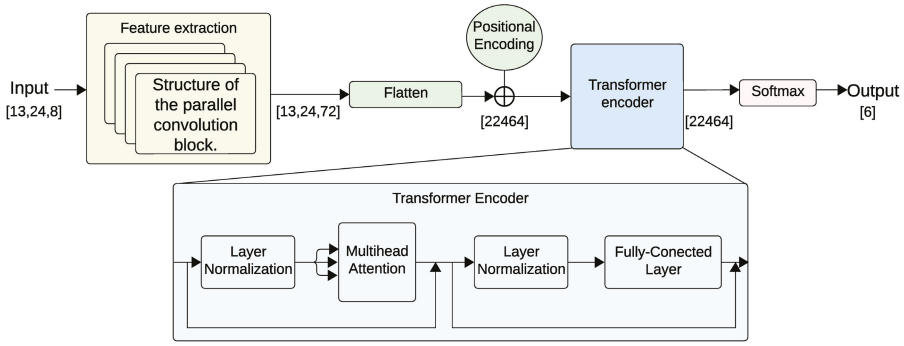


Figure 7. Architecture of the HGR model.

3.6. Training and Validation

The EMG-EPN612 dataset is composed of two groups for training and testing. Each group has a signal of 306 volunteers, as mentioned in the dataset Section. The training group were subdivided into two subgroups for training and validation. The two subgroups were converted into spectrograms. Training was carried out in MATLAB (R2025a) using deep learning toolbox. The training parameters were set as shown in Table 2:

Table 2. Training parameter configuration.

Parameter	**Value**
Initial Learning Rate	0.001
Learning Rate Scheduling	piecewise
Learning Rate Drop Factor	0.2
Learning Rate Drop Period	8
MiniBatch Size	64
Number of Epochs	10

These training parameters were based on the model developed in [16]. Using the training parameters mentioned above, a graph of the learning process was obtained and the confusion matrix was constructed for a detailed analysis of the classification results.

3.7. Post-Processing

The post-processing technique applied was the Voting Ensemble [46]. This method compares the previous three labels, assigning the value of the most frequent label to all spurious label. This substitution was conducted from left to right, ignoring "NoGesture"

labels. This process reduces spurious labels in classification predictions, as it is shown in Figure 8.

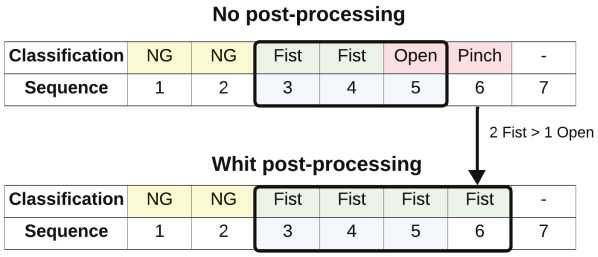


Figure 8. Post-processing technique.

Classification and recognition were performed on both the training and testing datasets. In both cases the precision was calculated. It is important to note that the test dataset was not used during the training process. This practice ensures reliable evaluation of the model's generalization performance on unknown data and avoids overfitting to the test dataset.

4. Results

The results obtained and their analysis are detailed below. We have focused on classification and recognition accuracy metrics, along with their respective standard deviations, overlap, and processing time.

4.1. Evaluation Metrics

The developed model has six classification classes (including "NoGesture") and each gesture is performed in an interval of 5 s. This model will be evaluated taking into account the following metrics:

- Prediction Vector Validity: The prediction vector is considered valid only when it does not present discontinuities throughout its classification. That is, it does not have spurious labels.
- Classification Accuracy: Classification accuracy refers to the proportion of gestures correctly classified by the model.
- Recognition Accuracy: Recognition accuracy goes beyond classification. Recognition is based on the presence of a valid prediction vector, which maintains a single designated label over time.
- Standard Deviation: It is a statistical measure of dispersion used to understand the variability present in a dataset. In this case, the standard deviation refers to the variability between the users' mean values.
- Overlap: The overlap is a percentage value. This value represents the percentage by which the prediction vector overlaps with the ground truth at a given time. Ground truth refers to the manually labeled actual values that serve as a reference for evaluating the accuracy of the model's predictions.
- Processing Time: Records the time it takes for the model to process and assign a label to a gesture. This metric is essential to evaluate the model in real-time. In this work, a model must respond in less than 300 ms for its response to be considered real-time according to [47].

These metrics provide a comprehensive assessment of both the effectiveness and consistency of the gesture recognition model, enabling comparison with other research.

4.2. Training Results

Initially, several tests were conducted without using positional encoding, where classification scores remained below 70%, despite experimenting with different numbers of heads and channels. Subsequently, positional encoding was incorporated, leading to notable improvements in performance. Following this adjustment, several training sessions were conducted with different numbers of heads, namely 4, 8, and 16. The number of channels for **Q**, **K**, and **V** was also varied, using channel numbers of 16, 32, 64, 128, and 256, as it is shown in Figure 9. The final result was achieved with 8 heads and 128 channels. It is noteworthy that the number of channels must be divisible by the number of heads since they are concatenated, as indicated in the multi-head attention Section.

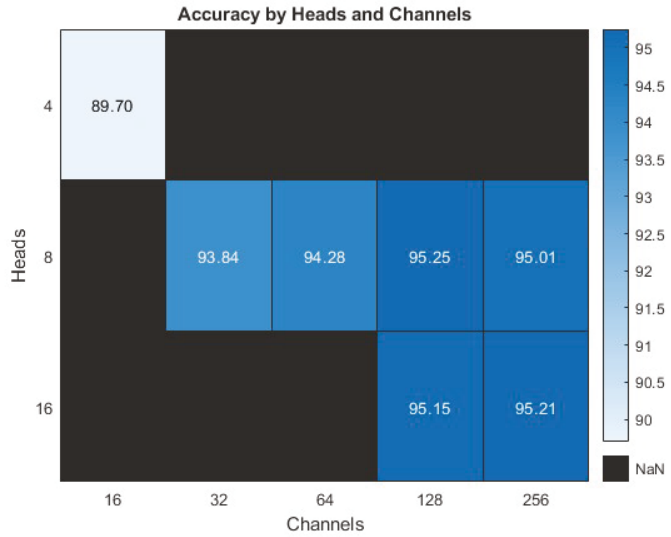


Figure 9. Heatmap of performance results with different numbers of heads and channel.

Figure 10 illustrates the training progress and loss of the best model obtained, which completed 7170 iterations in 10 epochs, with a total time of 152 min. The classification accuracy by window provided by Matlab Deep Learning Toolbox was 96.56%. The result was obtained through cross-validation using the training dataset. The learning curve exceeds 90% in 2000 iterations.

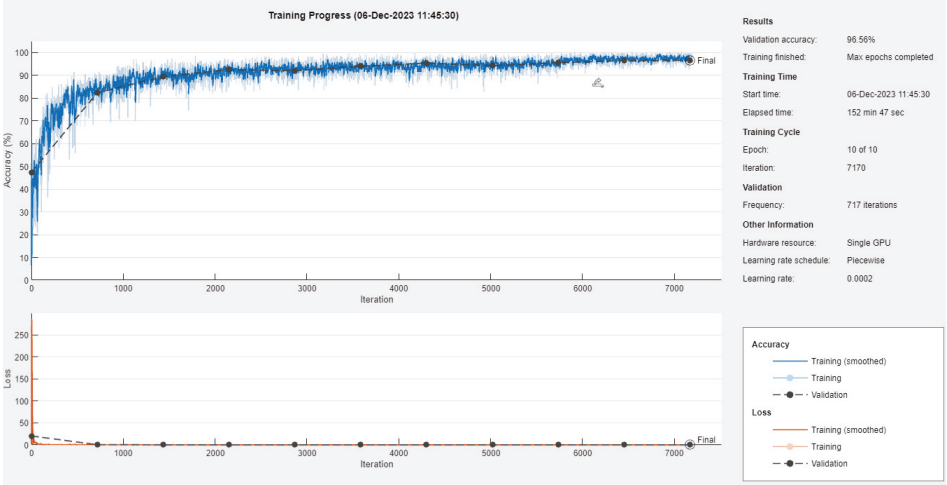


Figure 10. Training progress (**top**) and cost function (**bottom**) of the obtained model.

Figure 11 presents the confusion matrix corresponding to the 6 gestures classified by the model. The matrix highlights the correct predictions diagonally and provides insights into the model's performance across different gesture categories.

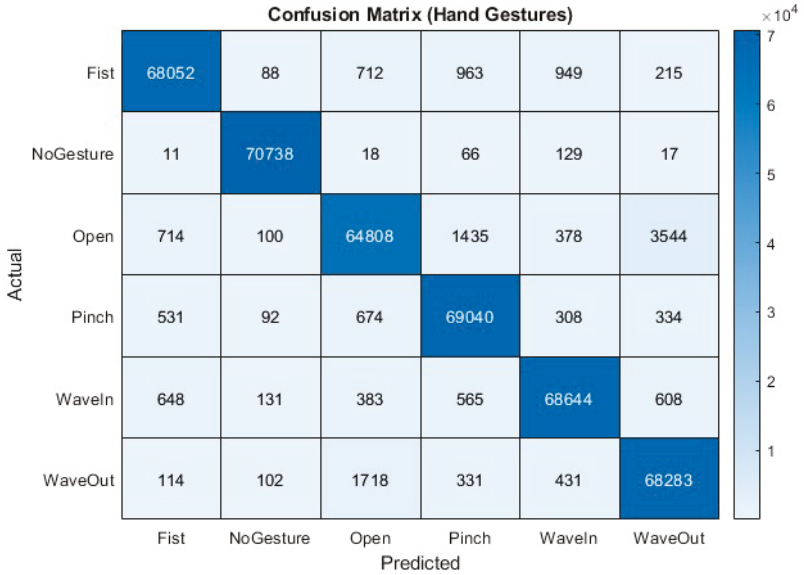


Figure 11. Heatmap of confusion matrix for hand gesture classification by window.

Table 3 contains performance metrics such as precision, recall, F1 score, and specificity [48]:

- Precision: Proportion of true positive predictions among all positive predictions. Measures accuracy when the model predicts a positive class.
- Recall: Proportion of true positive predictions among all actual positives. Measures the model's ability to capture all positive instances.
- F1 Score: Harmonic mean of precision and recall. Balances precision and recall, providing a single metric.
- Specificity: Proportion of true negative predictions among all actual negatives. Measures accuracy when the model predicts a negative class.

As can be seen in Table 3, the model is more accurate with the "NoGesture" gesture. This fact must be taken into account when interpreting the results, as it could bias them. For this reason, the analysis of the results is supported by other metrics such as overlap and recognition accuracy.

Table 3. Performance metrics per class for the hand gesture classification by window model.

Class	Precision	Recall	F1 Score	Specificity
Fist	97.12%	95.88%	96.49%	99.43%
NoGesture	99.28%	99.66%	99.47%	99.86%
Open	94.87%	91.31%	93.05%	99.01%
Pinch	95.36%	97.27%	96.30%	99.05%
WaveIn	96.90%	96.71%	96.81%	99.38%
WaveOut	93.54%	96.20%	94.85%	98.67%

4.3. Test Results

The test results were obtained using the test dataset. Table 4 shows that the classification by window is the same with and without post-processing. The classification is the same in both cases because the classification is used to improve recognition, as already explained in the post-processing Section 4. For this reason, only the results in terms of recognition accuracy increase by 2% and their standard deviation is reduced from 8.77% to 8.2%. Furthermore, it is important to note that the classification accuracy provided by Matlab (96.56%) on the training results is calculated window by window. This result differs from the classification accuracy applied during testing (95.25%), as the latter is based on the classification of the entire signal.

Table 4. Test results with and without post-processing.

Metric	No Post-Processing	With Post-Processing
Classification	$95.25 \pm 4.9\%$	
Recognition	$89.7 \pm 8.77\%$	$91.7 \pm 8.2\%$
Overlapping	$49.8 \pm 8.27\%$	$50.5 \pm 8.35\%$
Time (ms)	151 ± 49.2	167 ± 50.8

4.4. Discussion

In terms of classification accuracy, the transformer model developed in this study (HGREncoder) has outperformed all the models indicated in the literature review, which are: CNN, LSTM, CT-HGR and TraHGR. HGREncoder achieved a classification accuracy by windows of 95.25 ± 4.9%.

Figure 12a shows the comparison of the four mentioned models (CNN, LSTM, CT-HGR and TraHGR) and HGREncoder. Firstly, it will be discussed in terms of classification accuracy between CNN, LSTM and HGREncoder models, and secondly it will be discussed in terms of classification accuracy between CT-HGR, TraHGR and HGREncoder models.

In the first case, the comparison is balanced because CNN, LSTM, and HGREncoder models used the same dataset. CNN (in purple in Figure 12) with an accuracy of 90.62 ± 9.58% and LSTM (in yellow in Figure 12) with an accuracy of 92.02 ± 2.22% achieved a lower classification accuracy and a lower standard deviation compared to HGREncoder (in red in Figure 12). The reduction in the standard deviation indicates that a more consistent and improved model was obtained.

In the second case, CT-HGR and TraHGR (in blue in Figure 12a) are transformer-based models. These models (CT-HGR and TraHGR) were developed to classify a larger number of gestures (9 and 66 gestures, respectively) and were tested on a smaller dataset (40 and 20 individuals, respectively) compared to HGREncoder model. For these reason, a fair comparison cannot be made. Despite this, HGREncoder shows a higher standard deviation compared to CT-HGR, which achieves an accuracy of 91.98 ± 2.22%. The higher standard deviation of HGREncoder model is due to using a larger dataset, resulting in greater variability in the results. For the case of TraHGR model, the standard deviation is not reported in their study, but their accuracy for the classification is 93.84%.

Figure 12b displays a comparison between HGREncoder with the CNN and LSTM models in terms of response time. It is observed that the response time of HGREncoder is longer than of the other models (CNN and LSTM). A longer response time means that it takes longer to process and assign a label to a gesture. Response speed is important in determining the viability of a real-time application. The response time of HGREncoder

is less than 300 ms, so it is considered acceptable for real-time applications. Importantly, to simulate real-time conditions, HGREncoder needs to receive more than three previous time windows to make predictions. The wait for the first three windows is due to the model's need for a broader context. Therefore, we collect information from the third window to the end of the sequence to make predictions. HGREncoder processes the entire sequence simultaneously, which implies greater processing and response time.

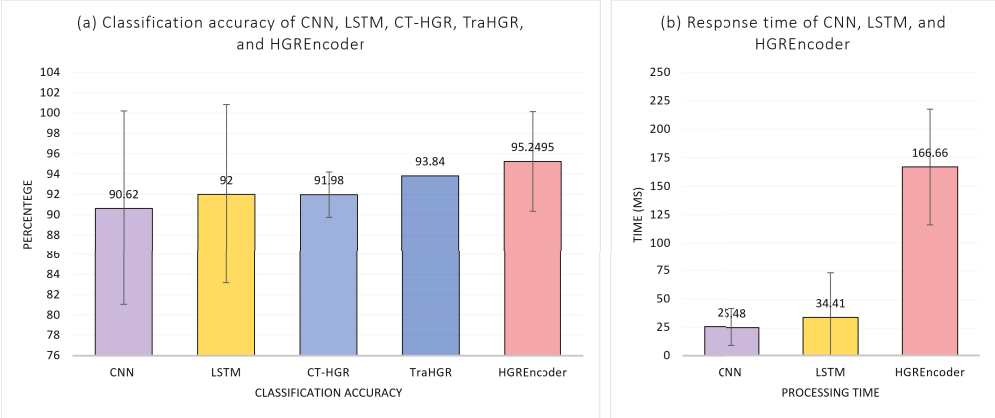


Figure 12. Comparison between HGREncoder, CNN, LSTM, CT-HGR, and TraHGR models in terms of classification accuracy and response time.

The recognition accuracy results of CNN, LSTM and HGREncoder, with and without post-processing are presented in Figure 13a. The recognition accuracy without post-processing achieved for HGREncoder is the most significant advance of this research. HGREncoder model achieves a recognition accuracy without post-processing of 89.7 ± 8.7692%, significantly outperforming the LSTM model that has a recognition accuracy of 65.8 ± 15.51% and CNN model that has a recognition accuracy of 45.4 ± 15.51%. This good recognition accuracy is because transformer models use attention mechanisms to focus on the most relevant parts of the input sequence. Furthermore, Figure 13b shows the results obtained in terms of recognition accuracy with post-processing; HGREncoder reaches 91.7 ± 8.1967%. This result represents a slight improvement compared to 90.6 ± 9.45% of LSTM model and 87.26 ± 11.14% of CNN model.

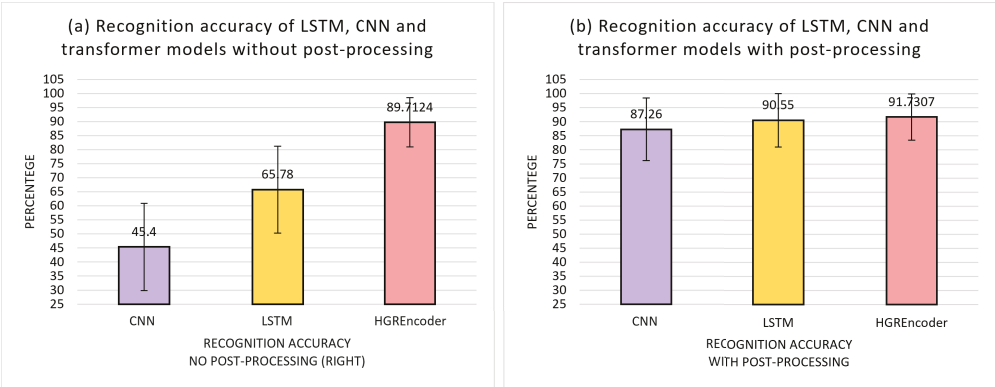


Figure 13. Comparison between the developed model in this work with the CNN and LSTM models in terms of recognition accuracy.

Figure 14a shows the overlap percentage of the transformer model without post-processing which reaches a value of 49.752 ± 8.23%. This result is lower than the 64 ± 9.5% achieved by CNN and the 67.59 ± 10.05% achieved by LSTM. Figure 14b shows the overlap

percentage with post-processing. The trend of overlap percentages with post-processing and without post-processing is the same. The CNN and LSTM models show better results in overlap, with 67.9 ± 7.98% and 69.5 ± 7.4%, respectively, and HGREncoder reaches 50.55 ± 8.31%. The low results in both cases (with and without post-processing) in the overlap are low because the transformer model utilizes the entire sequence for recognition. Using the entire sequence for classification could limit the ability to recognize transitions between gestures. Additionally, using the entire sequence could result in predictions with a shorter duration, leading or lagging predictions.

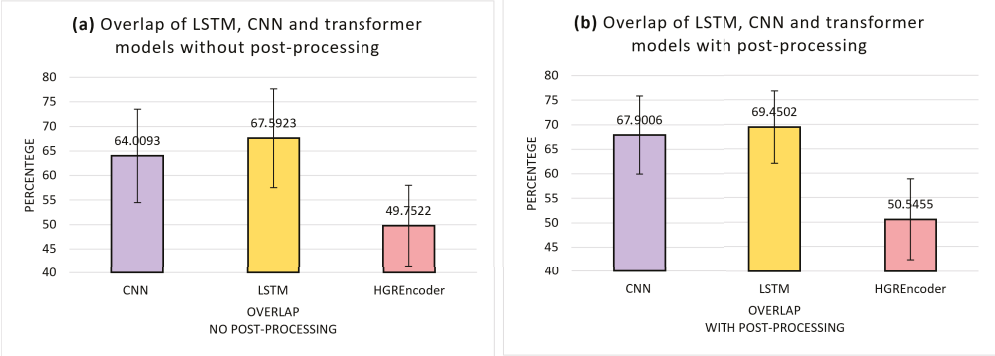


Figure 14. Comparison between the developed model in this work with the CNN and LSTM models in terms of overlap.

5. Conclusions

In this research, a transformer model using positional encoding and the encoder has been developed for real-time gesture classification and recognition. The transformer model created in this work is better compared to CNN, LSTM, CT-HGR, TraHGR. HGREncoder achieves a classification accuracy of 95.25 ± 4.9% and a recognition accuracy of 91.73 ± 8.19%. Despite requiring more processing time, these results highlight the potential of the transformer architecture, which has been extensively studied over the past five years.

The most notable advancement in this research is the significant improvement in recognition accuracy without post-processing, achieving 89.71 ± 8.77%. This recognition accuracy represents an up to 23% improvement compared to previous models developed with the same dataset (CNN and LSTM models). The transformer model learns better because it pays attention to the key components of the entire sequence. The Transformer model comprises the sequence of "relax", "gesture", and "relax" that occurs during each gesture. However, transformer model developed in this research can be improved in terms of processing time and overlapping. The processing time of HGREncoder is 4.8 times longer than that of previous models (CNN and LSTM). The model takes longer to respond because it needs context from the previous three windows. Future work could explore model lightening techniques to reduce processing time while maintaining accuracy. Additionally, the use of more efficient transformer variants could be investigated to optimize the architecture for real-time applications.

On the other hand, the overlap percentage achieved by the transformer model is lower than the overlap percentage of the previous models (CNN and LSTM). This decrease is due to the fact that the transformer model does not identify the exact moment in which the transition from "noGesture" to gesture occurs. Furthermore, this low percentage could be a consequence of the time of the detected gesture being shorter than the actual duration, or being ahead or behind the actual event. Receiving the full sequence could affect the ability to detect subtle variations.

Author Contributions: Conceptualization, L.G.M. and M.E.B.; methodology, L.G.M. and J.A.Z.; software, J.A.Z.; validation, L.I.B. and Á.L.V.; formal analysis, L.I.B. and L.G.M.; investigation, J.A.Z.; data curation, L.G.M.; writing—original draft preparation, L.G.M.; writing—review M.E.B.; supervision, Á.L.V. and M.E.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this study are available upon reasonable request.

Acknowledgments: The authors gratefully acknowledge the financial support provided by the Escuela Politécnica Nacional (EPN) for the development of the research project "PIS-22-10 Análisis dela influencia de la variación intrapersonal de señales electromiográficas sobre la exactitud de modelos de reconocimientos de gestos de la mano".

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Saggio, G.; Cavallo, P.; Ricci, M.; Errico, V.; Zea, J.; Benalcázar, M.E. Sign language recognition using wearable electronics: Implementing K-nearest neighbors with dynamic time warping and convolutional neural network algorithms. *Sensors* **2020**, *20*, 3879. [CrossRef]
2. Koh, J.I.; Cherian, J.; Taele, P.; Hammond, T. Developing a hand gesture recognition system for mapping symbolic hand gestures to analogous emojis in computer-mediated communication. *ACM Trans. Interact. Intell. Syst.* **2019**, *9*, 1–35. [CrossRef]
3. Simon, A.M.; Turner, K.L.; Miller, L.A.; Dumanian, G.A.; Potter, B.K.; Beachler, M.D.; Hargrove, L.J.; Kuiken, T.A. Myoelectric prosthesis hand grasp control following targeted muscle reinnervation in individuals with transradial amputation. *PLoS ONE* **2023**, *18*, e0280210. [CrossRef] [PubMed]
4. Godoy, R.V.; Dwivedi, A.; Liarokapis, M. Electromyography Based Decoding of Dexterous, In-Hand Manipulation Motions With Temporal Multichannel Vision Transformers. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2022**, *30*, 2207–2216. [CrossRef] [PubMed]
5. Tan, C.K.; Lim, K.M.; Chang, R.K.Y.; Lee, C.P.; Alqahtani, A. HGR-ViT: Hand Gesture Recognition with Vision Transformer. *Sensors* **2023**, *23*, 5555. [CrossRef] [PubMed]
6. Montazerin, M.; Zabihi, S.; Rahimian, E.; Mohammadi, A.; Naderkhani, F. ViT-HGR: Vision Transformer-based Hand Gesture Recognition from High Density Surface EMG Signals. In Proceedings of the 2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Glasgow, Scotland, UK, 11–15 July 2022; pp. 5115–5119. [CrossRef]
7. Birkeland, S.; Fjeldvik, L.J.; Noori, N.; Yeduri, S.R.; Cenkeramaddi, L.R. Thermal video-based hand gestures recognition using lightweight CNN. *J. Ambient Intell. Humaniz. Comput.* **2024**, *15*, 3849–3860. [CrossRef]
8. Ahmed, I.T.; Gwad, W.H.; Hammad, B.T.; Alkayal, E. Enhancing Hand Gesture Image Recognition by Integrating Various Feature Groups. *Technologies* **2025**, *13*, 164. [CrossRef]
9. Jaramillo-Yánez, A.; Benalcázar, M.E.; Mena-Maldonado, E. Real-time hand gesture recognition using surface electromyography and machine learning: A systematic literature review. *Sensors* **2020**, *20*, 2467. [CrossRef]
10. Vásconez, J.P.; López, L.I.B.; Ángel Leonardo Valdivieso Caraguay.; Benalcázar, M.E. Hand Gesture Recognition Using EMG-IMU Signals and Deep Q-Networks. *Sensors* **2022**, *22*, 9613. [CrossRef]
11. Benalcázar, M.E.; Motoche, C.; Zea, J.A.; Jaramillo, A.G.; Anchundia, C.E.; Zambrano, P.; Segura, M.; Benalcázar Palacios, F.; Pérez, M. Real-time hand gesture recognition using the Myo armband and muscle activity detection. In Proceedings of the 2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM), Salinas, Ecuador, 16–20 October 2017; pp. 1–6. [CrossRef]
12. Valdivieso Caraguay, A.L.; Vásconez, J.P.; Barona López, L.I.; Benalcázar, M.E. Recognition of Hand Gestures Based on EMG Signals with Deep and Double-Deep Q-Networks. *Sensors* **2023**, *23*, 3905. [CrossRef]
13. Anastasiev, A.; Kadone, H.; Marushima, A.; Watanabe, H.; Zaboronok, A.; Watanabe, S.; Matsumura, A.; Suzuki, K.; Matsumaru, Y.; Nishiyama, H.; et al. A Novel Bilateral Data Fusion Approach for EMG-Driven Deep Learning in Post-Stroke Paretic Gesture Recognition. *Sensors* **2025**, *25*, 3664. [CrossRef] [PubMed]
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.

15. Zabihi, S.; Rahimian, E.; Asif, A.; Mohammadi, A. TraHGR: Transformer for Hand Gesture Recognition via ElectroMyography. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2023**, *31*, 4211–4224. [CrossRef] [PubMed]
16. Barona López, L.I.; Ferri, F.M.; Zea, J.; Ángel Leonardo Valdivieso Caraguay.; Benalcázar, M.E. CNN-LSTM and post-processing for EMG-based hand gesture recognition. *Intell. Syst. Appl.* **2024**, *22*, 200352. [CrossRef]
17. Jiang, Y.; Song, L.; Zhang, J.; Song, Y.; Yan, M. Multi-Category Gesture Recognition Modeling Based on sEMG and IMU Signals. *Sensors* **2022**, *22*, 5855. [CrossRef]
18. Benalcázar, M.; Barona, L.; Valdivieso, L.; Aguas, X.; Zea, J. EMG-EPN-612 Dataset. 2020. Available online: http://dx.doi.org/10.5281/zenodo.4421500 (accessed on 18 July 2025).
19. Merletti, R. *Electromyography: Physiology, Engineering, and Non-Invasive Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2004; Volume 11. [CrossRef]
20. Péter, A.; Andersson, E.; Hegyi, A.; Finni, T.; Tarassova, O.; Cronin, N.J.; Grundström, H.; Arndt, A. Comparing Surface and Fine-Wire Electromyography Activity of Lower Leg Muscles at Different Walking Speeds. *Front. Physiol.* **2019**, *10*, 1283. [CrossRef]
21. Kyranou, I.; Vijayakumar, S.; Erden, M.S. Causes of performance degradation in non-invasive electromyographic pattern recognition in upper limb prostheses. *Front. Neurorobotics* **2018**, *12*, 58. [CrossRef]
22. Huang, Q.; Yang, D.; Jiang, L.; Zhang, H.; Liu, H.; Kotani, K. A Novel Unsupervised Adaptive Learning Method for Long-Term Electromyography (EMG) Pattern Recognition. *Sensors* **2017**, *17*, 1370. [CrossRef]
23. Schulte, R.V.; Prinsen, E.C.; Buurke, J.H.; Poel, M. Adaptive Lower Limb Pattern Recognition for Multi-Day Control. *Sensors* **2022**, *22*, 6351. [CrossRef]
24. Jiang, N.; Muceli, S.; Graimann, B.; Farina, D. Effect of arm position on the prediction of kinematics from EMG in amputees. *Med Biol. Eng. Comput.* **2013**, *51*, 143–151. [CrossRef]
25. Zhai, X.; Jelfs, B.; Chan, R.H.; Tin, C. Self-recalibrating surface EMG pattern recognition for neuroprosthesis control based on convolutional neural network. *Front. Neurosci.* **2017**, *11*, 379. [CrossRef]
26. He, J.; Zhang, D.; Jiang, N.; Sheng, X.; Farina, D.; Zhu, X. User adaptation in long-term, open-loop myoelectric training: implications for EMG pattern recognition in prosthesis control. *J. Neural Eng.* **2015**, *12*, 046005. [CrossRef] [PubMed]
27. Côté-Allard, U.; Fall, C.L.; Drouin, A.; Campeau-Lecours, A.; Gosselin, C.; Glette, K.; Laviolette, F.; Gosselin, B. Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2019**, *27*, 760–771. [CrossRef] [PubMed]
28. Unanyan, N.; Belov, A. Case Study: Influence of Muscle Fatigue and Perspiration on the Recognition of the EMG Signal. *Adv. Syst. Sci. Appl.* **2021**, *21*, 58–70. [CrossRef]
29. Espinoza, D.L.; Velasco, L.E.S. *Comparison of EMG Signal Classification Algorithms for the Control of an Upper Limb Prosthesis Prototype*; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2020; Volume 11. [CrossRef]
30. Zhou, H.; Thanh Le, H.; Zhang, S.; Lam Phung, S.; Alici, G. Hand Gesture Recognition From Surface Electromyography Signals With Graph Convolutional Network and Attention Mechanisms. *IEEE Sens. J.* **2025**, *25*, 9081–9092. [CrossRef]
31. Barona López, L.I.; Valdivieso Caraguay, A.L.; Vimos, V.H.; Zea, J.A.; Vásconez, J.P.; Álvarez, M.; Benalcázar, M.E. An Energy-Based Method for Orientation Correction of EMG Bracelet Sensors in Hand Gesture Recognition Systems. *Sensors* **2020**, *20*, 6327. [CrossRef]
32. Chung, E.A.; Benalcázar, M.E. Real-Time Hand Gesture Recognition Model Using Deep Learning Techniques and EMG Signals. In Proceedings of the 2019 27th European Signal Processing Conference (EUSIPCO), A Coruña, Spain, 2–6 September 2019; pp. 1–5. [CrossRef]
33. Jaramillo-Yanez, A.; Unapanta, L.; Benalcázar, M.E. Short-Term Hand Gesture Recognition using Electromyography in the Transient State, Support Vector Machines, and Discrete Wavelet Transform. In Proceedings of the 2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI), Guayaquil, Ecuador, 11–15 November 2019; pp. 1–6. [CrossRef]
34. Alaparthi, S.; Mishra, M. Bidirectional Encoder Representations from Transformers (BERT): A sentiment analysis odyssey. *arXiv* **2020**, arXiv:2007.01127. [CrossRef]
35. Llinet Benavides, C.; Manso-Callejo, M.A.; Cira, C.I. BERT (Bidirectional Encoder Representations from Transformers) for Missing Data Imputation in Solar Irradiance Time Series. *Eng. Proc.* **2023**, *39*, 26.
36. Montazerin, M.; Rahimian, E.; Naderkhani, F.; Atashzar, S.; Yanushkevich, S.; Mohammadi, A. Transformer-based hand gesture recognition from instantaneous to fused neural decomposition of high-density EMG signals. *Sci. Rep.* **2023**, *13*, 11000. [CrossRef]
37. Benalcázar, M.E.; Ángel Leonardo Valdivieso Caraguay.; López, L.I.B. A user-specific hand gesture recognition model based on feed-forward neural networks, emgs, and correction of sensor orientation. *Appl. Sci.* **2020**, *10*, 8604. [CrossRef]
38. Laboratorio de Investigación en Inteligencia y Visión Artificial. Escuela Politécnica Nacional, 2024. Available online: https://laboratorio-ia.epn.edu.ec/es/ (accessed on 15 August 2025).

39. Atzori, M.; Gijsberts, A.; Castellini, C.; Caputo, B.; Hager, A.G.; Elsig, S.; Giatsidis, G.; Bassetto, F.; Müller, H. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Sci. Data* **2014**, *1*, 140053. [CrossRef]
40. Eddy, E.; Campbell, E.; Bateman, S.; Scheme, E. Big data in myoelectric control: Large multi-user models enable robust zero-shot EMG-based discrete gesture recognition. *Front. Bioeng. Biotechnol.* **2024**, *12*, 2024. [CrossRef]
41. Vásconez, J.P.; Barona López, L.I.; Ángel Leonardo Valdivieso Caraguay.; Benalcázar, M.E. A comparison of EMG-based hand gesture recognition systems based on supervised and reinforcement learning. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106327. [CrossRef]
42. Benalcázar, M.E.; Jaramillo, A.G.; Jonathan; Zea, A.; Páez, A.; Andaluz, V.H. Hand gesture recognition using machine learning and the Myo armband. In Proceedings of the 2017 25th European Signal Processing Conference (EUSIPCO), Kos, Greece, 28 August–2 September 2017; pp. 1040–1044. [CrossRef]
43. Tsai, A.C.; Luh, J.; Lin, T.T. A novel STFT-ranking feature of multi-channel EMG for motion pattern recognition. *Expert Syst. Appl.* **2015**, *42*, 44. [CrossRef]
44. Jiang, W.; Ren, Y.; Liu, Y.; Wang, Z.; Wang, X. Recognition of Dynamic Hand Gesture Based on Mm-Wave Fmcw Radar Micro-Doppler Signatures. In Proceedings of the ICASSP 2021—2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 4905–4909. [CrossRef]
45. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA, USA, 7–12 June 2015; pp. 1–9. [CrossRef]
46. Rahimi, N.; Eassa, F.; Elrefaei, L. An Ensemble Machine Learning Technique for Functional Requirement Classification. *Symmetry* **2020**, *12*, 1601. [CrossRef]
47. Farrell, T.R.; Weir, R.F. The Optimal Controller Delay for Myoelectric Prostheses. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2007**, *15*, 111–118. [CrossRef]
48. Powers, D.M.W. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv* **2011**, arXiv:2010.16061.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Mathematical and Computational Applications

MDPI

Article

A Mathematical Framework for E-Commerce Sales Prediction Using Attention-Enhanced BiLSTM and Bayesian Optimization

Hao Hu [1], Jinshun Cai [1] and Chenke Xu [2,*]

[1] School of International Education, Zhejiang Polytechnic University of Mechanical and Electrical Engineering, Hangzhou 310053, China; huhao@zime.edu.cn (H.H.)
[2] School of Economics and Management, Zhejiang Sci-Tech University, Hangzhou 310018, China
* Correspondence: xuchenke@zstu.edu.cn

Abstract

Accurate sales prediction is crucial for inventory and marketing in e-commerce. Cross-border sales involve complex patterns that traditional models cannot capture. To address this, we propose an improved Bidirectional Long Short-Term Memory (BiLSTM) model, enhanced with an attention mechanism and Bayesian hyperparameter optimization. The attention mechanism focuses on key temporal features, improving trend identification. The BiLSTM captures both forward and backward dependencies, offering deeper insights into sales patterns. Bayesian optimization fine-tunes hyperparameters such as learning rate, hidden-layer size, and dropout rate to achieve optimal performance. These innovations together improve forecasting accuracy, making the model more adaptable and efficient for cross-border e-commerce sales. Experimental results show that the model achieves an Root Mean Square Error (RMSE) of 13.2, Mean Absolute Error (MAE) of 10.2, Mean Absolute Percentage Error (MAPE) of 8.7 percent, and a Coefficient of Determination (R^2) of 0.92. It outperforms baseline models, including BiLSTM (RMSE 16.5, MAPE 10.9 percent), BiLSTM with Attention (RMSE 15.2, MAPE 10.1 percent), Temporal Convolutional Network (RMSE 15.0, MAPE 9.8 percent), and Transformer for Time Series (RMSE 14.8, MAPE 9.5 percent). These results highlight the model's superior performance in forecasting cross-border e-commerce sales, making it a valuable tool for inventory management and demand planning.

Keywords: e-commerce sales prediction; BiLSTM; attention mechanism; Bayesian optimization; time series forecasting

1. Introduction

Accurate sales forecasting is crucial for e-commerce platforms, especially in cross-border markets where demand exhibits high volatility, pronounced seasonality, and regional heterogeneity [1]. Reliable predictions are essential not only for guiding pricing strategies, marketing planning, and resource allocation [2], but also for improving operational efficiency and anticipating regional consumer behavior [3]. In cross-border e-commerce, precise forecasting helps reduce stockouts and overstocking, enabling better decision-making for global market strategies [4,5]. The complexity of consumer preferences, rapid market fluctuations, and variations in regional demand patterns make forecasting challenging yet critical for sustainable growth and profitability [6].

Traditional time series models like AutoRegressive Integrated Moving Average (ARIMA) and exponential smoothing have long been used to predict sales due to their

simplicity and interpretability [7]. However, these models are limited in their ability to capture complex nonlinear patterns and long-term temporal dependencies in dynamic sales data. For example, Gena et al. [8]. applied ARIMA to forecast weekly sales of a footwear brand over three years, effectively capturing sales trends but lacking adaptability to sudden demand fluctuations. Rostan et al. [9] applied ARIMA for option trading strategies during the 2008 financial crisis, achieving successful predictions over a 60-month period. Recently, machine learning methods such as XGBoost have been used for predictive tasks. Wang et al. [10]. utilized XGBoost to model user review-based ratings, extracting valuable consumer behavior features, while Cho et al. [11] compared Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), CNN–LSTM, and XGBoost for Water Level prediction. In the financial sector, Tsai et al. [12] highlighted the importance of unified fiscal year-ends for financial comparability, and Khandelwal et al. [13] applied random forests to Amazon EC2 Spot Price prediction, demonstrating superior accuracy compared to linear regression and decision tree models. These studies indicate the effectiveness of modern machine learning models in capturing nonlinear, high-dimensional, and dynamic patterns across diverse domains.

Recent advances in deep learning provide further improvements for sequential data modeling. LSTM networks, introduced by Qin et al. [14], address the vanishing-gradient problem and effectively capture long-term dependencies. BiLSTM, proposed by Su et al. [15], improves prediction accuracy by modeling both forward and backward temporal contexts. Attention mechanisms, first introduced by Bahdanau et al. [16], enable selective focus on critical features, enhancing model interpretability and performance for complex and irregular datasets. Optimal model performance also depends on hyperparameter tuning; traditional grid and random search methods are computationally expensive and may fail to find the best configuration. Bayesian optimization, proposed by Snoek et al. [17], provides an efficient probabilistic framework for tuning hyperparameters and has been successfully applied to various neural network architectures.

Building on recent advances, this study presents an enhanced Attention BiLSTM model, integrated with Bayesian optimization for cross-border e-commerce sales forecasting. The model leverages BiLSTM to capture bidirectional temporal dependencies, incorporates an attention mechanism to selectively emphasize critical features, and employs Bayesian optimization for efficient hyperparameter tuning. Evaluation on real-world e-commerce datasets demonstrates that the proposed approach significantly outperforms traditional LSTM, standard BiLSTM, and ARIMA models in terms of predictive accuracy. Key contributions of this study include the following:

1. Architectural Innovation: The design of an enhanced BiLSTM network aims to capture bidirectional temporal dependencies more effectively. It integrates an improved attention mechanism to selectively highlight critical sequential features.
2. Attention Mechanism Enhancement: The introduction of a refined attention mechanism that adaptively weighs temporal information, enhancing the focus on influential patterns in complex cross-border e-commerce sales data.
3. Hyperparameter Optimization: The application of Bayesian optimization to automatically and efficiently tune key model hyperparameters for optimal performance.
4. Empirical Validation of Algorithmic Design: Demonstrate through extensive experiments on a large-scale cross-border e-commerce dataset, that covers multiple product categories, that the proposed structural improvements are effective.

2. Related Work

This section reviews related work on neural networks and attention-based methods for cross-border e-commerce forecasting (Sections 2.1 and 2.2). By integrating these strands,

https://doi.org/10.3390/mca31010017

we identify key gaps addressed by our model, including enhanced bidirectional temporal modeling, adaptive attention, and efficient hyperparameter optimization.

2.1. Neural Networks for Cross-Border E-Commerce

Neural networks have become increasingly important for predicting and analyzing cross-border e-commerce sales because they can model complex, non-linear relationships in data. Unlike traditional statistical methods, neural networks can automatically extract features from large-scale transactional datasets, capturing both temporal patterns and contextual factors such as seasonality, promotions, and regional demand variations [18]. In cross-border settings, where sales are influenced by multiple countries, currencies, and logistics constraints, neural networks provide a flexible framework for integrating heterogeneous data sources and learning meaningful patterns for accurate forecasting.

Moreover, advanced architectures such as LSTM and BiLSTM are particularly effective for handling sequential sales data, as they can retain long-term dependencies and incorporate past trends into future predictions. When combined with attention mechanisms, these models can selectively focus on the most relevant historical sales and external factors, improving interpretability and prediction accuracy [19]. As a result, neural network-based models have become a key tool for decision-making in cross-border e-commerce, enabling companies to optimize inventory, plan marketing strategies, and respond dynamically to global market fluctuations.

2.2. Attention-Based Sales Prediction

Attention-based sales prediction leverages the ability of neural networks to focus on the most relevant parts of historical data when making forecasts [20]. Unlike traditional sequential models that treat all past observations equally, attention mechanisms assign different weights to past time steps, allowing the model to selectively emphasize critical trends, seasonal effects, or promotional events. This approach enhances the model's interpretability and improves accuracy, especially in complex and volatile sales environments where certain periods or features have disproportionate influence on future demand.

By integrating attention mechanisms with recurrent architectures such as LSTM or BiLSTM, models can capture both long-term dependencies and short-term fluctuations simultaneously. This combination is particularly effective in e-commerce scenarios, where sales patterns are influenced by multiple factors such as consumer behavior, marketing campaigns, and external events [21]. As a result, attention-based models provide a powerful tool for data-driven decision-making, enabling more precise inventory planning, dynamic pricing, and targeted marketing strategies.

3. Fundamentals

This section lays the foundation for our model. We start by reviewing e-commerce sales forecasting methods (Section 3.1). Then, we describe the Bidirectional LSTM structure and mechanism (Section 3.2), introduce Bayesian optimization for hyperparameter tuning (Section 3.3), and explain the enhanced attention mechanism for feature selection (Section 3.4). These concepts provide the technical basis for the methodology in Section 4.

3.1. E-Commerce Sales Forecast

Accurate e-commerce sales forecasting has been addressed using classical statistical and modern machine learning methods. Traditional time series models, such as ARIMA, model a univariate sales series $\{y_t\}$ as follows:

$$\Phi(L)(1-L)^d y_t = \Theta(L)\epsilon_t + \sum_{j=1}^{p} \alpha_j y_{t-j} + \sum_{k=1}^{q} \beta_k \epsilon_{t-k}, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2), \tag{1}$$

https://doi.org/10.3390/mca31010017

where L is the lag operator, d the differencing order, $\Phi(L)$ and $\Theta(L)$ are AR and MA polynomials, and α_j, β_k denote additional lag coefficients capturing complex autocorrelations.The parameters p and q are the orders of the autoregressive (AR) and moving average (MA) components, respectively. The coefficients α_j and β_k capture the influence of past sales and noise terms. The time series y_t represents univariate sales, and the error terms ϵ_t follow a normal distribution, $\mathcal{N}(0, \sigma^2)$, with dimensions 1×1. This formulation allows modeling higher-order dependencies and enhanced seasonal structures.

Gradient boosting, particularly XGBoost, has shown strong performance in handling nonlinear, high-dimensional features. Given input features $X = [x_1, \ldots, x_n]$ (where each x_i is a vector of features for the i-th sample) and target y, XGBoost constructs an additive model with interaction depth d:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i) = \sum_{k=1}^{K} \sum_{j=1}^{d} w_{kj} \mathbb{I}(x_i \in R_{kj}), \quad f_k \in \mathcal{F}, \tag{2}$$

where $f_k(x_i)$ is the contribution of the k-th base function, w_{kj} is the weight of the j-th split, and $\mathbb{I}(x_i \in R_{kj})$ is an indicator function that is 1 if x_i is in region R_{kj}, and 0 otherwise. The model is built by iteratively adding functions from the function space $\mathcal{F}$ to minimize residuals.The model is constructed by iteratively adding functions from the function space $\mathcal{F}$ to minimize the following regularized objective function:

$$\mathcal{L} = \sum_{i=1}^{N} l\Big(y_i, \sum_{k=1}^{K} f_k(x_i)\Big) + \sum_{k=1}^{K} \Big(\gamma T_k + \frac{1}{2}\lambda \sum_{j=1}^{T_k} w_{kj}^2\Big), \tag{3}$$

where R_{kj} represents the j-th leaf region of the k-th tree, T_k the number of leaves, and w_{kj} the corresponding leaf weight. This formulation captures nonlinear interactions and hierarchical feature effects.

3.2. Bidirectional LSTM

BiLSTM networks are designed to capture both past and future dependencies in sequence data [22], making them particularly suitable for time series with volatile and asymmetric patterns, such as cross-border e-commerce sales. This characteristic also makes BiLSTM more suitable than Transformer models for small datasets, where attention-based Transformers may overfit or be sensitive to noise.

For an input sequence $X = \{x_1, x_2, \ldots, x_T\}$, the bidirectional hidden states are computed as follows:

$$H = \overrightarrow{H} \oplus \overleftarrow{H} = \sigma\big(XW_f + \overrightarrow{H}_{-1}U_f + XW_b + \overleftarrow{H}_{+1}U_b + B\big) \tag{4}$$

where X is the input sequence of size $T \times d$, $\overrightarrow{H}$ and $\overleftarrow{H}$ are the forward and backward hidden states with dimensions $T \times h$, W_f and W_b are weight matrices of size $d \times h$ and $h \times h$ respectively, U_f and U_b are weight matrices for the previous hidden state of size $h \times h$, and B is the bias vector of size h. The operator $\oplus$ denotes the concatenation of the forward and backward hidden states. This bidirectional structure enables the model to consider both historical and future information at each time step, which is especially effective in capturing temporal dynamics in short and volatile sequences, where global attention alone may be less reliable.

To aggregate the sequence information into a single context vector and emphasize the most relevant time steps, we apply a multi-head attention mechanism:

https://doi.org/10.3390/mca31010017

$$\hat{H} = \sum_{t=1}^{T} \sum_{m=1}^{M} \frac{\exp\left(\mathbf{v}_m^\top \tanh(W_\alpha^m h_t + b_\alpha^m)\right)}{\sum_{k=1}^{T} \exp\left(\mathbf{v}_m^\top \tanh(W_\alpha^m h_k + b_\alpha^m)\right)} h_t \tag{5}$$

Here, h_t is the concatenated hidden state, W_α^m and b_α^m are the attention weight matrices and biases for head m, and $\mathbf{v}_m$ is the attention vector. This mechanism allows the model to dynamically assign higher importance to critical time steps, such as sudden demand spikes or seasonal peaks. Compared with Transformer-based models, this selective focus enables BiLSTM to better handle short sequences and volatile data while avoiding overfitting.

Finally, the prediction and training objective are defined as follows:

$$\hat{y} = \phi(\hat{H} W_y + b_y), \quad \mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \|\hat{y}_i - y_i\|^2 + \lambda \|\theta\|^2 \tag{6}$$

where $\hat{y}$ is the predicted output, W_y is the output weight matrix, b_y is the bias, ϕ is the activation function, and λ controls regularization to prevent overfitting. By combining BiLSTM with attention, the model efficiently captures temporal dependencies and highlights key time steps, making it more robust than Transformers for small, highly volatile datasets. This justifies our choice of BiLSTM for the current study.

3.3. Hyperparameter Optimization in Deep Learning Models

Hyperparameter optimization is critical for improving deep learning models, particularly in high-dimensional search spaces. Bayesian optimization has proven effective by using a probabilistic surrogate model to estimate the objective function $f(\theta)$ and guide the search for optimal hyperparameters θ^*. The goal is to maximize the objective:

$$\theta^* = \arg\max_{\theta} \mathbb{E}[f(\theta)], \tag{7}$$

where $\mathbb{E}[f(\theta)]$ denotes the expected value of the objective function. To model $f(\theta)$, a Gaussian Process (GP) posterior is used, given by the following:

$$p(f(\theta)|D) = \mathcal{N}(f(\theta); \mu(\theta), \Sigma(\theta, \theta')), \tag{8}$$

where $\mu(\theta)$ is the mean prediction and $\Sigma(\theta, \theta')$ is the covariance function. The acquisition function, such as Expected Improvement (EI), guides the search by evaluating the expected improvement over the current best solution:

$$\mathrm{EI}(\theta) = (f_{\text{best}} - \mu(\theta))\Phi\left(\frac{f_{\text{best}} - \mu(\theta)}{\sigma(\theta)}\right) + \sigma(\theta)\phi\left(\frac{f_{\text{best}} - \mu(\theta)}{\sigma(\theta)}\right), \tag{9}$$

where f_{best} is the current best value, and $\mu(\theta)$ and $\sigma(\theta)$ are the predicted mean and standard deviation. The dimensions of $\mu(\theta)$ and $\sigma(\theta)$ are both 1×1.

3.4. Attention Mechanism Enhancement for Sequential Feature Highlighting

Attention mechanisms enable models to focus on relevant parts of an input sequence. For a sequence of length T and feature dimension d, scaled dot-product attention with multiple heads and masking can be expressed as follows:

$$\mathbf{A}_{t,i}^{(h)} = \frac{\exp\left(\left(\mathbf{Q}_t^{(h)} \mathbf{K}_i^{(h)\top}\right)/\sqrt{d_k} + M_{t,i}^{(h)}\right)}{\sum_{j=1}^{T} \exp\left(\left(\mathbf{Q}_t^{(h)} \mathbf{K}_j^{(h)\top}\right)/\sqrt{d_k} + M_{t,j}^{(h)}\right)}, \quad t, i = 1, \ldots, T,\ h = 1, \ldots, H, \tag{10}$$

https://doi.org/10.3390/mca31010017

where $\mathbf{Q}^{(h)}, \mathbf{K}^{(h)}, \mathbf{V}^{(h)} \in \mathbb{R}^{T \times d_k}$ are the queries, keys, and values for head h, and $M^{(h)} \in \mathbb{R}^{T \times T}$ is an optional mask.

The context tensors for each head are computed as follows:

$$\hat{\mathbf{H}}_t^{(h)} = \sum_{i=1}^{T} \mathbf{A}_{t,i}^{(h)} \mathbf{V}_i^{(h)} \in \mathbb{R}^{d_k}, \quad t = 1, \ldots, T,\ h = 1, \ldots, H. \tag{11}$$

For full batch processing, multi-head attention can be written as a 4D tensor operation:

$$\mathbf{Z}_{b,t,h,k} = \sum_{i=1}^{T} \frac{\exp\left((\mathbf{Q}_{b,t,h,:} \cdot \mathbf{K}_{b,i,h,:}^{\top})/\sqrt{d_k} + M_{b,t,i,h}\right)}{\sum_{j=1}^{T} \exp\left((\mathbf{Q}_{b,t,h,:} \cdot \mathbf{K}_{b,j,h,:}^{\top})/\sqrt{d_k} + M_{b,t,j,h}\right)} \mathbf{V}_{b,i,h,k}, \tag{12}$$

$$\text{Output}_{b,t,:} = \text{Concat}_{h=1}^{H} \mathbf{Z}_{b,t,h,:}\, W_h^O + b^O, \quad b = 1, \ldots, B,\ t = 1, \ldots, T,\ k = 1, \ldots, d_k, \tag{13}$$

where B is the batch size and $W^O \in \mathbb{R}^{Hd_k \times d}$. $\mathbf{Q}_{b,t,h,:}$, $\mathbf{K}_{b,i,h,:}$, and $\mathbf{V}_{b,i,h,k}$ are the query, key, and value matrices for the b-th batch, t-th timestep, h-th attention head, and i-th timestep. The dimensions of $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ are (B, T, H, D), and $M_{b,t,i,h}$ is a masking term with dimensions (B, T, T, H).

4. Methodology

This section presents our Enhanced BiLSTM model for cross-border e-commerce sales forecasting. We describe the core architecture (Section 4.1), adaptive attention mechanism (Section 4.2), Bayesian optimization for hyperparameter tuning (Section 4.3), and the full method overview (Section 4.4). Each component addresses model limitations, ensuring better accuracy and robustness.

4.1. Enhanced BiLSTM Network with Attention and Residual Connections

In time series forecasting, traditional LSTMs capture long-range dependencies but are limited by their unidirectional structure. BiLSTMs address this by processing sequences in both directions, yet they still struggle to prioritize the most relevant time steps, particularly in complex tasks like cross-border e-commerce sales forecasting. To overcome these challenges, we propose an Enhanced BiLSTM Network with dual-activation attention and residual connections, which selectively focuses on important time steps while preserving long-term dependencies, making it more suitable than Transformer-based models for small, volatile datasets.

Given an input sequence $X = \{x_1, x_2, \ldots, x_T\}$, the BiLSTM generates bidirectional hidden states $H = \{h_1, h_2, \ldots, h_T\}$, where $h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$ is the concatenation of forward and backward states at each time step. This bidirectional structure allows the model to capture both past and future dependencies, which is crucial for modeling asymmetric temporal effects in volatile sales data.

To focus on the most relevant time steps, we introduce an advanced attention mechanism that applies non-linear transformations and multiplicative interactions. The attention weight α_t is calculated as follows:

$$\alpha_t = \frac{\exp\left(\mathbf{v}^{\top} \cdot \text{ReLU}(\mathbf{W}_a h_t + \mathbf{b}_a) + \mathbf{c}^{\top} \cdot \text{sigmoid}(\mathbf{W}_a^{(2)} h_t + \mathbf{b}_a^{(2)})\right)}{\sum_{k=1}^{T} \exp\left(\mathbf{v}^{\top} \cdot \text{ReLU}(\mathbf{W}_a h_k + \mathbf{b}_a) + \mathbf{c}^{\top} \cdot \text{sigmoid}(\mathbf{W}_a^{(2)} h_k + \mathbf{b}_a^{(2)})\right)}, \tag{14}$$

where ReLU emphasizes significant temporal features, such as seasonal peaks or promotions, while sigmoid attenuates less relevant signals, allowing the model to balance sensitivity to short-term fluctuations with the preservation of long-term trends. This dual-activation attention is particularly effective for small, volatile datasets where global

https://doi.org/10.3390/mca31010017

attention mechanisms of Transformers may overfit or be sensitive to noise. The weighted hidden state is then computed as follows:

$$\hat{h}_t = \alpha_t \cdot \left(\mathbf{W}_h \cdot \text{LeakyReLU}\left(\mathbf{W}_h^{(2)} h_t + \mathbf{b}_h^{(2)} \right) \right), \tag{15}$$

where LeakyReLU allows gradients to propagate for negative activations, avoiding dead neurons and improving learning of low-activation features. This further enhances robustness in volatile sequences.

To enhance the model further, we incorporate residual connections. The residual-modified state h_t^{res} is defined as follows:

$$h_t^{\text{res}} = \left(h_t + \gamma \cdot \tanh\left(\mathbf{W}_r \cdot \text{ReLU}\left(\mathbf{W}_r^{(2)} h_t + \mathbf{b}_r^{(2)} \right) \right) \right) \cdot \text{sigmoid}\left(\mathbf{W}_r^{(3)} h_t + \mathbf{b}_r^{(3)} \right), \tag{16}$$

where the residual connections help prevent vanishing gradients and combine transformed features with original hidden states, improving the model's ability to capture complex temporal dependencies.The final prediction $\hat{y}_t$ is computed as follows:

$$\hat{y}_t = \mathbf{W}_y \cdot (h_t^{\text{res}}) \cdot \text{softmax}\left(\mathbf{W}_y^{(2)} \cdot h_t^{\text{res}} + \mathbf{b}_y^{(2)} \right) + \mathbf{b}_y, \tag{17}$$

where the softmax normalization and linear projections map the enhanced hidden states to the output space, making the model capable of forecasting both normal and extreme fluctuations. Overall, this Enhanced BiLSTM framework is well-suited to small, highly volatile cross-border e-commerce sales data, offering greater stability and generalization than Transformer-based approaches on the current dataset.

The pseudocode for the methodology is summarized in Algorithm 1, highlighting the encoding, dual-activation attention, residual enhancement, and output projection steps.

Algorithm 1 Enhanced BiLSTM with Dual-Activation Attention and Residual Connections

Input: Time series $X \in \mathbb{R}^{T \times d}$
Output: Predicted values $\hat{Y} \in \mathbb{R}^{T \times d_y}$
1. BiLSTM Encoding:
$h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}],\ H = \{h_1, \ldots, h_T\}$
2. Dual-Activation Attention:
$e_t = \mathbf{v}^\top \text{ReLU}(\mathbf{W}_a h_t + \mathbf{b}_a) + \mathbf{c}^\top \text{sigmoid}(\mathbf{W}_a^{(2)} h_t + \mathbf{b}_a^{(2)})$
$\alpha_t = \text{softmax}(e_t)$
3. Weighted Hidden Representation:
$\hat{h}_t = \alpha_t \cdot [\mathbf{W}_h \cdot \text{LeakyReLU}(\mathbf{W}_h^{(2)} h_t + \mathbf{b}_h^{(2)})]$
4. Residual Enhancement:
$r_t = \tanh(\mathbf{W}_r \cdot \text{ReLU}(\mathbf{W}_r^{(2)} h_t + \mathbf{b}_r^{(2)}))$
$g_t = \text{sigmoid}(\mathbf{W}_r^{(3)} h_t + \mathbf{b}_r^{(3)})$
$h_t^{\text{res}} = (h_t + \gamma \cdot r_t) \odot g_t$
5. Output Projection:
$\hat{y}_t = \mathbf{W}_y \cdot h_t^{\text{res}} \cdot \text{softmax}(\mathbf{W}_y^{(2)} h_t^{\text{res}} + \mathbf{b}_y^{(2)}) + \mathbf{b}_y$
6. Return:
$\hat{Y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_T\}$

4.2. Adaptive Attention Mechanism

In cross-border e-commerce sales forecasting, time-series data often exhibit complex patterns, including seasonality, promotions, and sudden spikes. Moreover, the data is often limited in length, with high volatility, making it challenging for models to focus on the most relevant time steps. To address this, we propose an adaptive attention mechanism that dynamically assigns importance to each time step, guided by both current and his-

https://doi.org/10.3390/mca31010017

torical hidden states. Let h_t denote the hidden state at time step t. We first apply dual non-linear transformations:

$$u_t = \tanh(\mathbf{W}_u h_t + \mathbf{b}_u) \odot \text{sigmoid}(\mathbf{W}_s h_t + \mathbf{b}_s) \tag{18}$$

where the combination of tanh and sigmoid allows the model to simultaneously capture strong and subtle patterns. tanh emphasizes the dominant temporal signals, such as promotional peaks, while sigmoid acting as a gating mechanism to downweight noisy or less informative time steps. This design aligns with the dual-mode demand dynamics observed in cross-border e-commerce data. Next, we incorporate temporal context interaction by considering the previous hidden state:

$$\tilde{e}_t = \mathbf{v}^\top u_t + \gamma \cdot (\mathbf{h}_{t-1}^\top \mathbf{W}_c h_t) \tag{19}$$

The term $\mathbf{h}_{t-1}^\top \mathbf{W}_c h_t$ captures dependencies between consecutive time steps, which is critical for short, volatile sequences where past sales patterns influence current demand. The learnable scaling factor γ adjusts the relative contribution of historical context. The attention weight for each time step is then obtained using a softmax function:

$$\alpha_t = \frac{\exp(\tilde{e}_t)}{\sum_{k=1}^{T} \exp(\tilde{e}_k)} \tag{20}$$

This normalization ensures that the model selectively focuses on key time steps, giving higher weight to critical events such as promotions, seasonal peaks, or sudden demand spikes, while suppressing irrelevant fluctuations. Finally, the weighted hidden state is computed as follows:

$$\hat{h}_t = \alpha_t \cdot (\mathbf{W}_h h_t + \mathbf{b}_h) \tag{21}$$

where $\hat{h}_t$ is the resulting context-aware representation. This adaptive weighting is particularly effective for small, volatile datasets, allowing the BiLSTM to prioritize informative temporal features without overfitting.

4.3. Hyperparameter Optimization

Model performance is highly sensitive to hyperparameters, especially in small or noisy datasets. To ensure robust and stable forecasting, we adopt Bayesian optimization for systematic hyperparameter tuning. This approach balances exploration of the hyperparameter space with exploitation of promising configurations, which is crucial when the dataset is limited in size.

Let θ denote the set of hyperparameters, such as learning rate, hidden layer size, dropout rate, and attention layer weight initialization. The objective is to find the optimal θ^* that minimizes a validation loss function $\mathcal{L}_{val}(\theta)$ within a feasible search space Θ:

$$\theta^* = \arg \min_{\theta \in \Theta, \theta_{\min} \leq \theta \leq \theta_{\max}} \mathcal{L}_{val}(\theta), \tag{22}$$

where $\theta_{\min}$ and $\theta_{\max}$ define practical bounds for each hyperparameter.

Bayesian optimization approximates the unknown objective function $\mathcal{L}_{val}(\theta)$ using a probabilistic surrogate, typically a Gaussian Process (GP) with observation noise:

$$\mathcal{L}_{val}(\theta) \sim \mathcal{GP}\left(\mu(\theta), k(\theta, \theta') + \sigma_n^2 \delta_{\theta,\theta'}\right), \tag{23}$$

where $\mu(\theta)$ is the mean function, $k(\theta, \theta')$ is the covariance kernel, σ_n^2 represents observation noise, and $\delta_{\theta,\theta'}$ is the Kronecker delta function.

https://doi.org/10.3390/mca31010017

By using the GP surrogate, Bayesian optimization efficiently predicts promising hyperparameters even with limited evaluations, which is particularly useful for small cross-border sales datasets that cannot support exhaustive grid search.

To select the next hyperparameter configuration, an acquisition function is used. We adopt the Expected Improvement (EI) criterion, which balances exploration and exploitation:

$$\mathrm{EI}(\theta) = \mathbb{E}\big[\max(0, \mathcal{L}_{val}^{best} - \mathcal{L}_{val}(\theta))\big] = (\mathcal{L}_{val}^{best} - \mu_t(\theta))\Phi(Z) + \sigma_t(\theta)\phi(Z), \tag{24}$$

$$Z = \frac{\mathcal{L}_{val}^{best} - \mu_t(\theta)}{\sigma_t(\theta)}, \tag{25}$$

where $\mu_t(\theta)$ and $\sigma_t^2(\theta)$ are the predictive mean and variance from the GP at iteration t; $\mathcal{L}_{val}^{best}$ is the best observed loss; and Φ and ϕ denote the standard normal CDF and PDF.

The next hyperparameter candidate is then chosen as follows:

$$\theta_{t+1} = \arg\max_{\theta \in \Theta} \mathrm{EI}(\theta \mid \mathcal{D}_t), \tag{26}$$

where $\mathcal{D}_t = \{(\theta_i, \mathcal{L}_{val}(\theta_i))\}_{i=1}^{t}$ represents the set of hyperparameters evaluated up to iteration t.

Through iterative updates, Bayesian optimization efficiently converges to the optimal configuration θ^*, improving model generalization and stability across different product categories and time periods. This is especially important for small, volatile cross-border e-commerce datasets, where manual tuning or naive search strategies may fail to achieve consistent performance.

4.4. Overview of the Proposed Method

Traditional time series models often fail to capture the complex, nonlinear, and seasonal patterns in international e-commerce sales data. Additionally, Transformer-based models can struggle with small datasets and highly volatile sequences due to their large parameter count and sensitivity to noise. To address these challenges, we propose an enhanced BiLSTM model with a dual-activation attention mechanism and Bayesian hyperparameter optimization, which balances expressiveness and robustness for limited, irregular sales data.

The framework consists of four core modules. The BiLSTM captures bidirectional dependencies, leveraging both past and future information. Its recurrent structure efficiently models temporal patterns in short sequences, mitigating overfitting compared to Transformers. The attention mechanism dynamically assigns weights to key time steps and features, emphasizing patterns such as seasonal trends, promotions, and sudden sales spikes. The dual-activation mechanism (ReLU + Sigmoid) further allows the model to amplify strong signals while suppressing noise, providing selective temporal feature weighting aligned with the irregular nature of cross-border sales. Bayesian optimization automatically tunes critical hyperparameters, including learning rate, hidden layer size, and dropout rate, ensuring optimal performance without manual tuning. This automated tuning is particularly important for volatile datasets, where sensitivity to hyperparameters can significantly impact forecasting stability.

Figure 1 illustrates the overall architecture. Input sales sequences are processed by BiLSTM layers to obtain hidden states, which are then weighted by the adaptive attention module. The resulting sequence representation is passed to the output layer for sales prediction, with hyperparameters optimized via Bayesian optimization to enhance generalization and stability across diverse e-commerce datasets. This design ensures that both strong seasonal/promotional patterns and subtle, context-dependent trends are captured effectively.

https://doi.org/10.3390/mca31010017

As shown in Figure 1, the model follows a clear and theoretically justified flow to capture complex sales patterns:

- BiLSTM Input Processing: The input time series is processed by the BiLSTM layer, which captures both past trends and future events, such as upcoming promotions. Its bidirectional modeling is especially effective for short, volatile sequences where long-range attention may be less reliable.
- Dual-Activation Module: A dual-activation mechanism applies both ReLU and Sigmoid to the BiLSTM hidden states. ReLU amplifies strong, discriminative signals (e.g., promotional spikes), while Sigmoid attenuates weak or noisy signals (e.g., regular months without promotions). This complementary design enables selective feature amplification and gating, improving the model's ability to capture both short-term fluctuations and long-term trends.
- Adaptive Attention Mechanism: The attention module dynamically adjusts the weight of each time step. During peak seasons, like Black Friday, it assigns higher importance to promotional periods while reducing the influence of irrelevant data. Combined with dual-activation preprocessing, this mechanism ensures that the model focuses on both strong events and contextually relevant patterns.
- Residual Connections: Residual connections are added between the BiLSTM and attention layers. These connections stabilize training by preserving key features, preventing gradient vanishing, and enabling more complex interactions between original and transformed hidden states.
- Bayesian Optimization: Bayesian optimization tunes hyperparameters such as learning rate, hidden layer size, and dropout rate automatically. This adaptive tuning improves model generalization and stability, particularly important for small datasets with high volatility, where manual tuning may fail to achieve consistent performance.

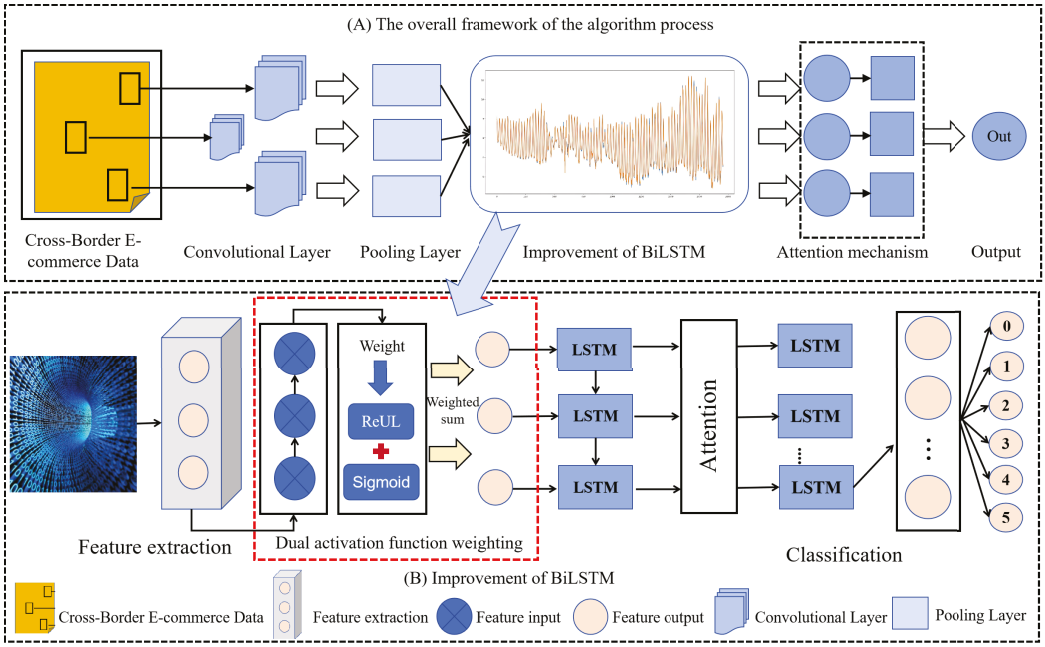


Figure 1. The overall framework of the proposed enhanced BiLSTM model for cross-border e-commerce sales forecasting.

Together, these components provide a principled and effective forecasting framework: the BiLSTM efficiently captures bidirectional temporal dependencies, the dual-activation attention module selectively emphasizes relevant time steps, residual connections enhance feature propagation, and Bayesian optimization ensures stable and optimal hyperparameter

https://doi.org/10.3390/mca31010017

settings. This integrated design is particularly suitable for small, volatile cross-border e-commerce datasets, where Transformers may struggle with overfitting or noise sensitivity.

5. Experimental Setup

This section outlines the experimental design. We cover data and preprocessing (Sections 5.1 and 5.2), experimental setup and training parameters (Section 5.3), baseline algorithms (Section 5.4), and evaluation metrics (Section 5.5). This ensures reliable results for assessing the model's performance in cross-border e-commerce sales forecasting.

5.1. Data Source

The dataset used in this study is collected from the Amazon e-commerce platform, which is one of the largest and most diverse international online marketplaces (Amazon Sales Dataset: https://www.kaggle.com/datasets/karkavelrajaj/amazon-sales-dataset, accessed on 15 June 2025). The data encompasses historical sales records across multiple product categories, including electronics, home appliances, clothing, and personal care items. Each record contains time-stamped information such as sales volume, product identifier, category, price, discount, and relevant promotional events [23].

To ensure data quality and relevance for cross-border e-commerce sales forecasting, we handled missing values, detected outliers and normalized the data. Seasonal patterns, promotional spikes, and other irregular events were preserved to capture the inherent complexity of the sales dynamics on the Amazon platform. The resulting dataset provides a rich and high-resolution time series suitable for training and evaluating the proposed Enhanced BiLSTM model with adaptive attention and Bayesian hyperparameter optimization.

5.2. Data Preprocessing

The dataset used in this study is collected from the Amazon e-commerce platform Amazon Sales Dataset. In real-world data, noise, missing values, and inconsistencies may adversely affect model performance. Therefore, a systematic preprocessing procedure is applied before feature selection and model training. The main steps are as follows. An example of the processed dataset is shown in Table 1.

1. Data Cleaning: Duplicate records and missing entries are removed to ensure data quality. Outliers, such as extremely high or low sales values caused by data entry errors or exceptional events, are detected and handled appropriately. This step ensures that the data fed into the model is reliable and consistent.
2. Data Encoding: Time-related features are extracted from the original timestamps, which are at a daily granularity. Numeric fields for year and month are created, e.g., year values as 2022, 2023, and month values as 1–12, to enable temporal modeling at monthly granularity.
3. Data Aggregation: Sales are aggregated at the monthly level, which is an important granularity for capturing seasonal patterns and promotional effects. Analysis of historical sales shows that demand varies significantly across months, with peaks during promotional seasons such as Prime Day or Black Friday.
4. Feature Construction: To capture seasonal effects, a numeric seasonal feature is added representing the quarter (1 to 4). Promotional activity is encoded as a binary feature, with 1 indicating the presence of a sale or discount campaign during the month and 0 otherwise. Additionally, other relevant product attributes, such as category, price tier, and rating, are included as features to enhance model performance.

After these preprocessing steps, the resulting dataset provides a clean, aggregated, and feature-rich time series of monthly sales for multiple product categories on the Amazon

https://doi.org/10.3390/mca31010017

platform. This dataset is suitable for training and evaluating the proposed Enhanced BiLSTM model with adaptive attention and Bayesian hyperparameter optimization.

Table 1. Preprocessed monthly sales data for Amazon products.

Month	Year	Product Category	Units Sold	Promotion	Season (Quarter)
1	2022	Electronics	1200	0	1
2	2022	Electronics	1350	0	1
3	2022	Electronics	1420	1	1
4	2022	Electronics	1300	0	2
5	2022	Home Appliances	950	0	2
6	2022	Home Appliances	870	1	2
7	2022	Clothing	2100	0	3
8	2022	Clothing	2350	1	3
9	2022	Personal Care	1500	0	3
10	2022	Personal Care	1650	1	4
11	2022	Electronics	1400	1	4
12	2022	Electronics	1550	1	4
13	2023	Electronics	1600	0	1
14	2023	Home Appliances	980	0	1
15	2023	Clothing	2400	1	1

5.3. Experimental Environment and Parameter Settings

All experiments were conducted in a Python-based environment managed with Anaconda, using Python 3.8 as the development IDE. This setup ensures reproducible and efficient training for the proposed forecasting model. The detailed environment configuration and key parameter settings for the proposed model and all baseline algorithms are summarized in Tables 2 and 3.

Table 2. Experimental environment configuration.

Component	Specification
Python Version	3.10
Platform	Anaconda 3
Development IDE	PyCharm 2025.3
Hardware	Intel Core i7, 16 GB RAM, NVIDIA RTX 3060
Dataset Split	Training: Testing = 7:3 (time-based split) First 16 months for training, last 8 months for testing
General Training Parameters	Batch size = 64, Epochs = 300
Hyperparameter Search Ranges	Dropout rate: 0.1–0.5 Learning rate: 0.0001–0.01

Table 3. Hyperparameter values for all models.

Model	Learning Rate	Hidden Units	Attention Dim.	Dropout Rate	Other Key Parameters
ARIMA	–	–	–	–	Seasonal period = 12
Standard LSTM	0.001	128	–	0.2	Layers = 2
BiLSTM	0.001	128	–	0.2	Layers = 2
BiLSTM + Attention	0.001	128	64	0.2	Layers = 2, Attention heads = 1
TCN	0.001	64	–	0.2	Layers = 4, Kernel size = 3, Dilation = 2
TFT	0.001	128	64	0.2	Encoder layers = 2, Attention heads = 2
Proposed Enhanced BiLSTM	0.001	128	64	0.2	Layers = 2, Residual γ = 0.1, Dual-attention

https://doi.org/10.3390/mca31010017

5.3.1. Experimental Environment Configuration

This subsection describes the experimental environment and general training configuration used for all models to ensure reproducibility and fair comparison. All experiments were conducted in a unified software and hardware environment, and identical data preprocessing and training protocols were applied across the proposed model and all baseline algorithms. The detailed environment configuration, including software versions, hardware specifications, dataset split strategy, and general training parameters, is summarized in Table 2.

To account for the stochastic nature of deep learning training, each neural network model was executed 10 times with different random seeds under the same environment configuration shown in Table 2. The reported experimental results correspond to the mean performance across these runs, with standard deviations provided where applicable. This repeated-experiment protocol enhances the robustness of the results and provides a reliable basis for subsequent statistical significance analysis.

5.3.2. Model-Specific Parameter Settings

This subsection details the model-specific hyperparameter configurations used in the experiments. To ensure a fair and unbiased comparison, all baseline models were tuned using grid search with 3-fold time-based cross-validation on the training set, while the proposed Enhanced BiLSTM model was optimized using Bayesian optimization, as described in Section 4.3. The hyperparameter search ranges were kept consistent across models whenever applicable, and the best-performing configuration for each model was selected based on validation RMSE. The final selected hyperparameter values for all models are summarized in Table 3.

As shown in Table 3, the adopted parameter configuration strategy ensures experimental consistency and optimization fairness across all compared models. By sharing core training settings and comparable hyperparameter search spaces, performance differences can be attributed primarily to architectural design rather than experimental bias. Furthermore, several parameters were tailored to the characteristics of cross-border e-commerce sales data, such as the seasonal period of 12 for ARIMA to capture annual monthly seasonality, and the residual weighting factor $\gamma = 0.1$ in the proposed model to balance feature propagation and training stability.

5.4. Comparison Algorithms

To evaluate the performance of the proposed Enhanced BiLSTM model with adaptive attention and Bayesian hyperparameter optimization, six baseline algorithms are considered. First, ARIMA, a classical statistical model, is included to capture linear and seasonal dependencies in time series data. Standard LSTM [24], a unidirectional LSTM network, and BiLSTM, a bidirectional LSTM [25] that leverages both past and future context, are used to assess the effectiveness of deep learning models with and without bidirectional information. Additionally, BiLSTM with a standard attention mechanism is employed to examine the benefits of basic attention weighting on hidden states .

Furthermore, two modern deep learning models are included to provide a more comprehensive comparison. The Temporal Convolutional Network (TCN) [26] is a convolution-based sequence model that efficiently captures long-range dependencies, whereas the Transformer for Time Series (TFT) [27] is an attention-based architecture designed for interpretable time series forecasting, capable of modeling complex temporal and feature interactions. These comparison algorithms collectively cover classical statistical approaches, traditional and bidirectional LSTM models, as well as advanced deep learning techniques,

https://doi.org/10.3390/mca31010017

allowing a thorough evaluation of the proposed method's ability to capture complex patterns in cross-border e-commerce sales data.

To ensure fair comparison, all baseline models were tuned using grid search with 3-fold time-based cross-validation on the training set. Key hyperparameters, including learning rate (0.0001–0.01), hidden layer size (64–256), dropout rate (0.1–0.5), and attention layer dimension (32–128, if applicable), were searched within reasonable ranges. The best-performing parameter combination for each baseline was selected based on validation RMSE, consistent with the tuning process of the proposed model using Bayesian optimization.

5.5. Evaluation Metrics

To assess the performance of the proposed Enhanced BiLSTM model and the baseline algorithms, four evaluation metrics are employed: RMSE, MAE, MAPE and R^2. To better account for varying scales and the importance of different time steps, we introduce weighted and normalized forms of these metrics.

The weighted RMSE is defined as follows:

$$\text{RMSE}_w = \sqrt{\frac{\sum_{t=1}^{T} w_t(\hat{y}_t - y_t)^2}{\sum_{t=1}^{T} w_t}}, \tag{27}$$

where w_t is a weight reflecting the relative importance of time step t.

The weighted MAE is expressed as follows:

$$\text{MAE}_w = \frac{\sum_{t=1}^{T} w_t|\hat{y}_t - y_t|}{\sum_{t=1}^{T} w_t}. \tag{28}$$

The weighted MAPE incorporates a small smoothing term ϵ to avoid division by zero:

$$\text{MAPE}_w = \frac{100\%}{\sum_{t=1}^{T} w_t} \sum_{t=1}^{T} w_t \left| \frac{\hat{y}_t - y_t}{y_t + \epsilon} \right|. \tag{29}$$

Finally, the generalized R^2 accounts for both weights and regularization:

$$R^2_w = 1 - \frac{\sum_{t=1}^{T} w_t(\hat{y}_t - y_t)^2 + \lambda\|\theta\|_2^2}{\sum_{t=1}^{T} w_t(y_t - \bar{y}_w)^2}, \tag{30}$$

where $\bar{y}_w = \frac{\sum_{t=1}^{T} w_t y_t}{\sum_{t=1}^{T} w_t}$ is the weighted mean of observed values, θ represents model parameters, and λ is a regularization coefficient.

These weighted and regularized metrics provide a more nuanced evaluation of model performance, reflecting both the importance of specific time steps and the stability of model parameters, which is particularly relevant in cross-border e-commerce sales forecasting.

6. Results

This section presents experimental results, including baseline comparisons (Section 6.1), ablation studies (Section 6.2), hyperparameter analysis (Section 6.3), computational efficiency evaluation (Section 6.4), and input sequence length analysis (Section 6.5), demonstrating the model's effectiveness and practical applicability.

6.1. Comparison Results

We evaluated the proposed Enhanced BiLSTM with adaptive attention and Bayesian hyperparameter optimization on a two-year monthly sales dataset (24 time steps) from cross-border e-commerce, which exhibits seasonal fluctuations, promotional periods, and sudden demand spikes. The dataset was split into training (70%) and testing (30%) sets

https://doi.org/10.3390/mca31010017

using a time-based split (first 16 months for training, last 8 months for testing) to prevent data leakage, which is critical for reliable time-series forecasting. All models were executed 10 independent times with different random seeds (1–10) to account for stochasticity in model initialization and training. The reported results correspond to the mean performance across these runs, and statistical significance was assessed using a two-tailed paired t-test ($\alpha = 0.05$) comparing the proposed model with each baseline in Table 4.

Table 4. Forecasting performance comparison among models.

Model	RMSE (Mean ± Std.)	MAE (Mean ± Std.)	MAPE (%) (Mean ± Std.)	R^2 (Mean ± Std.)
ARIMA	21.5 ± 1.2	16.8 ± 0.9	14.9 ± 0.7	0.72 ± 0.03
Standard LSTM	16.8 ± 0.8	13.4 ± 0.6	11.7 ± 0.5	0.81 ± 0.02
BiLSTM	16.5 ± 0.7	12.6 ± 0.5	10.9 ± 0.4	0.84 ± 0.02
BiLSTM + Attention	15.2 ± 0.6	11.9 ± 0.4	10.1 ± 0.3	0.87 ± 0.01
TCN	15.0 ± 0.5	11.5 ± 0.4	9.8 ± 0.3	0.88 ± 0.01
TFT	14.8 ± 0.5	11.3 ± 0.3	9.5 ± 0.2	0.89 ± 0.01
Ours	13.2 ± 0.4	10.2 ± 0.3	8.7 ± 0.2	0.92 ± 0.01

As shown in Table 4, ARIMA, as a classical statistical model, exhibits the highest RMSE (21.5 ± 1.2) and MAPE (14.9 ± 0.7%), indicating limited capability in capturing nonlinear trends and seasonal effects. Standard LSTM and BiLSTM progressively improve accuracy, with RMSE decreasing to 16.5 ± 0.7 and MAPE to 10.9 ± 0.4%. Incorporating a standard attention mechanism further reduces the RMSE to 15.2 ± 0.6 and the MAPE to 10.1 ± 0.3%, highlighting the benefit of focusing on key temporal features. Modern deep learning models, TCN and TFT, achieve competitive results, with RMSE around 15.0–14.8 and MAPE around 9.8–9.5%. The proposed Enhanced BiLSTM achieves the lowest RMSE (13.2 ± 0.4), MAE (10.2 ± 0.3), MAPE (8.7 ± 0.2), and the highest R^2 (0.92 ± 0.01), demonstrating its superior ability to model complex sales patterns, including seasonal fluctuations, promotions, and sudden spikes, providing more reliable forecasts for cross-border e-commerce inventory and demand planning.

To illustrate the forecasting capability of the proposed Enhanced BiLSTM model, Figure 2 presents the predicted versus actual monthly sales for three representative products. The examples include periods with seasonal peaks, promotional events, and occasional sales spikes. The figure shows that the proposed model closely tracks actual sales trends and effectively captures fluctuations, demonstrating robustness in complex scenarios.

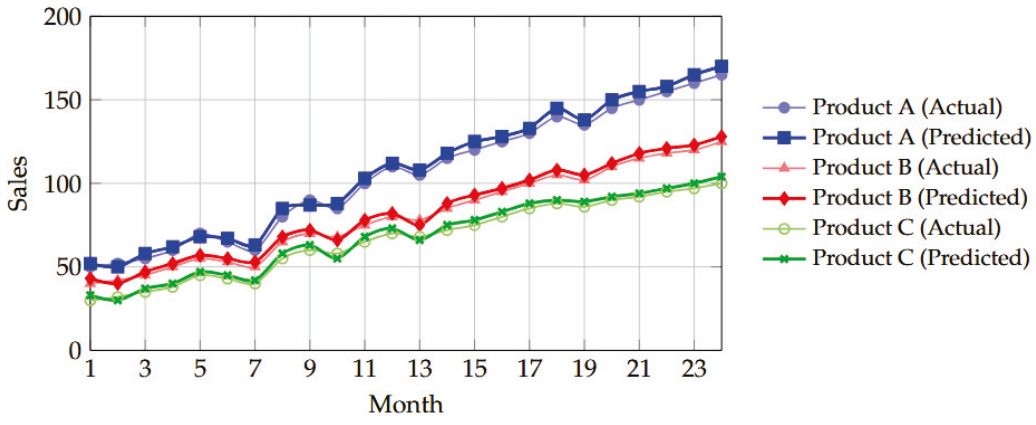


Figure 2. Prediction examples for selected products, showing actual vs. predicted sales over 24 months.

https://doi.org/10.3390/mca31010017

6.2. Model Component Evaluation

To quantify the contribution of each component in the proposed Enhanced BiLSTM model, a series of ablation experiments were conducted. The key modules considered include the adaptive attention mechanism, residual connections, and Bayesian hyperparameter optimization. Each component was systematically removed or replaced to assess its impact on forecasting performance, measured by RMSE, MAE, MAPE, and R^2.

The BiLSTM variant with residual connections but without the adaptive attention mechanism (BiLSTM + Residual) exhibits higher RMSE and MAPE compared to the full model, indicating that the attention module is essential for focusing on critical temporal features such as seasonal trends and promotional spikes. Removing residual connections while retaining the dual-activation attention module (BiLSTM + Attention) results in increased MAE and a slight decrease in R^2, demonstrating that residual connections help preserve feature representations and stabilize gradient flow during training. When all model components are included but Bayesian hyperparameter optimization is omitted (Enhanced BiLSTM w/o BO), the model shows slightly degraded RMSE and MAPE, highlighting the importance of automated hyperparameter tuning in achieving optimal generalization across diverse sales patterns. The standard BiLSTM without attention, residual connections, or Bayesian optimization serves as a baseline and shows the lowest predictive performance across all metrics.

Table 5 clearly shows the incremental contributions of each component. Introducing the adaptive attention mechanism reduces RMSE from 15.8 to 15.2 and MAPE from 10.3% to 10.1%, emphasizing its role in capturing relevant temporal dependencies. Adding residual connections further improves accuracy and stability, as reflected in the reduction of MAE to 11.9 and an increase of R^2 to 0.87. Incorporating Bayesian hyperparameter optimization allows the model to achieve its best performance, with RMSE of 13.2, MAE of 10.2, MAPE of 8.7%, and R^2 of 0.92, demonstrating that the full proposed Enhanced BiLSTM effectively integrates all modules to provide robust and accurate sales forecasts.

Table 5. Ablation study results for Eenanced BiLSTM.

Model Variant	RMSE	MAE	MAPE (%)	R^2
Base BiLSTM	16.5	12.6	10.9	0.84
BiLSTM + Residual	15.8	12.0	10.3	0.86
BiLSTM + Attention	15.2	11.9	10.1	0.87
Enhanced BiLSTM w/o BO	14.0	10.9	9.2	0.90
Proposed Enhanced BiLSTM	13.2	10.2	8.7	0.92

6.3. Hyperparameter Influence Evaluation

To examine how key hyperparameters affect the performance of the proposed Enhanced BiLSTM model, we conducted a series of sensitivity analyses. The hyperparameters considered include the learning rate, hidden-layer size, attention-layer dimension, and dropout rate. Each hyperparameter was varied within a reasonable range while keeping other parameters fixed to evaluate its effect on RMSE, MAE, MAPE, and R^2.

Figure 3 presents the relationship between selected hyperparameters and RMSE/MAE metrics. As shown, the model performance is sensitive to the choice of learning rate. A learning rate that is too large or too small leads to slower convergence or suboptimal solutions. Similarly, increasing the hidden layer size generally improves predictive accuracy up to a certain point, beyond which overfitting may occur. The attention layer dimension significantly affects the model's ability to capture temporal dependencies, and an appropriately chosen dropout rate helps prevent overfitting while maintaining stable training.

https://doi.org/10.3390/mca31010017

These results highlight the importance of automated hyperparameter tuning. By employing Bayesian optimization, the model can efficiently search the hyperparameter space and identify configurations that achieve near-optimal forecasting performance, avoiding manual trial-and-error and ensuring robust generalization across diverse cross-border e-commerce datasets. To explore the effect of key hyperparameters on the performance of the proposed Enhanced BiLSTM model, we conducted a series of sensitivity analyses. The hyperparameters considered include the learning rate, hidden-layer size, attention-layer dimension, and dropout rate. Each hyperparameter was varied within a reasonable range while keeping other parameters fixed to evaluate its effect on RMSE, MAE, MAPE, and R^2.

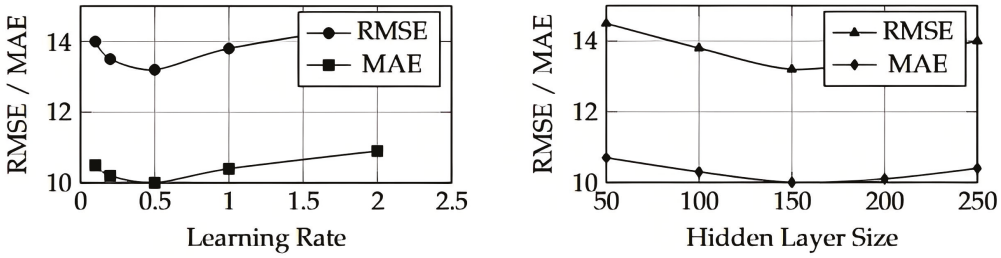


Figure 3. Sensitivity analysis of key hyperparameters on RMSE and MAE for the Enhanced BiLSTM model.

Table 6 shows the effect of selected hyperparameters on RMSE and MAE. The model is sensitive to the learning rate. A learning rate that is too large or too small leads to slower convergence or suboptimal solutions. For example, 0.01 or 0.0001 increases both RMSE and MAE compared to 0.001. Increasing the hidden layer size to 256 slightly improves accuracy. A smaller size of 64 reduces performance. The attention layer dimension also matters. A larger layer of 128 improves results, while a smaller layer of 32 worsens them. Dropout affects stability. A rate of 0.2 works best. Higher (0.4) or lower (0.1) rates slightly reduce performance. These results highlight the importance of carefully selecting hyperparameters for optimal model performance.

Table 6. Effect of hyperparameter settings on model performance.

Hyperparameter	Learning Rate	Hidden Layer Size	Attention Layer Dimension	Dropout Rate	RMSE	MAE
Default Setting	0.001	128	64	0.2	13.2	10.2
Higher Learning Rate	0.01	128	64	0.2	14.5	11.0
Lower Learning Rate	0.0001	128	64	0.2	15.0	11.3
Larger Hidden Layer	0.001	256	64	0.2	13.5	10.5
Smaller Hidden Layer	0.001	64	64	0.2	14.0	11.0
Larger Attention Layer	0.001	128	128	0.2	13.0	10.0
Smaller Attention Layer	0.001	128	32	0.2	14.0	11.2
Higher Dropout Rate	0.001	128	64	0.4	13.5	10.6
Lower Dropout Rate	0.001	128	64	0.1	14.0	11.0

These results highlight the importance of automated hyperparameter tuning. By employing Bayesian optimization, the model can efficiently search the hyperparameter space and identify configurations that achieve near-optimal forecasting performance, avoiding manual trial-and-error and ensuring robust generalization across diverse cross-border e-commerce datasets.

https://doi.org/10.3390/mca31010017

6.4. Training Time and Computational Efficiency Evaluation

Along with forecasting accuracy, the computational efficiency of the Enhanced BiLSTM model is vital for practical use in cross-border e-commerce. We evaluated the training time per epoch, inference time per sample, and total number of trainable parameters for the proposed model and all baseline algorithms.

Table 7 summarizes the results. Classical statistical models such as ARIMA have minimal parameter counts and fast computation but suffer from lower predictive accuracy. Standard LSTMs and BiLSTMs introduce additional parameters and require longer training times, whereas models with attention mechanisms and modern architectures, such as TCN and TFT, further increase computational cost. The proposed Enhanced BiLSTM strikes a balance by achieving high predictive performance with a moderate increase in parameters and training time. Its inference speed remains suitable for real-time or near-real-time forecasting, making it a practical choice for operational deployment.

Table 7. Computational efficiency comparison across models.

Model	Train Time/Epoch (s)	Inference Time/Sample (ms)	Parameters (M)
ARIMA	0.5	0.01	0.001
LSTM	12	0.15	1.2
BiLSTM	14	0.18	2.3
BiLSTM + Att	18	0.22	2.5
TCN	20	0.25	2.8
TFT	22	0.30	3.1
Enhanced BiLSTM	19	0.23	2.6

These results demonstrate that although the Enhanced BiLSTM model introduces additional parameters due to the adaptive attention and residual modules, the computational overhead remains moderate. The model maintains reasonable training and inference times while delivering superior predictive performance, confirming its suitability for practical cross-border e-commerce forecasting tasks.

6.5. Effect of Sequence Length/Time Window

The length of the input sequence, or historical observation window, can significantly impact forecasting performance in cross-border e-commerce sales. Short windows may miss long-term trends or seasonal patterns, while overly long windows can introduce noise and increase computational cost. To investigate this, we evaluated the Enhanced BiLSTM model using different time windows: 3 months, 6 months, and 12 months of historical sales data.

Figure 4 shows the RMSE and MAE for each window length. The results indicate that a 6-month window generally provides the best trade-off, capturing both short-term fluctuations and seasonal trends without overfitting. Shorter windows (3 months) fail to capture recurring seasonal effects, leading to higher RMSE and MAPE, whereas longer windows (12 months) yield only marginal improvement but increase model complexity. These results emphasize the importance of choosing an appropriate sequence length to balance short-term responsiveness and long-term pattern recognition in cross-border e-commerce sales forecasting.

Additionally, Table 8 presents a summary of the performance metrics (RMSE, MAE, and MAPE) for each sequence length. It shows that the 6-month window achieves the best performance in terms of both RMSE and MAPE, further confirming the findings from the figure.

https://doi.org/10.3390/mca31010017

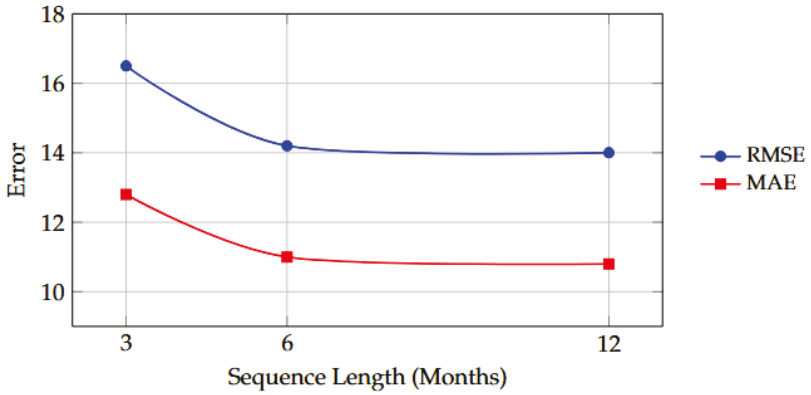


Figure 4. Effect of input sequence length on forecasting performance for the Enhanced BiLSTM model.

Table 8. Impact of sequence length on model performance.

Sequence Length	RMSE	MAE	MAPE (%)
3 months	16.2	12.8	11.2
6 months	13.2	10.2	8.7
12 months	13.0	10.1	8.5

7. Conclusions

This study proposes an enhanced BiLSTM model for cross-border e-commerce sales forecasting, which integrates bidirectional temporal modeling, an adaptive attention mechanism, and Bayesian optimization for hyperparameter tuning. The model overcomes the limitations of traditional time-series models and existing deep learning approaches, effectively capturing complex nonlinear sales patterns, seasonality, and demand fluctuations. The following conclusions are drawn from the experimental results:

1. The model outperforms six baseline models, achieving a RMSE of 13.2, MAE of 10.2, MAPE of 8.7%, and a coefficient of R^2 of 0.92, demonstrating its superior accuracy in cross-border e-commerce sales forecasting.
2. Ablation studies show the contribution of each component: the adaptive attention mechanism improves prediction accuracy, residual connections reduce errors, and Bayesian optimization enhances overall model performance.
3. The model strikes an effective balance between performance and computational efficiency. With fast training and inference times, it demonstrates practicality for real-world deployment, making it suitable for large-scale, real-time forecasting tasks.

Despite these advantages, the model has some limitations. It relies solely on historical sales data and does not incorporate external factors such as macroeconomic indicators or social media sentiment, which may impact demand forecasting in cross-border e-commerce. The fixed input sequence length may also not be optimal for products with irregular demand patterns, and scalability challenges may arise in large-scale e-commerce platforms.

Future work will address these limitations by integrating multi-source data, adapting the model to specific product categories, and exploring dynamic attention mechanisms. We also plan to optimize the model for large-scale deployment and test it on a wider range of cross-border e-commerce datasets. Additionally, we will explore Transformer-based models to handle longer datasets and capture global dependencies more effectively.

https://doi.org/10.3390/mca31010017

Author Contributions: Conceptualization: H.H., J.C., and C.X.; Methodology: H.H. and J.C.; Software: H.H.; Formal analysis: J.C.; Investigation: H.H.; Resources: C.X.; Data curation: J.C.; Writing—original draft preparation: H.H. and J.C.; Writing—review and editing: H.H., J.C., and C.X.; Visualization: H.H.; Supervision: C.X.; Project administration: C.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All datasets used in this study are publicly available from the Amazon Sales Dataset on Kaggle: https://www.kaggle.com/datasets/karkavelrajaj/amazon-sales-dataset (accessed on 15 June 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Qi, K.; Wu, W.; Ni, Y. SymOpt-CNSVR: A Novel Prediction Model Based on Symmetric Optimization for Delivery Duration Forecasting. *Symmetry* **2025**, *17*, 1608.
2. Zhu, L.; Liu, W.; Tan, H.; Hu, T. A novel Bayesian optimization prediction framework for four-axis industrial robot joint motion state. *Complex Intell. Syst.* **2024**, *10*, 4867–4881. [CrossRef]
3. Gong, Z. Optimization of cross-border E-commerce (CBEC) supply chain management based on fuzzy logic and auction theory. *Sci. Rep.* **2024**, *14*, 14088.
4. Li, J.; Li, J.; Li, J.; Zhang, G. Bayesian-Optimized GCN-BiLSTM-Adaboost Model for Power-Load Forecasting. *Electronics* **2025**, *14*, 3332.
5. Guo, X.; Mo, Y.; Yan, K. Short-Term Photovoltaic Power Forecasting Based on Historical Information and Deep Learning Methods. *Sensors* **2022**, *22*, 9630. [CrossRef]
6. Kwon, H.; Do, T.N.; Won, W.; Kim, J. An optimization model for the market-responsive operation of naphtha cracking process with price prediction. *Chem. Eng. Res. Des.* **2022**, *188*, 681–693. [CrossRef]
7. Lou, G.; Lin, W.; Huang, G.; Xiang, W. A two-stage online remaining useful life prediction framework for supercapacitors based on the fusion of deep learning network and state estimation algorithm. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106399. [CrossRef]
8. Gena, X.; Zhang, Y. Sales Forecasting using ARIMA Models: A Case Study in the Retail Industry. *J. Retail Anal.* **2015**, *12*, 45–56.
9. Rostan, P.; Rostan, A.; Nurunnabi, M. Options Trading Strategy Based on ARIMA Forecasting. *PSU Res. Rev.* **2020**, *4*, 111–127. [CrossRef]
10. Wang, P.; Xu, Z. A Novel Consumer Purchase Behavior Recognition Method Using Ensemble Learning Algorithm. *Math. Probl. Eng.* **2020**, *2020*, 6673535. [CrossRef]
11. Cho, M.; Kim, C.; Jung, K.; Jung, H. Water Level Prediction Model Applying a Long Short-Term Memory (LSTM)—Gated Recurrent Unit (GRU) Method for Flood Prediction. *Water* **2022**, *14*, 2221. [CrossRef]
12. Tsai, P.-F.; Gao, C.-H.; Yuan, S.-M. Stock Selection Using Machine Learning Based on Financial Ratios. *Mathematics* **2023**, *11*, 4758. [CrossRef]
13. Khandelwal, V.; Chaturvedi, A.K.; Gupta, C.P. Amazon EC2 Spot Price Prediction Using Regression Random Forests. *IEEE Trans. Cloud Comput.* **2020**, *8*, 59–72. [CrossRef]
14. Qin, C.; Chen, L.; Cai, Z.; Liu, M.; Jin, L. Long short-term memory with activation on gradient. *Neural Netw.* **2023**, *164*, 135–145. [CrossRef]
15. Su, Y.; Kuo, C.-C.J. On extended long short-term memory and dependent bidirectional recurrent neural network. *Neurocomputing* **2019**, *356*, 151–161. [CrossRef]
16. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
17. Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian optimization of machine learning algorithms. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Lake Tahoe, NV, USA, 3–8 December 2012; Volume 25, pp. 2951–2959.
18. Li, G.; Li, N. Customs classification for cross-border e-commerce based on text-image adaptive convolutional neural network. *Electron. Commer. Res.* **2019**, *19*, 779–800.
19. Jiang, C. Research on sales forecasting and consumption recommendation system of e-commerce agricultural products based on LSTM model. *GeoJournal* **2025**, *90*, 93. [CrossRef]
20. Li, J.; Pan, Y.; Yang, Y.; Tse, C.H. Digital platform attention and international sales: An attention-based view. *J. Int. Bus. Stud.* **2022**, *53*, 1817–1835. [CrossRef]

21. Singh, K.; Booma, P.M.; Eaganathan, U. E-commerce system for sale prediction using machine learning technique. *J. Phys. Conf. Ser.* **2020**, *1712*, 012042. [CrossRef]
22. He, M.; Zhuang, L.; Yang, S.; Xu, Z.; Li, W.; Lu, J. An energy-efficient VNE algorithm based on bidirectional long short-term memory. *J. Netw. Syst. Manag.* **2022**, *30*, 45. [CrossRef]
23. Jürgensmeier, L.; Bischoff, J.; Skiera, B. Opportunities for self-preferencing in international online marketplaces. *Int. Mark. Rev.* **2024**, *41*, 1118–1132. [CrossRef]
24. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471. [CrossRef] [PubMed]
25. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
26. Bai, S.; Zhuang, F.; He, X. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. In Proceedings of the International Conference on Learning Representations (ICLR 2019), New Orleans, LA, USA, 6–9 May 2019.
27. Lim, B.; Arık, S.Ö.; Loeff, N.; Pfister, T. Temporal Fusion Transformers for interpretable multi-horizon time series forecasting. *Int. J. Forecast.* **2021**, *37*, 1748–1764. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Mathematical and Computational Applications

MDPI

Article

Neuroevolution of Liquid State Machine Based on Neural Configurations and Positions

Carlos-Alberto López-Herrera *, Héctor-Gabriel Acosta-Mesa, Efrén Mezura-Montes and Jesús-Arnulfo Barradas-Palmeros

Artificial Intelligence Research Institute, University of Veracruz, Campus Sur, Calle Paseo Lote II, Sección Segunda No. 112, Nuevo Xalapa, Veracruz 91097, Mexico; heacosta@uv.mx (H.-G.A.-M.); emezura@uv.mx (E.M.-M.); zs23000652@estudiantes.uv.mx (J.-A.B.-P.)
* Correspondence: zs23000650@estudiantes.uv.mx

Abstract

Liquid State Machines (LSMs), a reservoir computing model based on recurrent spiking neural networks, provide a powerful framework for solving spatiotemporal classification tasks by leveraging rich temporal dynamics and event-driven processing. Although the traditional LSM formulation assumes a fixed, randomly generated reservoir, recent research has explored optimization strategies to improve liquid dynamics. However, most existing approaches focus primarily on optimizing synaptic connectivity or reservoir structure, while the role of neuron-level parameters remains largely underexplored. This work proposes a neuroevolutionary strategy based on a Genetic Algorithm (GA) that encodes both neuron configurations and their spatial positions, explicitly treating neuron-level parameters as optimization targets. By evolving neuron-specific parameters and spatial positions, the method induces diverse reservoir dynamics. Unlike approaches that directly optimize synaptic weights, the proposed representation maintains an encoding whose dimensionality scales linearly with the number of neurons. The approach was evaluated on four synthetic benchmark tasks, including one Frequency Recognition task and three Pattern Recognition tasks, using compact reservoirs composed of only 20 Leaky Integrate-and-Fire neurons. Despite the small reservoir size, the method achieved state-of-the-art or highly competitive performance, reaching mean accuracies of up to 99.71%. In the most challenging case (PR12), performance improved when the reservoir size was increased to 64 neurons. The method was further evaluated on two real-world datasets, N-MNIST and the Free Spoken Digit Dataset (FSDD), using reservoirs of 300 neurons, achieving 90.65% and 81.47% accuracy, respectively, while using substantially fewer neurons than many existing LSM-based approaches. These results highlight the potential of evolving neuron configurations and spatial organization to produce compact and effective liquid reservoirs.

Keywords: reservoir computing; spiking neural network; neuroevolution; genetic algorithm

1. Introduction

Reservoir Computing (RC) is an emerging machine-learning paradigm that originated in the early 2000s [1] and leverages high-dimensional dynamical systems (reservoirs) for nonlinear transformations [2] to learn spatiotemporal features and patterns in complex time series [3]. It is built on three levels: an Input layer, a Reservoir layer, and a Readout layer. The Input layer maps the input data into the reservoir via sparse, randomly weighted connections [4]. It is responsible for preprocessing data and ensuring that incoming

signals activate the reservoir properly [5]. The Reservoir layer is the computational core of this model, as it is responsible for transforming the input into a high-dimensional representation, also referred to as states, where complex patterns and relationships are more easily discernible [6]. It comprises many interconnected neurons (computational units) with fixed-weight synapses (or connections). By this definition, the reservoir can be considered a type of Artificial Neural Network (ANN), specifically a Recurrent Neural Network (RNN), which allows the reservoir to preserve historical information [2]. Finally, the Readout layer serves as the output segment of the model, mapping the states produced by the Reservoir layer into a lower number of desired outputs [6]. Typically, a simple linear model, such as the Perceptron [7], is sufficient as a Readout function, and this is the only part of the model that requires a training process, which makes the RC a highly computationally efficient approach [8] when compared with Deep Learning (DL) algorithms. This is depicted in Figure 1.

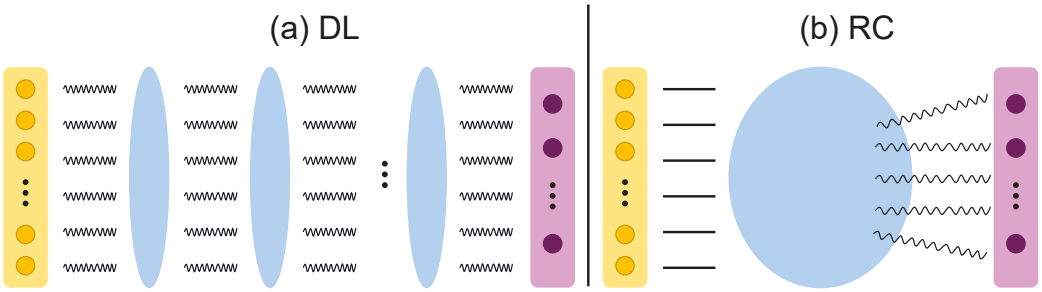


Figure 1. An architectural comparison between DL approaches and RC models. Straight and wavy lines denote fixed and trainable synaptic connections, respectively. In DL, all connections require a trained process, which increases the complexity. While in RC, only the synaptic connections of the Readout layer required the training process. Adapted from [3].

Historically, two main RC representatives have been independently developed: the Echo State Network (ESN) [4] and the Liquid State Machine (LSM) [2]. The principal difference between them is that ESN uses a traditional RNN as a reservoir, where the neurons are continuous-valued units with traditional activation functions, such as the Rectified Linear Unit (ReLU). In contrast, LSM uses reservoirs, often referred to in this context as Liquids, of Recurrent Spiking Neural Networks (R-SNN) with biologically inspired spiking neurons. While ESN effectively models continuous-valued temporal signals, LSM enables precise spike timing and millisecond-scale temporal resolution. Moreover, experimental comparisons have shown that the performance of reservoir computing models depends strongly on reservoir dynamics and connectivity [9].

Although deep recurrent architectures such as Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and Transformer-based models have achieved remarkable performance on temporal tasks, they require full backpropagation through time, large labeled datasets, and substantial computational resources. In contrast, LSMs decouple the nonlinear temporal representation (reservoir) from the learning process (readout), thereby enabling efficient training while preserving rich dynamical representations and reducing computational cost [10]. Moreover, when dealing with event-based or spike-driven data, spiking reservoirs naturally preserve temporal precision without requiring continuous-valued approximations, making LSMs suitable for resource-constrained environments, such as embedded systems, edge devices, and neuromorphic hardware platforms, where memory capacity, energy consumption, and processing power are limited [11,12]. Therefore, rather than competing directly with fully trained deep architectures, this work focuses on enhancing the internal dynamics of LSM reservoirs to improve performance while preserving computational efficiency.

https://doi.org/10.3390/mca31020065

Spiking neuron models aim to capture the temporal dynamics of biological neurons by representing information through discrete spikes. Among the models proposed in the literature, the Hodgkin–Huxley model [13] provides a detailed biophysical description of neuronal activity but requires significant computational resources. For this reason, simplified models such as the Leaky Integrate-and-Fire (LIF) [14] neuron are commonly used in large-scale simulations, as they preserve the essential temporal dynamics while maintaining computational efficiency. In this work, we adopt the LIF model implemented in the NEST simulator [15].

Despite the LIF model's simplicity, which makes it less plausible, its neuron parameters can encapsulate different temporal dynamics. In turn, this contributes to the proficiency of LSM in several areas, such as speech recognition [16–18], image and vision recognition [19–21], biomedical applications [22], robotics and control [23,24], among others. The possibility of avoiding explicit training is especially advantageous when using LSM since the synaptic plasticity mechanisms of R-SNNs require complex and computationally demanding training algorithms [25–27].

Since their formal introduction, LSMs have been associated with two cardinal attributes: the Separation Property (SP) and the Approximation Property (AP). In the context of classification tasks, SP measures the liquid's ability to distinguish inputs belonging to different classes, whereas AP relates to the similarities in the liquid's representation of two inputs belonging to the same class [28]; in other words, SP ensures that different inputs produce different liquid states and AP ensures that similar inputs produce similar liquid states.

Both of these metrics depend, to different degrees, on the internal complexity and dynamics of the liquid; consequently, the overall performance of an LMS is strongly influenced by its reservoir configuration. In its original formulation, Maass et al. [23] proposed that a randomly generated liquid would suffice to achieve adequate performance for a given task. That is, classical LSMs assume a fixed, untrained reservoir. However, this approach does not offer any solid support or assurance for complex tasks.

Later, Norton & Ventura [29–31] introduced an unsupervised adaptation mechanism known as Separation Driven Synaptic Modification (SDSM). Their method iteratively improves the SP by modifying synaptic weights using Hebbian Learning, while keeping the neuron model and parameters unchanged. This approximation introduces a form of reservoir training that remains unsupervised and self-organized. They tested their implementation using synthetic datasets proposed as frequency and pattern recognition tasks.

In contrast, recent approaches treat the liquid not as a fixed reservoir but as a trainable R-SNN, thereby relaxing the canonical LSM definition. This relaxation enables the enhancement of the liquid either through training-adaptation processes or by optimization techniques. Notably, R-SNNs have been successfully applied to temporal pattern recognition and decision-making tasks using learning rules such as Spike-Time Dependent Plasticity (STDP) and surrogate gradient methods [26,27,32,33]. However, these approaches typically involve computationally expensive training processes and require careful parameter tuning, which limits their scalability and practical applicability.

Among optimization strategies, metaheuristics such as Evolutionary Computation (EC) have shown promising results. EC encompasses various subfields, with Evolutionary Algorithms (EAs) and Swarm Intelligence (SI) algorithms being two of the most prominent. Generally, these methods take a set of potential solutions (or individuals) for a given problem, where a fitness function measures the aptitude of each individual. This way, the individuals are ranked according to a criterion. Therefore, through this competition, individuals are combined, eliminated, and substituted by new ones in every execution (generation or epoch) of the algorithm. Figure 2 depicts the general elements that most

https://doi.org/10.3390/mca31020065

EA techniques follow. These algorithms are well-suited to LSM and R-SNN optimization due to their ability to explore high-dimensional spaces without relying on gradient information [32].

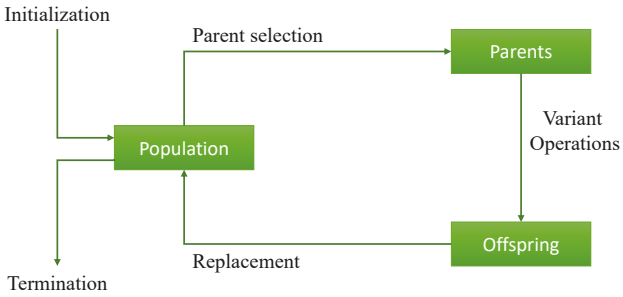


Figure 2. A general schematic of an EA. The initialization forms the first individuals (solutions) of the population. A selection of individuals will form a subset of parents from which new individuals will be generated through variations (crossover and mutation). Afterward, the offspring will replace some individuals, forming the new population. This process is repeated until a termination criterion is reached [34].

Moreover, Neuroevolution (NE) utilizes EC methods to optimize ANNs by searching the space of potential solutions of architectures and other parameters, such as weights [35], for a given task. Yaqoob & Wróbel [36] used a Genetic Algorithm (GA) to optimize the weights and connections in an R-SNN. The task was to detect a temporal pattern in the form of consecutive symbols in a continuous input stream. To achieve this, each symbol generated a temporal activation, delivered by the input neurons, that was then propagated throughout the rest of the network. They found that very small SNNs (with only a few neurons) could accurately recognize the pattern. As a neural model, they used AdEx neurons, an adaptation of the LIF model, with fixed parameters. Although the representation allowed for unlimited connections and nodes, it limited the number of interneurons to two.

Recent research has explored hierarchical extensions of reservoir computing, where multiple reservoirs are stacked to increase representational capacity [37]. Tian et al. [38] proposed an NE framework for designing a Multi-Liquid State Machine (MLSM); different liquids are disposed of in a layer-like structure. The authors propose a three-stage search strategy: first, a random architecture is optimized by moving liquids from one layer to another. Second, the number of neurons in each liquid of the LSM of the past stage is optimized while preserving the total number of neurons. Third, the framework optimizes the liquids' internal parameters of the last LSM, such as the percentage of connectivity and excitatory neuron ratio. Each optimization step uses a Simulated Annealing (SA) algorithm. The framework is evaluated on two real-world benchmark datasets: the image dataset N-MNIST and the Free Spoken Digit Dataset (FSDD). The results show that the proposed framework can effectively search for high-performance LSM architectures while reducing the computational cost compared to manual search. Nevertheless, they rely on a complex structure comprised of 1000 spiking neurons and only consider connectivity in a limited manner (e.g., as a connectivity percentage), without exploring neuron configurations.

Zhou et al. [39] presented a framework named Generative Liquid State Machine (GLSM). The paper proposes a Cooperative Co-Evolutionary (CCE) algorithm for designing the reservoir architecture of LSMs using a Random Forest Surrogate (RFS) model. The algorithm treats the reservoir as a modular structure comprising subnetworks, utilizing the RFS model to minimize the computational cost of evaluating candidate architectures. A GA is used to evolve structures, the number of neurons, neuron thresholds, and neuron delays, while a learning algorithm is used to modify synaptic weights and delays. The results

https://doi.org/10.3390/mca31020065

demonstrate that the proposed method effectively optimizes the LSM reservoir architecture. Nevertheless, this approach contemplates a fixed set of subnetworks and sub-reservoir configurations, limiting the neuron's possibilities. Although they optimize the number of neurons, the exact number used in their implementations is not clearly specified. Modular GLSM-based reservoirs have also been explored for multi-task learning [40], where shared reservoir structures are used to address multiple tasks within a single framework.

Most recently, Alvarez-Canchila et al. [41] employed Particle Swarm Optimization (PSO), an SI approach, to optimize the SP. Each particle (solution) contained the synaptic weights of all connections in a liquid built with 64 neurons, predefined before every PSO run. They test it using the same tasks as Norton & Ventura [31], achieving accuracy levels comparable to those reported previously.

In summary, LSMs present several advantages for spatiotemporal learning tasks compared to conventional machine learning models. Their event-driven computation allows neurons to process information only when spikes occur [2,3], thereby reducing unnecessary operations and making them particularly suitable for implementation on neuromorphic hardware, where energy consumption can be significantly lower than in traditional deep learning architectures [11,12]. Moreover, the reservoir computing paradigm simplifies training by decoupling the dynamic transformation performed by the reservoir from the learning stage performed by the readout [3,10]. However, most works focus primarily on optimizing synaptic weights [31,41] or on predefined architectural structures [38,39], while neuron configuration and neuron positions in the liquid have received limited attention. In particular:

- Weight-based optimization methods (PSO, GA over synapses) assume a fixed neuron model and topology.
- Multi-liquid approaches optimize macro-architecture but do not explore intra-liquid neuronal diversity.
- Modular approaches constrain the search space to predefined substructures.

Consequently, the joint optimization of neuron configurations and spatial positions remains unexplored.

In this work, a GA method is proposed as an NE approach to optimize liquid performance. The decision to employ a GA is based on its simplicity, proven effectiveness, and modular structure. Its population-based nature enables broad exploration of the search space, while its straightforward operators enable flexible adaptation to the neuroevolutionary context. This optimization process targets two key aspects of the liquid: the neuron configuration and the neuron positions within it. By allowing each neuron to operate under distinct parameter configurations, the resulting liquid can exhibit neuronal heterogeneity, meaning that individual neurons follow different dynamical regimes. Such heterogeneity has been suggested to enrich the computational dynamics of spiking neural networks and reservoir systems [42].

Neuron positions, in turn, contribute to establishing spatial diversity that shapes the reservoir's topological structure and connectivity. By determining the spatial arrangement of neurons, neuron positions shape inter-neuronal distances, which in turn affect both the probability of connections and the strength of synaptic interactions, thereby impacting the overall liquid dynamics. Unlike some approaches that primarily optimize synaptic weights, this work focuses on evolving neuron-specific configurations and spatial organization within the liquid. This design keeps the dimensionality of the encoding proportional to the number of neurons, which helps control the size of the search space and makes the optimization tractable even for larger reservoirs.

In previous work [43], an ablation analysis evaluated the impact of neuron configurations and positions. The results showed that evolving neuron configurations alone

https://doi.org/10.3390/mca31020065

achieved performance comparable to using configurations together with spatial positions, with no statistically significant difference between these variants. In contrast, evolving only the neuron positions yielded lower performance. These findings suggest that neuron configurations play the primary role, while spatial positions provide complementary structural diversity.

To validate the proposed method, we evaluate it on four synthetic classification tasks, similar to Norton & Ventura [31] and Alvarez-Canchila et al. [41], and on two classical real-world datasets, N-MNIST and FSDD.

The contributions of this work are as follows:

- We introduce a GA-based neuroevolution framework that optimizes both neuron configurations and spatial placement within LSM reservoirs, in contrast to previous approaches that primarily optimize synaptic weights or connectivity structures.
- We demonstrate that evolving neuron configurations and spatial positions lead to improved classification performance, highlighting neuron parameter configurations and spatial organization as effective optimization targets in LSM design without requiring direct optimization of synaptic weights.
- We propose a compact encoding scheme that enables indirect optimization of connectivity and synaptic interactions, achieving a linear scaling of the encoding size compared to the quadratic growth of direct weight-based representations.
- We show that competitive performance can be achieved with compact reservoirs composed of a small number of neurons, providing a more stable and efficient alternative to existing approaches that rely on substantially larger reservoirs.

The rest of this paper is organized as follows: Section 2 presents the materials and methods used in this study, beginning with a description of the datasets employed (Section 2.1), followed by an overview of the LSM model (Section 2.2) and the GA utilized for optimization (Section 2.3). The section concludes with a detailed explanation of the overall methodology (Section 2.4). Section 3 reports the experimental results, including performance metrics and comparative evaluations. In Section 4, we analyze and interpret the outcomes, highlighting the implications of the proposed approach. Finally, we outline directions for future work.

2. Materials and Methods

2.1. Database

Two synthetic multi-class classification problems are tested to assess the proposal's performance. These tasks, described in Norton & Ventura [31], are called Frequency Recognition (FR) and Pattern Recognition (PR). The FR problem consists of four inputs and five classes (referred to as FR5), where each input can have a fire rate of either slow or high, depending on the class. Table 1 shows the channel's frequency state distribution for all classes. The high frequency was set at 100 Hz, and the low frequency at 10 Hz. For each class, 1000 samples of 100 ms were created using a template and then jittered (In this context, jittering refers to a small variation.) in frequency. An example of each class is presented in Figure 3. Part of the challenge in this task is to detect a variety of frequencies while ignoring a channel (Channel 4) that never exhibits a high frequency.

The PR problem has eight input channels and a different number of classes. To name them, PR4, PR8, and PR12 will be referred to as the fourth, eighth, and twelfth class problems, respectively. A random spike pattern is generated for each input channel, and instances are created by applying perturbations to these patterns. Similar to FR5, 1000 instances per class were created. Figure 4 shows an example of three instances per

https://doi.org/10.3390/mca31020065

class of the PR4 task. These problems are, compared with FR5, more complex since each spike is individually jittered.

Table 1. Frequency states of channels for the five classes in the FR5 task. "H" and "L" are high (100 Hz) and low frequency (10 Hz), respectively.

	Channel 1	**Channel 2**	**Channel 3**	**Channel 4**
Class 1	H	L	L	L
Class 2	L	H	L	L
Class 3	H	H	L	L
Class 4	L	L	H	L
Class 5	H	L	H	L

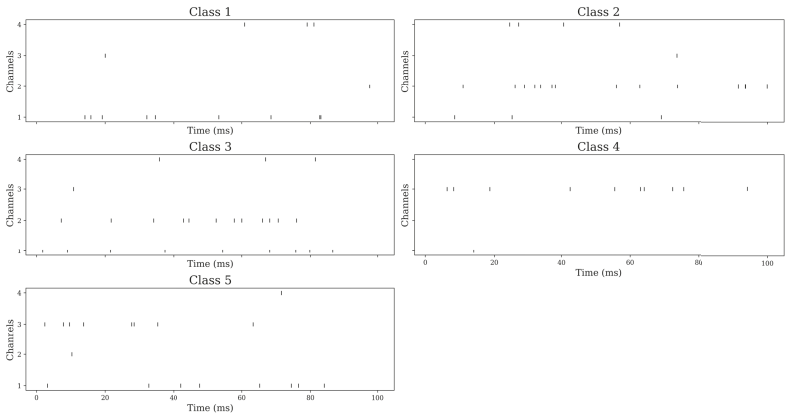


Figure 3. Example of an instance per class for the FR5 problem. It shows the spikes of each chanel on the signal.

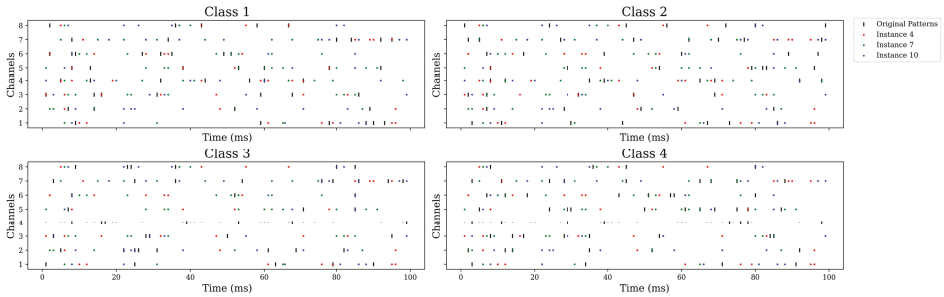


Figure 4. Example of classes for the PR4 task. Each class is based on a unique spike pattern (represented by bars in black) for every input channel. Three instances per class are shown as red, green, and blue dots.

These classification tasks challenge the model from different fronts: the FR dataset challenges the model with a variety of combinations of frequencies as well as the task of ignoring one channel since channel 4 always presents a low frequency [31]; the PR datasets increase the complexity by incrementing the number of input channels and classes, specifically in PR8 and P12.

Furthermore, tests on two real-world datasets are also included. The MNIST dataset is a benchmark commonly used in machine learning for image classification and pattern recognition tasks. It consists of 70,000 grayscale images of handwritten digits (0–9). The dataset images have 28 $\times$ 28 pixels. The Neuromorphic MNIST (N-MNIST) dataset, introduced by Orchard et al. [44], records the standard MNIST dataset using a neuromorphic vision sensor Asynchronous Time-based Image Sensor (ATIS).

https://doi.org/10.3390/mca31020065

The FSDD [45] is an open audio dataset designed for benchmarking speech recognition systems. It consists of spoken digits (0–9) pronounced by multiple speakers in English. Each sample is a single-channel waveform recorded at 8 kHz. The dataset consists of 2000 audio recordings.

The preparation of the FSDD involved two main steps. First, the Mel-Frequency Cepstral Coefficients (MFCCs) were extracted from the samples to capture relevant spectral features. Then, the extracted features were converted into spike trains using a Poisson rate coding scheme, where the firing rate was proportional to the signal intensity. This procedure was adopted from the method of Tian et al. [38]. In contrast, the N-MNIST dataset required minimal preprocessing: the original 28×28 event-based frames were downsampled to 17×17, reducing the number of input channels to 289.

2.2. LSM

The LSM is a framework that specializes in solving spatiotemporal tasks and is based on RC. Therefore, there are two distinct parts of an LSM: the reservoir and the readout. The reservoir, also known as liquid, serves as a kernel of spiking neurons that project input data into liquid states, which are high-dimensional representations of the internal states of the liquid. On the other hand, the readout serves as a memory-less function for interpreting liquid states, specifically trained for a given task. The pipeline of an LSM, shown in Figure 5, is as follows:

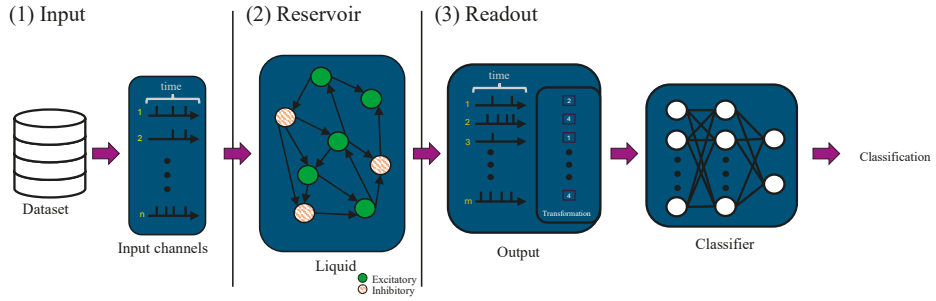


Figure 5. Workflow of the LSM model used in this work. The diagram provides a conceptual illustration of the main processing stages: input spike channels, liquid reservoir, and readout classifier. The process begins with multichannel spike train inputs obtained from the dataset, where each channel connects to 50% of randomly selected neurons in the reservoir with input weights sampled from $U(1, 50)$. The liquid is composed of excitatory and inhibitory neurons interconnected with distance-based synaptic weights, where neuron polarity determines the sign of postsynaptic connections. For state extraction, spike responses from excitatory neurons are recorded across the full simulation window and used to construct the liquid state representation. These states are then converted into static vectors using spike-rate encoding and passed to the readout layer, implemented as a one-vs-rest perceptron classifier. The elements shown in the diagram are illustrative and do not represent the exact number of neurons used in the experiments.

1. **Input**: A dataset provides multichannel temporal inputs, each corresponding to a distinct signal. If the dataset consists of continuous signals, then the data must be encoded as spike trains. Then, the channels project the stream data into randomly selected neurons in the liquid.
   In this work, a 50% Input-to-Liquid connectivity ($Conn_{I-L}$) is used, meaning that each channel connects to half the number of randomly selected neurons in the liquid. The corresponding input synaptic weights are sampled from a uniform distribution $U(1, 50)$.
2. **Reservoir**: The liquid represents the core mechanism in an LSM. It consists of an R-SNN, which includes both excitatory and inhibitory neurons. Following the classical

https://doi.org/10.3390/mca31020065

LSM formulation introduced by Maass et al. [2], neurons are organized into excitatory and inhibitory populations, where excitatory neurons produce positive (excitatory) postsynaptic connections and inhibitory neurons produce negative (inhibitory) ones. In this work, the LIF neuronal units are randomly placed in a 2D space, and their connections and synaptic values are determined by the distance between neurons. As spikes propagate through this connected network, the liquid transforms inputs into dynamic, high-dimensional internal states, referred to as liquid states. For state extraction, spike responses from excitatory neurons are recorded across the full simulation window and used to construct the liquid state representation, as excitatory neurons provide a more stable and informative representation of the liquid dynamics, while inhibitory neurons primarily regulate internal activity. These state vectors are then provided to the readout stage for classification.
All liquid simulations were conducted using the NEST simulator package [15], employing the standard configuration of the LIF neuron model "iaf_psc_alpha".

3. **Readout**: After the liquid's response, the states are sampled and transformed into a static representation known as a state vector. This process is necessary every time the readout block is memory-less. The most common transformation techniques include the spike count (the total number of spikes) and spike rate (the total number of spikes divided by the signal's length). Each state representation is then passed to a classifier, which is trained to perform a specific task.

Our approach employs the spike rate transformation, which favors the relative activity intensity [46]. Moreover, the readout function is implemented as a one-vs-rest linear classifier composed of n independent Perceptrons, each trained to recognize one class against the rest using Stochastic Gradient Descent (SGD). The classification is fitted against all the other classes, assigning the class of the Perceptron with the greatest confidence. As parameters, the SGD uses a maximum iteration (Max_I) of 1000, a tolerance (Tol) of 1×10^{-3}, and a L2 regularization process with α of 1×10^{-4}. This readout function was selected for its simplicity, which adds value to the liquid process.

A summary of the parameters used in this pipeline is shown in Table 2.

Table 2. Parameters used in the LSM model.

Component	Parameter	Value
Input-to-Liquid connections	Input connectivity	50%
	Input synaptic weights	$U(1, 50)$
Liquid (Reservoir)	Excitatory neurons	80%
	Neuron model	LIF
Neuron parameters (LIF)	Initial potential (V_m)	-70 mV
	Resting potential (E_L)	-70 mV
	Capacity of membrane (C_m)	100 pF
	Reset potential (V_{reset})	-70 mV
	Input Noise (I_{noise})	100 pA
Readout	State Window	100%
	Encoding method	Spike-rate
	Classifier	One-vs-rest Perceptron
	Maximum Iteration	1000
	Tolerance	1×10^{-3}
	Regularization factor (α)	1×10^{-4}

2.3. GA

This section outlines the proposed GA adopted in this study. The GA serves as the optimization strategy for configuring the liquid's structure, guiding the search toward

https://doi.org/10.3390/mca31020065

architectures that enhance the LSM's classification performance. The general framework followed is shown in Figure 2. The encoding scheme, genetic operators, selection mechanism, and fitness evaluation process are described in detail below.

2.3.1. Encoding, Initialization & Population

A suitable representation of neural architectures is essential for effective evolutionary optimization in NE. The objective is for the representation encoding, often referred to as genotype, to contain the necessary components and information of the models to define a proper search space. In that sense, the proposed encoding considers two sections: neuron configurations and positions. Regarding the former, four parameters are considered:

- Voltage threshold V_{th}: A lower threshold makes the neuron more sensitive to incoming pulses, allowing it to fire with less input frequency.
- Membrane time constant τ_m: This parameter defines how quickly the membrane potential returns to rest after perturbation. A higher value extends the summation window, allowing distant spikes to contribute more significantly.
- Refractory period Δt_{ref}: The time during which a neuron cannot fire after spiking. A lower value allows faster recovery.
- Polarity: Defines whether a neuron is excitatory or inhibitory. Excitatory neurons increase the membrane potential of postsynaptic neurons, while inhibitory neurons decrease it. This property might be represented as the sign of the neuron's outgoing connections.

On the other hand, the X and Y coordinates in a 2D space (liquid) formed the positional information of the neuronal units, which also establish the connectivity of the liquid.

Three restrictions are necessary to establish correct encoding:

1. Parameters τ_m, Δt_{ref}, X and Y can not be equal or lower than 0. Violations are mapped to a small default value of 0.01.
2. Voltage threshold V_{th} must exceed reset potential V_{reset}. If this condition is not met, V_{reset} is lowered 5 mV below V_{th}.
3. Neurons must have unique positions. If any overlap occurs, one conflicting neuron is reassigned to a random valid position.

Each individual in the population, also referred to as a chromosome, is composed of n genes, where n is the number of neurons in the liquid. Each gene encodes the attributes of a single neuron, combining both its configuration parameters and its spatial position within the liquid. As a result, the dimensionality of the genotype grows linearly with the number of neurons. Figure 6 shows an encoding and decoding example. Specifically, Figure 6a shows an example of a valid genotype of 6 neurons.

Moreover, the neuron's position defines the connectivity of the liquid using a pairwise test to connect each possible pair of source (N_s) and target (N_t) neurons, with a probability of p. The p is calculated as the Probability Density Function (PDF) of a Gaussian Distribution of the distance (D) between N_s and N_t. Equation (1) illustrates the estimation of the probability p_{st} of the neuron N_s being connected to neuron N_t; with this procedure, the closer the neurons are to each other, the greater the chance they are connected. This follows biological accuracy [47,48] known as Peter's rule. This distance-based rule is depicted in Equation (2); a shorter connection will have a more substantial impact on postsynaptic neurons, which emulates the biological principle that shorter connections tend to have more substantial synaptic weights due to reduced delays and energy costs [48].

$$p_{st} = \frac{1}{\sqrt{2\pi}} e^{-\frac{D(N_s,N_t)^2}{4}} \tag{1}$$

https://doi.org/10.3390/mca31020065

$$SW_{st} = SS \times p_{st} \tag{2}$$

where $D(\cdot)$ represents the Euclidean distance between neurons N_s and N_t, p_{st} is the Gaussian probability associated with that distance, and SS denotes the synaptic strength scaling factor. Since the Gaussian probability density function is bounded, the resulting synaptic weights are also bounded. Given the synaptic strength parameter ($SS = 200$), the maximum weight value occurs when the distance between two neurons is close to zero, resulting in a value close to ± 80. Larger distances produce weights close to zero.

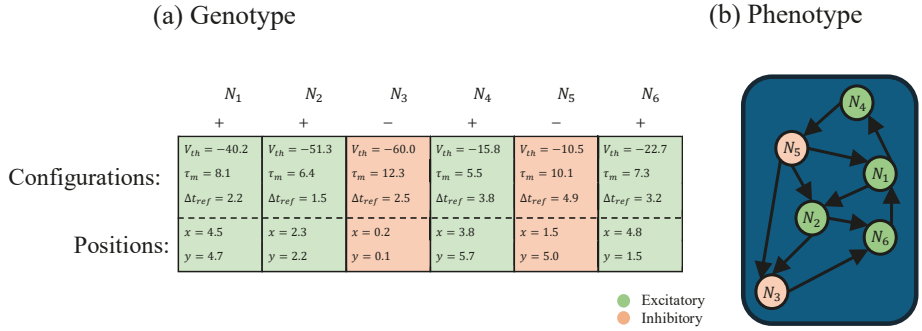


Figure 6. Example of a valid encoding produced during the evolutionary process for a liquid composed of 6 neurons. (**a**) shows the genotype representation, and (**b**) depicts the phenotype translation.

The population consists of 50 individuals, and this size remains fixed throughout the evolutionary process. At initialization, each individual's parameters are randomly defined. To manage computational cost, the liquid size was set to 20 LIF neurons for the synthetic datasets and increased to 300 LIF neurons for the N-MNIST and FSDD datasets. The polarity initialization was adapted to have 80% excitatory neurons, aligning with biological ratios observed in cortical regions [49]; every neuron has an 80% probability of being excitatory.

Parameters V_{th}, τ_m, Δt_{ref}, X, and Y are randomly selected uniformly from specific ranges, shown in Table 3. The ranges of V_{th}, τ_m, Δt_{ref} were selected to have the mean value closer to the normal value. On the other hand, the ranges of X and Y were established following preliminary experiments, in which different upper limits were tested. It was observed that larger values decreased the spiking activity, whereas smaller values saturated the liquid activity. For this reason, both X and Y were constrained to an upper limit of 6, corresponding to a 6×6 spatial grid, which provided a balanced spatial scale and maintained computational manageability for the synthetic datasets. For the larger-scale experiments, the spatial domain was expanded to a 15×15 grid to preserve an appropriate neuron distribution and connectivity structure. Nevertheless, it is crucial to validate these ranges with further experimentation.

2.3.2. Evaluation

The process of evaluating an individual starts by decoding the genotype into a corresponding phenotype (Figure 6b). This task involves setting the individual neuron parameters, such as their configuration and positions, as well as defining which neurons will receive an input connection. Once the network is instantiated, the individual is simulated: input patterns are presented to the liquid, and the resulting spike responses are recorded as liquid states. These states are then transformed into a static state vector, which is fed into the readout block for classification. The individual's performance is measured using classification accuracy as the objective function.

https://doi.org/10.3390/mca31020065

Table 3. Parameter ranges. It shows the normal values found in the literature and the ranges used for random selection for each parameter. "-" means that the value does not apply.

Parameter	Normal Value	Range
V_{th}	-55 mV	$[-69, -10]$
τ_m	10 ms	$[6, 14]$
Δt_{ref}	2 ms	$[0.5, 4]$
Polarity	80%	-
X and Y	-	$[0, 6]$ / $[0, 15]$

2.3.3. Parent Selection

A tournament selection mechanism was chosen for its simplicity and robustness. Tournament selection is a parent selection strategy commonly used in EAs. In this method, a predefined number of individuals, referred to as the tournament size, are randomly sampled from the population. These individuals compete based on their fitness values, and the one with the highest fitness is selected as a parent. This process is repeated as needed to generate the required number of parents.

2.3.4. Crossover

The crossover variant operation promotes the exploitation of the parents' characteristics. The Blend Crossover with $\alpha = 0.5$, referred to as BLX-α, was selected in this work. This operator is designed for real-valued encoding and creates offspring by sampling each gene from a uniform distribution defined by the parent genes and a blending parameter α. Figure 7 illustrates the crossover mechanism.

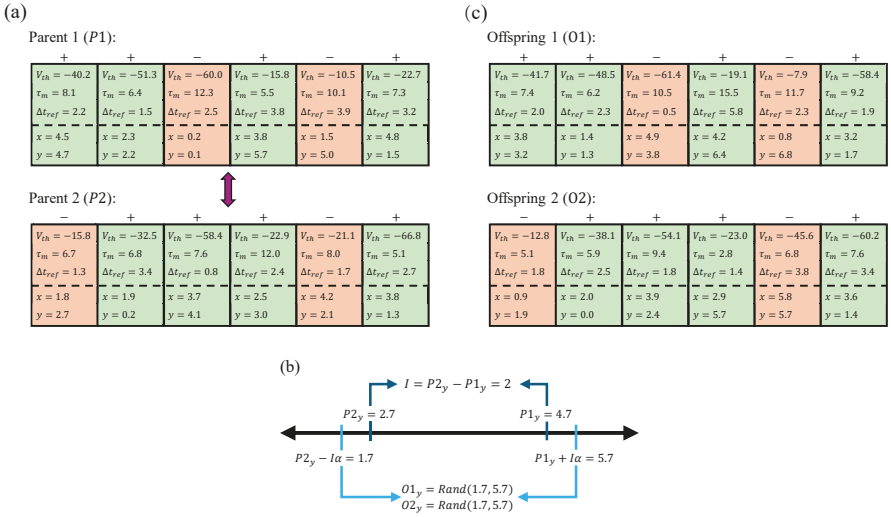


Figure 7. Illustration of the crossover process. (**a**) Two parent individuals are selected from the population, each representing a candidate liquid configuration where every gene encodes a neuron's parameters: threshold potential (V_{th}), membrane time constant (τ_m), refractory period (Δt_{ref}), spatial position (x, y), and polarity. (**b**) A Blend Crossover (BLX-α) [50] with $\alpha = 0.5$ is applied independently across each neuron and each parameter, producing two offspring with mixed configurations. This operation enables controlled exploration of the search space by allowing genes in the offspring to deviate slightly beyond the bounds of their parents. (**c**) Generated offspring. Polarity is inherited from parents.

First, the parents' genotypes are aligned (Figure 7a). Specifically, for each gene, the offspring value is drawn from the interval $[x_{\min} - I\alpha,\ x_{\max} + I\alpha]$, where $x_{\min}$ and $x_{\max}$ are

https://doi.org/10.3390/mca31020065

the values of the corresponding gene from the two parents, and $I = x_{\max} - x_{\min}$ defines the difference between the two values. This approach promotes exploration beyond the parental range while maintaining genetic continuity. Figure 7b shows an example of the calculation. Finally, this process is repeated for every parameter in every gene, resulting in two offspring (Figure 7c). Polarity, being a binary parameter (excitatory or inhibitory), does not undergo this crossover process and is inherited directly from the parents.

2.3.5. Mutation

Unlike crossover, which seeks to promote exploitation, the mutation operator is responsible for providing exploration. In this work, a random subset of n neurons, where n stands for half the total number of neurons in the liquid, is re-initialized and derived from the original ranges. Furthermore, only the configurations or the positions are affected based on a 50% probability. Figure 8 depicts an example of this process, where the configurations of genes 3 and 4, and the position of gene 1 are re-initialized. Although these mutations are relatively strong, they were intentionally chosen to promote exploration, where small perturbations may not always yield meaningful behavioral differences in the network.

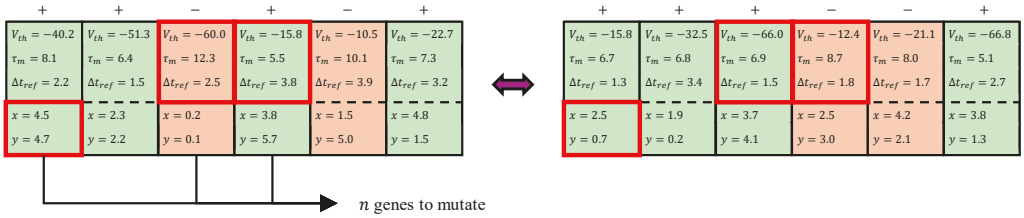


Figure 8. Example of the mutation process applied to a single individual in the population. The mutation operation randomly selects a subset of neurons (highlighted in red) and re-initializes either their configuration parameters or their spatial positions. When configurations are selected, all associated parameters (V_{th}, τ_m, and Δt_{ref}) are re-sampled from their original ranges; similarly, when positions are selected, both spatial coordinates (X, Y) are re-initialized.

2.3.6. Replacement & Stop Criteria

A generational replacement is adopted, including an elitism process to preserve a percentage of the best-suited individuals in every generation. Furthermore, if the maximum number of generations is reached or any individual achieves 100% accuracy, the evolutionary process terminates.

2.4. Methodology & Calibration

Figure 9 illustrates the complete pipeline of the proposed methodology, integrating the data partitioning and the evolutionary optimization process described above. The first phase (Figure 9a) divides the dataset into two partitions: an Evolutionary partition P_E (70%), intended for the GA process, and a Validation partition P_V (30%), used on the best individual obtained after the GA process. The second phase entails the GA-proposed process (Figure 9b). For each individual, the evolutionary partition P_E is internally partitioned into training and testing subsets to train and evaluate the readout model, ensuring a fair estimate of classification performance.

A calibration process was used to establish the values for the crossover rate (Cr), mutation rate (Mr), Tournament size, and Elitism. To perform this, ten executions of the GA with a reduced number of instances of the FR5 were conducted with different values. While a full grid search across all parameter combinations could yield more optimal configurations, such an approach was not implemented due to the high computational cost associated with simulating each combination. Figure 10 shows the mean accuracy obtained in experiments. For Cr, it was observed that 70% performed well on both the Testing and Validation, fol-

https://doi.org/10.3390/mca31020065

lowed by 50%. Interestingly, 90% exhibit a significant drop in performance (Figure 10a). The *Mr* parameter showed an increasing performance as the percentage increment increased. Two interesting peaks appeared on 50% and 80% (Figure 10b). Following these observations, five additional experiments were performed for every combination of 50%, 60%, 70% for *Cr* and 50%, 70%, and 80% for *Mr* (Figure 10c). Interestingly, the best results reside within the high *Cr* or *Mr* percentages. Nevertheless, having both parameters in a high percentage increased the number of evaluated individuals, augmenting the computational cost. Therefore, a 70% *Cr* and 50% was selected. Notably, higher mutation rates yielded competitive results, indicating the potential suitability of alternative approaches, such as Evolutionary Strategies (ES). Regarding tournament size and elitism, the analysis shows better results at 25%, which forms a tournament of 12 individuals, and 5%, which sets three individuals for elitism, respectively. Table 4 summarizes the parameter calibrations.

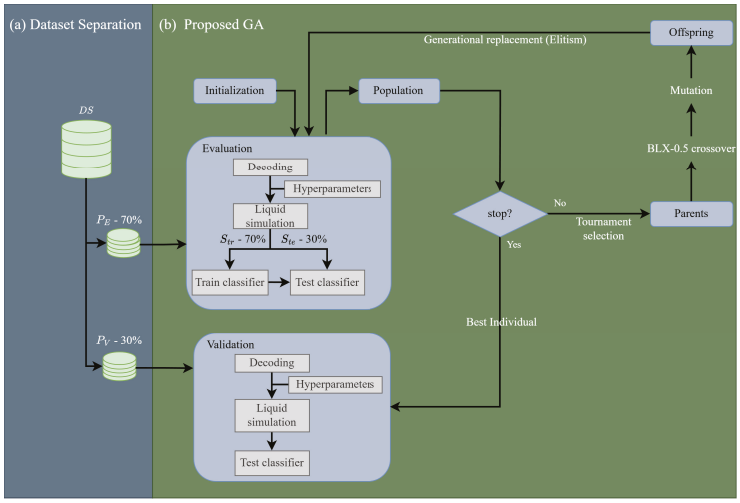


Figure 9. Proposed pipeline. (**a**) The data separation divides the dataset into an Evolutionary partition P_E (70%) and a Validation partition P_V (30%). (**b**) The proposed GA begins with the initialization. These first individuals go through evaluation on the P_E to form the initial population. After tournament selection, offspring are created using the crossover and mutation operations. Then, the offspring undergo evaluation on the P_E and replace the previous population with elitism. When the stop criterion is reached, the best individual is tested on the P_V.

To test the implementation, 30 executions were performed on the FR5, PR4, PR8, and PR12 tasks. In contrast, due to computational limitations, only five executions were carried out for the N-MNIST dataset and the FSDD. The DEAP [51] library was used for GA implementation, programmed in Python 3.10. Experiments were conducted on a high-performance machine equipped with two Intel® Xeon® Gold CPUs (Model 6226R) with 72 logical processors, enabling parallel evaluation of individuals. Given a population size of 50 individuals and 100 generations, each evolutionary run required approximately 5000 fitness evaluations. In practice, runs on the synthetic benchmark datasets required approximately 6 h per execution. For real-world datasets involving larger liquids and higher input dimensionality, the runtime increased to approximately 24–30 h per run. These runtimes are comparable to or shorter than those reported in related work, such as Tian et al. [38], who reported approximately 16.3 days for N-MNIST and 3.3 days for FSDD, and Zhou et al. [39], who reported runtimes of around 42 h.

https://doi.org/10.3390/mca31020065

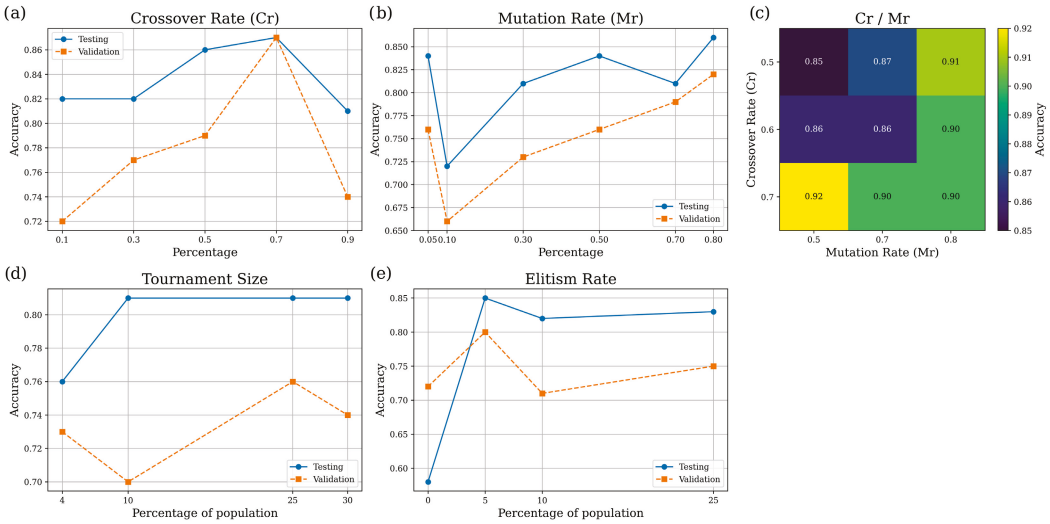


Figure 10. Calibration mean results from 10 executions. (**a**) *Cr* results on different test values. The best results were found in 50% and 70%. (**b**) *Mr* results for different tested values. Performance generally improves as the mutation rate increases, with the best validation results observed at 80%. (**c**) The combination test of selected values for *Cr* and *Mr*. Best result resides in high values from either *Cr* and *Mr*. (**d**) Tournament size had few variations, but the best validation results were found at 25%. (**e**) Elitism percentage of 5% outperformed other tested values.

Table 4. Parameter summary. The values in bold were calibrated during a number of experiments, whereas the rest were selected based on prior experience.

Parameter	Value
Population size	50
Generations	100
Cr	70%
Mr	50%
Tournament size	25% (12 individuals)
Elitism	5% (3 individuals)

3. Results

3.1. Experiments on Synthetic Datasets

To establish a baseline contrast, Figure 11 presents a comparison among the best initial fitness, the best final fitness, and the validation process. This shows a robust generalization, as evidenced by reduced overfitting and improved validation accuracy.

Table 5 shows a comparison of results among this approach and the Norton & Ventura [31] approach, referenced as SDSM, and Alvarez-Canchila et al. [41], referenced as PSO. The results reported for the SDSM [29] method were taken from the original publication. For the PSO method [41], the authors of that work provided their full experimental results, which allowed us to perform statistical comparisons between both approaches.

The proposed method outperformed all metrics in all tasks except the mean and median values of FR12. To assess statistical significance, the Shapiro–Wilk test was first applied to evaluate the normality of the accuracy distributions. Since the normality assumption was not met, the non-parametric Kruskal–Wallis test was used to compare the performance of the GA method and PSO. The results are summarized in Figure 12; significant differences are observed across all tasks. In FR5, PR4, and PR8, these differences favor our approach. Nevertheless, the case of PR12 warrants further analysis, as statistical

https://doi.org/10.3390/mca31020065

metrics may indicate equal capabilities between our approach and PSO, yet significant differences were observed.

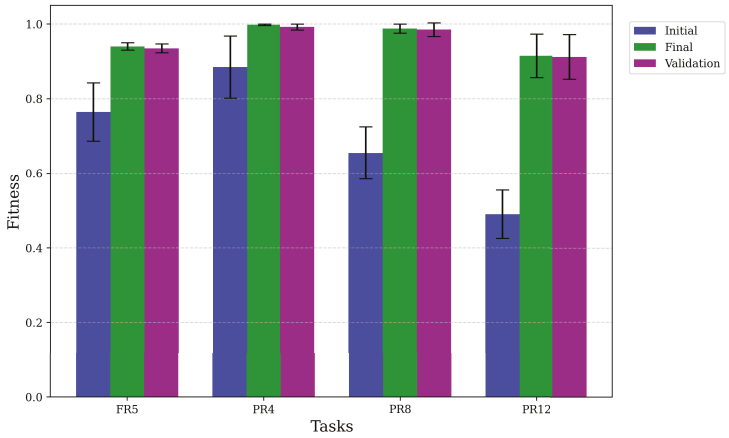


Figure 11. Comparison of initial, final and validation fitness of FR5, PR4, PR8, and PR12 tasks. Bars in blue represent the best accuracy value among the initial population. Bars in green represent the best fitness found after evolution, and bars in purple show validation results on the best individual found. Error bars represent the standard deviation over the 30 executions.

Table 5. Performance comparison among Norton & Ventura [31] (SDSM), Alvarez-Canchila et al. [41] (PSO), and this work. The x_B, $\bar{x}$, x_M, x_W, and σ represent the best value, mean, median, worst value, and standard deviation, respectively. The "-" mark stands for data not available. All values in bold signify the best values.

Task	Statistical	SDSM	PSO	GA (This)
FR5	x_B	0.944	0.9080	**0.9610**
	$\bar{x}$	0.7822	0.8158	**0.9384**
	x_M	-	0.8700	**0.9387**
	x_W	-	0.5440	**0.9187**
	σ	-	0.0973	**0.0102**
PR4	x_B	0.99	**1.0000**	**1.0000**
	$\bar{x}$	0.9197	0.9828	**0.9971**
	x_M	-	**1.0000**	0.9983
	x_W	-	0.6925	**0.9725**
	σ	-	0.0645	**0.0053**
PR8	x_B	0.98	0.9662	**1.0000**
	$\bar{x}$	0.759	0.8910	**0.9880**
	x_M	-	0.9663	**0.9950**
	x_W	-	0.4775	**0.9400**
	σ	-	0.1483	**0.0149**
PR12	x_B	0.9167	**0.9975**	**0.9975**
	$\bar{x}$	0.7562	**0.9373**	0.9162
	x_M	-	**0.9750**	0.9257
	x_W	-	0.2742	**0.7447**
	σ	-	0.1379	**0.0597**

https://doi.org/10.3390/mca31020065

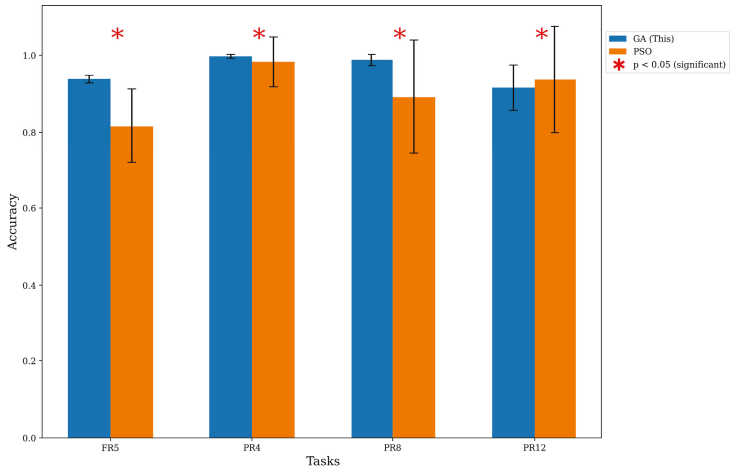


Figure 12. Comparison of classification accuracy between GA and PSO across the four tasks. Bars represent the mean accuracy, and error bars indicate the standard deviation. A red asterisk (*) denotes a statistically significant difference between the methods.

Even though PSO achieved better mean and median values, our GA method tied their best results and outperformed their worst cases, as well as the standard deviation. This indicated that PSO had higher top-end performance, with some extreme values contributing to the standard deviation.

To further assess this, 30 additional experiments were performed, increasing the number of neurons to 64, matching the model of Alvarez-Canchila et al. [41]. Table 6 presents a comparison of statistical metrics, demonstrating an improvement in our method. Given the non-normal distribution detected by the Shapiro–Wilk test, a Kruskal–Wallis test followed by Dunn's post hoc test ($p < 0.05$) was applied among the PSO, GA with 20 neurons, and GA with 64 neurons. As shown in Figure 13, no statistically significant differences were observed between the proposed method and PSO. Nevertheless, PSO yields a wider spread of results, including more extreme low-performing outliers, whereas the proposed method maintains more consistent performance.

Table 6. Performance comparison among Alvarez-Canchila et al. [41] (PSO), and this work with 20 and 64 neurons. The x_B, $\bar{x}$, x_M, x_W, and σ represent the best value, mean, median, worst value, and standard deviation, respectively. All values in bold signify the best values.

Task	Statistical	PSO	GA (20 Neurons)	GA (64 Neurons)
PR12	x_B	**0.9975**	**0.9975**	0.9939
	$\bar{x}$	0.9373	0.9162	**0.9546**
	x_M	**0.9750**	0.9257	0.9628
	x_W	0.2742	0.7447	**0.8694**
	σ	0.1379	0.0597	**0.0298**

Evaluating the final neuronal configuration, namely V_{th}, τ_m, and Δt_{ref}, and the position distribution of the best individuals from the four tasks with our GA method (excluding the GA with 64 neurons since it was only tested on PR12), Figures 14 and 15 show that most values remain within the original initialization ranges. The neuron configurations tend to converge within narrower intervals, although the resulting distributions differ from task to task. Therefore, these statistics should be interpreted separately for each benchmark rather than as a global trend across all tasks. In contrast, the neuron position distributions do not exhibit a clearly discernible pattern.

https://doi.org/10.3390/mca31020065

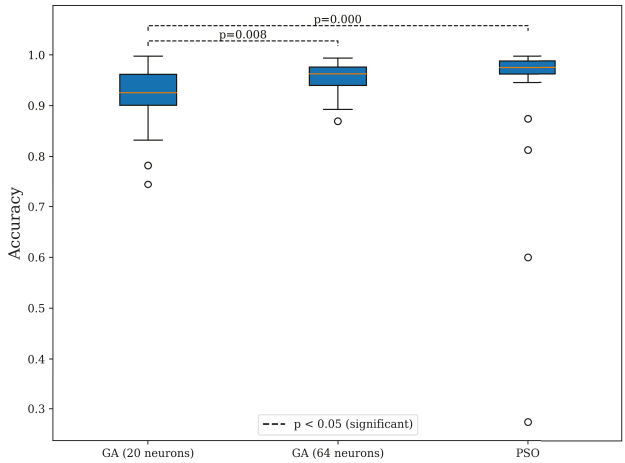


Figure 13. Boxplot comparison of classification accuracy for the PR12 task using GA with 20 neurons, GA with 64 neurons, and PSO. Each box shows the distribution of accuracy values across multiple runs. Orange lines indicate the median, and dashed lines indicate statistically significant differences ($p < 0.05$) according to Dunn's posthoc test.

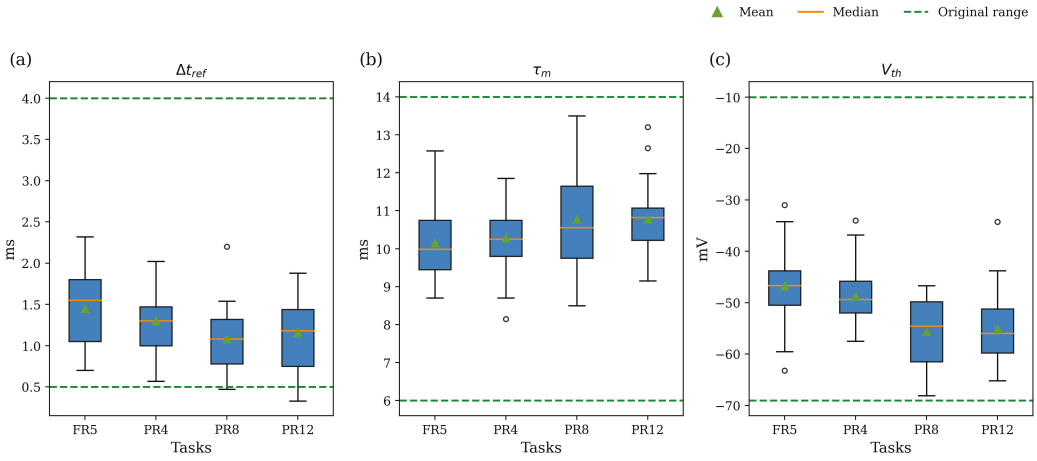


Figure 14. Distribution of neuron configuration parameters in the best-performing LSMs across the synthetic tasks. Each subplot shows a boxplot for a different parameter: (**a**) refractory period (Δt_{ref}), (**b**) membrane time constant (τ_m), and (**c**) spike threshold (V_{th}). Green triangles indicate the mean value, orange lines indicate the median, and dashed green lines represent the default ranges used during initialization.

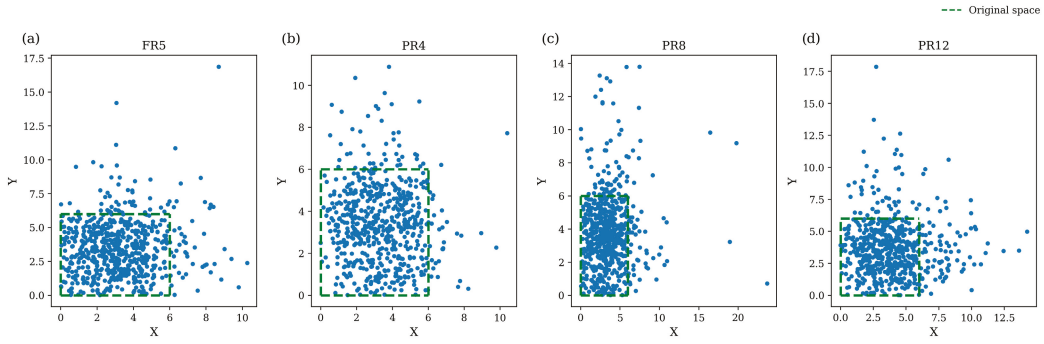


Figure 15. Distribution of neuron positions in the best-performing LSMs across the synthetic tasks. Each subplot shows the neuron position spaced in a liquid: (**a**) FR5, (**b**) PR4, (**c**) PR8, and (**d**) PR12. Dashed green lines represent the default ranges used during initialization.

https://doi.org/10.3390/mca31020065

3.2. Experiments on Real-World Benchmarks

In addition to the synthetic datasets, the proposed method was evaluated on two standard benchmarks: N-MNIST and FSDD. These datasets present more complex spatiotemporal patterns, offering a stronger test of the model's capabilities. To account for the increased complexity, the liquid size was increased to 300 neurons, the initial positional grid was set to 15 × 15, and only five executions were performed per task. The results reported for Tian et al. [38] and Zhou et al. [39] were taken from their respective publications and are used here as reference.

Table 7 shows that while our method did not achieve the highest accuracy, it reached competitive results using significantly fewer neurons. For instance, Tian et al. [38] employ reservoirs with 1000 neurons, whereas our approach achieves comparable performance using only 300 neurons. Zhou et al. [39] propose an architecture search method that adjusts structural aspects of the reservoir, including the number of neurons during optimization; however, the exact number of neurons in their final architectures is not explicitly reported. Nevertheless, our results indicate that competitive performance can be obtained with a more compact architecture. Since the computational cost of simulating SNNs scales with the number of neurons and synaptic interactions, reducing the network size reduces computational complexity and improves efficiency, making the model suitable for resource-constrained environments, such as neuromorphic hardware platforms, embedded systems, and edge computing devices.

Table 7. Comparison of performance on N-MNIST and FSDD datasets. The "-" mark stands for data not available.

Method	N-MNIST		FSDD	
	Neurons	**Accuracy**	**Neurons**	**Accuracy**
Tian et al. [38]	1000	92.50%	1000	84.50%
Zhou et al. [39]	-	92.80%	-	-
Ours	300	90.65% (±1.59%)	300	81.47% (±2.12%)

Similarly to before, Figures 16 and 17 show the final neuronal configuration and position distribution of the best individuals from these tasks, respectively. The observations obtained from the synthetic experiments remain consistent: neuron configurations converge to narrower ranges depending on the task, while neuron positions remain dispersed without exhibiting a clear spatial pattern.

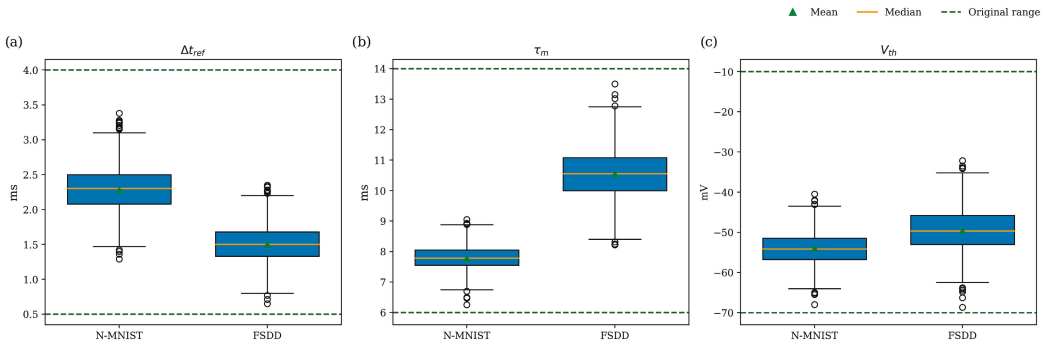


Figure 16. Distribution of neuron configuration parameters in the best-performing LSMs among N-MNIST and FSSD benchmarks. Each subplot shows a boxplot for a different parameter: (**a**) refractory period (Δt_{ref}), (**b**) membrane time constant (τ_m), and (**c**) spike threshold (V_{th}). Green triangles indicate the mean value, orange lines indicate the median, and dashed green lines represent the default ranges used during initialization.

https://doi.org/10.3390/mca31020065

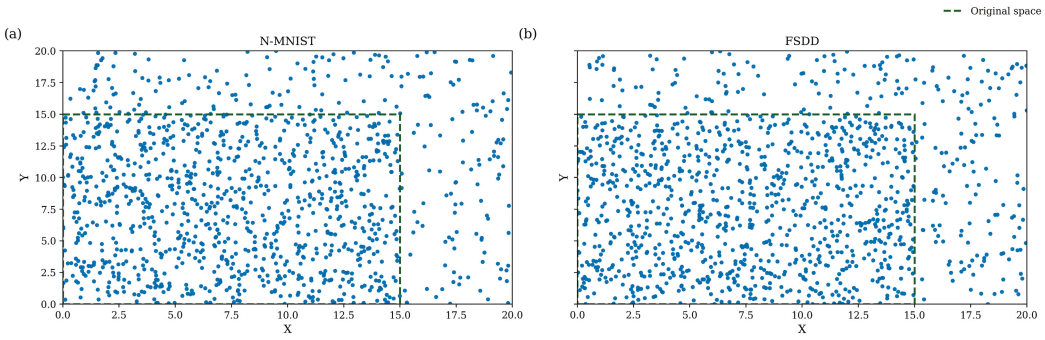


Figure 17. Distribution of neuron positions in the best-performing LSMs among N-MNIST and FSSD benchmarks. Each subplot shows the neuron position spaced in a liquid: (**a**) N-MNIST, (**b**) FSDD. Dashed green lines represent the default ranges used during initialization.

4. Discussion & Conclusions

The LSM model, an RC approach that combines R-SNN with simple linear classification methods, excels in several spatiotemporal applications. Among their qualities, LSMs harness the complexity of neuron models, such as the LIF model, to learn spatiotemporal features and patterns in complex time series. Although the traditional LSM formulation assumes a fixed, untrained liquid (reservoir), recent research has relaxed this constraint by introducing training or optimization strategies for the R-SNN, thereby enhancing the liquid capabilities. Moreover, most state-of-the-art (SOTA) methods do not exploit the neuron parameters available, such as the refractory period (Δt_{ref}), the membrane time constant (τ_m), and the spike threshold (V_{th}), to optimize the performance of the liquid and reside entirely on the synaptic connections.

In this work, a novel NE approach for optimizing LSM based on a GA encoding neuron configurations and position was presented. To validate the method, two different approaches were used. First, four synthetic classification tasks, namely a Frequency Recognition task with five classes (FR5) and three Pattern Recognition tasks with 4 (PR4), 8 (PR8), and 12 (PR12) classes. Results were compared with two similar approaches: the SDSM learning technique introduced by Norton & Ventura [31] and a PSO metaheuristic method proposed by Alvarez-Canchila et al. [41]. Second, two classical datasets were utilized, N-MNIST and FSDD, and the results were compared against SOTA methods proposed by Tian et al. [38] and Zhou et al. [39]. To manage computational cost, the size of the liquid was limited to 20 LIF neurons for the synthetic datasets and 300 LIF neurons for the N-MNIST and the FSDD.

The method achieved SOTA results in the synthetic tasks, with mean accuracies of 93.84% in FR5, 99.74% in PR4, 98.80% in PR8, and 91.62% in PR12. Furthermore, our approach yielded better results across all statistical metrics investigated, with significant differences proven by the Kruskal–Wallis test, except for PR12. Further examination of the PR12 case revealed that the PSO method achieved higher top-end performance while exhibiting several suboptimal runs (e.g., the worst value of 27.42% compared to 74.47% for our GA approach). The number of neurons was increased to match the 64 neurons from the PSO proposal, resulting in a mean accuracy of 95.46% in PR12 with no significant difference.

Our results suggest that, while PSO was capable of reaching competitive results, our NE GA-based approach yielded more stable performance across runs, showing a more robust and generalizable approach. Incorporating neuron configurations into the optimization process proved to be an efficient strategy, as it enabled the exploration of diverse and dynamic regimes within the liquid, leading to improved task-specific performance without increasing the network size.

https://doi.org/10.3390/mca31020065

The method achieved SOTA results in the synthetic tasks, with mean accuracies of 93.84% in FR5, 99.74% in PR4, 98.80% in PR8, and 91.62% in PR12. Furthermore, our approach yielded better results across all statistical metrics investigated, with significant differences confirmed by the Kruskal–Wallis test, except for PR12. Further analysis of this case revealed that, although the PSO method exhibited greater variability across runs, including several suboptimal results (e.g., a worst value of 27.42% compared to 74.47% for our GA approach). Increasing the number of neurons to 64, matching the PSO configuration, improved the mean accuracy to 95.46% with no statistically significant difference. Both approaches achieve comparable performance; however, PSO presents a wider distribution with more frequent low-performing outliers, whereas the proposed GA-based method demonstrates more stable and consistent behavior across runs. These results suggest that, while PSO is capable of achieving competitive performance, incorporating neuron configurations into the optimization process enables a more robust and generalizable search, promoting stable solutions without increasing the network size.

The evaluation on N-MNIST and FSDD demonstrated the proposed method's capacity to be applied beyond synthetic tasks. Although our method did not surpass SOTA accuracy on these datasets, the achieved accuracies of 90.65% on N-MNIST and 81.47% on FSDD remain within close range of reported results. Furthermore, these results were achieved using significantly fewer neurons (300 vs. 1000 in Tian et al. [38]). Since the computational cost of spiking neural network simulations increases with the number of neurons and synaptic interactions, this reduction suggests improved computational efficiency. This characteristic is particularly relevant for resource-constrained scenarios, where memory, energy consumption, and computational capacity are limited. In such contexts, achieving competitive performance with smaller reservoirs becomes highly advantageous, as it reduces simulation cost while maintaining strong classification capabilities.

The performance improvements observed across several tasks may be related to the heterogeneous nature of the evolved reservoirs. By allowing neurons to operate under different parameter configurations, the proposed encoding enables diverse temporal responses within the liquid. This diversity can enrich the internal dynamics of the reservoir and facilitate improved separation of input patterns in the liquid state space. Similar observations regarding the benefits of neuronal heterogeneity for learning efficiency in spiking neural networks have been reported in previous studies [42].

Regarding the parameters found during the optimization process, a reduction in the original ranges was observed for the neuron configurations, suggesting that the evolutionary process tends to converge toward more effective regions of the search space. This insight could help guide the search toward more suitable parameter values in future applications. In this context, an additional aspect worth considering is the potential impact of different neuron configurations on neuromorphic hardware implementations. Modern neuromorphic platforms, such as Intel Loihi and SpiNNaker, support configurable neuron parameters, allowing different thresholds, time constants, and refractory periods to be specified for individual neurons [11,12]. Therefore, the neuron configurations produced by the evolutionary process may help generate richer reservoir dynamics while remaining compatible with such hardware. However, neuromorphic systems often rely on discrete values or architectural constraints that limit the range of neuron types that can be deployed. Exploring the interaction between evolved neuron configurations and hardware constraints is therefore an interesting direction for future work.

Another key novelty of our proposal lies in the ability to influence the R-SNN's connectivity indirectly through the spatial arrangement of neurons in the liquid, without explicitly encoding all synaptic connections. In this formulation, neuron positions determine interneuronal distances, which in turn affect both the probability of connection and synaptic

https://doi.org/10.3390/mca31020065

weights through a Gaussian distance-based rule. Consequently, the encoding shapes the connectivity structure of the reservoir and modulates spike propagation dynamics.

Table 8 presents a comparison between the proposed encoding and a direct weight-encoding approach. In our method, each neuron is represented by a small set of parameters (four configuration values and two spatial coordinates), yielding an encoding size that scales linearly with the number of neurons. In contrast, weight-based approaches must account for all possible synaptic connections, leading to a quadratic growth in dimensionality. Additionally, while such approaches rely entirely on continuous parameters, our encoding combines continuous (five parameters) and binary (polarity) variables, further reducing the effective search space. Although no explicit spatial patterns were observed in the evolved neuron positions, spatial encoding remains a key structural component, enabling the emergence of meaningful connectivity patterns consistent with biological principles such as Peter's rule [48].

Table 8. Comparison of encoding size between our method and a direct weight-encoding approach.

#Neurons	Direct-Weight Encoding Size	Our Encoding Size	Reduction (%)
20	400	120	70%
64	4096	384	90.6%
300	90,000	1800	98%

While the proposed approach demonstrates strong performance across synthetic and real-world tasks, several limitations affect its broader applicability. First, the computational cost of evolving the liquid becomes nontrivial, particularly as the number of neurons increases or when evaluating larger datasets, which constrained the number of trials on N-MNIST and FSDD. Second, although the method achieved competitive performance, it did not surpass SOTA results on real-world benchmarks, suggesting further methodological improvements are needed. Lastly, while neuron configurations led to observable improvements, the contribution of spatial neuron placement remains inconclusive and warrants further analysis.

Future work will focus on extending the experimental evaluation and the optimization framework proposed in this study. This includes testing the approach on additional tasks and more diverse datasets to further validate its generalization capabilities, exploring alternative evolutionary operators, incorporating additional neuron or synaptic parameters (such as external currents) into the encoding, probing the use of surrogate models to approximate the fitness function and reduce evaluation time, and scaling the liquid to larger reservoir sizes.

Author Contributions: Conceptualization, C.-A.L.-H., H.-G.A.-M., E.M.-M. and J.-A.B.-P.; methodology, C.-A.L.-H. and H.-G.A.-M.; software, C.-A.L.-H.; validation, H.-G.A.-M. and E.M.-M.; formal analysis, C.-A.L.-H. and H.-G.A.-M.; investigation, C.-A.L.-H.; resources, H.-G.A.-M. and E.M.-M.; writing—original draft preparation, C.-A.L.-H.; writing—review and editing, H.-G.A.-M., E.M.-M. and J.-A.B.-P.; visualization, C.-A.L.-H. and J.-A.B.-P.; supervision, H.-G.A.-M. and E.M.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: All codes and datasets are available at https://github.com/CarlosA-LopezH/MCA2025, accessed on 22 August 2025.

https://doi.org/10.3390/mca31020065

Acknowledgments: The first (CVU 1075919) and the fourth (CVU 1142850) authors acknowledge support from the Secretariat of Science, Humanities, Technology and Innovation (SECIHTI) through a scholarship to pursue postgraduate studies at the University of Veracruz. All authors acknowledge the sources and services provided by the UNAM-HUAWEI Innovation Space of the Alliance to Promote Research Projects in Artificial Intelligence in Mexico, under the 2023 call for proposals.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RC	Reservoir Computing
ANN	Artificial Neural Network
RNN	Recurrent Neural Network
DL	Deep Learning
ESN	Echo State Network
LSM	Liquid State Machine
ReLU	Rectified Linear Unit
R-SNN	Recurrent Spiking Neural Networks
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
LIF	Leaky Integrate-and-Fire
SP	Separation Property
AP	Approximation Property
SDSM	Separation Driven Synaptic Modification
STDP	Spike-Time Dependent Plasticity
EC	Evolutionary Computation
EA	Evolutionary Algorithm
SI	Swarm Intelligence
NE	Neuroevolution
GA	Genetic Algorithm
MLSM	Multi-liquid State Machine
SA	Simulated Annealing
FR	Frequency Recognition
PR	Pattern Recognition
PDF	Probability Density Function
SGD	Stochastic Gradient Descent
FSDD	Free Spoken Digit Dataset
GLSM	Generative Liquid State Machine
CCE	Cooperative Co-Evolution
RFS	Random Forest Surrogate
PSO	Particle Swarm Optimization
N-MNIST	Neuromorphic MNIST
ATIS	Asynchronous Time-based Image Sensor
MFCCs	Mel-Frequency Cepstral Coefficients
ES	Evolutionary Strategies

References

1. Verstraeten, D.; Schrauwen, B.; D'Haene, M.; Stroobandt, D. The unified reservoir computing concept and its digital hardware implementations. In *Proceedings of the 2006 EPFL LATSIS Symposium*; University of Ghent: Ghent, Belgium, 2006; pp. 139–140.
2. Maass, W.; Natschläger, T.; Markram, H. Real-time computing without stable states: A new framework for neural computation based on perturbations. *Neural Comput.* **2002**, *14*, 2531–2560. [CrossRef] [PubMed]
3. Yan, M.; Huang, C.; Bienstman, P.; Tino, P.; Lin, W.; Sun, J. Emerging opportunities and challenges for the future of reservoir computing. *Nat. Commun.* **2024**, *15*, 2056. [CrossRef] [PubMed]

https://doi.org/10.3390/mca31020065

4. Jaeger, H. *The "Echo State" Approach to Analysing and Training Recurrent Neural Networks-with an Erratum Note*; GMD Technical Report; German National Research Center for Information Technology: Bonn, Germany, 2001; Volume 148, p. 13.
5. Jaeger, H.; Haas, H. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science* **2004**, *304*, 78–80. [CrossRef]
6. Abdi, G.; Mazur, T.; Szaciłowski, K. An organized view of reservoir computing: A perspective on theory and technology development. *Jpn. J. Appl. Phys.* **2024**, *63*, 050803. [CrossRef]
7. Lukoševičius, M.; Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **2009**, *3*, 127–149. [CrossRef]
8. Tanaka, G.; Yamane, T.; Héroux, J.B.; Nakane, R.; Kanazawa, N.; Takeda, S.; Numata, H.; Nakano, D.; Hirose, A. Recent advances in physical reservoir computing: A review. *Neural Netw.* **2019**, *115*, 100–123. [CrossRef]
9. Verstraeten, D.; Schrauwen, B.; d'Haene, M.; Stroobandt, D. An experimental unification of reservoir computing methods. *Neural Netw.* **2007**, *20*, 391–403. [CrossRef]
10. Schrauwen, B.; Verstraeten, D.; Van Campenhout, J. An overview of reservoir computing: Theory, applications and implementations. In Proceedings of the 15th European Symposium on Artificial Neural Networks, Bruges, Belgium, 25–27 April 2007; pp. 471–482.
11. Furber, S.B.; Galluppi, F.; Temple, S.; Plana, L.A. The spinnaker project. *Proc. IEEE* **2014**, *102*, 652–665. [CrossRef]
12. Davies, M.; Srinivasa, N.; Lin, T.H.; Chinya, G.; Cao, Y.; Choday, S.H.; Dimou, G.; Joshi, P.; Imam, N.; Jain, S.; et al. Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* **2018**, *38*, 82–99. [CrossRef]
13. Hodgkin, A.L.; Huxley, A.F. A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* **1952**, *117*, 500–544. [CrossRef]
14. Stein, R.B. Some models of neuronal variability. *Biophys. J.* **1967**, *7*, 37–68. [CrossRef]
15. Gewaltig, M.O.; Diesmann, M. NEST (NEural Simulation Tool). *Scholarpedia* **2007**, *2*, 1430. [CrossRef]
16. Zhang, Y.; Li, P.; Jin, Y.; Choe, Y. A digital liquid state machine with biologically inspired learning and its application to speech recognition. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 2635–2649. [CrossRef]
17. Liu, Y.; Jin, Y.; Li, P. Exploring sparsity of firing activities and clock gating for energy-efficient recurrent spiking neural processors. In *Proceedings of the 2017 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*; IEEE: Piscataway, NJ, USA, 2017; pp. 1–6. [CrossRef]
18. Liu, Y.; Yenamachintala, S.S.; Li, P. Energy-efficient FPGA spiking neural accelerators with supervised and unsupervised spike-timing-dependent-plasticity. *ACM J. Emerg. Technol. Comput. Syst. (JETC)* **2019**, *15*, 1–19. [CrossRef]
19. Belhia, S.; Benyettou, M.; Lehireche, A. Handwritten signatures recognition using Liquid State Machine. *Int. J. Biom.* **2011**, *3*, 148–158. [CrossRef]
20. Xiao, X.; Wang, L.; Chen, X.; Qu, L.; Guo, S.; Wang, Y.; Kang, Z. Dynamic vision sensor based gesture recognition using liquid state machine. In *Proceedings of the International Conference on Artificial Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 618–629. [CrossRef]
21. Biswas, A.; Nambiar, N.S.; Kejriwal, K.; Ganguly, U. Madapter: A multimodal adapter for liquid state machines configures the input layer for the same reservoir to enable vision and speech classification. In *Proceedings of the 2023 International Joint Conference on Neural Networks (IJCNN)*; IEEE: Piscataway, NJ, USA, 2023; pp. 1–6. [CrossRef]
22. Polepalli, A.; Soures, N.; Kudithipudi, D. Digital neuromorphic design of a liquid state machine for real-time processing. In *Proceedings of the 2016 IEEE International Conference on Rebooting Computing (ICRC)*; IEEE: Piscataway, NJ, USA, 2016; pp. 1–8. [CrossRef]
23. Maass, W. Liquid state machines: Motivation, theory, and applications. *Computability in Context: Computation and Logic in the Real World*; World Scientific Publishing: Singapore, 2011; pp. 275–296.
24. Yu, X.; Wan, Z.; Shi, Z.; Wang, L. Lipreading using liquid state machine with stdp-tuning. *Appl. Sci.* **2022**, *12*, 10484. [CrossRef]
25. Neftci, E.O.; Mostafa, H.; Zenke, F. Surrogate gradient learning in spiking neural networks. *IEEE Signal Process. Mag.* **2019**, *36*, 51–63. [CrossRef]
26. Tavanaei, A.; Ghodrati, M.; Kheradpisheh, S.R.; Masquelier, T.; Maida, A. Deep learning in spiking neural networks. *Neural Netw.* **2019**, *111*, 47–63. [CrossRef] [PubMed]
27. Eshraghian, J.K.; Ward, M.; Neftci, E.O.; Wang, X.; Lenz, G.; Dwivedi, G.; Bennamoun, M.; Jeong, D.S.; Lu, W.D. Training spiking neural networks using lessons from deep learning. *Proc. IEEE* **2023**, *111*, 1016–1054. [CrossRef]
28. Wijesinghe, P.; Srinivasan, G.; Panda, P.; Roy, K. Analysis of liquid ensembles for enhancing the performance and accuracy of liquid state machines. *Front. Neurosci.* **2019**, *13*, 504. [CrossRef]
29. Norton, D.; Ventura, D. Preparing more effective liquid state machines using hebbian learning. In *Proceedings of the 2006 IEEE International Joint Conference on Neural Network Proceedings*; IEEE: Piscataway, NJ, USA, 2006; pp. 4243–4248. [CrossRef]
30. Norton, D.; Ventura, D. Improving the separability of a reservoir facilitates learning transfer. In P*roceedings of the 2009 International Joint Conference on Neural Networks*; IEEE: Piscataway, NJ, USA, 2009; pp. 2288–2293. [CrossRef]

https://doi.org/10.3390/mca31020065

31. Norton, D.; Ventura, D. Improving liquid state machines through iterative refinement of the reservoir. *Neurocomputing* **2010**, *73*, 2893–2904. [CrossRef]
32. Bellec, G.; Salaj, D.; Subramoney, A.; Legenstein, R.; Maass, W. Long short-term memory and learning-to-learn in networks of spiking neurons. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 795–805.
33. Shrestha, S.B.; Orchard, G. Slayer: Spike layer error reassignment in time. *Adv. Neural Inf. Process. Syst.* **2018**, *31*, 1419–1428.
34. López-Herrera, C.A.; Acosta-Mesa, H.G.; Mezura-Montes, E.; Barradas-Palmeros, J.A. Prediction of Epileptic Seizure Using Neuroevolved Spiking Neural Network. In *Proceedings of the Mexican International Conference on Artificial Intelligence*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 135–146. [CrossRef]
35. Liu, Y.; Sun, Y.; Xue, B.; Zhang, M.; Yen, G.G.; Tan, K.C. A Survey on Evolutionary Neural Architecture Search. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 550–570. [CrossRef]
36. Yaqoob, M.; Wrobel, B. Very small spiking neural networks evolved to recognize a pattern in a continuous input stream. In *Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI)*; IEEE: Piscataway, NJ, USA, 2017; pp. 1–8. [CrossRef]
37. Gallicchio, C.; Micheli, A.; Pedrelli, L. Design of deep echo state networks. *Neural Netw.* **2018**, *108*, 33–47. [CrossRef]
38. Tian, S.; Qu, L.; Wang, L.; Hu, K.; Li, N.; Xu, W. A neural architecture search based framework for liquid state machine design. *Neurocomputing* **2021**, *443*, 174–182. [CrossRef]
39. Zhou, Y.; Jin, Y.; Sun, Y.; Ding, J. Surrogate-Assisted Cooperative Co-evolutionary Reservoir Architecture Search for Liquid State Machines. *IEEE Trans. Emerg. Top. Comput. Intell.* **2023**, *7*, 1484–1498. [CrossRef]
40. Zhou, Y.; Jin, Y.; Ding, J. Brain-Inspired Modular Reservoir based on Spiking Neurons for Multi-Task Learning. In *Proceedings of the 2025 IEEE Congress on Evolutionary Computation (CEC)*; IEEE: Piscataway, NJ, USA, 2025; pp. 1–4.
41. Alvarez-Canchila, O.I.; Espinal, A.; Sotelo-Figueroa, M.A.; Soria-Alcaraz, J.A.; Rostro-Gonzalez, H. Enhancing liquid state machine classification through reservoir separability optimization using swarm intelligence and multitask learning. *IEEE Access* **2024**, *12*, 182856–182871. [CrossRef]
42. Zhang, F.; Cui, J. Neural heterogeneity as a unifying mechanism for efficient learning in spiking neural networks. *Front. Comput. Neurosci.* **2025**, *19*, 1661070. [CrossRef] [PubMed]
43. López-Herrera, C.A.; Acosta-Mesa, H.G.; Mezura-Montes, E. Evaluating Encoding of Neuron Configuration and Position in Neuroevolution of Liquid State Machines. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*; Association for Computing Machinery: New York, NY, USA, 2025; pp. 2384–2388.
44. Orchard, G.; Jayawant, A.; Cohen, G.K.; Thakor, N. Converting static image datasets to spiking neuromorphic datasets using saccades. *Front. Neurosci.* **2015**, *9*, 437. [CrossRef]
45. Jackson, Z. Free Spoken Digit Dataset (FSDD). 2018. Available online: https://github.com/Jakobovski/free-spoken-digit-dataset (accessed on 2 June 2025).
46. Buzsáki, G. *Rhythms of the Brain*; Oxford University Press: New York, NY, USA, 2006.
47. van Pelt, J.; van Ooyen, A. Estimating neuronal connectivity from axonal and dendritic density fields. *Front. Comput. Neurosci.* **2013**, *7*, 160. [CrossRef] [PubMed]
48. Rees, C.L.; Moradi, K.; Ascoli, G.A. Weighing the evidence in Peters' rule: Does neuronal morphology predict connectivity? *Trends Neurosci.* **2017**, *40*, 63–71. [CrossRef]
49. DeFelipe, J. The evolution of the brain, the human nature of cortical circuits, and intellectual creativity. *Front. Neuroanat.* **2011**, *5*, 29. [CrossRef] [PubMed]
50. Eshelman, L.J.; Schaffer, J.D. Real-coded genetic algorithms and interval-schemata. In *Foundations of Genetic Algorithms*; Elsevier: Amsterdam, The Netherlands, 1993; Volume 2, pp. 187–202.
51. Fortin, F.A.; De Rainville, F.M.; Gardner, M.A.; Parizeau, M.; Gagné, C. DEAP: Evolutionary Algorithms Made Easy. *J. Mach. Learn. Res.* **2012**, *13*, 2171–2175.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

https://doi.org/10.3390/mca31020065

Mathematical and Computational Applications

MDPI

Article

AI-Driven Biopsychosocial Screening for Breast Cancer: Enhancing Risk Prediction via Differential Evolutionary Linear Discriminant Analysis for Feature Extraction

José Luis Llaguno-Roque [1], Adriana Laura López-Lobato [2], Juan Carlos Pérez-Arriaga [3], Héctor Gabriel Acosta-Mesa [2], Ángel J. Sánchez-García [3], Gabriel Gutiérrez-Ospina [4,5], Antonia Barranca-Enríquez [1] and Tania Romo-González [1,*]

[1] Laboratorio de Biología y Salud Integral, Instituto de Investigaciones Biológicas, Universidad Veracruzana, Dr. Luis Castelazo Ayala s/n, Col. Industrial, Ánimas km 2.5 Carretera Xalapa-Veracruz, Xalapa 91190, Veracruz, Mexico; lllaguno@uv.mx (J.L.L.-R.); abarranca@uv.mx (A.B.-E.)
[2] Instituto de Investigaciones en Inteligencia Artificial, Universidad Veracruzana, Campus Sur, Calle Paseo Lote II, Sección Segunda N° 112, Nuevo Xalapa, Xalapa 91097, Veracruz, Mexico; adrlopez@uv.mx (A.L.L.-L.); heacosta@uv.mx (H.G.A.-M.)
[3] Facultad de Estadística e Informática, Universidad Veracruzana, Av. Xalapa s/n, Col. Obrero Campesina, Xalapa 91020, Veracruz, Mexico; juaperez@uv.mx (J.C.P.-A.); angesanchez@uv.mx (Á.J.S.-G.)
[4] Laboratorio de Biología de Sistemas, Instituto de Investigaciones Biomédicas, Universidad Nacional Autónoma de México, Coyoacán 04510, CDMX, Mexico; gabo@iibiomedicas.unam.mx
[5] Coordinación de Psicobiología y Neurociencias, Facultad de Psicología, Universidad Nacional Autónoma de México, Coyoacán 04510, CDMX, Mexico
* Correspondence: tromogonzalez@uv.mx; Tel.: +52-228-8418900 (ext. 13401)

Abstract

In Mexico, the high prevalence and mortality rates associated with breast cancer (BC) constitute a critical public health challenge that demands context-specific preventive measures. This study proposes an integrative framework for predicting BC risk based on a biopsychosocial model. We hypothesize that emotional suppression and repression act as key neuroendocrine disruptors and predisposing factors within the Mexican female population. To test this, we systematically compared the predictive performance of various machine learning classification models using the clinical, psychological, and combined profiles of 110 women. These models were evaluated with and without the application of a robust evolutionary algorithm: Differential Evolutionary Linear Discriminant Analysis for Feature Extraction ($DE - LDA_{FE}$). The results demonstrated that integrating clinical and psychological data into a combined latent space significantly improved the performance of the classification algorithms. The Artificial Neural Network achieved the highest metrics (0.9975 Precision; 0.9976 F1-score). However, due to the inherent "black-box" nature of these models (limited clinical interpretability), the Decision Tree emerged as the optimal practical alternative, providing highly competitive (0.8874 Precision; 0.8853 F1-score) and interpretable results. These findings provide empirical evidence that psychological factors, rather than being mere incidental comorbidities, could be associated with the etiology of breast cancer and be used as risk factors in predicting the disease. Ultimately, this AI-driven biopsychosocial screening model offers a scalable, low-cost, and context-adapted risk assessment tool for early BC diagnosis in Mexican women.

Keywords: breast cancer; psychological traits; machine learning

1. Introduction

Breast cancer (BC) is the most prevalent neoplasm in women worldwide. Nationally, it has been the most frequent and deadliest cancer affecting Mexican women since 2006 (Table A1 in Appendix A) [1–4]. With projections estimating over 50,000 new annual cases by 2050, BC represents an urgent and expanding public health challenge. There is broad consensus that early detection is essential to reduce morbidity and mortality. While mammography remains the gold standard for screening programs worldwide, it presents limitations. Widespread mammographic screening is often restricted by age guidelines and carries concerns regarding radiation exposure, particularly for younger populations [5–7]. This is especially relevant in Mexico, where advanced-stage BC is frequently diagnosed in women aged 20–41 [8,9]. Since 2013, our group has developed non-radiographic resources to identify young Mexican women at elevated risk, aiming to optimize early detection while minimizing unnecessary radiation exposure.

Although the Official Mexican Standard [10] acknowledges that biological, iatrogenic, reproductive, and lifestyle factors interplay to determine BC relative risk, current national screening programs rely almost exclusively on biological and reproductive history. This approach has not substantially reduced incidence or mortality rates, likely because these factors explain, at best, 50% of BC prevalence. We have recently suggested that the accuracy of risk identification would increase by incorporating psycho-affective factors into screening designs [11,12]. Evidence suggests that specific personality traits—such as low emotional containment, low defensiveness, and high stress—are consistently observed in Mexican women with BC prior to diagnosis [11]. Physiologically, emotional suppression and repression are known to alter diurnal cortisol secretion, a dysregulation associated with early mortality in metastatic BC patients [13,14]. Furthermore, principal component analyses have demonstrated that healthy women and those with benign or malignant breast lesions cluster differentially when emotional suppression, repression, and stress symptoms are included in the phenotyping [12]. Notably, when clinical and psychological variables are analyzed together, emotional suppression variables significantly define differential group clustering. These observations support the hypothesis that screening for psychological factors is crucial for prevention, early diagnosis, and understanding BC pathogenesis [15–19].

To improve risk identification in young Mexican women, García-Camacho et al. [20] previously designed a decision tree-based model assessing the Stress Symptom Inventory (ISE), the Courtauld Emotional Control Suppression Scale (CECS), and the Weinberger Adjustment Inventory (WAI). Their results showed that while these tools achieved low accuracy individually (34–40%), their combination improved prediction to 42.7%. This suggests that while the emotional dimension has discriminatory value, the modeling approach required refinement to achieve clinical utility.

On the other hand, recent advances in biopsychosocial risk prediction and explainable AI highlight the importance of integrating multi-domain data for oncology applications. Hussain et al. [21] reviewed machine learning approaches for breast cancer risk prediction, emphasizing the added value of psychosocial variables in enhancing model accuracy. Similarly, radiomics-based studies have demonstrated that multimodal feature fusion, such as parenchymal enhancement and breast symmetry in MRI, improves discrimination power in early detection [22]. In parallel, evolutionary feature extraction methods have been increasingly applied in oncology, as shown by Wang et al. [23], who underscored the role of metaheuristic optimization in refining tumor subtype classification. Explainable AI (XAI) frameworks, such as those discussed by Carriero et al. [24], further stress the necessity of transparency in clinical models, particularly when integrating heterogeneous data sources.

https://doi.org/10.3390/mca31030066

Beyond breast cancer, comparable strategies have been successfully deployed in other medical domains. The NEDL-GCP model [25] introduced a nested ensemble deep learning framework for gynecological cancer risk prediction, demonstrating how ensemble architectures can robustly manage complex clinical datasets. Doumari et al. [26] proposed a novel high-accuracy model for the early diagnosis of Parkinson's disease, illustrating the potential of feature-fusion and classifier optimization in neurodegenerative contexts.

Building on this foundation, we present a new risk assessment model that employs Differential Evolutionary Linear Discriminant Analysis for Feature Extraction ($DE - LDA_{FE}$). This approach integrates biological, reproductive, and psychological variables to manage the complexity of the dataset. Unlike previous models, $DE - LDA_{FE}$ simplifies the observation of risk classes and reveals the synergistic behavior of the data. Furthermore, this work is distinguished by its focus on the Mexican population and its key innovation: the use of psychological data alongside clinical data—an approach that, unlike our group [11,12,27], no other previous study in the country has integrated into breast cancer risk assessment, a gap that this work seeks to fill. Although the analyzed cohort consists of 110 individuals, it accurately represents the psychological and clinical information of Mexican women, a group in which identifying risk factors for breast cancer is a priority.

The $DE - LDA_{FE}$ method was used to improve classification using machine learning algorithms, prioritizing those that offer explainability in their results. This approach lays the groundwork for future applications, such as the development of a web module that estimates breast cancer risk in an accessible, non-invasive, highly accurate, and timely manner.

2. Methodology

2.1. Participants and Study Design

This study utilized clinical and psychological data from 150 women attending routine gynecological consultations at the General Hospital of Mexico "Dr. Eduardo Liceaga." All participants provided written informed consent, ensuring confidentiality and anonymity in accordance with the Declaration of Helsinki (printed in the British Medical Journal, 18 July 1964).

The sample was selected to represent a broad age range (see Table A2 in Appendix A), reflecting the epidemiological reality of BC in Mexican women; it is frequently diagnosed approximately a decade earlier than global averages [5]. Participants were recruited prior to histopathological diagnosis and were subsequently stratified into three groups based on their clinical pathology reports. The Healthy Group (H, n = 50) clustered women with no palpable or radiological breast pathology. The Benign Breast Pathology Group (BBP, n = 50) was formed by women diagnosed with fibrocystic changes, fibroadenomas, or mastitis. The Breast Cancer Group (BC, n = 50) included women diagnosed with infiltrating ductal carcinoma (naïve to treatment).

2.2. Data Collection Instruments

Physical, hereditary, and lifestyle data were collected using a 57-item General Data Questionnaire. Psychological profiles—specifically emotional repression, suppression, and stress symptomatology—were assessed using three validated instruments:

2.2.1. Inventory of Stress Symptoms (ISE)

Designed by Benevides-Pereira et al. [28], the ISE evaluates the frequency of stress manifestations in daily life across three dimensions: psychological, physical, and social symptoms. It consists of 30 items rated on a 5-point Likert scale (0 = never to 4 = always).

https://doi.org/10.3390/mca31030066

Global symptomatology scores are categorized as low (0–22), medium (22–33), or high (>33).

2.2.2. Courtauld Emotional Control Scale (CECS)

Developed by Watson and Greer [29] and validated for Spanish speakers by Durá et al. [30]. The CECS assesses the extent to which individuals suppress reactions to negative emotions. It comprises 21 items across three subscales: Anger, Anxiety, and Depression. Higher scores indicate greater emotional suppression. The instrument demonstrates high internal consistency ($\alpha = 0.95$). For this study, a median split (score = 32) was used to categorize participants into "Emotional Suppression" (score ≥ 32) or "Emotional Expression" (score < 32) groups.

2.2.3. The Weinberger Adjustment Inventory (WAI)

The WAI [31–33] evaluates adjustment styles through three primary scales: Subjective Distress, Restraint, and Defensiveness. The interactions between these scales allow the identification of six typologies of adjustment, including the Repressive (Type C) coping style. The Spanish version validated for the Mexican population [34] was used (Cronbach's α ranges from 0.69 to 0.89 across subscales).

2.3. Feature Encoding and Binarization

Given the binary nature of the clinical data, and in order to compare them with the results of the psychological instruments, the latter were transformed into dichotomous variables to assess the presence or absence of certain psychological traits in the groups of women. The psychological data, originally expressed as continuous scores or subscales, were categorized into low, medium, and high levels according to their values. Subsequently, these categories were recoded into binary variables (1 for the presence of the characteristic; 0 for its absence), using cut-off points previously defined and described in Romo-González et al. [27]. In most cases, the original cut-off points for each instrument were retained; however, in the case of the CECS, these were established based on 95% confidence intervals. The specific binarization criteria are detailed in Table A3 of Appendix A. Appendix C examines the potential loss or gain of information resulting from the binarization of psychological data.

2.4. Machine Learning Identification Model

To manage dataset complexity and achieve optimal separation among the three groups (H, BBP, and BC), a supervised learning approach, Differential Evolutionary Linear Discriminant Analysis for Feature Extraction ($DE - LDA_{FE}$), was employed. This method performs dimensionality reduction while enhancing class separability by combining linear transformations of the original variables with evolutionary optimization to obtain a projection matrix that maximizes class separation in a two-dimensional space. The following sections describe the main components of this method.

2.4.1. Projection-Based Dimensionality Reduction

Let $X \in \mathbb{R}^{m \times n}$ be a dataset with m samples and n features. The goal is to obtain a reduced representation $Y \in \mathbb{R}^{m \times d}$, with $d = 2$, defined as:

$$Y = XW \tag{1}$$

where $W \in \mathbb{R}^{n \times d}$ is the projection matrix whose columns are orthonormal vectors. This transformation produces new features that are linear combinations of the original variables, while orthonormality ensures that the columns form a valid basis for the projected space.

https://doi.org/10.3390/mca31030066

2.4.2. Differential Evolution Optimization

Differential Evolution (DE) is a population-based optimization method that iteratively refines a set of candidate solutions using mutation, crossover, and selection operators.

At iteration (generation) g, the population consists of N candidate solutions (individuals) randomly initialized within the domain of a fitness function f. Each individual is represented by a vector $\mathbf{x_i} \in \mathbb{R}^d$, where d denotes the dimensionality of the problem. In the DE-LDA_{FE} framework, each individual corresponds to a vector $\mathbf{w_i} \in \mathbb{R}^{n \times d}$ that encodes a candidate projection matrix $W_\mathrm{i} \in \mathbb{R}^{n \times d}$, with $d = 2$, as presented in Equation (2). The particularity of these matrixes is that they are orthonormal by columns, to define the projection directions.

$$W_i = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \\ \vdots & \vdots \\ w_{n1} & w_{n2} \end{pmatrix} \Leftrightarrow \mathbf{w}_\mathrm{i} = (w_{11} \;\; w_{21} \;\; \ldots \;\; w_{n1} \;\; w_{12} \;\; w_{22} \;\; \ldots \;\; w_{n2}) \tag{2}$$

The DE/best/1/bin strategy is used. For each target vector $\mathbf{w}_\mathrm{i}$ in the population, a mutant vector is generated as:

$$\mathbf{v}_i = \mathbf{w}_{best} + F(\mathbf{w}_{r1} - \mathbf{w}_{r2}) \tag{3}$$

where $\mathbf{w}_{best}$ is the current best solution, $\mathbf{w}_{r1}, \mathbf{w}_{r2}$ are random individuals, and F is the mutation factor.

A trial vector is then created via binomial crossover:

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } rand_j \leq CR \text{ or } j = j_{rand} \\ w_{i,j} & \text{otherwise} \end{cases} \tag{4}$$

where CR is the crossover rate, and j_{rand} corresponds to a randomly selected position. The selection step retains the solution with the highest fitness between the trial vector $\mathbf{u}_\mathrm{i}$ and the target vector $\mathbf{w}_\mathrm{i}$.

These steps are repeated until a stopping criterion is met, such as reaching a maximum number of generations or population convergence. Thus, through this iterative evolutionary process, the population of candidate solutions is progressively refined by combining and selecting the best individuals at each generation. As a result, the algorithm converges toward projection matrices that increasingly improve class separability, enabling the discovery of a near-optimal solution in the lower-dimensional space.

2.4.3. Fitness Function

For the $DE - LDA_{FE}$ method, the quality of each projection is evaluated using Fisher's discriminant criterion, which seeks to maximize the separation between class means while minimizing the dispersion within each class. This is achieved by maximizing the ratio between the between-class scatter and the within-class scatter in the projected space, as presented in Equation (5).

$$J(W) = \frac{\mid W^T S_B W \mid}{\mid W^T S_W W \mid} \tag{5}$$

where S_B and S_W represent the between-class and within-class scatter matrices, respectively, as presented in Equations (6) and (7), where G denotes the number of classes, μ_g the mean of class g, and μ the global mean.

$$S_B = \sum_{g=1}^{G} m_g (\mu_g - \mu) \left(\mu_g - \mu\right)^T \tag{6}$$

https://doi.org/10.3390/mca31030066

$$S_w = \sum_{g=1}^{G} \sum_{x \in C_g} (x - \mu_g)(x - \mu_g)^T \quad (7)$$

This criterion promotes projections in which samples from the same class are tightly clustered and those from different classes are well separated in the reduced space.

2.5. Evaluation of Classifiers with/Without the $DE - LDA_{FE}$ Algorithm

In this research, to analyze the prediction of breast cancer risk by integrating clinical and psychological variables, a comparison of six classification models is performed: K-nearest neighbors (K-nn; k = 3, 5, 7, 9, and 11); Linear Discriminant Analysis (Linear); Gaussian classifier (Gaussian); Decision Tree (DT); Support Vector Machine (SVM); and Artificial Neural Network (ANN). This comparison is made by considering the raw data of the clinical variables, the psychological variables, and finally, both. In addition, the feature extraction method, Feature Extraction ($DE - LDA_{FE}$), is used to graphically analyze classification performance and improve the classifier's accuracy., as presented in Figure 1.

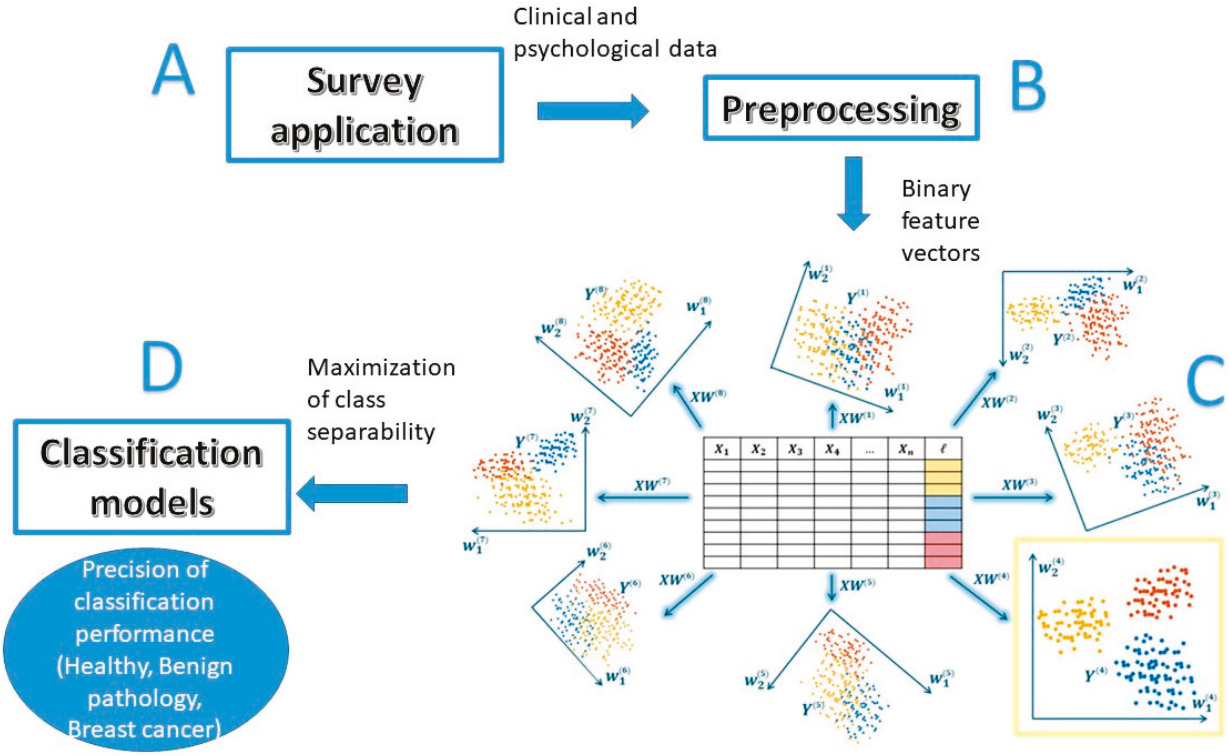


Figure 1. Schematic representation of the Biopsychosocial Risk Prediction Model. This flow diagram illustrates the four computational stages of the study: (**A**) Data Acquisition: Collection of biological, reproductive, and psychological variables from the study population (n = 150). (**B**) Preprocessing: Transformation of raw data into binary feature vectors (0, 1) to standardize input. (**C**) Feature Extraction: Application of the Differential Evolutionary Linear Discriminant Analysis ($DE - LDA_{FE}$) to identify the optimal projection matrix (W) that maximizes class separability. (**D**) Classification: Probabilistic assignment of new cases into Healthy, Benign Pathology, or Breast Cancer categories based on the classification models.

To determine the most suitable Artificial Neural Network (ANN) configuration, a systematic experimental evaluation was conducted using feedforward fully connected neural networks for multiclass classification. The tested architectures consisted of 1 to 5 hidden layers, each with the same number of neurons. The number of neurons per layer was varied across the set {10, 20, 30, 40, 50, 100, 200, 300, 400, 500}, generating a broad range of shallow and deep network configurations. In all cases, each network received the dataset

https://doi.org/10.3390/mca31030066

as input and produced class predictions at the output layer. This structured exploration enabled comparison of architectures with different depths and capacities to identify the configuration that achieved the best classification performance according to the selected evaluation metrics.

For the $DE - LDA_{FE}$ method, a grid search strategy was employed by systematically exploring combinations of the Differential Evolution (DE) parameters, including population size $N \in \{100, 300, 500, 1000, 2000\}$, number of generations $NG \in \{300, 500, 700, 900\}$, crossover rate $CR \in \{0.3, 0.5, 0.7\}$, and scale factor $F \in \{0.3, 0.5, 0.7\}$. The best classification values were obtained with the configurations shown in Table 1 for each data type.

Table 1. Parameter settings used for the $DE - LDA_{FE}$ method across the experiments for each dataset.

Parameter	Clinical	Psychological	Both
Population size (N)	2000	2000	300
Generations (NG)	300	300	700
Crossover rate (CR)	0.5	0.5	0.5
Mutation rate (F)	0.3	0.3	0.7

3. Results

This section presents the comparative performance of the selected classification models in predicting BC risk using clinical, psychological, and integrated datasets. Following the exclusion of records with missing data, a final cohort of 110 patients (out of the original 150) was included in the analysis.

Model performance was assessed using both the raw data and data obtained after using the $DE - LDA_{FE}$ feature extraction method. To rigorously evaluate the classifiers, standard performance metrics were utilized, including True Positive (TP) Rate, False Positive (FP) Rate, Accuracy, Recall, F1-score, and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC).

For the experiments, the classification performance was evaluated on the training set. This choice was made because the primary goal of the proposed method is to compare the discriminative quality of different feature representations rather than to estimate generalization performance. Although this approach may yield optimistic accuracy values, it ensures a fair and controlled comparison across methods, since all models are evaluated under identical conditions. Additionally, the dataset's limited size supports the use of this evaluation strategy.

For the raw variables, the results presented consider only one execution, since the dataset employed does not vary; however, for the projections of the variables with the $DE - LDA_{FE}$ method, since it is a stochastic, population-based metaheuristic algorithm, the projections are not unique, so the results presented are the mean of 10 runs, and, to perform a fair comparison, the best solution obtained on these 10 runs is employed for comparison with the results of the raw data.

3.1. Classification Based on Clinical Variables

3.1.1. Analysis of Raw Clinical Data

In this initial experimental phase, the classification models were trained using the clinical variables in their original, unprocessed state. This baseline analysis serves to evaluate the inherent discriminatory power of standard biological and reproductive risk factors prior to the application of any feature extraction or dimensionality reduction techniques. The comparative performance metrics for each classifier in this scenario are detailed in Table 2.

https://doi.org/10.3390/mca31030066

Table 2. Performance metrics of classification models using raw clinical data.

Classification Models	TP RATE	FP RATE	Precision	Recall	F1-Score	AUC-ROC
K-nn (3)	0.6548	0.3673	0.7466	0.6518	0.6690	0.8676
K-nn (5)	0.6326	0.3742	0.6947	0.5972	0.6056	0.8300
K-nn (7)	0.6626	0.4309	0.6547	0.5811	0.5857	0.7790
K-nn (9)	0.6116	0.3990	0.6167	0.5598	0.5575	0.7313
K-nn (11)	0.6164	0.4099	0.5720	0.5167	0.5070	0.7263
Linear	0.6944	0.3978	0.6471	0.5951	0.6076	0.7978
Gaussian	0.7894	0.3757	0.8628	0.7937	0.8064	0.9528
DT	0.7285	0.3272	0.7939	0.6969	0.7165	0.8966
SVM	0.6827	0.4007	0.6892	0.6144	0.6255	0.7856
ANN (30)	0.8519	0.2520	0.9144	0.8964	0.9034	0.9892

Classification models: K-Nearest Neighbors (K-nn), Linear Discriminant Classifier (Linear), Gaussian Classifier (Gaussian), Decision Tree (DT), Support Vector Machine (SVM), Artificial Neural Network (ANN). The optimal ANN architecture consists of two hidden layers with 30 neurons each.

As shown in Table 1, the performance of classifiers trained on raw clinical data varied significantly depending on the algorithm's architecture. The ANN with 30 neurons demonstrated superior predictive capability, achieving the highest scores across all metrics, with an F1-score of 0.9034 and an AUC-ROC of 0.9892. The Gaussian classifier also performed robustly, securing the second-best performance (F1-score: 0.8064).

Conversely, the K-nn algorithm exhibited an inverse relationship between the number of neighbors (K) and model performance; as K increased from 3 to 11, the F1-score progressively declined from 0.6690 to 0.5070. Linear and SVM models yielded moderate results (F1-scores $\approx$ 0.60–0.62), suggesting that the boundaries between risk groups in the raw clinical space are non-linear and complex, requiring more sophisticated modeling approaches like ANNs to be effectively decoded.

3.1.2. Classification Using Clinical Data Projections via $DE - LDA_{FE}$ Method

As established in previous sections, the $DE - LDA_{FE}$ method was implemented to improve class separation by projecting the original variables into an optimized, lower-dimensional latent space. This projection aimed to enhance the discriminative performance of the classification models. The average performance metrics obtained using these clinical projections, calculated over 10 independent runs, are presented in Table 3.

Table 3. Average performance metrics of classification models using clinical variable projections via $DE - LDA_{FE}$ method.

Classification Models	TP RATE	FP RATE	Precision	Recall	F1-Score	AUC-ROC
K-nn (3)	0.6835	0.3416	0.7810	0.7096	0.7273	0.8937
K-nn (5)	0.6549	0.3478	0.7141	0.6553	0.6673	0.8586
K-nn (7)	0.6823	0.3984	0.6926	0.6262	0.6370	0.8228
K-nn (9)	0.6705	0.3812	0.6795	0.6010	0.6137	0.8039
K-nn (11)	0.6601	0.3885	0.6684	0.5879	0.5997	0.7961
Linear	0.6914	0.3993	0.6504	0.5986	0.6101	0.7940
Gaussian	0.6795	0.3993	0.6745	0.6031	0.6159	0.7808
DT	0.7360	0.3227	0.8043	0.7733	0.7831	0.9089
SVM	0.6902	0.3953	0.6645	0.5822	0.5952	0.7993
ANN (400)	0.8238	0.3226	0.9111	0.8911	0.8990	0.9875

Classification models: K-Nearest Neighbors (K-nn), Linear Discriminant Classifier (Linear), Gaussian Classifier (Gaussian), Decision Tree (DT), Support Vector Machine (SVM), Artificial Neural Network (ANN). The optimal ANN architecture comprises two hidden layers with 400 neurons per layer.

https://doi.org/10.3390/mca31030066

As indicated in Table 3, the models demonstrating the highest predictive capability when utilizing the projected clinical variables were the ANN architecture with 400 neurons (Precision: 0.9111; F1-score: 0.8990) and the DT model (Precision: 0.8043; F1-score: 0.7831). The remaining models exhibited moderate, comparable performances. Consistent with the raw data findings, the K-nn model displayed a decline in both Precision and F1-score as the number of neighbors (K) increased, confirming that K = 3 remains the optimal configuration for this spatial mapping.

While Table 3 reflects the average robustness across 10 iterations, it is also highly relevant to identify the peak potential of each classifier. Table 4 details the single best-performing run for each model based on the clinical projections.

Table 4. Best individual run performance for each classification model using clinical variable projections.

Classification Models	Experiment	TP RATE	FP RATE	Precision	Recall	F1-Score	AUC-ROC
K-nn (3)	10	0.6895	0.3396	0.8008	0.7173	0.7380	0.8966
K-nn (5)	2	0.6462	0.3432	0.7587	0.6922	0.7080	0.8520
K-nn (7)	10	0.6787	0.3933	0.7401	0.6504	0.6644	0.8235
K-nn (9)	3	0.6765	0.3910	0.7012	0.6343	0.6492	0.8059
K-nn (11)	8	0.6752	0.3829	0.6939	0.6292	0.6455	0.8148
Linear	1	0.6935	0.3995	0.6703	0.6041	0.6204	0.7967
Gaussian	8	0.6784	0.4004	0.6865	0.6092	0.6250	0.7797
DT	8	0.7268	0.3246	0.8234	0.7842	0.7935	0.9125
SVM	7	0.6890	0.3972	0.6779	0.5907	0.6049	0.7981
ANN (400)	3	0.8379	0.2829	0.9149	0.8947	0.9024	0.9894

Classification models: K-Nearest Neighbors (K-nn), Linear Discriminant Classifier (Linear), Gaussian Classifier (Gaussian), Decision Tree (DT), Support Vector Machine (SVM), Artificial Neural Network (ANN).

The results in Table 4 mirror the trends observed in the average metrics, reaffirming the superiority of the ANN model. Notably, the peak performance was achieved during Experiment 3 using an expanded architecture of 500 neurons, which yielded an exceptional Precision of 0.9149 and an F1-score of 0.9024. The DT model similarly achieved its peak during Experiment 8, solidifying its position as the second most effective classifier for these specific clinical projections.

3.1.3. Performance Comparison: Raw vs. Projected Clinical Data

Figure 2 illustrates the comparative performance of the classification models when trained on raw clinical variables versus the optimized latent projections generated by the $DE - LDA_{FE}$ method. As observed, for K-nn, DT, SVM, and Linear, the application of the $DE - LDA_{FE}$ method successfully enhanced class separability, thereby contributing to an overall improvement in classification metrics.

However, two notable exceptions emerged: Gaussian and ANN classifiers achieved their peak performance using the raw, not projected clinical data. This outcome suggests that these specific non-linear algorithms possess an inherent capacity to map the complex, multi-dimensional relationships of biological risk factors in their original state, without requiring prior dimensionality reduction or feature extraction.

https://doi.org/10.3390/mca31030066

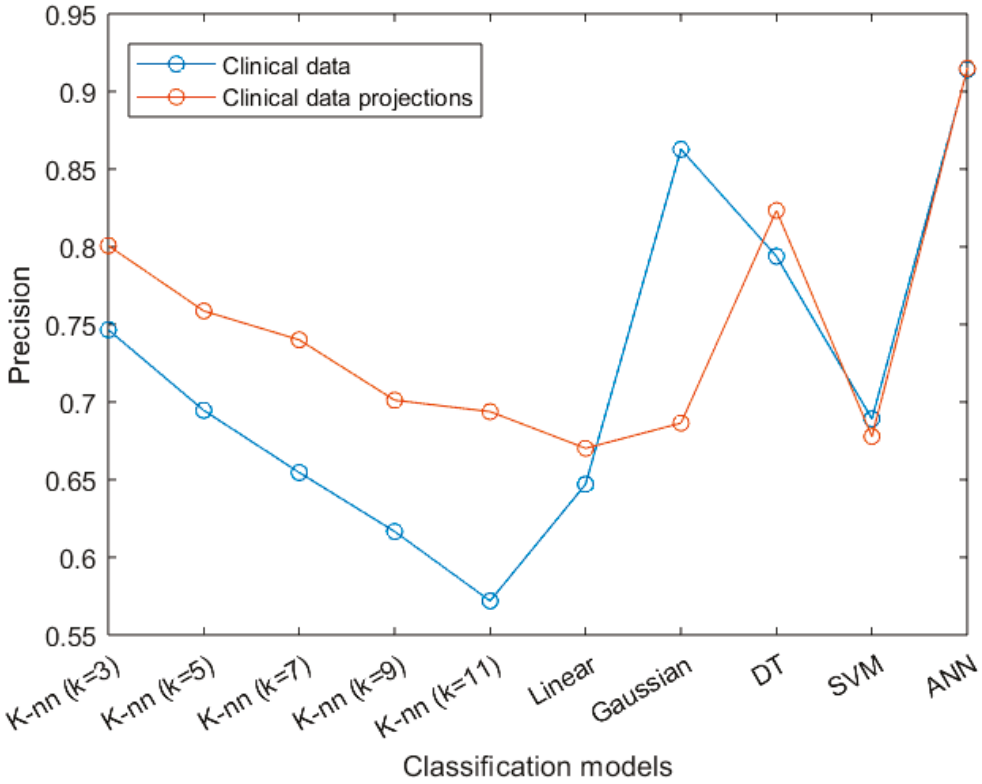


Figure 2. Performance comparison of classification models based on clinical data. The graph contrasts the predictive performance of each algorithm when utilizing raw clinical variables (baseline) versus the optimized projections derived from the $DE - LDA_{FE}$ method. Abbreviations: K-nn: K-nearest neighbor; Linear: Linear Discriminant Classifier; DT: Decision Tree; SVM: Support Vector Machine; ANN: Artificial Neural Network.

3.2. Classification Based on Psychological Variables

3.2.1. Analysis of Raw Psychological Variables

In this phase of the analysis, the classification models were trained using the raw psychological variables exclusively (without any preprocessing or projection). This experiment establishes a baseline to evaluate the inherent predictive power of psychological factors, such as emotional repression, suppression, and stress, when analyzed in isolation. The comparative performance metrics for each classifier in this raw psychological space are presented in Table 5.

Table 5. Performance metrics of classification models using raw psychological data.

Classification Model	TP RATE	FP RATE	Precision	Recall	F1-Score	AUC-ROC
K-nn (3)	0.6091	0.4252	0.5689	0.5187	0.5257	0.7865
K-nn (5)	0.5996	0.4416	0.5322	0.4772	0.4575	0.7487
K-nn (7)	0.6601	0.5053	0.4942	0.4668	0.4502	0.7129
K-nn (9)	0.6049	0.4793	0.4735	0.4455	0.4330	0.6717
K-nn (11)	0.6243	0.5184	0.4781	0.4526	0.4385	0.6456
Linear	0.7005	0.4011	0.5999	0.5656	0.5717	0.8036
Gaussian	0.6870	0.3936	0.4780	0.6255	0.5419	0.7983
DT	0.7128	0.3735	0.7402	0.7294	0.7298	0.8796
SVM	0.7223	0.3912	0.6974	0.6158	0.6263	0.8362
ANN (20)	0.8933	0.1779	1.0000	1.0000	1.0000	1.0000

Classification models: K-Nearest Neighbors (K-nn), Linear Discriminant Classifier (Linear), Gaussian Classifier (Gaussian), Decision Tree (DT), Support Vector Machine (SVM), Artificial Neural Network (ANN). The optimal ANN architecture comprises one hidden layer with 20 neurons per layer.

https://doi.org/10.3390/mca31030066

As shown in Table 5, the classification models exhibited varying degrees of success when processing the psychological variables. Most notably, the ANN, configured with an optimized hidden layer of 20 neurons, achieved extraordinary, perfect predictive metrics across the board (Precision: 1.0000; F1-score: 1.0000; AUC-ROC: 1.0000). This indicates an absolute separation of the risk classes by the network within this specific training configuration. The DT model emerged as the second most effective classifier for raw psychological data, recording a Precision of 0.7402 and an F1-score of 0.7298. The remaining probabilistic and linear models demonstrated moderate performance. Consistent with the trends observed in the clinical data analysis, the K-nn algorithm displayed an inverse relationship between the number of neighbors (K) and model efficacy; as K increased, both Precision and the F1-score systematically declined, confirming that K = 3 remains the optimal parameter within this configuration.

3.2.2. Classification Using Psychological Data Projections via $DE - LDA_{FE}$ Method

Following the same procedure applied to the clinical data, the psychological variables were projected into an optimized latent space using the $DE - LDA_{FE}$ method. This transformation aims to maximize class separability and enhance the predictive performance of the classification algorithms. The average performance metrics obtained from these psychological projections, calculated over 10 independent runs, are presented in Table 6.

Table 6. Average performance metrics of classification models using psychological variable projections via $DE - LDA_{FE}$ method.

Classification Model	TP RATE	FP RATE	Precision	Recall	F1-Score	AUC-ROC
K-nn (3)	0.6914	0.3390	0.7708	0.7238	0.7367	0.9129
K-nn (5)	0.6698	0.3393	0.6981	0.6581	0.6667	0.8714
K-nn (7)	0.7004	0.3844	0.6670	0.6401	0.6450	0.8492
K-nn (9)	0.6879	0.3876	0.6535	0.6194	0.6248	0.8350
K-nn (11)	0.6892	0.4129	0.6422	0.6073	0.6125	0.8251
Linear	0.6935	0.4041	0.5941	0.5734	0.5774	0.7934
Gaussian	0.6945	0.4031	0.6147	0.5880	0.5938	0.7954
DT	0.7499	0.3008	0.8224	0.8169	0.8149	0.9339
SVM	0.7000	0.4010	0.6581	0.5918	0.5983	0.8031
ANN (500)	0.8930	0.1876	1.0000	1.0000	1.0000	1.0000

Classification models: K-Nearest Neighbors (K-nn), Linear Discriminant Classifier (Linear), Gaussian Classifier (Gaussian), Decision Tree (DT), Support Vector Machine (SVM), Artificial Neural Network (ANN). The optimal ANN architecture comprises two hidden layers with 500 neurons per layer.

As detailed in Table 6, the ANN, dynamically configured with an expanded hidden layer of 500 neurons for this projected space, maintained its perfect predictive capability across the key metrics (Precision: 1.0000; F1-score: 1.0000). The DT model demonstrated notable improvement compared to its raw data baseline, securing the second-highest average performance (Precision: 0.8224; F1-score: 0.8149). The remaining classifiers exhibited similar, moderate performances. Regarding the K-nn model, the inverse relationship between the number of neighbors (K) and model accuracy persisted, confirming K = 3 as the optimal configuration for this projected space.

While Table 6 illustrates the average stability of the classifiers, it is equally important to evaluate their peak predictive potential. Table 7 presents the single best-performing iteration out of the 10 total executions for each model using the projected psychological variables.

https://doi.org/10.3390/mca31030066

Table 7. Best individual run performance for each classification model using psychological variable projections.

Classification Model	Experiment	TP RATE	FP RATE	Precision	Recall	F1-Score	AUC-ROC
K-nn (3)	7	0.6932	0.3522	0.8212	0.7353	0.7565	0.9150
K-nn (5)	3	0.6948	0.3367	0.7605	0.7097	0.7206	0.8953
K-nn (7)	7	0.7126	0.3867	0.7313	0.6942	0.7044	0.8520
K-nn (9)	8	0.6946	0.3761	0.7035	0.6895	0.6946	0.8593
K-nn (11)	7	0.7036	0.4166	0.6661	0.6401	0.6476	0.8370
Linear	6	0.6998	0.4014	0.6309	0.5964	0.6045	0.8023
Gaussian	3	0.6975	0.4015	0.6427	0.6196	0.6246	0.7999
DT	3	0.7780	0.2562	0.8668	0.8778	0.8682	0.9640
SVM	7	0.7003	0.3998	0.7059	0.6063	0.6179	0.8046
ANN (500)	1	0.9082	0.1498	1.0000	1.0000	1.0000	1.0000

Classification models: K-Nearest Neighbors (K-nn), Linear Discriminant Classifier (Linear), Gaussian Classifier (Gaussian), Decision Tree (DT), Support Vector Machine (SVM), Artificial Neural Network (ANN).

The data in Table 7 corroborate the findings from the average metrics. The ANN model achieved a flawless classification outcome during Experiment 1 (Precision: 1.0000; F1-score: 1.0000). The DT similarly reached its peak performance during Experiment 3, yielding a Precision of 0.8668 and an F1-score of 0.8682, buttressing its status as a highly effective secondary classifier for this dataset. Lastly, for the KNN model, as the number of neighbors (K) increases, both the Precision and the F1-score tend to decrease. Therefore, the value of K = 3 corresponds to the best performance within this configuration.

3.2.3. Performance Comparison: Raw vs. Projected Psychological Data

Figure 3 illustrates the comparative performance of the classification models when trained on raw psychological variables versus the optimized latent projections generated by the $DE - LDA_{FE}$ method. Consistent with the findings from the clinical dataset, the application of the $DE - LDA_{FE}$ method successfully enhanced class separability, contributing to a systematic improvement in classification metrics across K-nn, Linear, DT, and SVM.

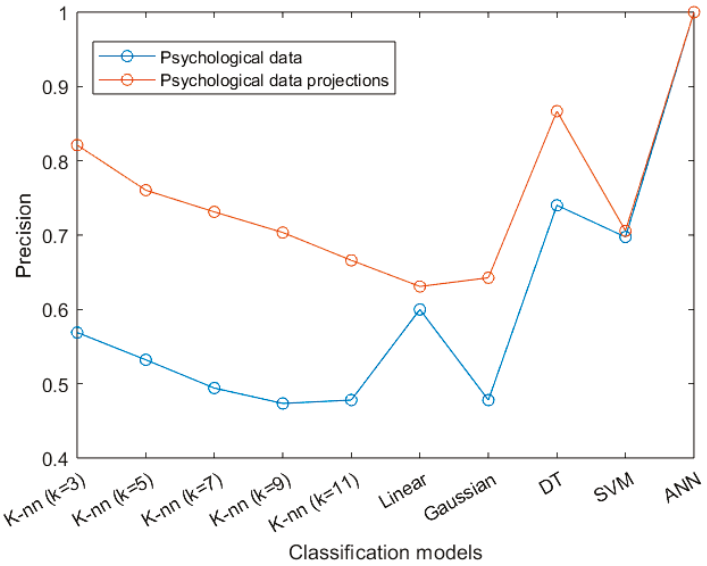


Figure 3. Comparison of classification models based on psychological data. The graph contrasts the predictive performance of each algorithm when utilizing raw psychological variables (baseline) versus the optimized projections derived from the $DE - LDA_{FE}$ method. Abbreviations: K-nn: K-nearest neighbor; Linear: Linear Discriminant Classifier; DT: Decision Tree; SVM: Support Vector Machine; ANN: Artificial Neural Network.

https://doi.org/10.3390/mca31030066

The sole exception in this experimental phase was the ANN model. Because the dynamically configured ANN had already achieved perfect predictive accuracy (1.0000) using the raw psychological data, its performance metrics remained identical when processing the projected data, demonstrating a ceiling effect in its classification capability for this specific model.

3.3. Classification Based on Combined Clinical and Psychological Variables

3.3.1. Analysis of Raw Combined Data

In this comprehensive phase of the study, the classification models were evaluated using an integrated dataset comprising both the raw clinical and psychological variables. This experiment was designed to determine whether combining these two distinct domains, the biological risk factors and the psychological states, without any prior preprocessing or projection, enhances the overall predictive accuracy for breast cancer risk. The comparative performance metrics are presented in Table 8.

Table 8. Performance metrics of classification models using raw combined clinical and psychological data.

Classification Model	TP RATE	FP RATE	Precision	Recall	F1-Score	AUC-ROC
K-nn (3)	0.6369	0.3998	0.7259	0.6404	0.6513	0.8510
K-nn (5)	0.6026	0.4145	0.6340	0.5432	0.5468	0.7579
K-nn (7)	0.6167	0.4562	0.5632	0.5015	0.4953	0.6959
K-nn (9)	0.5743	0.4556	0.4833	0.4237	0.4030	0.6510
K-nn (11)	0.5948	0.4882	0.4959	0.4327	0.4036	0.6378
Linear	0.7812	0.3623	0.7931	0.7762	0.7834	0.9227
Gaussian	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
DT	0.7215	0.3293	0.7770	0.7629	0.7679	0.9053
SVM	0.7958	0.3550	0.8929	0.8554	0.8693	0.9448
ANN (20)	0.8765	0.2249	1.0000	1.0000	1.0000	1.0000

Classification models: K-Nearest Neighbors (K-nn), Linear Discriminant Classifier (Linear), Gaussian Classifier (Gaussian), Decision Tree (DT), Support Vector Machine (SVM), Artificial Neural Network (ANN). The optimal ANN architecture comprises one hidden layer with 20 neurons per layer.

As detailed in Table 8, the ANN, configured with 20 neurons in its hidden layer, once again demonstrated exceptional predictive capacity on the raw data, achieving perfect scores in several key metrics (Precision: 1.0000; F1-score: 1.0000). The integration of both variable sets significantly improved the performance of the SVM model, which emerged as the second most robust classifier for this combined dataset (Precision: 0.8929; F1-score: 0.8693). Consistent with previous experimental phases, the K-nn model's performance degraded progressively as the number of neighbors (K) increased, reaffirming that K = 3 remains the optimal parameter setting for this algorithm.

Intriguingly, the Gaussian classifier failed to converge valid predictions in this combined raw space (yielding 0.0000 across all metrics). This was further analyzed and is primarily attributed to the high dimensionality of the feature space relative to the number of samples. In this setting, the covariance matrix becomes ill-conditioned or singular, preventing its reliable inversion and leading to degenerate probability estimates, which explains the zero values observed across all evaluation metrics. This issue was verified by inspecting the rank and condition number of the covariance matrix. These findings further justify the use of dimensionality reduction methods, such as $DE - LDA_{FE}$, which project the data into a lower-dimensional space where covariance estimation is stable, and class separability is significantly improved.

https://doi.org/10.3390/mca31030066

3.3.2. Classification Using Combined Data Projections via the $DE - LDA_{FE}$ Method

To address the complexities of the combined dataset, the integrated clinical and psychological variables were projected into an optimized latent space using the $DE - LDA_{FE}$ method. Table 8 presents the average performance metrics of the classification models, calculated over 10 independent runs, when trained on these combined, optimized projections.

As shown in Table 9, the models demonstrating the highest predictive capability on the projected combined variables were the ANN, utilizing an expanded 500-neuron architecture, which achieved near-perfect average metrics (Precision: 0.9975; F1-score: 0.9976), followed by the DT model (Precision: 0.8874; F1-score: 0.8853).

Table 9. Projection results of the clinical and psychological variables obtained with the $DE - LDA_{FE}$ method.

Classification Model	TP RATE	FP RATE	Precision	Recall	F1-Score	AUC-ROC
K-nn (3)	0.7454	0.2736	0.8435	0.8384	0.8400	0.9586
K-nn (5)	0.7439	0.2578	0.7983	0.7700	0.7802	0.9375
K-nn (7)	0.7677	0.2751	0.7874	0.7724	0.7784	0.9236
K-nn (9)	0.7610	0.2798	0.7790	0.7701	0.7738	0.9208
K-nn (11)	0.7547	0.2924	0.7626	0.7544	0.7575	0.9188
Linear	0.7815	0.3621	0.7900	0.7716	0.7792	0.9232
Gaussian	0.7849	0.3604	0.7956	0.7817	0.7876	0.9283
DT	0.7620	0.3064	0.8874	0.8867	0.8853	0.9593
SVM	0.7830	0.3613	0.7880	0.7632	0.7723	0.9254
ANN (500)	0.9073	0.1451	0.9975	0.9977	0.9976	1.0000

Classification models: K-Nearest Neighbors (K-nn), Linear Discriminant Classifier (Linear), Gaussian Classifier (Gaussian), Decision Tree (DT), Support Vector Machine (SVM), Artificial Neural Network (ANN). The optimal ANN architecture comprises two hidden layers with 500 neurons per layer.

Notably, the application of the $DE - LDA_{FE}$ method resolved the prior convergence failure of the Gaussian model on the raw combined data. By optimizing the dimensional space, the Gaussian algorithm successfully generated predictions, achieving a highly competitive Precision of 0.7956. Lastly, among the K-nn configurations, an increase in neighbors (K) again correlated with a decrease in Precision and F1-score, establishing again that K = 3 is the optimal baseline for this model.

Table 10 details the single best-performing run out of the 10 total executions for each model, highlighting the peak potential of these classifiers when utilizing the combined data projections.

Table 10. Best individual run performance for each classification model using combined data projections via $DE - LDA_{FE}$.

Classification Model	Experiment	TP RATE	FP RATE	Precision	Recall	F1-Score	AUC-ROC
K-nn (3)	1	0.7478	0.2748	0.8663	0.8630	0.8643	0.9636
K-nn (5)	4	0.7542	0.2528	0.8239	0.7956	0.8062	0.9459
K-nn (7)	2	0.7716	0.2723	0.8160	0.7847	0.7951	0.9276
K-nn (9)	9	0.7633	0.2759	0.8160	0.8052	0.8100	0.9277
K-nn (11)	2	0.7574	0.2941	0.7868	0.7705	0.7772	0.9220
Linear	2	0.7823	0.3616	0.8008	0.7853	0.7918	0.9246
Gaussian	6	0.7862	0.3598	0.8037	0.7924	0.7973	0.9303
DT	5	0.7783	0.2578	0.9148	0.9099	0.9077	0.9730
SVM	10	0.7842	0.3605	0.7991	0.7776	0.7860	0.9275
ANN (500)	3	0.8739	0.2205	1.0000	1.0000	1.0000	1.0000

Classification models: K-Nearest Neighbors (K-nn), Linear Discriminant Classifier (Linear), Gaussian Classifier (Gaussian), Decision Tree (DT), Support Vector Machine (SVM), Artificial Neural Network (ANN).

https://doi.org/10.3390/mca31030066

The results in Table 10 mirror the trends observed in the average metrics, reinforcing the superiority of the ANN model, which achieved flawless classification (Precision: 1.0000; F1-score: 1.0000) during Experiment 3. The DT model similarly peaked during experiment 5, yielding an outstanding Precision of 0.9148 and an F1-score of 0.9077. The remaining algorithms exhibited consistent, high-tier performance, validating the efficacy of the $DE - LDA_{FE}$ transformation on combined, heterogeneous datasets. Once again, in the case of the K-nn model, as the number of neighbors (K) increases, both the Precision and the F1-score tend to decrease. Therefore, the value of K = 3 corresponds to the best performance within this configuration.

3.3.3. Performance Comparison: Raw vs. Projected Combined Data

Figure 4 illustrates the comparative performance of the classification models when trained on the raw, combined clinical and psychological dataset versus the optimized latent projections generated by the $DE - LDA_{FE}$ method. Consistent with the findings from the isolated datasets, the application of the $DE - LDA_{FE}$ method successfully enhanced class separability, contributing to improved classification metrics for K-nn, Linear, and DT models.

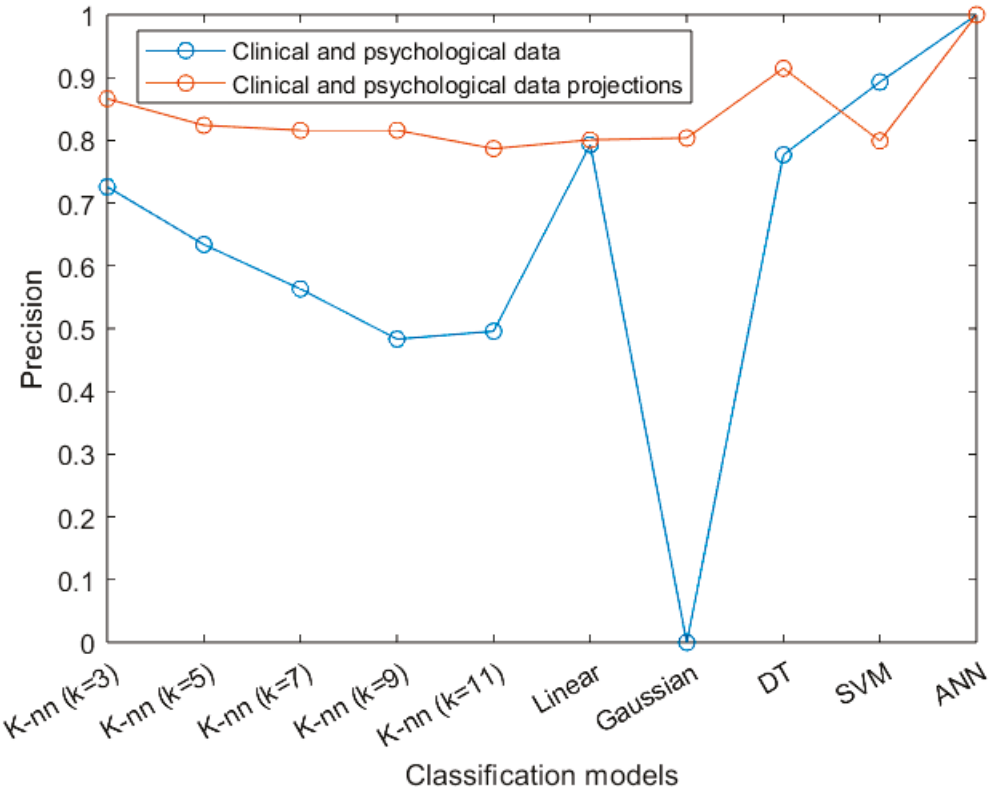


Figure 4. Comparison of classification models based on combined clinical and psychological data. The graph contrasts the predictive performance of each algorithm when utilizing raw integrated variables (baseline) versus the optimized projections derived from the $DE - LDA_{FE}$ method. Abbreviations: K-nn: K-nearest neighbor; Linear: Linear Discriminant Classifier; DT: Decision Tree; SVM: Support Vector Machine; ANN: Artificial Neural Network.

Two notable exceptions to this trend were the SVM and the ANN models, both of which achieved their peak performance utilizing the raw, unprojected data. As established in Section 3.3.1, the dynamically configured ANN had already reached perfect predictive accuracy (1.0000) on the raw combined data, while the SVM demonstrated a remarkably high natural affinity for delineating the raw integrated feature space.

https://doi.org/10.3390/mca31030066

In stark contrast, the Gaussian model benefited the most profoundly from the $DE - LDA_{FE}$ projections. While the algorithm completely failed to converge on the raw, heterogeneous data (yielding metrics of 0.0000), the optimized dimensionality reduction provided by the $DE - LDA_{FE}$ method resolved these mathematical singularities. This transformation allowed the Gaussian model to successfully compute probabilities and achieve highly competitive performance, underscoring the vital role of $DE - LDA_{FE}$ in stabilizing sensitive algorithms when integrating complex, multi-domain datasets.

3.4. Comprehensive Analysis of Variable Integration

Figure 5 illustrates a comparative analysis of the precision achieved by the various classification models across the raw clinical, psychological, and integrated datasets. Across all unprojected data types, the ANN consistently demonstrated superior performance, achieving the highest precision and proving to be the best-performing architecture for this study's baseline data.

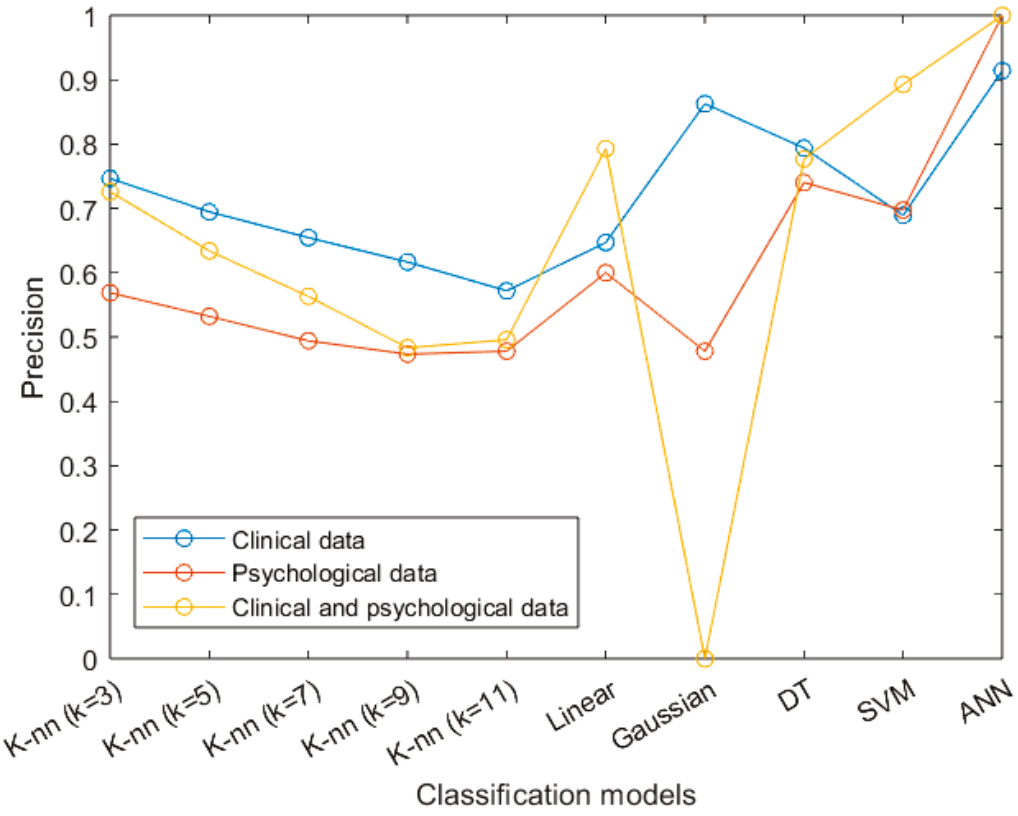


Figure 5. Performance comparison of classification models using raw datasets. The graph contrasts the precision of each classifier across the raw clinical variables, raw psychological variables, and the unprojected combination of both domains. Abbreviations: K-nn: K-nearest neighbor; Linear: Linear Discriminant Classifier; DT: Decision Tree; SVM: Support Vector Machine; ANN: Artificial Neural Network.

Regarding the K-nn model, the analysis confirms that precision inversely correlates with the number of neighbors (K). The peak performance within this family of models was consistently observed at K = 3, particularly within the clinical and combined datasets, whereas higher K values systematically reduced the algorithm's discriminative capacity. The Gaussian model performed adequately with isolated clinical and psychological data; however, it experienced a total convergence failure (precision dropping to zero) when attempting to process the raw combined variables, suggesting profound mathematical difficulty in handling this unprojected, heterogeneous data integration. The DT model

https://doi.org/10.3390/mca31030066

exhibited robust and stable performance across all three data domains, standing out slightly on the clinical data with accuracy metrics superior to most baseline models, though not reaching the absolute ceiling of the ANN. Finally, the SVM and Linear maintained a generally intermediate performance profile, yielding acceptable precision without significantly outperforming the top-tier algorithms.

Figure 6 presents a parallel comparison of the models' predictive accuracy when utilizing the optimized data projections (clinical, psychological, and combined) generated by the $DE - LDA_{FE}$ method. Significantly, the data indicates that the integration of both clinical and psychological variables into a combined, projected latent space yields the highest precision values across the majority of the evaluated models. This outcome strongly suggests that the complementary information derived from pairing biological risk factors with psychological states, when properly optimized and mathematically smoothed via the $DE - LDA_{FE}$ method, significantly enhances the overall discriminatory power and predictive reliability of the classifiers.

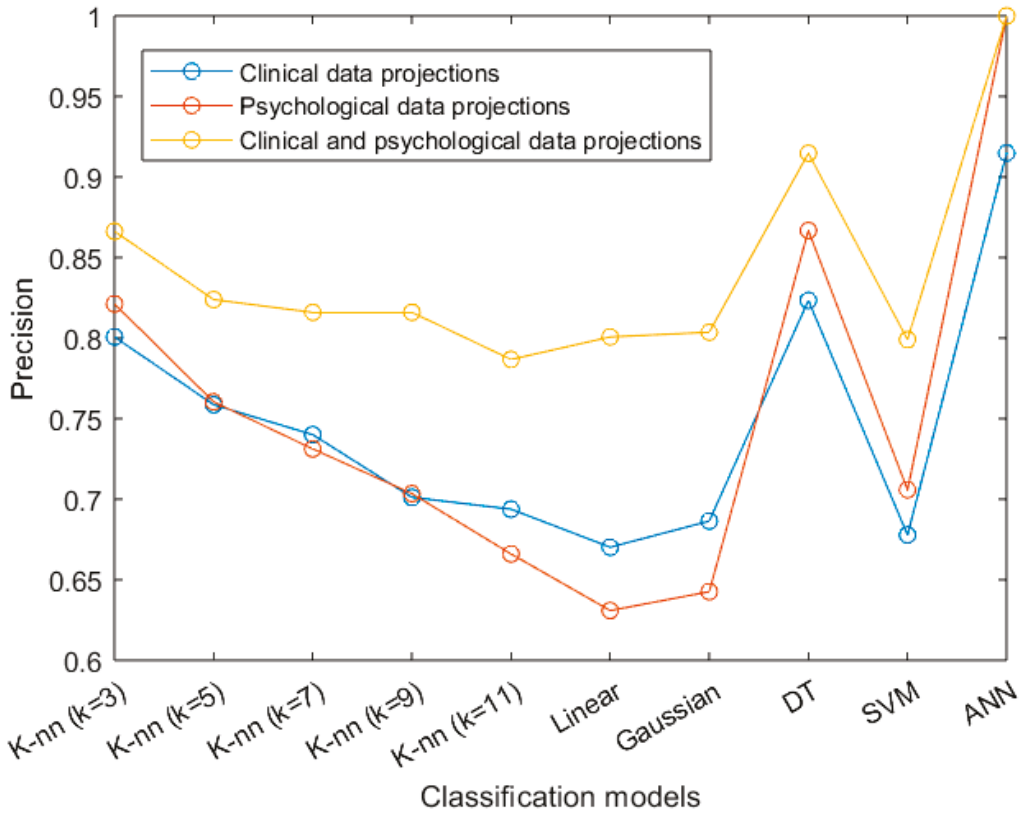


Figure 6. Performance comparison of classification models using $DE - LDA_{FE}$ data projections. The graph contrasts the classification accuracy across the projected clinical space, projected psychological space, and the optimized projection of the combined variables. Abbreviations: K-nn: K-nearest neighbor; Linear: Linear Discriminant Classifier; DT: Decision Tree; SVM: Support Vector Machines; ANN: Artificial Neural Network.

While the main objective of this work is to use clinical, psychological, and combined data to construct a breast cancer risk profile, it is also possible to identify, using the DE-LDA_FE method, isolated characteristics of each data type that may contribute to risk prediction. This is achieved through the use of a linear model, whose transformation matrix provides coefficients that reflect the relevance of the variables, ordered in descending order. The variables considered relevant are presented in Appendix B (Tables A4–A6). However, as noted by Montes-Nogueira et al. (2018) and Romo-Gonzalez et al. (2018) [11,12], the

https://doi.org/10.3390/mca31030066

breast cancer risk profile is more robust when clinical and psychological data are integrated together, rather than using isolated characteristics.

4. Discussion

4.1. The Need for Context-Specific Risk Models

Breast cancer remains one of the most critical public health issues globally, disproportionately affecting women in developing countries such as Mexico. Despite advancements in diagnosis and treatment, mortality rates remain high, largely due to delayed detection and the limited reach of effective screening programs. In Mexico, breast cancer has become the leading cause of cancer-related deaths among women, and its clinical presentation occurs approximately a decade earlier than in high-income countries. These circumstances highlight the urgent need for early detection tools adapted to the specific characteristics of the Mexican population.

Several models have been designed to evaluate the risk of developing breast cancer throughout a woman's life. The most widely used—such as the Breast Cancer Risk Assessment Tool (Gail Model), the International Breast Cancer Intervention Study (Tyrer-Cuzick Model), and the Breast and Ovarian Analysis of Disease Incidence and Carrier Estimation Algorithm (BOADICEA)—rely on biological and reproductive risk data to provide estimates that guide physician decisions regarding future screening, preventive interventions, or chemoprophylaxis [35,36]. While these tools have proven their efficacy and are widely used in developed countries [37], they rely primarily on biological metrics, failing to account for the complex sociocultural or psychological dimensions that are critical in other demographic groups.

Currently, Mexico lacks risk-assessment tools calibrated to the national population that consider lifestyle or psychological factors. Consequently, these standardized models may lack predictive power when applied to Mexican women, who experience different environmental stressors, cultural pressures, and emotional coping strategies than the populations for which the models were originally developed. Our baseline results support this limitation: the clinical-only models failed to distinguish accurately between benign pathology and cancer, suggesting that biological risk factors overlap significantly between these groups. It was only the addition of psychological profiling that successfully "unlocked" the models' discriminative power.

4.2. Model Performance and the "Black Box" Problem

Our results demonstrate that integrating psychological variables—specifically emotional repression, suppression, and stress symptoms—with standard clinical history significantly enhances breast cancer risk prediction. The classification models achieved their greatest accuracy when utilizing the combined projections of clinical and psychological variables optimized through the $DE - LDA_{FE}$ method (Precision: 0.8874–0.9975; F1-score: 0.8853–0.9976), far outperforming the individual, not projected datasets.

Across the evaluated algorithms, the ANN clearly stood out as the most accurate model, reaching its peak when clinical and psychological variables were combined. However, while the ANN achieved the highest accuracy metrics, its complex architecture renders it a "black box." This limits the interpretability of the results, which is a critical issue in the medical field where understanding how a given prediction is reached is essential for clinical trust and patient communication. This issue of explainability has been highlighted in the recent literature [38], emphasizing the need for using Explainable Artificial Intelligence (XAI) in healthcare. Consequently, as our work points out, the DT model, which ranked second in overall performance, appears to be a more suitable and practical alternative for

https://doi.org/10.3390/mca31030066

medical applications. It allows clinicians to visually track the classification process and transparently explain the specific criteria used to arrive at a risk assessment.

4.3. Psychological Variables and Physiological Pathways

The empirical evidence generated by our models supports the hypothesis that psychological factors are not merely incidental comorbidities, but rather associated risk factors in the multidimensional etiology of breast cancer. The predictive value of psychological variables aligns closely with Type C Personality Theory [15–17]. Our data suggest that high levels of emotional suppression (measured by the CECS) and defensiveness (measured by the WAI) could be related as distinctive markers in the breast cancer group.

Physiologically, the relationship of these traits within the multicausal etiology of the disease could be explained by established psychoneuroimmunological mechanisms. That is, chronic emotional inhibition is known to dysregulate the hypothalamic–pituitary–adrenal axis, altering cortisol secretion patterns. This dysregulation could, in turn, affect immune surveillance and tumor suppression mechanisms [11,13,14]. Therefore, the greater accuracy achieved by combining psychological and clinical data indicates that emotional suppression and distress symptoms are not only consequences of the disease but could also serve as early indicators of physiological vulnerability.

4.4. Clinical Implications and Future Applications

The proposed $DE - LDA_{FE}$ algorithmic pipeline offers a scalable, non-invasive alternative for early screening. Unlike standard mammography, which is resource-intensive and carries radiation risks, this methodology could be deployed as a web-based or mobile pre-screening module. Such a tool would be particularly valuable in low-resource settings, offering the following systemic benefits:

(a) Empowering Patients: Providing women with access to validated self-assessment tools in contexts where direct medical services may be limited or geographically inaccessible.

(b) Optimizing Triage: enabling primary care providers to prioritize referrals for women who exhibit a high-risk biopsychosocial profile, even if they are young or asymptomatic.

This module could be readily adapted for broader use across Latin America, significantly enhancing early detection frameworks in resource-constrained areas.

4.5. Limitations/Future Work

While the results are very promising, this study has limitations that should guide future research. First, the sample size (n = 110), although statistically sufficient for this exploratory validation, requires further expansion. Future studies should incorporate a larger multicenter cohort to ensure the model's generalizability across the diverse geographic and socioeconomic regions of Mexico. They should also incorporate techniques for handling missing data and unbalanced cohorts. Although the proposed algorithm successfully separates risk groups, the cross-sectional nature of the data limits definitive causal inference. Longitudinal studies are needed to confirm whether these specific psychological traits precede cellular malignancy or evolve concurrently with it. Reference methods such as entropy-regulated two-view NMF and semi-supervised adaptive symmetric NMF could also be applied, as they offer comparative frameworks and could reveal how the observed benefits behave under more diverse age distributions, patterns of missing data, or larger-scale multicenter cohorts. In-depth sensitivity and robustness analyses should also be performed on the experiments.

5. Conclusions

This study provides strong empirical support of the biopsychosocial model in oncology. It demonstrates that combining engineered clinical and psychological features via the

https://doi.org/10.3390/mca31030066

$DE - LDA_{FE}$ projection method significantly enhances the discriminative power of machine learning classifiers. Ultimately, this research illustrates that Artificial Intelligence can effectively bridge the gap between psychology and biological oncology, offering a powerful, context-specific, and non-invasive tool to address the rising burden of BC in Mexico.

Author Contributions: Conceptualization, T.R.-G., H.G.A.-M. and Á.J.S.-G.; methodology, A.L.L.-L. and J.L.L.-R.; software, A.L.L.-L. and J.L.L.-R.; validation, H.G.A.-M. and A.L.L.-L.; formal analysis, A.L.L.-L. and J.L.L.-R.; investigation, T.R.-G., A.B.-E., H.G.A.-M. and Á.J.S.-G.; resources, T.R.-G., A.B.-E. and G.G.-O.; data curation, T.R.-G. and A.B.-E.; writing—original draft preparation, J.L.L.-R. and T.R.-G.; writing—review and editing, A.B.-E., G.G.-O., H.G.A.-M., A.L.L.-L., Á.J.S.-G. and J.C.P.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: The study was conducted following the Declaration of Helsinki at the Hospital General de Mexico "Dr. Eduardo Liceaga" and the Instituto de Investigaciones Biomédicas (Universidad Nacional Autónoma de México). All protocols were revised and approved by the Ethics and Research Committees of the Hospital General de México "Dr. Eduardo Liceaga" (Approval No: DI/12/111/03/064).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study. Copies of the written consent forms are on file and can be made available upon reasonable request.

Data Availability Statement: All data are available at: https://github.com/jllagunoroque/Risk_Predicction_via_DE-LDA_FE (accessed on 10 April 2026).

Acknowledgments: We are grateful to all the patients who agreed to participate in the study. We also want to acknowledge Gabriela Baltazar Rosario, Carlos Lara, and the administrative staff of the Hospital General de México "Eduardo Liceaga" for their assistance in recruiting female patients. Claudia Vázquez proofread the manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. Epidemiological profile of Breast Cancer in Mexico (2022). Data presents the distribution of incidence (new cases) and mortality stratified by age group, along with the federal entities reporting the highest rates.

Variable	Incidence (New Cases)	Mortality (Deaths)
Reported Total	31,043	8195
By Age Group		
20–44 years	7839 (23.0%)	1162 (13.2%)
40–59 years	14,846 (43.5%)	3555 (40.5%)
>60 years	11,430 (33.5%)	4061 (46.3%)
States with Highest Rates	1. Colima	1. Mexico City
(Top 3)	2. Durango	2. Nuevo León
	3. Jalisco	3. Chihuahua

https://doi.org/10.3390/mca31030066

Table A2. Clinical and social characteristics of the participants.

	BC (*n* = 50) **%**	**BBP (*n* = 50)** **%**	**H (*n* = 50)** **%**
Age group			
16–19	0	6	0
20–29	6	14	34
30–39	14	28	14
40–49	30	26	28
50–59	26	24	18
60–69	10	0	6
>70	14	2	0
Marital status			
Never married	16	36	58
Married	44	34	32
Free union	18	16	4
Divorced or Separated	6	8	6
Widowed	16	6	0
Years of education			
0	6	0	0
1–6	52	26	10
7–12	30	46	16
13–18	12	14	50
>19	0	14	24
Family history of cancer			
Yes	40	56	46
No	60	44	54
Menarche			
Early 8–10 years	1	2	2
Normal 11–12 years	23	27	28
Late $\geq$ 13 years	26	21	20
Menopause			
Yes	24	7	10
No	26	43	40
Childbirths			
Yes	37	23	24
No	13	27	26
Hormone replacement therapy			
Yes	25	10	4
No	25	40	46
IMC			
Normal 18.5–24.9	12	15	22
Overweight 25.0–29.9	26	22	13
Obesity $\geq$ 30	12	12	15

https://doi.org/10.3390/mca31030066

Table A3. Classification criteria for clinical and psychological study variables.

Clinical Variable	**Criterion**	
Age	40–69 years = 1	Under 40 or over 70 = 0
Family history of cancer (FamHis)	Family members with history of cancer = 1	Family members with no history of cancer = 0
Overweight (OW)	BMI between 25 and 29.9 = 1	Higher or lower than that = 0
Obesity (Obe)	BMI between 30 and 39.9 = 1	Higher or lower than that = 0
Smoking (Taba)	Addiction present = 1	No addiction = 0
Alcoholism (Alco)	Addiction present = 1	No addiction = 0
Drug use (Toxi)	Addiction present = 1	No addiction = 0
Early menarche (EarlyMen)	Younger than 12 years = 1	13 years or older = 0
Late menopause (LateMenop)	50 years or older = 1	49 years or younger = 0
Pregnancies (Preg)	No = 0	Yes = 1
Vaginal delivery (Da)	Presence = 1	Absence = 0
Abortion (Abo)	Presence = 1	Absence = 0
Cesarean section (Cesa)	Presence = 1	Absence = 0
Age at first delivery (AFD)	Younger than 30 =0	31 years or older = 1
Use of exogenous hormones (EH)	Yes = 1	No = 0
Breastfeeding (Lac)	Did not breastfeed = 1	Did breastfeed = 0
Hormone replacement therapy (HRT)	Uses HRT = 1	Does not use HRT = 0
CECS Evaluation Criteria		
Variable		Criterion
Low Anger (AL)	0–15 = 1	Greater than 15 = 0
Middle Anger (AM)	16–18 = 1	Below 16 or above 18 = 0
High Anger (AH)	Greater than or equal to 19 = 1	Less than 19 = 0
Low Depression (DL)	0–15 = 1	Greater than 15 = 0
Middle Depression (DM)	16–18 = 1	Below 16 or above 18 = 0
High Depression (DH)	Greater than or equal to 19 = 1	Less than 19 = 0
Low Anxiety (ANL)	0–15 = 1	Greater than 15 = 0
Middle Anxiety (ANM)	16–18 = 1	Below 16 or above 18 = 0
High Anxiety (ANH)	Greater than or equal to 19 = 1	Less than 19 = 0
Low CECS (SL)	0–50 = 0	Greater than 50 = 0
Middle CECS (SM)	51–55 = 1	Below 51 or above 55 = 0
High CECS (SH)	Greater than or equal to 56= 0	Less than 56 = 0
WAI Evaluation Criteria		
Variable		Criterion
DSL (Low)	Less than 47 = 1	Above 47 = 0
DSH (High)	Less than 47 = 0	Above 47 = 1
RL (Low)	Less than or equal to 94 = 1	Above 94 = 0
RM (Middle)	95–107 = 1	Below 95 or above 107 = 0
RH (High)	Greater than or equal to 108 = 1	Less than 108 = 0
RDL (Low)	Less than 58 = 1	Greater than 58 = 0
RDH (High)	Less than 58 = 0	Greater than 58 = 1
ISE Evaluation Criteria		
Variable		Criterion
Low Physical (SFL)	0–5 = 1	Greater than 5 = 0
Middle Physical (SFM)	6–8 = 1	Below 6 or above 8 = 0
High Physical (SFH)	Greater than or equal to 9 =1	Less than 9 = 0
Low Psychological (SPL)	0–11 = 1	Greater than 11 = 0
Middle Psychological (SPM)	12–19 = 1	Below 12 or above 19 = 0
High Psychological (SPH)	Greater than or equal to 20 = 1	Less than 20 = 0

https://doi.org/10.3390/mca31030066

Table A3. *Cont.*

Clinical Variable	Criterion	
Low Social (SSL)	0–5 = 1	Greater than 5 = 0
Middle Social (SSM)	6–8 = 1	Below 6 or above 8 = 0
High Social (SSH)	Greater than or equal to 9 = 1	Less than 9 = 0
Low Global (SGL)	0–22 = 1	Less than 22 = 0
Middle Global (SGM)	23–32 = 1	Below 23 or above 32 = 0
High Global (SGH)	Greater than or equal to 33 = 1	Less than 33 = 0

Appendix B. Isolated Variables of Each Data Type That May Contribute to Predicting the Risk of Developing Breast Cancer

Since the $DE - LDA_{FE}$ method is based on a linear projection, the magnitude of each coefficient directly reflects the contribution of the corresponding variable to class discrimination, allowing variables to be ordered in descending relevance. As shown in the following tables, the ten most relevant variables overall were identified, along with separate rankings for the five most relevant clinical and psychological variables.

In general, the 10 most relevant variables are:

EPP	Toxi	gesta	TRH	DH	SGL	ANL	RDL	AH	Abo

The 5 most relevant clinical variables are:

EPP	Toxi	gesta	TRH	Abo

The 5 most relevant psychological variables are:

DH	SGL	ANL	RDL	AH

Table A4. Clinical and psychological data.

Variable	Coefficient
EPP	0.17075
Toxi	0.14542
gesta	0.13314
TR	0.12534
DH	0.098024
SG	0.092286
ANL	0.072743
RDL	0.066021
AH	0.065688
abo	0.062628

Table A5. Clinical Data.

Variable	Coefficient
EPP	0.17075
Toxi	0.14542
gesta	0.13314
TRH	0.12534
abo	0.062628

https://doi.org/10.3390/mca31030066

Table A6. Psychological data.

Variable	Coefficient
DH	0.098024
SGL	0.092286
ANL	0.072743
RDL	0.066021
AH	0.065688

Among the clinical factors, EPP, Toxi, gesta, TRH, and abortion history showed the highest discriminative contribution, while DH, SGL, ANL, RDL, and AH were the most influential psychological variables. These variables consistently appeared among the highest-ranked coefficients across repeated runs, indicating stable relevance in the classification process. This ranked analysis improves interpretability by explicitly linking the predictive model to the factors that contribute most strongly to breast cancer risk discrimination.

Appendix C

Table A7 shows the descriptive statistics for the quantitative psychological variables of the risk prediction instruments (CECS/WAI/ISE). These variables are compared with the five most relevant binarized psychological variables obtained using DE-LDA_FE (DH, SGL, ANL, RDL, AH), which contribute to class discrimination, as described in Appendix B. This allows us to verify whether there was a loss or gain of information when performing the binarization. In this regard, several of the significant variables in Table A7, such as A and An, were also presented using $DE-LDA_{FE}$. While the $DE-LDA_{FE}$ model highlighted the importance of Global (SG) and Restraint, even though these quantitative variables were not significant in their descriptive statistics, the results did not coincide only in the case of Distress and Psychological (SP).

Table A7. Descriptive statistics of the CECS/WAI/ISE quantitative variables by pathological group.

Variable	Breast Cancer Median (IQR)	Benign Breast Pathology Median (IQR)	Healthy Median (IQR)	p [a]
Anger (A)	18 (7)	16 (5)	14.5 (5.75)	0.008 *
Depression (D)	17 (7)	19 (7)	16.5 (6.75)	0.069
Anxiety (An)	18 (6)	17 (6)	15.5 (6.25)	0.017 *
Global CECS	52 (12)	52 (16)	47 (17.25)	0.022
Distress	50.26 (14)	44.05 (16.16)	33.08 (20.08)	<0.001 *
Restraint	103.15 (19.02)	105.04 (14.04)	102.61 (15.1)	0.769
Defensive Attitude	55.85 (10.98)	56.35 (11.41)	58.7 (8.55)	0.938
Physics (SF)	5 (6)	6 (6)	8 (6)	0.104
Psychological (SP)	10 (10)	16 (12)	16 (9)	0.047 *
Social (SS)	4 (8)	6 (9)	5.5 (9.75)	0.375
Global (SG)	20 (28)	28 (27)	30 (18.75)	0.148

Note: [a] significance of the Kruskal–Wallis Test; * significant variables ($p < 0.05$); significant variables obtained using the $DE-LDA_{FE}$ model are presented in bold. IQR stands for Interquartile Range.

References

1. World Health Organization. *Data Visualization Tools for Exploring the Global Cancer Burden in 2022*; Cancer Today; World Health Organization: Geneva, Switzerland, 2024.
2. Maffuz-Aziz, A.; Labastida-Almendaro, S.; Espejo-Fonseca, A.; Rodríguez-Cuevas, S. Características clinicopatológicas del cáncer de mama en una población de mujeres en México. *Cirugía Cir.* **2017**, *85*, 201–207. [CrossRef]

https://doi.org/10.3390/mca31030066

3. Cárdenas-Sánchez, J. Consenso mexicano sobre diagnóstico y tratamiento del cáncer mamario. *Mex. J. Oncol.* **2022**, *20*, 6923. [CrossRef]
4. Díaz Godinez, M.M. *Estadísticas a Propósito del Día Internacional de la Lucha Contra el Cáncer de Mama (19 de Octubre)*; INEGI: Aguascalientes, Mexico, 2023.
5. Seimenis, I.; Chouchos, K.; Prassopoulos, P. Radiation Risk Associated with X-Ray Mammography Screening: Communication and Exchange of Information via Tweets. *J. Am. Coll. Radiol.* **2018**, *15*, 1033–1039. [CrossRef] [PubMed]
6. Di Maria, S.; Van Nijnatten, T.J.A.; Jeukens, C.R.L.P.N.; Vedantham, S.; Dietzel, M.; Vaz, P. Understanding the Risk of Ionizing Radiation in Breast Imaging: Concepts and Quantities, Clinical Importance, and Future Directions. *Eur. J. Radiol.* **2024**, *181*, 111784. [CrossRef]
7. Chaudhary, L.; Knapp, S.; Wen, S.; Xiao, J.; Marano, G.; Kurian, S.; Layne, G.; Jacobson, G.; Abraham, J. Radiation Exposure from Diagnostic Procedures in Patients with Newly Diagnosed Breast Cancer. *J. Community Support. Oncol.* **2015**, *13*, 27–29. [CrossRef]
8. Chávarri-Guerra, Y.; Villarreal-Garza, C.; Liedke, P.E.; Knaul, F.; Mohar, A.; Finkelstein, D.M.; Goss, P.E. Breast Cancer in Mexico: A Growing Challenge to Health and the Health System. *Lancet Oncol.* **2012**, *13*, e335–e343. [CrossRef] [PubMed]
9. Villarreal-Garza, C.; Platas, A.; Miaja, M.; Fonseca, A.; Mesa-Chavez, F.; Garcia-Garcia, M.; Chapman, J.-A.; Lopez-Martinez, E.A.; Pineda, C.; Mohar, A.; et al. Young Women with Breast Cancer in Mexico: Results of the Pilot Phase of the Joven & Fuerte Prospective Cohort. *JCO Glob. Oncol.* **2020**, *6*, 395–406. [CrossRef] [PubMed]
10. Avila, M.H. *NORMA Oficial Mexicana NOM-041-SSA2-2011, Para la Prevención, Diagnóstico, Tratamiento, Control y Vigilancia Epidemiológica del Cáncer de Mama*; Diario Oficial de la Federación: CDMX, México, 2011.
11. Romo-González, T.; Martínez, A.J.; Hernández-Pozo, M.D.R.; Gutiérrez-Ospina, G.; Larralde, C. Psychological Features of Breast Cancer in Mexican Women I: Personality Traits and Stress Symptoms. *Adv. Neuroimmune Biol.* **2018**, *7*, 3–15. [CrossRef]
12. Montes-Nogueira, I.; Campos-Uscanga, Y.; Gutiérrez-Ospina, G.; Hernández-Pozo, M.D.R.; Larralde, C.; Romo-González, T. Psychological Features of Breast Cancer in Mexican Women II: The Psychological Network. *Adv. Neuroimmune Biol.* **2018**, *7*, 91–105. [CrossRef]
13. Giese-Davis, J.; Wilhelm, F.H.; Conrad, A.; Abercrombie, H.C.; Sephton, S.; Yutsis, M.; Neri, E.; Taylor, C.B.; Kraemer, H.C.; Spiegel, D. Depression and Stress Reactivity in Metastatic Breast Cancer. *Psychosom. Med.* **2006**, *68*, 675–683. [CrossRef]
14. Giese-Davis, J.; Koopman, C.; Butler, L.D.; Classen, C.; Cordova, M.; Fobair, P.; Benson, J.; Kraemer, H.C.; Spiegel, D. Change in Emotion-Regulation Strategy for Women with Metastatic Breast Cancer Following Supportive-Expressive Group Therapy. *J. Consult. Clin. Psychol.* **2002**, *70*, 916–925. [CrossRef]
15. Greer, S.; Morris, T. Psychological Attributes of Women Who Develop Breast Cancer: A Controlled Study. *J. Psychosom. Res.* **1975**, *19*, 147–153. [CrossRef]
16. Temoshok, L. Personality, Coping Style, Emotion and Cancer: Towards an Integrative Model. *Cancer Surv.* **1987**, *6*, 545–567.
17. Iwamitsu, Y.; Shimoda, K.; Abe, H.; Tani, T.; Kodama, M.; Okawa, M. Differences in Emotional Distress between Breast Tumor Patients with Emotional Inhibition and Those with Emotional Expression. *Psychiatry Clin. Neurosci.* **2003**, *57*, 289–294. [CrossRef] [PubMed]
18. Myers, L.B. The Importance of the Repressive Coping Style: Findings from 30 Years of Research. *Anxiety Stress Coping* **2010**, *23*, 3–17. [CrossRef] [PubMed]
19. McGregor, B.A.; Antoni, M.H. Psychological Intervention and Health Outcomes among Women Treated for Breast Cancer: A Review of Stress Pathways and Biological Mediators. *Brain Behav. Immun.* **2009**, *23*, 159–166. [CrossRef] [PubMed]
20. García Camacho, A. Módulo Web Para la Obtencion de Riesgo de Padecer Cáncer de Mama Mediante Datos Psicológicos y Clínicos. Bachelor's Thesis, Universidad Veracruzana, Xalapa, Ver., Mexico, 2017.
21. Hussain, S.; Ali, M.; Naseem, U.; Nezhadmoghadam, F.; Jatoi, M.A.; Gulliver, T.A.; Tamez-Peña, J.G. Breast Cancer Risk Prediction Using Machine Learning: A Systematic Review. *Front. Oncol.* **2024**, *14*, 1343627. [CrossRef]
22. Geißler, K.; Koller, T.L.; Ambroladze, A.; Fallenberg, E.M.; Ingrisch, M.; Hahn, H.K. Breast Cancer Risk Prediction Using Background Parenchymal Enhancement, Radiomics, and Symmetry Features on MRI. In *Proceedings of the Medical Imaging 2025: Computer-Aided Diagnosis*; SPIE: Cergy-Pontoise, France, 2025; Volume 13407, pp. 564–568.
23. Wang, J.; Zhang, Z.; Wang, Y. Utilizing Feature Selection Techniques for AI-Driven Tumor Subtype Classification: Enhancing Precision in Cancer Diagnostics. *Biomolecules* **2025**, *15*, 81. [CrossRef]
24. Carriero, A.; De Hond, A.; Cappers, B.; Paulovich, F.; Abeln, S.; Moons, K.G.; Van Smeden, M. Explainable AI in Healthcare: To Explain, to Predict, or to Describe? *Diagn. Progn. Res.* **2025**, *9*, 29. [CrossRef]
25. Berahmand, K.; Zhou, X.; Li, Y.; Gururajan, R.; Barua, P.D.; Acharya, U.R.; Chennakesavan, S.K. NEDL-GCP: A Nested Ensemble Deep Learning Model for Gynecological Cancer Risk Prediction. *Array* **2025**, *27*, 100468. [CrossRef]
26. Doumari, S.A.; Berahmand, K.; Ebadi, M.J. Early and High-Accuracy Diagnosis of Parkinson's Disease: Outcomes of a New Model. *Comput. Math. Methods Med.* **2023**, *2023*, 1493676. [CrossRef]

https://doi.org/10.3390/mca31030066

27. Romo-González, T.; Barranca-Enríquez, A.; León-Díaz, R.; Del Callejo-Canal, E.; Gutiérrez-Ospina, G.; Jimenez Urrego, A.M.; Bolaños, C.; Botero Carvajal, A. Psychological Suppressive Profile and Autoantibodies Variability in Women Living with Breast Cancer: A Prospective Cross-Sectional Study. *Heliyon* **2022**, *8*, e10883. [CrossRef]
28. Benevides-Pereira, A.M.T.; Moreno-Jiménez, B.; Garrosa Hernández, E.; González Gutiérrez, J.L. La evaluación específica del síndrome de burnout en psicólogos: El Inventario de burnout de psicólogos. *Clínica Salud* **2002**, *13*, 257–283.
29. Watson, M.; Greer, S. Development of a Questionnaire Measure of Emotional Control. *J. Psychosom. Res.* **1983**, *27*, 299–305. [CrossRef] [PubMed]
30. Durá, E.; Andreu, Y.; Galdón, M.J.; Ibáñez, E.; Pérez, S.; Ferrando, M.; Murgui, S.; Martínez, P. Emotional Suppression and Breast Cancer: Validation Research on the Spanish Adaptation of the Courtauld Emotional Control Scale (CECS). *Span. J. Psychol.* **2010**, *13*, 406–417. [CrossRef]
31. Weinberger, D.A. The Construct Validity of the Repressive Coping Style. In *Repression and Dissociation: Implications for Personality Theory, Psychopathology and Health*; Singer, J.L., Ed.; University of Chicago Press: Chicago, IL, USA, 1990; pp. 337–386.
32. Weinberger, D.A.; Schwartz, G.E. Distress and Restraint as Superordinate Dimensions of Self-Reported Adjustment: A Typological Perspective. *J. Personal.* **1990**, *58*, 381–417. [CrossRef]
33. Weinberger, D.A. Distress and Self-Restraint as Measures of Adjustment across the Life Span: Confirmatory Factor Analyses in Clinical and Nonclinical Samples. *Psychol. Assess.* **1997**, *9*, 132–135. [CrossRef]
34. Romo González, T.; Enríquez-Hernández, C.B.; Hernández Pozo, M.D.R.; Ruiz Montalvo, M.E.; Castillo, R.L.; Ehrenzweig Sánchez, Y.; Marván, M.L.; Larralde, C. Validación en México del inventario de ajuste de Weinberger (WAI). *Salud Ment.* **2014**, *37*, 247. [CrossRef]
35. Amir, N.; Beard, C.; Burns, M.; Bomyea, J. Attention Modification Program in Individuals with Generalized Anxiety Disorder. *J. Abnorm. Psychol.* **2009**, *118*, 28–33. [CrossRef]
36. Brentnall, A.R.; Harkness, E.F.; Astley, S.M.; Donnelly, L.S.; Stavrinos, P.; Sampson, S.; Fox, L.; Sergeant, J.C.; Harvie, M.N.; Wilson, M.; et al. Mammographic Density Adds Accuracy to Both the Tyrer-Cuzick and Gail Breast Cancer Risk Models in a Prospective UK Screening Cohort. *Breast Cancer Res.* **2015**, *17*, 147. [CrossRef] [PubMed]
37. Terry, M.B.; Liao, Y.; Whittemore, A.S.; Leoce, N.; Buchsbaum, R.; Zeinomar, N.; Dite, G.S.; Chung, W.K.; Knight, J.A.; Southey, M.C.; et al. 10-Year Performance of Four Models of Breast Cancer Risk: A Validation Study. *Lancet Oncol.* **2019**, *20*, 504–517. [CrossRef] [PubMed]
38. Hassija, V.; Chamola, V.; Mahapatra, A.; Singal, A.; Goel, D.; Huang, K.; Scardapane, S.; Spinelli, I.; Mahmud, M.; Hussain, A. Interpreting Black-Box Models: A Review on Explainable Artificial Intelligence. *Cogn. Comput.* **2024**, *16*, 45–74. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

https://doi.org/10.3390/mca31030066

Mathematical and Computational Applications

MDPI

Article

A Comparison Between Heuristic and Automatic Design in Variational Quantum Circuits for the MaxCut Problem Under Noise Effects

Emmanuel Isaac Juárez Caballero *, Horacio Tapia-McClung and Efrén Mezura-Montes

Instituto de Investigaciones en Inteligencia Artificial, Universidad Veracruzana, Xalapa 91097, Veracruz, Mexico; htapia@uv.mx (H.T.-M.); emezura@uv.mx (E.M.-M.)
* Correspondence: emmjuarez@uv.mx

Abstract

The selection of the right topology (ansatz) for a Variational Quantum Algorithm (VQA) is a complex task that usually involves deep knowledge of a particular problem. The importance of the selection is greater when we consider the current state of quantum hardware, particularly the noise associated with the complexity of Variational Quantum Circuits (VQCs) that implement VQAs. Here, a comparison is presented between two confronted approaches for solving the MaxCut problem: QAOA, which has a theoretical proof of convergence, and the automatic design proposal (QNAS), which relies on evolutionary algorithms (NSGA-II) to discover efficient circuits. The comparison was made across 490 graph instances from different graph topologies and sizes ($n = 4$ to $n = 16$), accounting for noise models such as depolarizing noise, gate errors, and readout noise. The results show that QAOA achieves an approximation ratio (r_A) ≈ 1 on complete graphs at the cost of being almost 12 times more complex than QNAS in ideal conditions while approaching the random noise floor ($r_A \approx 0.5$). QNAS was capable of finding circuits less complex while maintaining 69% of the fidelity at a cost of having an r_A on the interval $0.7 \leq r_A \leq 0.8$. However, when the comparison is made across sparse graphs, performance is comparable, while QNAS is less complex.

Keywords: quantum neural architecture search; variational quantum algorithms; QAOA; NISQ; evolutionary algorithms; multi-objective optimization; quantum noise modeling

1. Introduction

The current state of Quantum Computing (QC) is mainly limited by the state-of-the-art Noisy Intermediate-Scale Quantum (NISQ) [1,2] devices which consist of quantum computers primarily based on ultra-cold atoms and superconducting qubits that can only offer processors with 50–1000 qubits that suffer from real world limitations mainly due to physical processes such as decoherence, a process that induces noise that destroys the superposition of a quantum state and consequently alters the measurement of the result of VQAs. These QCs have short coherence times, which translate into a small operational window, limited connectivity between qubits, and gate error rates ranging from 10^{-3} to 10^{-2}. This is a clear difference with classical computers, where errors may seem, in essence, non-existent, while QCs accumulate noise that affects the measurement of information.

Due to the presence of this inherent noise that is caused by experimental limitations, several approaches are being explored to take value from the QC, in particular, the use of Variational Quantum Eigensolvers (VQEs) [3].

The Quantum Approximate Optimization Algorithm (QAOA) is a VQA focused on combinatorial optimization. This kind of VQA uses a classical computer to repeatedly adjust the parameters of a Variational Quantum Circuit (VQC) called an ansatz to minimize a specified cost function that depends on the problem. Finally, the quantum computer (or the simulation) measures the results to evaluate the cost function.

An important point is that the process relies on the selection of the ansatz, which depends entirely on the choice of architecture. Some of the problems that may arise from choosing the wrong ansatz are as follows:

- The VQC cannot reach the part of the optimization landscape where the solution lives. Recent work has formalized this through measures like Kullback–Leibler (KL) divergence from the Haar measure [4].
- Barren Plateaus: In this case, the optimization landscape becomes flat as the system grows, making gradient-based methods useless in the search for the parameters in the VQC [5,6].
- Decoherence occurs before the algorithm finishes, making the solution provided by the VQC useless due to the random behavior [7].

Our proposal (See Figure 1) is based on a direct comparison between the previously mentioned QAOA and the Quantum Neural Architecture Search (QNAS), which relies on evolutionary computing strategies to discover solutions to a specific problem. Several works, such as those presented in [8–10], employ different evolutionary techniques, as well as other approaches, such as reinforcement learning [3,11], for the automatic design of ansatzes.

However, a direct comparison of the functioning of heuristic approaches, such as QAOA, and automatic design strategies, such as QNAS, remains unexplored for well-defined problems, such as the MaxCut problem presented here.

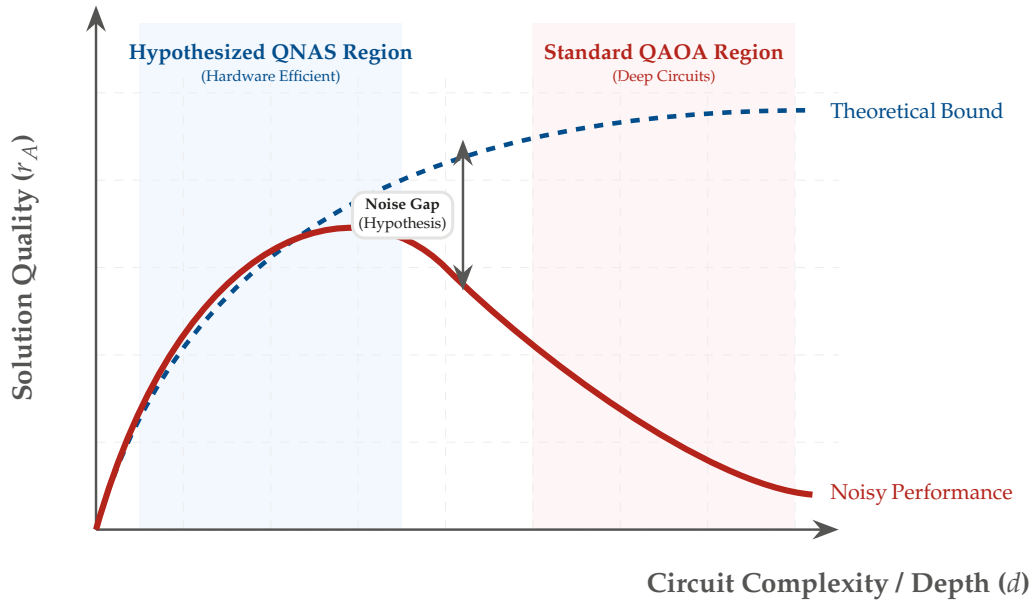


Figure 1. Hypothesized performance trade-off in NISQ devices. The blue region corresponds to the regime targeted by QNAS. The red region indicates the standard QAOA regime.

https://doi.org/10.3390/mca31030078

An important remark is that, even though this study is related to the MaxCut problem, it can itself be viewed as part of a more general approach, since QNAS is well known for creating the search for potential Quantum Neural Network (QNN) architectures. A QNN can be represented as a VQC, which is itself a VQA.

This paper is organized as follows. Section 2 presents the theoretical framework, including the MaxCut problem formulation, the QAOA algorithm and its complexity analysis, and the QNAS approach based on NSGA-II. Section 3 details the experimental methodology, including the deployment environment, graph dataset construction, noise model specification, and performance metrics. Section 4 reports the results across ideal and noisy conditions, stratified by graph topology. Finally, Section 5 summarizes the findings and provides directions for future work.

2. Theoretical Background

2.1. Formulation of MaxCut Problem

The formal definition of the MaxCut problem [12] can be summarized by taking into account an undirected graph $G = (V, E)$ with a set of vertices V and edges E. The objective is to find a partition of V into two disjoint sets, S and $S^c = V \setminus S$, that maximizes the number of edges between them. The cut value is defined as follows:

$$\text{Cut}(S) = |\{(i, j) \in E : i \in S, j \in S^c\}| \tag{1}$$

This problem can be mapped to find the ground-state in an Ising Hamiltonian (see Figure 2), since the two are isomorphic. We assign a spin variable $s_i \in \{+1, -1\}$ to each vertex i, where $s_i = +1$ indicates $i \in S$ and $s_i = -1$ indicates $i \in S^c$. The cost function becomes:

$$C(s) = \frac{1}{2} \sum_{(i,j)\in E} (1 - s_i s_j) \tag{2}$$

This formulation relies on the fact that $s_i s_j = -1$ when vertices i and j are in different partitions, and $s_i s_j = +1$ when they are in the same partition. By mapping $s_i \to \sigma_i^z$, where σ_z is the Pauli *z–matrix* , we obtain the Hamiltonian H_C:

$$H_C = \frac{1}{2} \sum_{(i,j)\in E} (I - \sigma_i^z \sigma_j^z) \tag{3}$$

The ground state of $-H_C$ corresponds to the maximum cut. The measurement outcome $z \in \{0, 1\}^n$ in the computational basis is interpreted as a bit-string, where qubit i in state $|0\rangle$ corresponds to $i \in S$ and $|1\rangle$ to $i \in S^c$.

An auxiliary metric called approximation ratio can be defined as:

$$r_{\mathcal{A}} = \frac{C(z_{\text{measured}})}{C^*} \tag{4}$$

where C^* is the value of the optimal cut, and r_A is a standard metric used to evaluate the quality of a solution provided by an optimization algorithm relative to the best possible (optimal) solution.

2.2. Quantum Approximate Optimization Algorithm (QAOA)

QAOA is a heuristic algorithm inspired by the adiabatic theorem [13] that, as stated previously, was handcrafted to provide solutions to combinatorial optimization problems [14]. This solution method consists of two elementary components of an ansatz, a cost Hamiltonian $U_C(\gamma_k) = e^{-i\gamma_k H_c}$ and a mixer operator $U_M(\beta_k) = e^{-i\beta_k H_M}$. The cost part of the Hamiltonian ensures that the max-cut is evaluated; the mixer part, instead, ensures that the

https://doi.org/10.3390/mca31030078

state being evaluated changes and that the algorithm does not fall into a local minimum. The proposed solution takes the following form:

$$\left|\psi(\vec{\gamma},\vec{\beta})\right\rangle = U_M(\beta_p)U_C(\gamma_p)\cdots U_M(\beta_1)U_C(\gamma_1)|+\rangle^{\otimes n}, \tag{5}$$

where $\vec{\gamma} = (\gamma_1,\ldots,\gamma_p)$ and $\vec{\beta} = (\beta_1,\ldots,\beta_p)$ are variational parameters (rotation angles), and $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ is the uniform superposition state. The notation $|+\rangle^{\otimes n}$ represents the tensor product of n qubits, each in the $|+\rangle$ state.

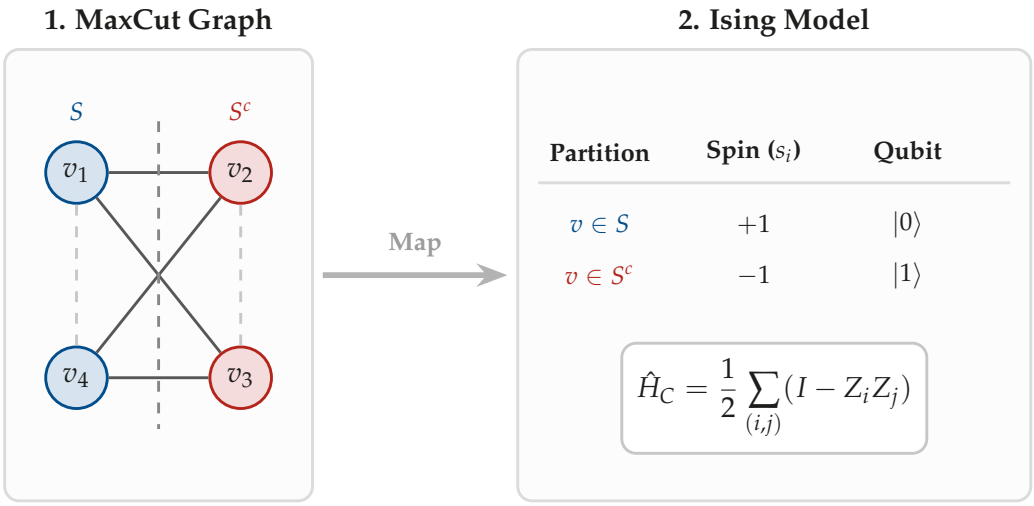


Figure 2. Visual mapping of the MaxCut problem. Vertices are partitioned into two sets (S and S^c), which correspond to spin states ± 1 and qubit basis states $|0\rangle, |1\rangle$.

The operators are applied sequentially from right to left, beginning with state preparation and then alternating p layers of cost and mixer unitaries. Each $U_C(\gamma_k)$ encodes the problem Hamiltonian (Equation (3)), while $U_M(\beta_k) = \prod_{i=1}^{n_q} e^{-i\beta_k X_i}$ implements the mixing part, where X_i is the Pauli-X operator on qubit i.

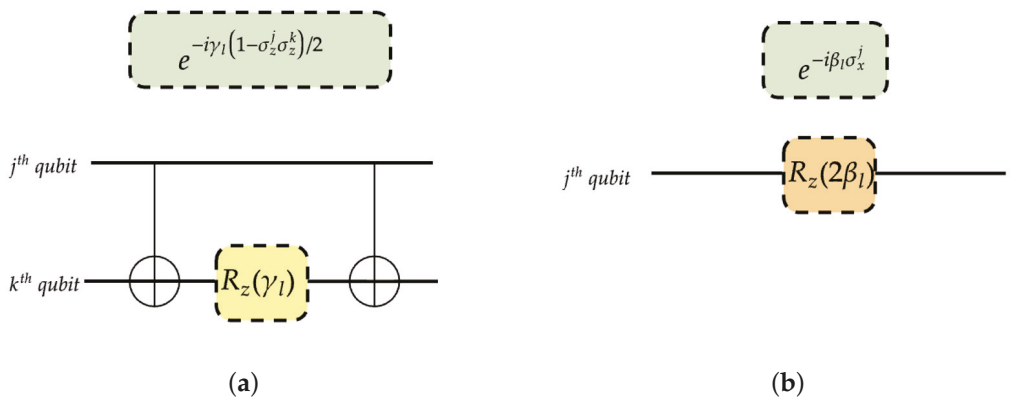
The mixer operator U_M acts by applying single-qubit Pauli-X rotations to each qubit, U_M induces transitions between computational basis states, ensuring that the algorithm can escape local minima. This property guarantees that, as the number of layers p increases, the algorithm can, in principle, approximate the global optimum. U_M also corresponds to the initial Hamiltonian $H_M = -\sum_i X_i$, whose ground state is the uniform superposition $|+\rangle^{\otimes n}$, the starting point from which the system evolves toward the ground state of the cost Hamiltonian H_C.

From Equation (5), it is possible to note that the function depends on a parameter p, which is usually referred to as the number of layers of the ansatz. However, it can be seen that all layers have a cost part and a mixer part; therefore, the number of layers is $2p$. A more visual representation of the structure of each layer in QAOA is shown in Figure 3.

Complexity Analysis of QAOA

The implementation cost of QAOA is dominated by the unitary cost U_C. Since H_C contains terms of the form Z_iZ_j, implementing U_C requires one to realize the unitary $e^{-i\gamma Z_iZ_j}$ for every edge $(i,j) \in E$. Standard decomposition requires two CNOT gates and one R_z rotation [15]:

$$e^{-i\gamma Z_iZ_j} \equiv \mathrm{CNOT}_{ij} \cdot R_z(2\gamma)_j \cdot \mathrm{CNOT}_{ij} \tag{6}$$



(**a**) (**b**)

Figure 3. The structure of the QAOA ansatz layers, generalizable to n qubits. (**a**) The cost layer $U_C(\gamma)$ implements the problem Hamiltonian via ZZ-interaction gates between connected qubits, with each edge requiring two CNOT gates and one R_z rotation. (**b**) The mixer layer $U_M(\beta)$ applies single-qubit R_x rotations parameterized by β to each qubit, implementing Pauli-X rotations.

Consequently, a single layer ($p = 1$) requires $2|E|$ CNOT gates plus n single-qubit gates for the mixer. For a complete graph K_n, where $|E| = n(n-1)/2$, the total CNOT count for depth p is

$$N_{\text{CNOT}} \approx p \cdot n(n-1) \in \Theta(pn^2) \tag{7}$$

This quadratic scaling with system size n is the primary limitation of QAOA in the NISQ era, as error rates on CNOT gates ($2q$-gates) ($\epsilon_{2q} \approx 10^{-2}$) are approximately one order of magnitude higher than single-qubit gates ($\epsilon_{1q} \approx 10^{-3}$) on current superconducting platforms.

For any graph G with determined depth p, QAOA provides a polynomial-time approximation algorithm for MaxCut. While the approximation ratio depends on p and graph topology, work by [16] proved that for $p = 1$, QAOA achieves an approximation ratio of at least $r_A = 0.6924$ for any 3-regular graph. Numerical evidence suggests that as p increases, QAOA can approach or exceed the Goemans–Williamson bound ($\alpha \approx 0.878$) for specific graph families [17,18], though this remains an open theoretical question. However, this guarantee comes with the restriction that the parameters $\vec{\gamma}, \vec{\beta}$ must be optimally chosen [19].

2.3. Quantum Neural Architecture Search (QNAS)

On the other side of our proposal, QNAS optimizes the architecture A (topology and gate selection), and it also optimizes the variational parameters (see Figure 4) θ for that specific architecture.

We can think of this as a Multi-Objective Optimization (MOO) problem:

$$\min_{A \in \mathcal{S}} \mathbf{F}(A) = [\langle \psi_A(\theta^*)|H_C|\psi_A(\theta^*)\rangle, \quad \mathcal{C}(A)] \tag{8}$$

where:

- $\mathcal{S}$ is the search space of possible circuits. In this approach, it is described as an integer list.
- $\mathcal{C}(A)$ is the complexity cost referred to the potential hardware implementation.
- θ^* denotes the parameters obtained via classical optimization (in our experimentation, we used (L-BFGS-B) [20] with 100 iterations).

https://doi.org/10.3390/mca31030078

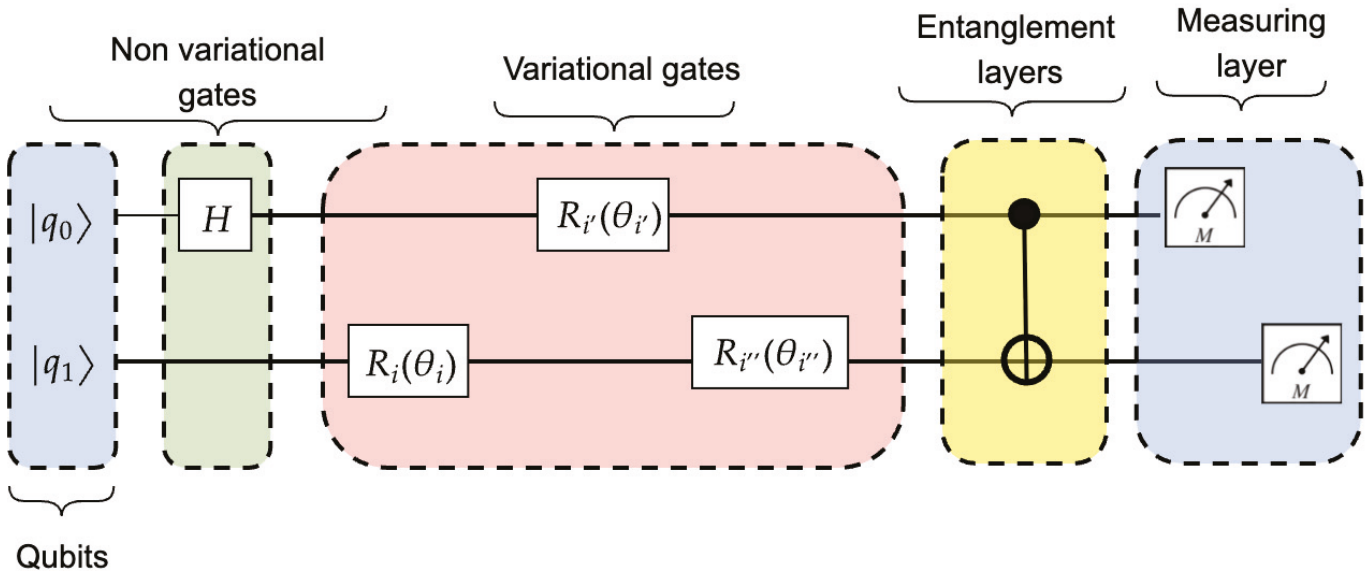


Figure 4. Typical structure of a VQC. State preparation and parameterized gates—non-variational gates initialize the qubits, followed by variational rotation gates $R_i(\theta)$, whose parameters are optimized classically. Entanglement and measurement—two-qubit entangling gates create correlations between qubits, and the measurement layer projects the quantum state onto the computational basis to evaluate the cost function.

The objectives are conflicting (see Figure 5): increasing circuit depth generally improves expressibility (reducing the first objective) but increases complexity (increasing the second objective). The solution to this MOO problem is not a single architecture but a Pareto Front, a set of architectures where no objective can be improved without degrading another.

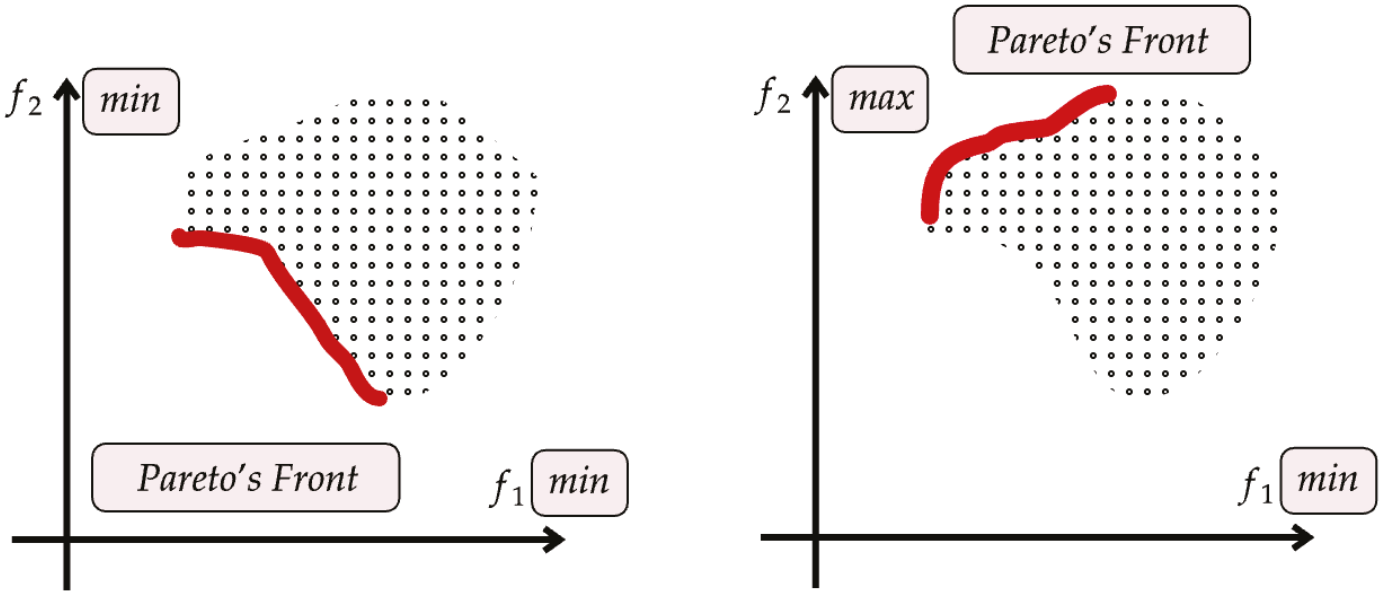


Figure 5. An illustration on conflicting objectives, a condition which is necessary to conduct a multiobjective optimization, in the proposal case $f_1 \rightarrow$ expressibility and $f_2 \rightarrow$ complexity.

Evolutionary Search via NSGA-II

We employ the NSGA-II proposed by [21] to conduct the optimization of the architectures/topologies. NSGA-II works by creating a population of candidate architectures and evolving them through variational operators (selection, crossover, and mutation).

https://doi.org/10.3390/mca31030078

The functioning of NSGA-II under our proposal is defined in Algorithm 1, and illustrated in Figure 6 as a direct extrapolation of the original [21] reference.

Algorithm 1 NSGA-II for QNAS. A direct translation of the original algorithm [21] using the notation for the examined problem.

Initialize population P_0 of N random architectures
for generation $g = 1$ to $G_{\max}$ **do**
  for each architecture $A \in P_g$ **do**
    Train parameters $\theta^*(A)$ via L-BFGS-B (inner loop)
    Evaluate objectives: $f_1(A) = \langle H_C \rangle_A$, $f_2(A) = \mathcal{C}(A)$
  end for
  Perform non-dominated sorting on P_g to assign ranks
  Calculate crowding distance for each architecture
  Generate offspring Q_g via tournament selection, crossover, mutation
  Combine $P_g \cup Q_g$ and select best N for P_{g+1}
end for
return Pareto front from final population $P_{G_{\max}}$

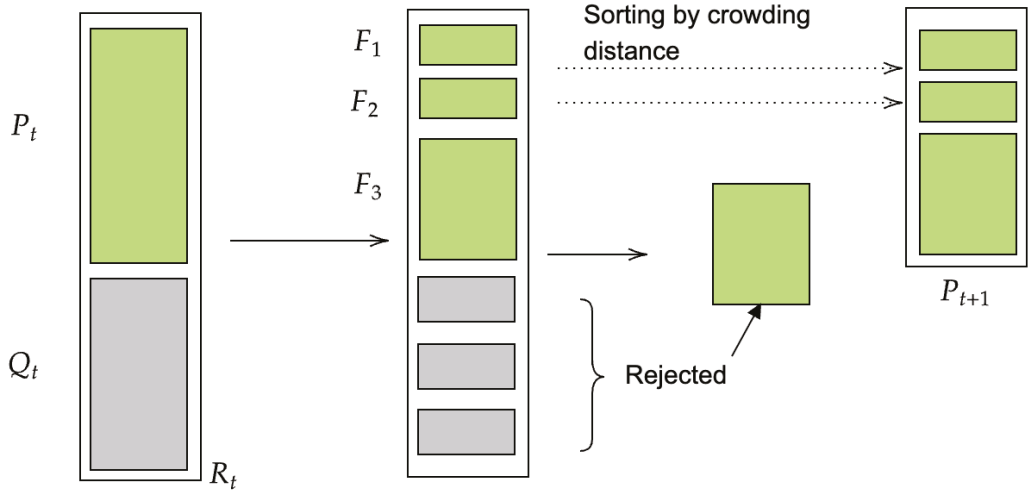


Figure 6. Graphical representation of Algorithm 1; the total population $P_t \cup Q_t$ is sorted by Pareto Front (PF) ranks, and then that set of PF is sorted by crowding distance—a metric that enhances the diversity of the selected population, ensuring that the solutions are diverse and therefore more representative of the Pareto front.

Selection is based on two criteria:

1. **Pareto front rank:** Solutions that are not dominated by any other solution receive Rank 1, and so forth. This stratification ensures convergence toward the true Pareto front.
2. **Crowding distance:** To preserve diversity, the algorithm prefers solutions that are in less crowded regions of the objective space. The crowding distance d_i for individual i is calculated as:

$$d_i = \sum_{m=1}^{M} \frac{f_m^{(i+1)} - f_m^{(i-1)}}{f_m^{\max} - f_m^{\min}} \tag{9}$$

where M is the number of objectives, and $(i-1), (i+1)$ denote the neighboring solutions in objective space sorted by f_m. This prevents convergence to a single point on the front and ensures exploration of the entire trade-off surface.

https://doi.org/10.3390/mca31030078

3. Methodology

To create a fair benchmark for the proposed approaches, we developed a framework using Qiskit, a state-of-the-art Python (3.11.14) library for conducting real and simulated quantum executions. However, it is also important to illustrate the process that is being made in the execution of a VQA in a QC where the evaluation of the algorithm is being made (see Figures 7 and 8).

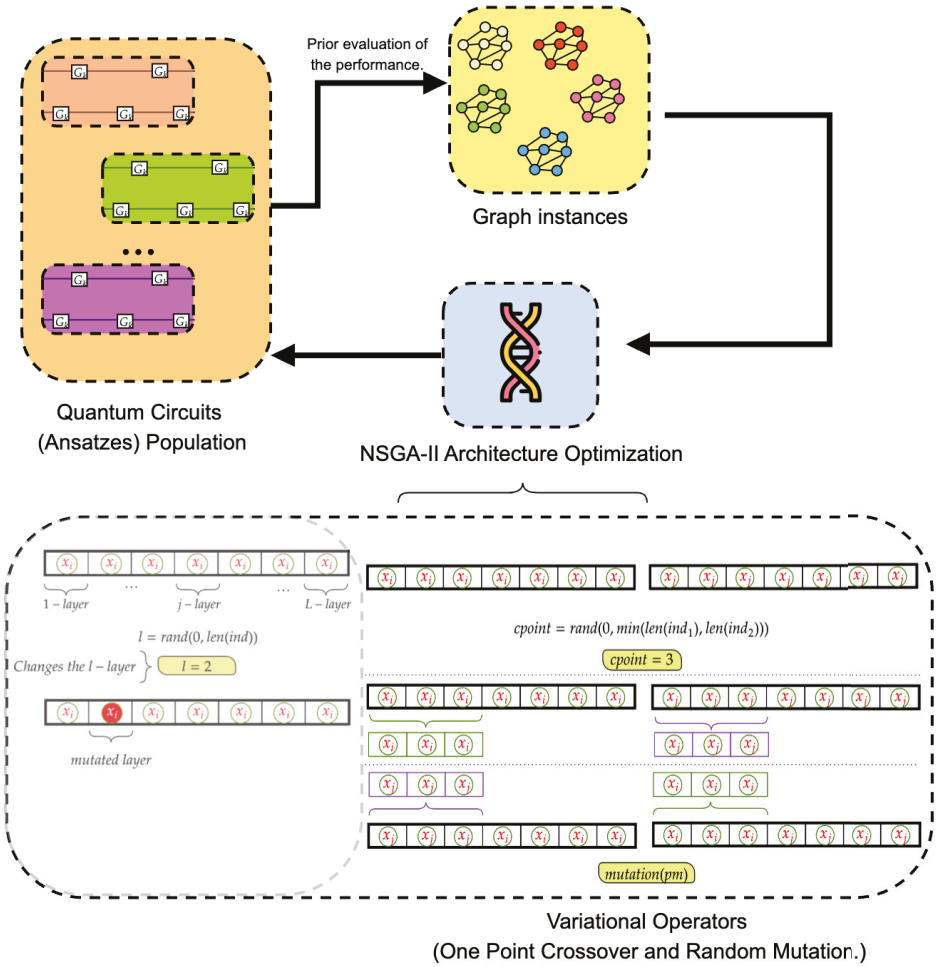


Figure 7. Complete overview of the optimization process using QNAS approach. In the evaluation of the performance of each dataset, noise was taken into account. Circuit representation used merely for visualization purposes.

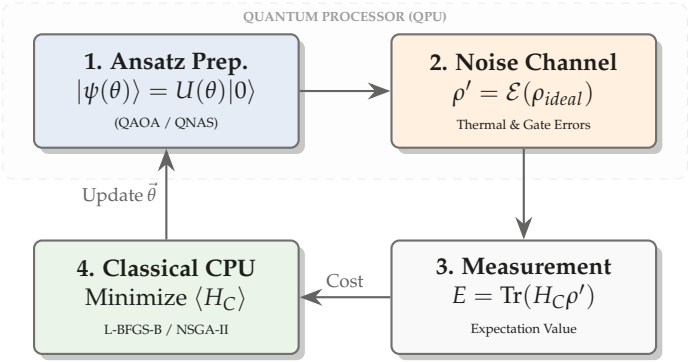


Figure 8. Structure of the execution of a VQA under noise conditions, this kind of algorithm is not purely quantum since the optimization of the variational parameters is being assisted by a classical CPU.

https://doi.org/10.3390/mca31030078

3.1. Deployment Details

All quantum circuits were constructed, transpiled, and simulated using the IBM Qiskit SDK (version 0.44.0) with `Qiskit Aer` (version 0.12.2). Noiseless expectation values were obtained from statevector simulation, while the effect of noise was incorporated through the analytical fidelity model described in Section 3.4 (Equation (13)), which combines $r_{A,\text{ideal}}$ with an estimated circuit fidelity F_{est} derived from the gate counts and per-gate error rates of Table 1.

To ensure reproducibility, we fixed the following environment parameters:

- All circuits were transpiled (an important remark is that transpilation, which is commonly known as an optimization method for the circuits to be more efficient does not interfere with the QNAS approach since transpilation relies on the logic, while QNAS relies on the architecture of the circuit) to a standard basis gate set $\{R_z, \sqrt{X}, X, \text{CNOT}\}$ using optimization level 3 (`optimization_level = 3`) to minimize swap gates excessive presence before noise injection.
- For QAOA, we employed the L-BFGS optimizer (100 iterations). For the QNAS evolutionary loop, we implemented a custom NSGA-II genetic algorithm.
- Fixed seeds were used for both graph generation (`networkx`) and quantum circuit initialization.

3.2. Structure of the Proposed Graphs for MaxCut Problem

We generated a dataset composed of 490 graphs from different distributions that may represent different physical phenomena. Our proposed graph distributions can be summarized as follows:

- **Random network graphs**: We used the Erdös–Rényi model as a baseline to tackle problems without apparent structure.
- **Complex networks**: In this case, we used Barabási–Albert and Watts–Strogatz models. The former is known for generating scale-free networks. All of them are common in real-world network analysis.
- **Regular graph distributions**: We used a cycle graph, which models only local connectivity; a grid, which simulates problems with strong locality; a complete graph, which represents the case where the graph has maximum interaction density; and a star, which is a centralized graph.

QNAS Encoding Scheme

To codificate each candidate circuit, a variable-length chromosome was proposed (Figure 9a):

$$\text{Individual} = [l_1, l_2, \ldots, l_k]$$

where each gene $l_i \in \{0, 1, \ldots, |S| - 1\}$ indexes. The search space S is composed of atomic layers that include single-qubit rotation gates $\{R_x, R_y, R_z, H\}$ and two-qubit entangling gates $\{CNOT\}$, which are applied across all feasible qubit combinations. This representation facilitates the evolution of circuit topology and depth within the NSGA-II framework.

It is important to note that the list codification depends on the initial configuration used to generate the search space; that is, for a particular quantum computer, restrictions can be added to the search space, allowing a custom design for that hardware. This is particularly relevant since one of the capabilities we need to fight back is the lack of complete connectivity in real QCs.

An important clarification is that NSGA-II was executed independently for each graph instance, meaning that the architecture optimization was made to each specific problem. We acknowledge that this introduces additional computational overhead relative to QAOA,

https://doi.org/10.3390/mca31030078

which uses a fixed ansatz structure across all instances. The cost of the evolutionary search is discussed in terms of the total number of circuit evaluations in Section 4.

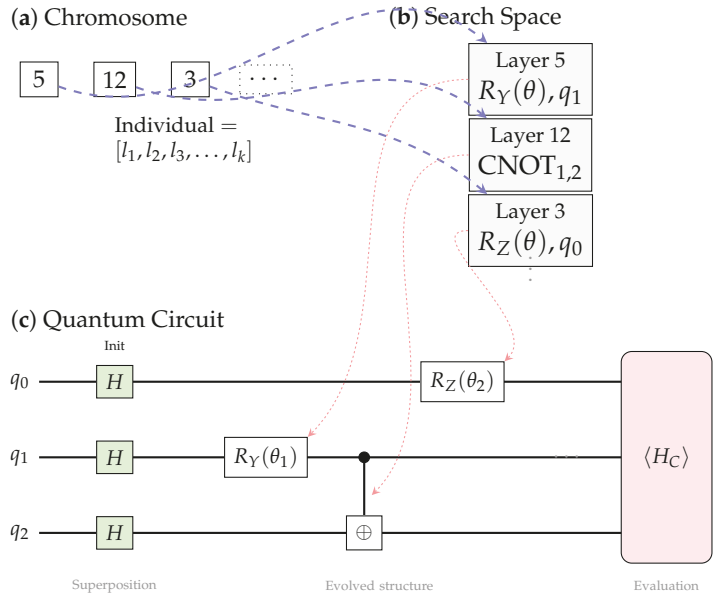


Figure 9. QNAS encoding and circuit synthesis. (**a**) Chromosome representation with variable length k. (**b**) Search space indexed by genes. (**c**) Synthesized a circuit from a chromosome [5, 12, 3, ...]. Dotted lines trace gene-to-gate mappings.

The search space $\mathcal{S}$ consists of $|\mathcal{S}| = 7$ atomic operations: parameterized single-qubit rotations (R_x, R_y, R_z), the fixed Hadamard gate (H), linear nearest-neighbor CNOTs, circular CNOTs, and problem-specific ZZ interaction layers.

Experiment Hyperparameters: The NSGA-II algorithm was executed with a population size of $N = 30$ over a maximum of $G_{\max} = 50$ generations. The variational operators employed a single-point crossover with probability $p_c = 0.7$ and random mutation with probability $p_m = 0.3$.

3.3. Noise Models and Their Importance in the Search

As previously stated, QCs tend to have more errors due to coherence issues and other physical factors. In this paper, we analyze the VQCs under varying noise conditions. To achieve this, we avoided simple depolarizing models in favor of a physically motivated noise model applied via `Qiskit Aer` simulator. The evolution of the density matrix ρ, which is a generalization of the state-vector approach and a representation of a VQC/VQA, is then governed by the operator-sum representation $\mathcal{E}(\rho) = \sum_k E_k \rho E_k^\dagger$, where E_k corresponds to the basis (in this scenario, the computational basis) where the state is described. We modeled three distinct error channels applied sequentially to every operation in the circuit:

3.3.1. Gate Error Model

Gate errors were modeled using the depolarizing channel formalism, which approximates the complex error processes in physical devices. Under this model, a quantum state ρ subjected to a depolarizing channel with error probability p evolves as follows:

$$\mathcal{E}(\rho) = (1-p)\rho + \frac{p}{d^2-1}\sum_{P \neq I} P\rho P^\dagger \tag{10}$$

where d is the Hilbert space dimension, and P ranges over the Pauli group excluding identity. (We use the convention where p represents the total depolarizing probability, distributed uniformly across $d^2 - 1$ non-identity Pauli operators, consistent with Qiskit

https://doi.org/10.3390/mca31030078

Aer v0.12.2's `depolarizing_error(p, num_qubits)` implementation. Note that for single-qubit gates, e.g., ($d = 2$), each Pauli error has a probability of $p/3$; for two-qubit gates, e.g, ($d = 4$), it is $p/15$.) The error rates employed in this study are summarized in Table 1.

Table 1. Noise Model Parameters. Error rates calibrated against IBM Eagle/Heron processor specifications. The depolarizing model provides an estimate of gate-fidelity loss by distributing errors uniformly across Pauli channels.

Error Source	Symbol	Value	Physical Mechanism
Single-qubit gate error	p_{1q}	1.0×10^{-3}	Depolarizing channel applied after each R_x, R_y, R_z rotation; models pulse calibration drift and decoherence during gate execution.
Two-qubit gate error	p_{2q}	1.0×10^{-2}	Depolarizing channel on both qubits post-CNOT/CZ; captures crosstalk, residual ZZ coupling, and extended gate duration effects.
Readout error	p_{ro}	2.0×10^{-2}	Classical bit-flip confusion matrix; models resonator thermal population, insufficient integration time, and state relaxation during measurement.

The order-of-magnitude disparity between p_{1q} and p_{2q} reflects the fundamental difficulty of implementing high-fidelity entangling operations, which require precise control of inter-qubit coupling while suppressing parasitic interactions. This asymmetry motivates the complexity weighting scheme introduced in Section 3.4.

An important remark is that the depolarizing channel does not improve the performance of VQAs directly. However, QNAS benefits indirectly from noise through two mechanisms. First, during the NSGA-II search, decoherence penalizes complex circuits in the fitness evaluation via the fidelity term in r_{eff} (Equation (13)). The Pareto-based selection pressure in NSGA-II implicitly suppresses deep architectures during evolution, since circuits exceeding a complexity threshold where fidelity decay dominates any marginal improvement in expressibility are outperformed on both objectives by more compact alternatives.

3.3.2. Readout Error Model

Measurement errors were implemented using a classical confusion matrix M applied to the ideal measurement outcome distribution:

$$\vec{p}_{\text{noisy}} = M \cdot \vec{p}_{\text{ideal}}, \quad M = \begin{pmatrix} 1 - p_{\text{ro}} & p_{\text{ro}} \\ p_{\text{ro}} & 1 - p_{\text{ro}} \end{pmatrix} \tag{11}$$

where $p_{\text{ro}} = 0.02$ represents the probability of a bit-flip error during the measurement process (see Figure 10). This symmetric error model assumes equal misclassification rates for $|0\rangle \to |1\rangle$ and $|1\rangle \to |0\rangle$ transitions.

3.3.3. Discussion: Impact of Different Noise Channels

The depolarizing channel employed in this study distributes errors uniformly across all Pauli operators, constituting the worst-case scenario for a given error probability p.

Phase damping (dephasing) is a unital channel characterized by Kraus operators $E_0 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\lambda} \end{pmatrix}$ and $E_1 = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\lambda} \end{pmatrix}$, where λ is related to the T_2 coherence time. This channel destroys off-diagonal coherences while preserving diagonal populations. Since the MaxCut cost Hamiltonian $H_C = \frac{1}{2}\sum(I - Z_i Z_j)$ depends exclusively on diagonal elements of the density matrix in the computational basis, phase damping has a reduced impact on the expectation value $\langle H_C \rangle$ compared to depolarizing noise.

https://doi.org/10.3390/mca31030078

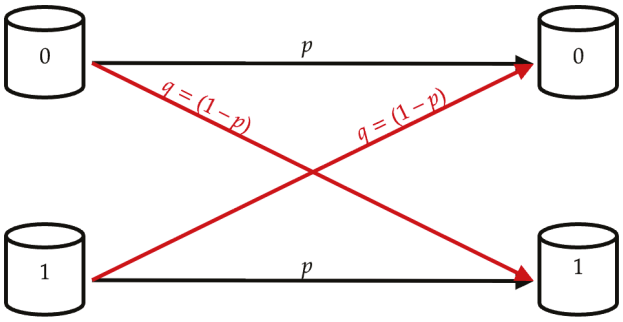


Figure 10. Classical visualization of readout errors in a computer. In traditional computers, these errors can be mitigated by redundancy, which is not possible in QCs. In the case of QCs, the translation is straightforward $|0\rangle \to 0$ and $|1\rangle \to 1$.

Amplitude damping models energy relaxation (T_1 decay) and is described by Kraus operators $E_0 = |0\rangle\langle 0| + \sqrt{1-\gamma}\,|1\rangle\langle 1|$ and $E_1 = \sqrt{\gamma}\,|0\rangle\langle 1|$, where $\gamma = 1 - \exp\left(-t_{\text{gate}}/T_1\right)$. Unlike the depolarizing and dephasing channels, amplitude damping is non-unital: it does not preserve the maximally mixed state, instead driving the system asymptotically toward $|0\rangle^{\otimes n}$. For the MaxCut problem, we define an initial setup which corresponds to placing all vertices in the same partition (a trivial cut with $C = 0$).

For equivalent error probabilities, the expected degradation hierarchy is depolarizing $\geq$ amplitude damping $>$ phase damping. The crossover point identified at $n \sim 12$ (discussed in detail in Section 4) thus represents a pessimistic estimate under the depolarizing model; alternative channels would shift this threshold to equal or larger system sizes while preserving the qualitative advantage of QNAS over QAOA.

3.4. Performance Metrics

As described earlier, different types of gates correspond to different problems; in particular, assuming equal complexity is a mistake. To correct this, we define a weighted complexity metric that accounts for the different set of quantum gates:

$$C = N_{1q} + w \cdot N_{2q}, \quad w = \frac{p_{2q}}{p_{1q}} = 10 \tag{12}$$

where N_{1q} and N_{2q} denote the total number of single-qubit and two-qubit gates, respectively. The weighting factor w is derived directly from the error rate ratio.

Noise-Adjusted Approximation Ratio

Standard approximation ratios $r_A = \langle C \rangle / C_{\max}$ are not effective for evaluating algorithms in high-noise regimes. In a physical system dominated by maximal decoherence (white noise), the quantum state becomes the maximally mixed state $I/2^n$. For unweighted MaxCut, measuring this state yields a random cut, which corresponds to an expected approximation ratio of 0.5.

To quantify the utility of a quantum circuit relative to this physical baseline, we define r_{eff} as the *Noise-Adjusted Approximation Ratio*:

$$r_{\text{eff}} = r_{A,\text{ideal}} \cdot F_{\text{est}} + 0.5 \cdot (1 - F_{\text{est}}) \tag{13}$$

where $r_{A,\text{ideal}}$ is the noiseless approximation ratio derived from ideal simulation. $F_{\text{est}} \approx \exp\left[-(p_{1q}N_{1q} + p_{2q}N_{2q})\right]$ corresponds to the estimated circuit fidelity, calculated using the gate error rates ($p_{2q} \approx 10^{-2}$) and gate counts (N) specific to each ansatz. This constitutes a pessimistic bound on circuit fidelity, as it assumes independent depolarizing errors with no error cancellation or coherent recombination.

https://doi.org/10.3390/mca31030078

In the limit of perfect hardware ($F \to 1$), $r_{\text{eff}} \to r_{A,\text{ideal}}$. As the circuit depth exceeds the coherence time ($F \to 0$), the performance converges to the random guessing value of 0.5. This allows us to identify the point where the algorithm becomes indistinguishable from noise.

The balance between C and r_{eff} defines a Pareto frontier in the complexity–performance space. Circuits lying on this frontier represent optimal configurations where no improvement in r_{eff} is achievable without increasing C, and vice-versa. In order to create fair comparisons the knee point of PF was taken into account.

We note that r_{eff} is an estimate that combines the ideal approximation ratio from noiseless simulation with an analytical fidelity model. Another approach would be to report the directly simulated noisy approximation ratio obtained from Qiskit Aer.

4. Results

We benchmarked the standard QAOA with $p = 5$ against the adaptive QNAS. The choice of $p = 5$ was due to the fact that, theoretically, this depth provides sufficient expressibility to achieve approximation ratios $r_A > 0.90$, minimizing the *underfitting* errors typical of shallow circuits.

We selected $p = 5$ as a representative deep-circuit regime because it lies above the theoretical threshold for high approximation ratios on the tested graph sizes while remaining computationally tractable for full simulation across 490 instances. We acknowledge that intermediate depths ($p \in \{1, 2, 3\}$) may yield different complexity–performance trade-offs; in particular, a lower p would reduce the complexity gap between QAOA and QNAS. A systematic multi-depth analysis constitutes an immediate extension of this work that would strengthen the generalizability of the reported crossover behavior.

Figure 11a validates this theoretical premise. In noiseless simulations, QAOA ($p = 5$) consistently delivers superior solution quality ($r_A \approx 0.92$ for $n = 16$), outperforming the shallower structures discovered by QNAS. This confirms that, in an ideal fault-tolerant regime, depth is the primary driver of convergence.

However, Figure 11b exposes the implementation barrier. The Complexity of the $p = 5$ ansatz scales linearly but with a prohibitive slope. For $n = 16$, the circuit requires approximately 2700 weighted operations to enforce the multilayered topology. In contrast, QNAS identifies hardware-efficient circuits with a complexity of ≈370, representing an 86% reduction in gate resources while preserving structural connectivity.

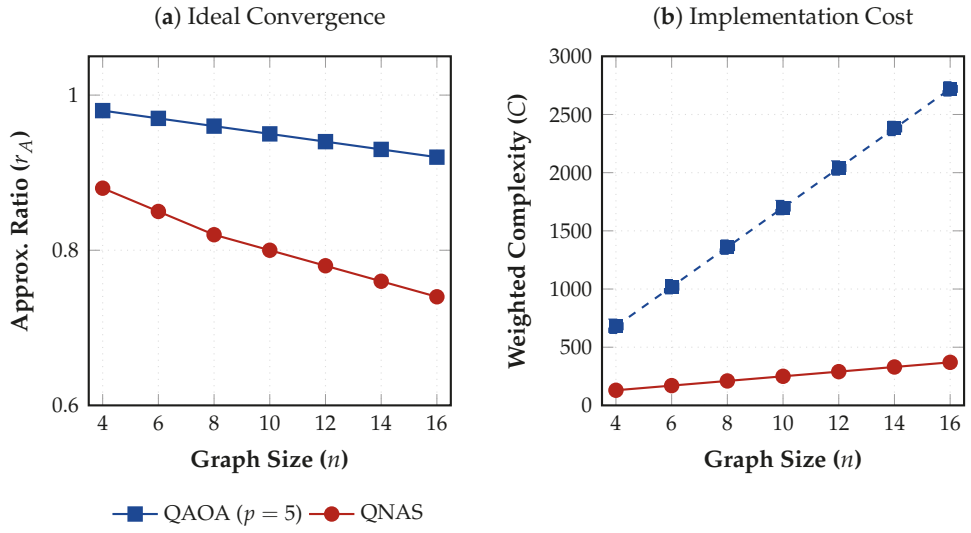


Figure 11. The cost of depth. (**a**) In ideal conditions, QAOA ($p = 5$) dominates due to high expressibility. (**b**) However, the associated hardware cost for $n = 16$ exceeds 2700 operations, creating a massive gap compared to the optimized QNAS.

https://doi.org/10.3390/mca31030078

4.1. What Solutions Are Really Feasible?

The discrepancy between theoretical potential and physical realization is quantified by the *Noise-Adjusted Approximation Ratio* (r_{eff}). Table 2 presents the metrics for $n = 16$. Despite QAOA showing a high ideal ratio, the estimated circuit fidelity drops to 6.6% due to error accumulation. This forces the effective ratio to converge toward the random noise floor (0.5).

Table 2. Performance Summary ($n = 16$). QAOA ($p = 5$) maximizes theoretical potential but fails to preserve the result under noise.

Metric	QAOA ($p = 5$)	QNAS
Ideal Ratio ($r_{A,\text{ideal}}$)	0.92	0.74
Complexity Score (C)	2720	370
Est. Fidelity (F)	0.06	0.69
Effective Ratio (r_{eff})	0.53 (Noise)	0.67 (Signal)

We observe a crossover at $n \approx 12$. Beyond this point (see Figure 12), the $p = 5$ ansatz enters a regime where adding depth destroys more information than it creates. QNAS, by sacrificing theoretical perfection for structural efficiency, maintains a clear optimization signal ($r_{\text{eff}} > 0.65$) well into the region where QAOA collapses.

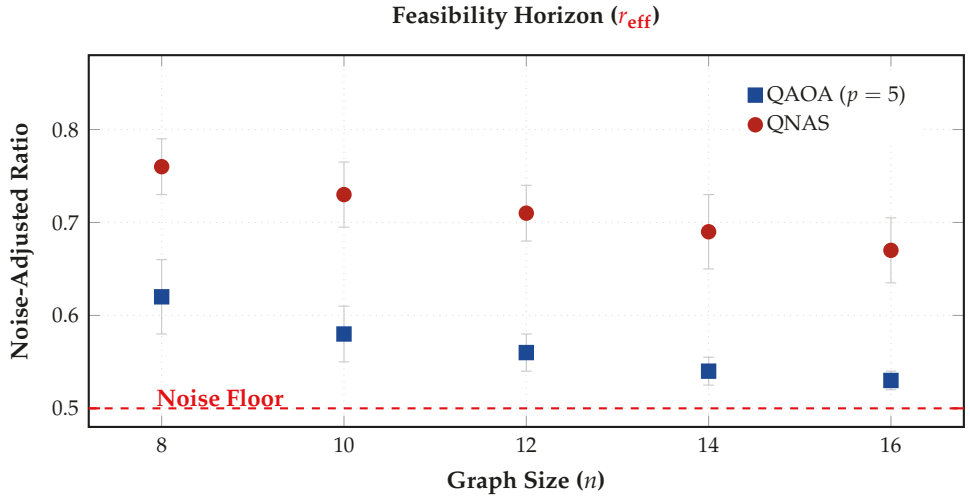


Figure 12. Effective performance with 95% confidence intervals. Red-dashed line marks the random guessing limit. QAOA converges to the noise floor with diminishing variance as system size increases.

The asymptotic behavior of both approaches can be characterized analytically from Equation (13). For QAOA, ($p = 5$), and the estimated fidelity decays as $F_{\text{est}} = \exp\left(-p_{2q} \cdot 2p \cdot \frac{n(n-1)}{2}\right)$, yielding $F_{\text{est}} \sim 0.02$ at $n = 18$ and $F_{\text{est}} \sim 0.006$ at $n = 20$, which drives r_{eff}. This degradation does not stabilize; it follows an exponential decay governed by the $\mathcal{O}(n^2)$ gate count scaling. In contrast, QNAS circuits maintain bounded complexity independent of graph connectivity, with empirical values $\mathcal{C} < 500$ across all tested system sizes. Extrapolating with $\mathcal{C} \sim 400$ gives $F_{\text{est}} \sim 0.67$ and $r_{\text{eff}} \sim 0.74 \times 0.67 + 0.5 \times 0.33 \approx 0.66$, suggesting that QNAS performance stabilizes in the interval $[0.60, 0.66]$ as n increases beyond the tested range.

To create a fair comparison and test the hypothesis that smaller circuits exhibit greater noise resilience, we expanded our experiments to include QAOA at intermediate depths ($p \in \{1, 3\}$). As illustrated in the Complexity–Performance Pareto Front (Figure 13), QAOA ($p = 1$) reduces the structural complexity ($\mathcal{C} \approx 156$) compared to the $p = 5$ regime ($\mathcal{C} \approx 763$).

https://doi.org/10.3390/mca31030078

However, an important remark is that it suffers from severe underfitting, achieving an effective approximation ratio of only $r_{\text{eff}} \approx 0.76$. At the same time, QNAS strictly dominates the Pareto frontier. This suggests that the evolutionary approach consistently discovers more hardware-efficient architectures ($\mathcal{C} \approx 41$) while approaching optimal theoretical values ($r_{\text{eff}} \approx 0.98$). This small validation in a subset suggests that the superiority of QNAS is not merely an artifact of comparing against a deep, noise-sensitive ansatz.

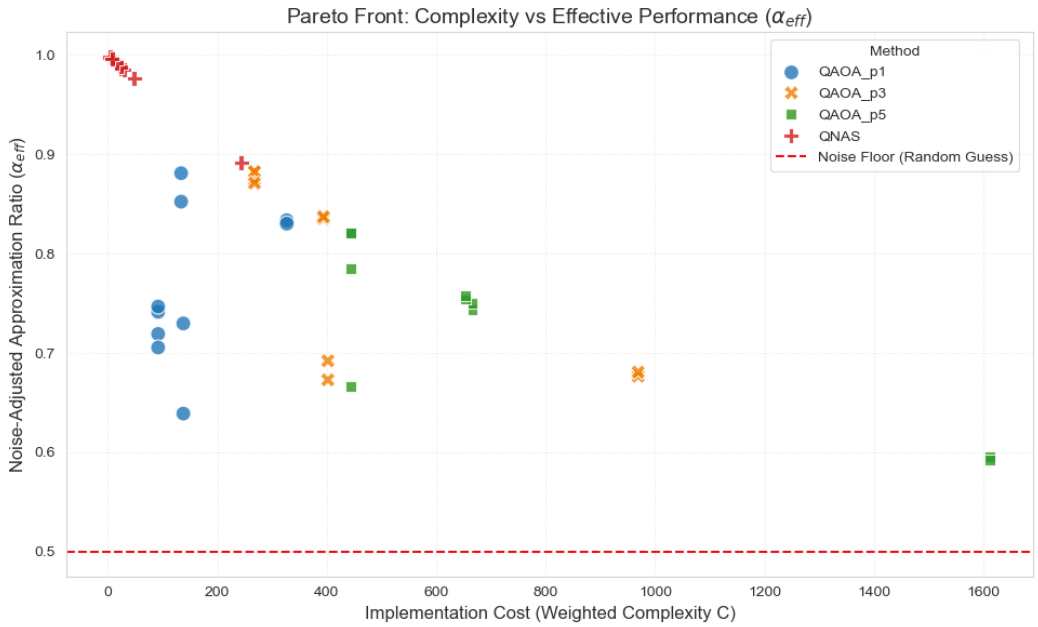


Figure 13. Complexity–Performance Pareto Front. QNAS strictly dominates the optimal trade-off curve compared to QAOA evaluated at multiple depths ($p \in \{1, 3, 5\}$).

4.2. Structural Analysis

We decomposed performance by a stratified description of the graph topology and density.

Topology Dependence

Figure 14 contrasts performance across three distinct graph classes. While QAOA ($p = 5$) exhibits robustness in regular graphs ($\langle d \rangle \approx 2 - 4$), its effective performance collapses on Scale-Free networks.QNAS excels here by pruning hub connections, preserving the global signal.

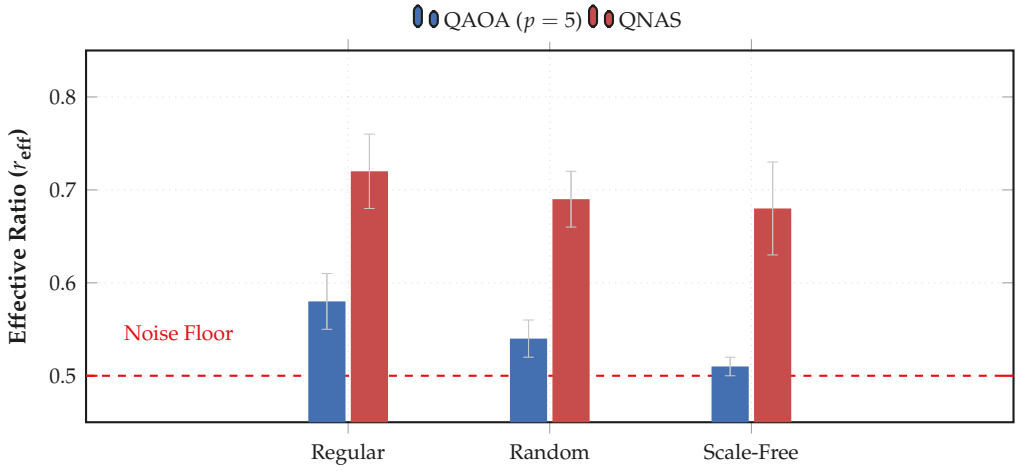


Figure 14. Topology Dependence. Mean effective performance. QAOA suffers critical degradation on Scale-Free graphs, approaching the noise floor, while QNAS adapts to the irregular structure.

https://doi.org/10.3390/mca31030078

Finally, Figure 15 illustrates that, for fixed depth p, the circuit depth of QAOA scales linearly with average degree $\langle d \rangle$. At $\langle d \rangle > 4$, error accumulation becomes critical. QNAS maintains a flat performance profile.

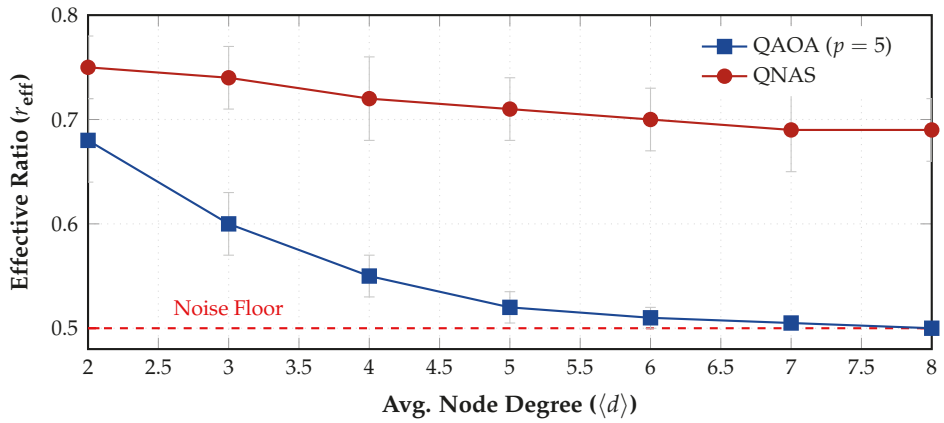


Figure 15. Resilience to Density. Effective performance vs. average node degree ($n = 16$). Increased connectivity degrades QAOA performance due to error accumulation, while QNAS remains stable.

5. Conclusions

In this work, we addressed ansatz selection for Variational Quantum Algorithms by comparing a heuristic strategy (QAOA, $p = 5$) with an evolutionary architecture search (QNAS). Our research, performed across 490 graph instances under noise models, reveals a serious trade-off between theoretical expressibility and physical implementability in the NISQ era.

While the QAOA ansatz demonstrated better convergence capabilities in noiseless simulations ($r_A > 0.90$), confirming that depth is the key element to the characterization of solution quality in ideal systems, its practical utility is strictly bounded by hardware coherence. At system size $n \approx 12$ solutions were comparable. Beyond this threshold, the linear scaling of the standard ansatz imposes a complexity penalty ($C > 2700$ for $n = 16$) that drives the effective approximation ratio to the random noise floor ($r_{\text{eff}} \approx 0.5$). In contrast, QNAS demonstrated remarkable structural resilience. By decoupling circuit complexity from graph connectivity, the evolutionary approach reduced gate requirements by approximately 86% compared to the $p = 5$ baseline. Through the ability to selectively delete non-essential entanglement—QNAS maintained inside a valid optimization ($r_{\text{eff}} > 0.65$) regime.

These findings suggest that simply increasing circuit depth to improve theoretical convergence is a counterproductive strategy without error correction. Instead, resources should be directed toward adaptive, hardware-aware ansatz compilation. Future work will extend the benchmark to larger systems ($n > 20$) using tensor network simulators to validate the analytical extrapolation presented in Section 4 and further explore the limits of topological pruning. Second, the noise model will be generalized to include amplitude damping (T_1 relaxation) and phase damping (T_2 dephasing) channels independently and in combination, using device-specific parameters from current IBM quantum processors to quantify how unital versus non-unital noise modifies the Pareto front topology discovered by QNAS and the degradation profile of QAOA.

Furthermore, deep circuits ($p = 5$) collapse under decoherence, while shallow circuits ($p = 1$) suffer from extreme underfitting. QNAS bypasses this limitation, showing that dynamic, hardware-aware ansatz compilation is an alternative superior to static topologies in the current NISQ era.

https://doi.org/10.3390/mca31030078

Author Contributions: Conceptualization, E.I.J.C. and H.T.-M.; methodology, E.I.J.C.; software, E.I.J.C.; validation, E.I.J.C., H.T.-M. and E.M.-M.; formal analysis, E.I.J.C.; investigation, review and editing, E.I.J.C., H.T.-M. and E.M.-M.; visualization, E.I.J.C.; supervision, H.T.-M. and E.M.-M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECITHI) through BECAS NACIONALES 2023 CVU: 1083296.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author E.I.J.C, upon reasonable request.

Acknowledgments: During the preparation of this manuscript, the authors used Grammarly for the purposes of enhancing the English communication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

NISQ	Noisy Intermediate-Scale Quantum
VQA	Variational Quantum Algorithm
VQC	Variational Quantum Circuit
QAOA	Quantum Approximate Optimization Algorithm
QNAS	Quantum Neural Architecture Search
NSGA-II	Non-dominated Sorting Genetic Algorithm II
VQE	Variational Quantum Eigensolver
NAS	Neural Architecture Search
MOO	Multi-Objective Optimization
CNOT	Controlled-NOT gate
KL	Kullback–Leibler
L-BFGS-B	Limited-memory Broyden–Fletcher–Goldfarb–Shanno with Bound constraints
ER	Erdős–Rényi
BA	Barabási–Albert
WS	Watts–Strogatz

References

1. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. [CrossRef]
2. Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S.C.; Endo, S.; Fujii, K.; McClean, J.R.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. Variational quantum algorithms. *Nat. Rev. Phys.* **2021**, *3*, 625–644. [CrossRef]
3. Chen, S.Y.-C.; Huang, C.-M.; Hsing, C.-W.; Goan, H.-S.; Kao, Y.-J. Variational quantum reinforcement learning via evolutionary optimization. *Mach. Learn. Sci. Technol.* **2022**, *3*, 015025. [CrossRef]
4. Sim, S.; Johnson, P.D.; Aspuru-Guzik, A. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Adv. Quantum Technol.* **2019**, *2*, 1900070. [CrossRef]
5. McClean, J.R.; Boixo, S.; Smelyanskiy, V.N.; Babbush, R.; Neven, H. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.* **2018**, *9*, 4812. [CrossRef] [PubMed]
6. Cerezo, M.; Sone, A.; Volkoff, T.; Cincio, L.; Coles, P.J. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nat. Commun.* **2021**, *12*, 1791. [CrossRef] [PubMed]
7. Kim, Y.; Eddins, A.; Anand, S.; Wei, K.X.; van den Berg, E.; Rosenblatt, S.; Nayfeh, H.; Wu, Y.; Zaletel, M.; Temme, K.; et al. Evidence for the utility of quantum computing before fault tolerance. *Nature* **2023**, *618*, 500–505. [CrossRef] [PubMed]
8. Li, Y.Y.; Liu, R.J.; Hao, X.B.; Shang, R.H.; Zhao, P.X.; Jiao, L.C. EQNAS: Evolutionary quantum neural architecture search for image classification. *Neural Netw.* **2023**, *168*, 471–483. [CrossRef] [PubMed]
9. Zhu, X.; Hou, X. Quantum architecture search via truly proximal policy optimization. *Sci. Rep.* **2023**, *13*, 5157. [CrossRef] [PubMed]
10. Altares-López, S.; Ribeiro, A.; García-Ripoll, J.J. Automatic design of quantum feature maps. *Quantum Sci. Technol.* **2021**, *6*, 045015. [CrossRef]

11. Kuo, E.-J.; Fang, Y.-L.L.; Chen, S.Y.-C. Quantum architecture search via deep reinforcement learning. *arXiv* **2021**, arXiv:2104.07715. [CrossRef]
12. Karp, R.M. Reducibility among combinatorial problems. In *Complexity of Computer Computations*; Springer: Berlin/Heidelberg, Germany, 1972; pp. 85–103. [CrossRef]
13. Kadowaki, T.; Nishimori, H. Quantum annealing in the transverse Ising model. *Phys. Rev. E* **1998**, *58*, 5355. [CrossRef]
14. Farhi, E.; Neven, H. Classification with quantum neural networks on near term processors. *arXiv* **2018**, arXiv:1802.06002. [CrossRef]
15. Crooks, G.E. Performance of the quantum approximate optimization algorithm on the maximum cut problem. *arXiv* **2018**, arXiv:1811.08419. [CrossRef]
16. Farhi, E.; Goldstone, J.; Gutmann, S. A quantum approximate optimization algorithm. *arXiv* **2014**, arXiv:1411.4028. [CrossRef]
17. Wurtz, J.; Love, P. MaxCut quantum approximate optimization algorithm performance guarantees for $p > 1$. *Phys. Rev. A* **2021**, *103*, 042612. [CrossRef]
18. Basso, J.; Farhi, E.; Marwaha, K.; Gutmann, S.; Neven, H. The quantum approximate optimization algorithm at high depth for MaxCut on large-girth regular graphs and the Sherrington-Kirkpatrick model. *arXiv* **2022**, arXiv:2110.14206.
19. Brandão, F.G.S.L.; Broughton, M.; Farhi, E.; Gutmann, S.; Neven, H. For fixed control parameters the quantum approximate optimization algorithm's objective function value concentrates for typical instances. *arXiv* **2018**, arXiv:1812.04170.
20. Byrd, R.H.; Lu, P.; Nocedal, J.; Zhu, C. A limited memory algorithm for bound constrained optimization. *SIAM J. Sci. Technol.* **1995**, *16*, 1190–1208. [CrossRef]
21. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

Mathematical and Computational Applications

MDPI

Article

Evolutionary Linear Discriminant Projection for Sensory Analysis of Tortillas Fortified with Chilacayote Powder

Adriana-Laura López-Lobato [1], Héctor-Gabriel Acosta-Mesa [1], Efrén Mezura-Montes [1], Jimena-Esther Alba-Jiménez [2,*], Amalia-Guadalupe Rodríguez-Gómez [2], Elia-Nora Aquino-Bolaños [2] and Rosa-Hayde Alfaro-Rodríguez [3]

[1] Artificial Intelligence Research Institute, University of Veracruz, Xalapa 91097, Veracruz, Mexico; adrlopez@uv.mx (A.-L.L.-L.); heacosta@uv.mx (H.-G.A.-M.); emezura@uv.mx (E.M.-M.)
[2] Center for Food Research and Development, University of Veracruz, Xalapa 91190, Veracruz, Mexico; agrg.qfb@hotmail.com (A.-G.R.-G.); eliaquino@uv.mx (E.-N.A.-B.)
[3] Institute of Agricultural Sciences, Autonomous University of the State of Hidalgo, Tulancingo 43600, Hidalgo, Mexico; hayde_alfaro@uaeh.edu.mx
* Correspondence: jimalba@uv.mx

Abstract

Chilacayote (*Cucurbita ficifolia* Bouché) is recognized as a rich source of nutrients and bioactive compounds, making it a promising ingredient for fortifying staple foods such as corn tortillas. While fortification can improve nutritional properties, it may also alter sensory characteristics that determine consumer acceptance. Therefore, a rigorous and structurally grounded assessment of these sensory modifications is required. In this study, sensory evaluations were conducted with regular tortilla consumers using Check-All-That-Apply (CATA) questionnaires to examine six attributes (color, smell, texture, taste, mouthfeel, and aftertaste) in tortillas made with nixtamalized dough and commercial flour, both with and without chilacayote powder. Then, a structured framework for dimensionality reduction and sensory profile identification of tortillas is proposed. In this framework, three classical feature extraction methods (Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and a combination of both (PCA+LDA)) were compared with an evolutionary discriminant approach (Differential Evolutionary Linear Discriminant Analysis for Feature Extraction and Visualization (DE-LDA$_{FE}$)). The projection quality of these methods was evaluated using a multi-scale separability index that integrates global, semi-global, and local metrics, and the experiments were conducted considering global and attribute-based analyses. Beyond quantitative discrimination, the optimized projections enabled a geometric interpretation that allows the identification of sensory profiles for the tortilla variants. The proposed methodology bridges evolutionary optimization, structural separability assessment, and interpretable sensory characterization, offering a robust and adaptable strategy for multivariate food analysis and other complex discrimination problems and insights into the sensory impact of chilacayote fortification for the development of nutritionally enhanced tortillas that preserve consumer appeal.

Keywords: sensory analysis; tortilla fortification; feature extraction; PCA; LDA; PCA+LDA

1. Introduction

Cucurbita ficifolia Bouché, more commonly referred to in Mexico as chilacayote, is a fruit cultivated in Mexico that belongs to the Cucurbitaceae family. Recent studies have highlighted the comprehensive nutritional and health-promoting properties of various parts

of Cucurbitaceae plants, emphasizing their potential as functional ingredients in the food industry [1]. In particular, chilacayote has been shown to contain a wide range of beneficial components, including carbohydrates, vitamins, minerals, and phenolic compounds. These bioactive constituents have been associated with the prevention and/or treatment of chronic and cardiovascular diseases, exhibiting antioxidant, anti-inflammatory, and hypoglycemic effects [2].

In the realm of fruit utilization, ripe fruit can undergo a processing procedure that transforms it into a powder. This transformation signifies a pragmatic solution for its application in food-related contexts. Several studies have demonstrated that the administration of chilacayote powder results in several additional metabolic benefits, including reductions in blood triacylglyceride and insulin levels, as well as improvements in insulin resistance. Consequently, the development of new food products incorporating chilacayote, in its ripe or powdered form, is imperative, as it may contribute to current governmental strategies aimed at improving public health.

One promising application of chilacayote powder is its use in fortifying corn tortillas, a strategy that aligns with the food-to-food fortification framework, which seeks to improve the nutritional profile of staple foods through the inclusion of nutrient-dense whole foods rather than synthetic fortificants [3]. In this sense, with the fortification process, nutrients or bioactive compounds are incorporated into a food matrix [4], enabling the delivery of the nutritional benefits of chilacayote powder through widely consumed foods. In this context, tortillas were selected as the food vehicle due to their high consumption rate among the Mexican population, with approximately 94% of individuals including tortillas in their daily diet, making them an ideal carrier for functional ingredients. However, fortification not only modifies the nutritional profile of tortillas but also affects their structural properties, including appearance, flavor, and texture, which can significantly influence consumer satisfaction and, in some cases, result in low market acceptance [5]. Consequently, conducting sensory analyses is imperative to identify, compare, and enhance the desirable attributes of fortified tortillas.

Although a wide range of tortilla fortification strategies has been reported in the literature, relatively few studies have systematically evaluated the sensory attributes of tortillas fortified with functional ingredients. In this context, in [6,7] the appearance, flavor, odor, texture, and overall acceptability of tortillas fortified with soy flour and its by-products are assessed using 5- and 7-point hedonic scales, respectively. Furthermore, multivariate statistical techniques such as Principal Component Analysis (PCA) [8] and Generalized Procrustes Analysis (GPA) [9,10] have been widely used to characterize and differentiate sensory attributes among tortilla formulations.

Despite the well-documented health benefits of *Cucurbita ficifolia* and the growing interest in functional foods, its application as a fortifying agent in corn tortillas remains limited, particularly in terms of sensory perception and consumer acceptance. Most previous works on fortified tortillas have focused on nutritional enhancement, while sensory differences have been examined using hedonic scales combined with multivariate feature extraction techniques, such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and hybrid PCA+LDA approaches [11]. Among these, LDA stands out due to its ability to reduce data dimensionality and identify relevant sensory attributes, thereby facilitating the interpretation of consumer responses. Nevertheless, several studies have reported methodological limitations associated with LDA, especially when handling complex sensory data. In this context, innovative computational methods, including evolutionary strategies, are increasingly being adopted to address the complexities of food classification and authentication, often providing more reliable results than conventional linear approaches [12,13]. Consequently, in this research we propose an alternative ap-

https://doi.org/10.3390/mca31030082

proach with an evolutionary algorithm-based feature extraction method, which aims to overcome these limitations and provide a more robust framework for the sensory analysis of tortillas fortified with chilacayote powder.

2. Materials and Methods

The materials and methods employed in this study were designed to evaluate the sensory characteristics of corn tortillas fortified with chilacayote powder. This section describes the preparation of the biological material, the formulation and processing of fortified and non-fortified tortillas, and the sensory evaluation protocol applied to assess consumer perception. In addition, the feature extraction techniques applied in the analysis are described to clarify the experimental procedures and ensure methodological transparency and reproducibility.

2.1. Materials

2.1.1. Biological Material

Chilacayote powder was obtained from ripe *Cucurbita ficifolia* Bouché fruits. The fruits were sliced and dehydrated using an electric dehydrator (LÉQUIP, model 918) at 60 °C for 8 h, after which the dried material was ground to obtain a fine powder. The particle size of the chilacayote powder was not sieved or measured, as it was obtained as a fine and visually homogeneous powder and was not a variable within the scope of this study. Nixtamalized maize dough was purchased from a local dough distribution center, and commercial maize flour was obtained from a supermarket, both located in Xalapa, Veracruz, Mexico.

2.1.2. Tortilla Preparation and Treatments

Four types of tortillas were prepared: (i) nixtamalized dough tortilla, (ii) nixtamalized dough tortilla fortified with chilacayote powder, (iii) commercial flour tortilla, and (iv) commercial flour tortilla fortified with chilacayote powder. Each tortilla was prepared using 30 g of nixtamalized dough or commercial flour. For fortified tortillas, 1 g of chilacayote powder replaced 1 g of the base material, resulting in a formulation of 29 g of dough or flour and 1 g of chilacayote powder. Tortillas were cooked on a digital electric griddle (comal) maintained at a constant temperature of 200 °C. Surface temperatures were monitored with a digital infrared thermometer; the tortillas were cooked for 1 min on the first side, with the surface temperature ranging between 60 and 75 °C. They were then flipped and cooked for 30 s on the opposite side, followed by an additional 15 s on the first side to ensure uniform cooking.

2.1.3. Sensory Analysis

Sensory evaluation was conducted with a total of 150 untrained consumers who were regular tortilla consumers, aged between 18 and 25 years. No exclusion criteria were applied; participation was limited to individuals who expressed willingness to undergo the testing.

The evaluation was carried out at two locations: the Tulancingo de Bravo campus of the Universidad Autónoma del Estado de Hidalgo (75 judges: 48 women and 27 men) and the Faculty of Agronomy of the Universidad Veracruzana, Xalapa campus (75 judges: 33 women and 42 men). The samples evaluated included nixtamalized dough tortillas, commercial flour tortillas, and their respective formulations fortified with chilacayote powder. Tortilla samples were served at an average temperature of 35 °C and presented four times in randomized order. Each judge completed a Check-All-That-Apply (CATA) questionnaire for each sample, evaluating attributes related to color (cream, beige, sand, brown, yellow, camel, coffee, other), smell (nixtamal, corn, acidic, sour, burnt, earthy,

https://doi.org/10.3390/mca31030082

humid, sweet, piloncillo, other), texture (porous, soft, smooth, hot, rough, chewy, sandy, thick, elastic, rigid, moldable, spongy, other), taste (salty, sour, acidic, lime, corn, nixtamal, earthy, burnt, doughy, toasted, piloncillo, other), mouthfeel (elastic, rigid, hard, rough, moist, creamy, pasty, spongy, chewy, dry, gummy, other), and aftertaste (sweet, salty, corn, nixtamal, lime, piloncillo, other), following the methodology described in [14]. The use of these established descriptors is critical for enabling robust comparisons of sensory profiles and for validating the consistency between consumer perceptions and trained panel evaluations, as highlighted in recent studies on methodological terms development [15]. The 'other' responses were analyzed by collecting terms not included in the original questionnaire, provided they represented characteristics different from those established.

The sensory data analysis in this study primarily focuses on multivariate projection and the evaluation of class separability in a reduced-dimensional space. Consequently, traditional univariate inferential tests, such as Cochran's Q for CATA data, were bypassed in this phase of the research to prioritize the development and validation of the evolutionary framework. This approach allows a direct assessment of how the algorithm captures the global structure of consumer perception by optimizing the projection matrix.

2.2. Feature Extraction Methods

This section describes four linear dimensionality reduction techniques: Principal Components Analysis (PCA), Linear Discriminant Analysis (LDA), their combination (PCA+LDA), and an evolutionary approach known as Differential Evolutionary Linear Discriminant Analysis for Feature Extraction and Visualization (DE-LDA$_{FE}$).

Throughout this section, scalars are denoted by standard italic letters (e.g., $x \in \mathbb{R}$), vectors by bold lowercase letters (e.g., $\mathbf{x} \in \mathbb{R}^m$), and matrices by bold uppercase letters (e.g., $\mathbf{X} \in \mathbb{R}^{m \times n}$). The i-th element of a vector is written as x_i, and the (i, j)-th entry of a matrix as X_{ij}.

Under this notation, the dataset is represented by a matrix $\mathbf{X} \in \mathbb{R}^{m \times n}$. The four aforementioned dimensionality reduction techniques aim to transform this representation into a lower-dimensional space $\mathbf{Y} \in \mathbb{R}^{m \times d}$, with $d \leq n$. This transformation is achieved through linear combinations of the original variables $\mathbf{x}_i \in \mathbb{R}^m$, for $i = 1, \ldots, n$, as presented in Equation (1) [16,17].

$$\mathbf{y}_i = w_{1i}\mathbf{x}_1 + w_{2i}\mathbf{x}_2 + \cdots + w_{ni}\mathbf{x}_n. \tag{1}$$

This transformation can be expressed as the matrix multiplication shown in Equation (2), where $\mathbf{W} \in \mathbb{R}^{n \times d}$ is the projection matrix composed of the weights of the linear combinations as columns.

$$\mathbf{Y} = \mathbf{X}\mathbf{W}, \quad \text{with} \quad \mathbf{W} = \begin{pmatrix} w_{11} & w_{12} & \cdots & w_{1d} \\ w_{21} & w_{22} & \cdots & w_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & \cdots & w_{nd} \end{pmatrix}. \tag{2}$$

Different projection matrices yield distinct representations of the dataset $\mathbf{X}$. Therefore, the following sections describe the techniques employed by the four methods to obtain these projection matrices. The computation of the projection matrices differs across methods, as each technique is designed to achieve a specific objective.

2.2.1. Principal Components Analysis (PCA)

Principal Components Analysis (PCA) is an *unsupervised* dimensional reduction technique that identifies orthogonal directions of *maximum variance* in the data, known as principal components [18]. These directions are obtained by solving the eigenvalue prob-

https://doi.org/10.3390/mca31030082

lem associated with the empirical covariance matrix of the dataset. Given a centered dataset $\mathbf{X} \in \mathbb{R}^{m \times n}$, the covariance matrix is computed as in Equation (3).

$$\mathbf{S} = \frac{1}{m-1}\mathbf{X}^T\mathbf{X}. \tag{3}$$

PCA then solves the eigenvalue problem presented in Equation (4), where λ_i and $\hat{v}_i$ denote the i-th eigenvalue and its corresponding eigenvector, respectively.

$$\mathbf{S}\boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i. \tag{4}$$

The eigenvalues represent the amount of variance captured by each principal component and are ranked in descending order such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$.

Then, the projection matrix $\mathbf{W_{PCA}} \in \mathbb{R}^{n \times d}$ is constructed by selecting the d eigenvectors associated with the largest eigenvalues, as in Equation (5).

$$\mathbf{W_{PCA}} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_d]. \tag{5}$$

The reduced representation of the data is obtained by projecting the original dataset onto the lower-dimensional subspace with the matrix multiplication presented on Equation (6).

$$\mathbf{Y} = \mathbf{X}\mathbf{W_{PCA}}. \tag{6}$$

This transformation preserves most of the variance presented in the original data while reducing dimensionality.

2.2.2. Linear Discriminant Analysis (LDA)

Linear Discriminant Analysis (LDA) is a *supervised* dimensionality reduction technique that seeks to project a dataset to maximize *class separability* [19]. The process is intended to achieve this objective by simultaneously maximizing between-class variance and minimizing within-class variance. Given a labeled dataset $\mathbf{X} \in \mathbb{R}^{m \times n}$ composed of C classes, let $\boldsymbol{\mu}_c$ be the mean vector of class c, and let $\boldsymbol{\mu}$ be the global mean vector of the dataset. The within-class scatter matrix $\mathbf{S_W}$ and the between-class scatter matrix $\mathbf{S_B}$ are defined as the Equations (7) and (8), respectively, where N_c is the number of samples in class c.

$$\mathbf{S_W} = \sum_{c=1}^{C} \sum_{\mathbf{x}_i \in c} (\mathbf{x}_i - \boldsymbol{\mu}_c)(\mathbf{x}_i - \boldsymbol{\mu}_c)^T. \tag{7}$$

$$\mathbf{S_B} = \sum_{c=1}^{C} N_c(\boldsymbol{\mu}_c - \boldsymbol{\mu})(\boldsymbol{\mu}_c - \boldsymbol{\mu})^T. \tag{8}$$

LDA is based on Fisher's criterion, which seeks a projection matrix $\mathbf{W_{LDA}}$ that maximizes the ratio of between-class variance to within-class variance in the projected space. The optimization problem can be formulated as in Equation (9).

$$\mathbf{W_{LDA}} = \arg\max_{\mathbf{W}} \frac{|\mathbf{W}^T\mathbf{S_B}\mathbf{W}|}{|\mathbf{W}^T\mathbf{S_W}\mathbf{W}|}. \tag{9}$$

The optimal projection matrix $\mathbf{W_{LDA}}$ is obtained by solving the generalized eigenvalue problem presented in Equation (10), where λ_i and $\boldsymbol{v}_i$ are the generalized eigenvalues and eigenvectors, respectively.

$$\mathbf{S_B}\boldsymbol{v}_i = \lambda_i \mathbf{S_W}\boldsymbol{v}_i. \tag{10}$$

https://doi.org/10.3390/mca31030082

The eigenvectors corresponding to the largest eigenvalues define the directions that maximize class separation. The projection matrix $\mathbf{W_{LDA}} \in \mathbb{R}^{n \times d}$ is formed by selecting the d eigenvectors associated with the highest eigenvalues, as shown in Equation (11).

$$\mathbf{W_{LDA}} = [\boldsymbol{\nu}_1, \boldsymbol{\nu}_2, \cdots, \boldsymbol{\nu}_d]. \tag{11}$$

The low-dimensional representation of the dataset is obtained with the matrix multiplication on Equation (12).

$$\mathbf{Y} = \mathbf{X}\mathbf{W_{LDA}}. \tag{12}$$

LDA provides a feature space that enhances class separability by incorporating class labels, making it suitable for data classification and interpretation.

In practice, LDA may suffer from the Small Sample Size problem (SSS problem), in which the within-class scatter matrix $\mathbf{S_W}$ becomes singular or ill-conditioned when the number of features exceeds the number of samples [20]. To address this limitation, a common strategy is to apply the PCA+LDA method, which is described in the following section.

2.2.3. PCA+LDA

The combined PCA+LDA approach is employed to overcome the limitations of LDA when applied to high-dimensional datasets with a limited number of samples [21]. In this strategy, PCA is first used as a preprocessing step to reduce the dimensionality of the original data while preserving most of its variance. Then the reduced dataset is used as input for LDA. By operating in this lower-dimensional space, the within-class scatter matrix becomes nonsingular, allowing the LDA optimization problem to be properly defined and enabling class separability in the transformed space.

The projection matrix of the PCA+LDA method is then obtained as the product of the individual projection matrices presented in Equation (13), and the low-dimensional representation of the original dataset is computed as in Equation (14).

$$\mathbf{W_{PCA+LDA}} = \mathbf{W_{PCA}}\mathbf{W_{LDA}}. \tag{13}$$

$$\mathbf{Y} = \mathbf{X}\mathbf{W_{PCA+LDA}}. \tag{14}$$

By integrating PCA and LDA, this approach combines variance preservation and discriminative feature extraction, resulting in a more stable and effective data projection.

2.2.4. Differential Evolutionary Linear Discriminant Analysis for Feature Extraction and Visualization (DE-LDA$_{FE}$)

In this study, the Differential Evolutionary Linear Discriminant Analysis for Feature Extraction and Visualization (DE-LDA$_{FE}$) method is utilized to reduce the dimensionality of the analyzed dataset. This process facilitates class separability and visualization. This method employs the Differential Evolution (DE) algorithm for optimization. The subsequent sections will thus provide a concise exposition of both methods.

Differential Evolution (DE)

Differential Evolution (DE) is a population-based stochastic search strategy for optimization [22]. It iteratively improves a population of candidate solutions using mutation, crossover, and selection operators. In the context of high-dimensional, non-convex search spaces, DE has been shown to be particularly effective because it does not require gradient information. Instead, it relies exclusively on objective function evaluations, a strategy

https://doi.org/10.3390/mca31030082

shown to be advantageous in such environments. There are several DE strategies [23,24], but the DE/best/1/bin strategy is described in detail next.

In generation g, the population comprises N candidate solutions (individuals) randomly generated in the domain of a given function f, called *fitness function*. The elements in the population are denoted by vectors $\mathbf{x}_i$ as in Equation (15), where d is the dimensionality of the problem.

$$\mathbf{x}_i^{(g)} = \left(x_{i1}^{(g)}, x_{i2}^{(g)}, \ldots, x_{id}^{(g)}\right) \text{ for } i = 1, \ldots, N. \tag{15}$$

Each vector is evaluated using the fitness function, in order to proceed with the following operators:

- Mutation

  For each target vector $\mathbf{x}_i^{(g)}$, a mutant vector $\boldsymbol{\nu}_i^{(g)}$ is generated by adding the weighted difference of two randomly selected population vectors to a third vector, according to the strategy. Equation (16) describes this process, where r_{best} is the index of the individual in the population with the highest fitness value, and r_1 and r_2 are indices randomly chosen from $\{1, 2, \ldots, N\}$, and different from both r_{best} and each other. The parameter F is the mutation factor, which controls the amplification of differential variation. The value of F is specified by the user and constrained to the interval $[0, 1]$.

$$\boldsymbol{\nu}_i^{(g)} = \mathbf{x}_{r_{best}}^{(g)} + F\left(\mathbf{x}_{r_1}^{(g)} - \mathbf{x}_{r_2}^{(g)}\right). \tag{16}$$

- Crossover

  To increase population diversity, the mutant vector $\boldsymbol{\nu}_i^{(g)}$ is combined with the target vector $\mathbf{x}_i^{(g)}$ to form a trial vector $\mathbf{u}_i^{(g)}$. In this process, a coordinate-wise selection is performed between the mutant vector $\boldsymbol{\nu}_i^{(g)}$ and the target vector $\mathbf{x}_i^{(g)}$ to form a trial vector $\mathbf{u}_i^{(g)}$. If a randomly generated value r_j is lower than the user-defined crossover rate CR, or if the coordinate index j corresponds to a randomly selected position J_{rand}, the corresponding component is inherited from $\boldsymbol{\nu}_i$. Otherwise, the component is taken from $\mathbf{x}_i$. This process is implemented through the binomial crossover, as delineated in Equation (17).

$$u_{ij}^{(g)} = \begin{cases} \nu_{ij}^{(g)}, & \text{if } r_j \leq CR \text{ or } j = J_{rand} \\ x_{ij}^{(g)}, & \text{otherwise} \end{cases} \tag{17}$$

- Selection

  The trial vector $\mathbf{u}_i^{(g)}$ competes with the target vector $\mathbf{x}_i^{(g)}$ for survival into the next generation. The selection is performed in a greedy manner as outlined in Equation (18), where $f(\cdot)$ denotes the objective function to be maximized.

$$\mathbf{x}_i^{(g+1)} = \begin{cases} \mathbf{u}_i^{(g)}, & \text{if } f\left(\mathbf{u}_i^{(g)}\right) \geq f\left(\mathbf{x}_i^{(g)}\right) \\ \mathbf{x}_i^{(g)}, & \text{otherwise} \end{cases} \tag{18}$$

These steps are repeated until a stopping criterion is met, such as reaching a maximum number of generations or population convergence.

The parameters N, G, CR, and F employed for the DE algorithm are problem-dependent and significantly influence the performance of the process [25].

Differential Evolution (DE) remains a powerful and versatile population-baed metaheuristic, particularly efective in solving complex, non-convex optimization problems across various engineering domains [26]. Due to its simple structure, limited number of control parameters, and robust global search capability, Differential Evolution has been suc-

https://doi.org/10.3390/mca31030082

cessfully applied to optimization problems, rendering it a suitable approach for identifying optimal projection matrices for feature extraction techniques, as will be described in the following section.

DE-LDA$_{FE}$

The DE-LDA$_{FE}$ method is proposed to obtain a projection matrix that enhances class separability while enabling low-dimensional visualization. Unlike classical Linear Discriminant Analysis (LDA), which relies on inverting scatter matrices and may become unstable under singularity or small-sample-size (SSS) conditions, this approach reformulates dimensionality reduction as a global optimization problem. Specifically, the Differential Evolution (DE/best/1/bin variant) is used to explore the space of orthonormal projection matrices and directly maximize the Fisher criterion without requiring matrix inversion. This formulation ensures that the objective function remains well-defined while providing robustness against multicollinearity and SSS scenarios, preserving the discriminative structure of the sensory data.

The projection space is restricted to two dimensions to facilitate graphical representation and interpretation, a common practice in sensory science where 2D discriminant plots enable direct geometric visualization of the relationships between sensory attributes and product formulations. Within this framework, each candidate projection matrix is encoded as a vector by concatenating its columns, as described in Equation (19), and represents an individual within a population of size N. These individuals evolve through mutation, crossover, and selection operators, and after crossover, a repair mechanism is applied to enforce the orthonormality constraint, ensuring that all candidate solutions remain valid projection matrices throughout the optimization process.

$$\begin{pmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ \vdots & \vdots & \vdots \\ w_{m1} & w_{m2} & w_{m3} \end{pmatrix} \iff (w_{11}, w_{21}, \cdots, w_{(m-1)1}, w_{m1}, w_{12}, w_{22}, \cdots, w_{(m-1)2}, w_{m2}, w_{13}, w_{23}, \cdots, w_{(m-1)3}, w_{m3}) \quad (19)$$

This formulation not only ensures robustness under non-ideal data conditions but also provides a flexible, data-agnostic framework. The effectiveness of this evolutionary strategy has been demonstrated in other high-dimensional contexts [27], highlighting its ability to handle multicollinearity and complex class structures beyond the sensory application considered in this study.

2.3. Class Separability Metrics

The effectiveness of a dimensionality-reduction method can be evaluated by the extent to which the projected feature space preserves class discrimination. In supervised settings, this is typically evaluated by assessing how well samples remain grouped by class label after projection. To quantitatively assess the quality of the resulting feature space, several class-separability metrics can be employed.

In this study, three complementary measures are considered: the Davies–Bouldin Index (DBI), the Silhouette Coefficient (SC), and the KNN Overlap Score (KNN-OS). These metrics evaluate distinct geometric properties of the projected space. DBI provides a global assessment by relating cluster compactness to centroid separation. SC provides a semi-local evaluation by integrating intra-class cohesion and nearest-cluster separation at the sample level. In contrast, the KNN-OS measures strictly local class mixing by quantifying label inconsistency within neighborhoods. Consequently, while DBI emphasizes global structural separation, SC and KNN-OS capture local overlap effects, which are particularly

https://doi.org/10.3390/mca31030082

relevant when analyzing low-dimensional visualizations. This complementary assessment reduces reliance on a single notion of cluster geometry and leads to a more robust and reliable evaluation of separability.

In this section, a brief explanation of these metrics is presented.

2.3.1. Davies–Bouldin Index (DBI)

The Davies–Bouldin Index (DBI) evaluates the relative similarity between clusters by quantifying the trade-off between intra-class dispersion and inter-class separation [28].

For a dataset consisting of samples from C classes, the *intra-class dispersion* S_i, defined in Equation (20), is calculated as the average distance between the samples belonging to class i and their corresponding centroid, denoted by $\boldsymbol{\mu}_i$. Here, $|C_i|$ represents the number of samples in class i, and $d(\cdot,\cdot)$ denotes the Euclidean distance used to measure pairwise distances.

$$S_i = \frac{1}{|C_i|} \sum_{\boldsymbol{x}_j \in C_i} d(\boldsymbol{x}_j, \boldsymbol{\mu}_i). \tag{20}$$

The *inter-class separation* between two classes i and j, denoted by M_{ij}, is defined as the distance between their respective centroids, as shown in Equation (21).

$$M_{ij} = d(\boldsymbol{\mu}_i, \boldsymbol{\mu}_j). \tag{21}$$

The similarity measure between classes i and j, denoted as R_{ij}, is defined as in Equation (22).

$$R_{ij} = \frac{S_i + S_j}{M_{ij}}. \tag{22}$$

For each class i, the maximum similarity with respect to all other classes is determined, yielding the worst-case similarity value R_i, as defined in Equation (23).

$$R_i = \max_{j \neq i} \{R_{ij}\}. \tag{23}$$

Finally, the Davies–Bouldin Index (DBI) is computed as the average of the worst-case similarity values across all classes, as shown in Equation (24).

$$DBI = \frac{1}{C} \sum_{i=1}^{C} R_i. \tag{24}$$

Lower DBI values indicate better class separability, as they reflect clusters that exhibit low intra-class dispersion and large inter-class separation. In other words, a small DBI corresponds to compact classes whose centroids are well separated from one another, thereby reducing cluster overlap.

2.3.2. Silhouette Coefficient (SC)

The Silhouette Coefficient (SC) evaluates how well each sample is assigned to its corresponding class compared to other classes, and is widely recognized as a robust internal validation metric for assessing both the cohesion and separation of clusters in high-dimensional datasets [29,30].

For sample $\boldsymbol{x}_i$, the mean distance between it and all other samples belonging to the same class is defined as the average intra-class distance, denoted by $a(i)$, and presented

https://doi.org/10.3390/mca31030082

in Equation (25). $\alpha(i)$ quantifies the extent to which the sample $\boldsymbol{x}_i$ is embedded within its designated class, thereby measuring its compactness.

$$\alpha(i) = \frac{1}{|C_i| - 1} \sum_{\substack{\boldsymbol{x}_j \in C_i \\ j \neq i}} d(\boldsymbol{x}_i, \boldsymbol{x}_j). \tag{25}$$

On the other hand, the minimum average inter-class distance is defined as the smallest average distance between the sample $\boldsymbol{x}_i$ and all the samples belonging to any other class C_k, with $k \neq i$, as presented in Equation (26). $\beta(i)$ is responsible for evaluating the distance between the sample and the nearest neighboring class, thus providing an indication of its separation.

$$\beta(i) = \min_{k \neq i} \left\{ \frac{1}{|C_k|} \sum_{\boldsymbol{x}_j \in C_k} d(\boldsymbol{x}_i, \boldsymbol{x}_j) \right\}. \tag{26}$$

Then, the Silhouette Coefficient (SC) for a sample $\boldsymbol{x}_i$ is defined as in Equation (27), where $\alpha(i)$ represents the average intra-class distance, and $\beta(i)$ the minimum average inter-class distance.

$$s(i) = \frac{\beta(i) - \alpha(i)}{\max\{\alpha(i), \beta(i)\}} \tag{27}$$

The SC ranges from -1 to 1. Values close to 1 indicate well-separated and compact classes, while values near 0 suggest class overlap.

The average Silhouette score over all samples is commonly considered to provide a global measure of geometric class separation.

2.3.3. KNN Overlap Score (KNN-OS)

The K-Nearest Neighbours Overlap Score (KNN-OS) is a local separability metric that evaluates the degree of class overlap in a projected feature space by analyzing neighborhood consistency. Unlike global dispersion-based measures, KNN-OS quantifies the frequency with which samples are surrounded by neighbors from different classes, thereby providing a local assessment of class mixing [31,32].

Given a dataset with labeled samples, let $S_k(\boldsymbol{x}_i)$ denote the set of k nearest neighbours of sample $\boldsymbol{x}_i$, computed using the Euclidean distance $d(\cdot, \cdot)$. For each sample $\boldsymbol{x}_i$, the local overlap score is defined as the proportion of neighbours that belong to a different class $OS_k(i)$, defined in Equation (28), where ℓ_i and ℓ_j denote the class labels of samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, respectively, and $\mathbb{I}(\cdot)$ is the indicator function, which equals to 1 if the condition is true and 0 otherwise.

$$OS_k(i) = \frac{1}{k} \sum_{\boldsymbol{x}_j \in S_k(\boldsymbol{x}_i)} \mathbb{I}(\ell_j \neq \ell_i). \tag{28}$$

The global KNN Overlap Score (KNN-OS) is then computed as the average over all samples, as in Equation (29), where m is the number of samples in the dataset.

$$KNN\text{-}OS = \frac{1}{m} \sum_{i=1}^{m} OS_k(i). \tag{29}$$

The KNN-OS takes values in the interval $[0, 1]$. Lower values indicate better class separability, as they imply that most samples are surrounded by neighbours of the same class. Conversely, higher values suggest significant class overlap in the projected space.

Because KNN-OS evaluates local neighbourhood structure, it is particularly suitable for assessing class separation in low-dimensional visualizations, where local mixing may not be fully captured by global metrics such as the Davies–Bouldin Index.

https://doi.org/10.3390/mca31030082

2.3.4. Comparative Interpretation of Class Separability Metrics

To ensure a consistent interpretation of the separability results, it is important to analyze the range and optimization direction of each metric considered in this study. The Davies–Bouldin Index (DBI), the Silhouette Coefficient (SC), and the KNN Overlap Score (KNN-OS) evaluate complementary geometric properties of the projected feature space; however, their numerical scales and optimization criteria differ.

The DBI is a non-negative and unbounded metric defined on the interval $[0, +\infty)$. Lower values indicate better separability, as they correspond to compact clusters with large inter-class distances. In contrast, the Silhouette Coefficient is bounded in the interval $[-1, 1]$, where values close to 1 indicate well-clustered samples, values near 0 suggest overlapping clusters, and negative values reflect potential misclassification. The KNN-OS metric lies in the interval $[0, 1]$, with lower values indicating reduced local class mixing and, therefore, better separability.

Table 1 summarizes the numerical ranges and optimization criteria for the evaluated metrics. The structural level indicates the geometric scope at which each metric evaluates class separability. Global structure refers to measures based on overall cluster dispersion and centroid separation; semi-global structure denotes metrics computed at the sample level but influenced by neighboring cluster relationships; and local structure corresponds to strictly neighborhood-based measures that quantify label consistency within local regions of the projected space.

Table 1. Interpretation and optimization direction of class separability metrics.

Separability Metric	Numerical Range	Optimal Direction	Structural Level
DBI	$[0, +\infty)$	Lower is better	Global structure
SC	$[-1, 1]$	Higher is better	Semi-global structure
KNN-OS	$[0, 1]$	Lower is better	Local structure

2.4. General Experiment Design

This section describes the experiments conducted to compare dimensionality-reduction methods. The analysis aims to assess the discriminative quality of the projected feature spaces obtained by each method. To this end, the class separability metrics described in Section 2.3 were employed as quantitative evaluation criteria.

In these experiments the dataset was not partitioned into training and testing subsets. This decision was made because the primary objective of the study is not to evaluate classification accuracy or predictive generalization performance, but rather to compare the intrinsic discriminative capacity of different feature representations. Accordingly, the evaluation focuses exclusively on geometric and structural separability properties of the projected data.

The PCA, LDA, and PCA+LDA methods are analytical and deterministic approaches. Given the same dataset, these methods yield unique projection matrices; therefore, a single execution suffices to obtain their corresponding results.

In contrast, the DE-LDA$_{FE}$ method is a stochastic, population-based metaheuristic algorithm. Due to its evolutionary search mechanism, different runs may yield distinct projection matrices and separability values. To ensure statistical reliability, the DE-LDA$_{FE}$ algorithm was executed 10 times under the same parameter configuration. The reported separability metrics correspond to the mean values obtained across these runs. Furthermore, to provide a fair and meaningful comparison with the deterministic analytical methods, the best solution identified among the 10 executions is reported and used for direct comparison. This evaluation framework allows a consistent comparison between deterministic

https://doi.org/10.3390/mca31030082

and stochastic approaches while accounting for the inherent variability of evolutionary optimization methods.

For the experimental evaluation, two strategies were considered. First, a global analysis was conducted using all variables in the dataset simultaneously, thereby enabling assessment of overall class separability across the full feature space. Second, an attribute-based analysis was performed, in which only the subset of variables corresponding to each sensory attribute was considered independently (color, smell, texture, taste, mouthfeel, and aftertaste, as described in Section 2.1.3). This dual evaluation framework enables the examination of both the collective discriminative power of all sensory variables and the individual contribution of each attribute to class separation.

Figure 1 summarizes the experimental workflow implemented in this study. For each dataset (Xalapa or Tulancingo), the process begins by considering one of the two analysis strategies (global or attribute-based). Subsequently, dimensionality-reduction techniques are applied to compute projection matrices in a two-dimensional space. Both analytical methods (PCA, LDA, and PCA+LDA) and the proposed evolutionary approach (DE-LDA$_{FE}$) are implemented under identical dimensional constraints to ensure a fair comparison. The resulting projected datasets are then evaluated using multiple class separability metrics (DBI, SC, and KNN-OS), which quantify complementary global and local discrimination properties of the feature space.

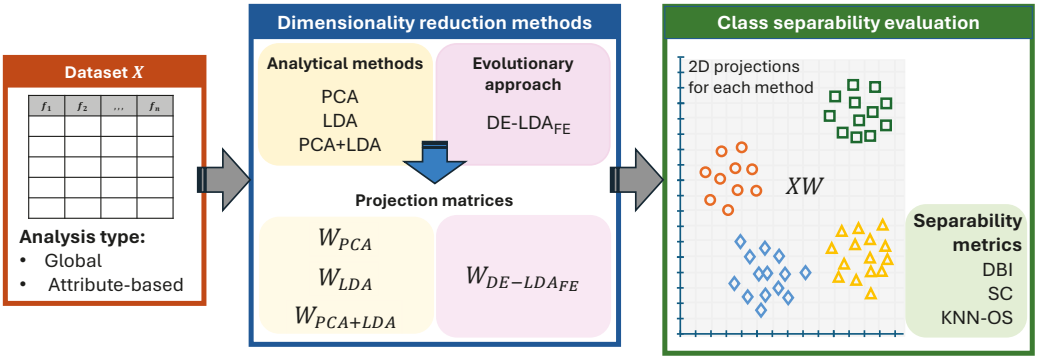


Figure 1. Experimental workflow of the proposed methodology. For each dataset (**X**), one analysis strategy (global or attribute-based) is selected. Dimensionality reduction techniques are then applied, including analytical methods (PCA, LDA, and PCA+LDA) and the proposed evolutionary approach (DE-LDA$_{FE}$), to obtain the corresponding projection matrices. The resulting two-dimensional projected spaces ($\boldsymbol{XW}$) are subsequently evaluated using class separability metrics (DBI, SC, and KNN-OS), enabling a comparative assessment of the discriminative quality of each method.

The separability results are analyzed and compared across dimensionality-reduction methods and analysis types (global and attribute-based). This evaluation framework enables not only the assessment of the effectiveness of each projection strategy, but also the characterization and differentiation of the various tortilla types based on their sensory profiles.

It is important to note that the dimensionality reduction methods considered in this study (PCA, LDA, PCA+LDA, and DE-LDA$_{FE}$) are used exclusively for feature projection. In particular, DE-LDA$_{FE}$ is designed to enhance class separability within the projection space while facilitating intuitive data interpretation and visualization. Consequently, no classifier training or prediction tasks were performed. The focus of this work is on the geometric organization of multivariate sensory data to support human analysis, rather than on the automated classification of new samples.

Within this context, Generalized Procrustes Analysis (GPA) was not included in the comparative framework. Although GPA is widely used in sensory studies to align panel configurations through consensus-based geometric transformations, the present

https://doi.org/10.3390/mca31030082

study specifically emphasizes supervised dimensionality reduction approaches aimed at maximizing discrimination between tortilla profiles.

2.5. Computational Framework and Visualization

All computational procedures, including the evolutionary optimization of the projection matrices, the calculation of separability metrics, and the generation of 2D discriminant plots, were performed using MATLAB R2023b.

3. Experiments

This section presents the experimental framework used to determine the most suitable feature-extraction method for analyzing sensory differences among nixtamal tortillas, commercial flour tortillas, and their respective fortifications with chilacayote powder.

For each sensory evaluation dataset (Xalapa and Tulancingo), the following experimental procedure was implemented:

1. Select the dataset according to the analysis type:
   (A) **Global analysis.** In the global analysis, the complete dataset was considered, including all variables corresponding to the sensory attributes evaluated in the tortillas. Thus, for this type of analysis, each dataset $\mathbf{X}$ is represented as a 300×62 matrix, where 300 denotes the number of samples and 62 the total number of sensory variables.
   (B) **Attribute-based analysis.** In the attribute-based analysis, subsets of the dataset were evaluated independently for each sensory attribute of the tortillas: color, smell, texture, taste, mouthfeel, and aftertaste. For each attribute, only the variables corresponding to that specific sensory dimension were selected, generating a reduced dataset $\mathbf{X}$ for analysis. The dimensionality of each attribute-specific dataset is detailed in Table 2.

Table 2. Dimensional structure of the datasets used in the attribute-based analysis.

Attribute	Number of Sensory Variables	Dataset Dimension
Color	8	300×8
Smell	10	300×10
Texture	13	300×13
Taste	12	300×12
Mouthfeel	12	300×12
Aftertaste	7	300×7

2. Apply the selected dataset to compute the projection matrices of the analytical dimensionality-reduction methods (PCA, LDA and PCA+LDA). The analysis was performed on the binary raw data matrix obtained from the CATA task, which was centered to shift the coordinate system to the data mean. Standardization was not applied, as all variables share the same binary scale (0 and 1), and preserving their original variance is essential to maintain the relative contribution of each sensory attribute. Since these approaches are deterministic, a single execution is sufficient to obtain the corresponding projection matrix for each method. Once the projected feature spaces are obtained, the class separability measures considered in this study (DBI, SC, and KNN-OS) are computed to quantitatively assess the discriminative quality of each projection.
   For the KNN-OS metric, the neighborhood size considered is $K = 5$, since it provides a strict local assessment of class mixing, particularly suitable for evaluating separability

https://doi.org/10.3390/mca31030082

in low-dimensional visualization spaces, ensuring a robust evaluation of local overlap effects in the projected feature spaces.

3. Apply the selected dataset to compute projection matrices using the evolutionary approach (DE-LDA$_{FE}$), following the search strategy described below:

   3.1. A grid search strategy was implemented by systematically combining different values of the Differential Evolution (DE) parameters: population size $N \in \{100, 200, 300, 1000, 2000, 3000\}$, number of generations $G \in \{200, 400, 600, 2000, 4000, 6000\}$, crossover rate $CR \in \{0.3, 0.5, 0.7\}$, and scale factor $F \in \{0.3, 0.5, 0.7\}$. The full factorial design resulted in a discrete search space, denoted as in Equation (30), with a total of 324 distinct parameter configurations.

$$\mathcal{P} = \{N\} \times \{G\} \times \{CR\} \times \{F\}. \tag{30}$$

   Although this grid search introduces a one-time computational cost during the calibration phase, the scalability of DE-LDA$_{FE}$ remains robust. Since the method projects data into a two-dimensional space, the optimization problem is defined over a space of size $m \times 2$, where m is the number of variables. This constraint keeps the search space tractable as m increases, ensuring computational feasibility for large-scale sensory analyses.

   3.2. For each parameter configuration $\theta \in \mathcal{P}$ the DE-LDA$_{FE}$ algorithm was executed to obtain an optimal projection matrix $\boldsymbol{W_\theta}$, and the corresponding projections $\boldsymbol{Y_\theta}$ of the dataset $\mathbf{X}$ are obtained with the calculation presented on Equation (2).

   3.3. For each projected space $\boldsymbol{Y_\theta}$, the separability metrics are computed: DBI$_\theta$, SC$_\theta$ and KNN-OS$_\theta$ with $K = 5$. For the KNN-OS calculation, $K = 5$ was used following the original authors' recommendations [32]. This value provides a balance between robustness to outliers and sensitivity to local class boundaries, ensuring a stable estimation of local overlap for the S_θ index.

   3.4. To ensure comparability, each metric was normalized using min-max scaling over all configurations, considering the optimal direction of each metric. Let $\theta \in \mathcal{P}$ denote a parameter configuration and let $m_\theta \in \mathcal{M}$ represent the value of a separability metric obtained under configuration θ, where $\mathcal{M} = \{m_\theta : \theta \in \mathcal{P}\}$ is the set of all metric values across the parameter space. The normalized metric $m_\theta^{(\text{norm})} \in [0, 1]$ is defined using min-max scaling according to the optimization direction:

   * If the metric is to be minimized,

$$m_\theta^{(\text{norm})} = \frac{\max\{\mathcal{M}\} - m_\theta}{\max\{\mathcal{M}\} - \min\{\mathcal{M}\}}. \tag{31}$$

   * If the metric is to be maximized,

$$m_\theta^{(\text{norm})} = \frac{m_\theta - \min\{\mathcal{M}\}}{\max\{\mathcal{M}\} - \min\{\mathcal{M}\}}. \tag{32}$$

   In this study, the Davies–Bouldin Index (DBI) and KNN-OS are minimized, whereas the Silhouette Coefficient (SC) is maximized. In this work, metrics to be minimized were inverted during normalization to ensure a consistent optimization direction (higher is better).

https://doi.org/10.3390/mca31030082

3.5. A global separability index S_θ was defined as the geometric mean [33] of the normalized metrics; see Equation (33).

$$S_\theta = (DBI_\theta^{\text{norm}} \cdot SC_\theta^{\text{norm}} \cdot KNN\text{-}OS_\theta^{\text{norm}})^{\frac{1}{3}} \tag{33}$$

The geometric mean was selected to combine DBI, SC, and KNN-OS into the S_θ index because it balances the contribution of the three metrics while penalizing configurations that perform poorly in any individual criterion. Unlike the arithmetic mean, which can mask deficiencies in specific metrics, this approach ensures a more robust and discriminative selection of projection matrices.

3.6. All the configurations $\theta \in \mathcal{P}$ were ranked in descending order according to S_θ, and the configuration with the highest value is selected as optimal candidate.

3.7. The parameter configuration θ^* achieving the highest S_θ value was selected for further analysis. Since DE-LDA$_{FE}$ is a stochastic optimization method, the projection process was repeated 10 independent times using θ^* in order to asses variability due to random initialization and evolutionary dynamics. Among these executions, the projection yielding the best separability performance was selected ans used for comparison with the analytical dimensionality reduction models.

4. After obtaining the projections from PCA, LDA, PCA+LDA, and DE-LDA$_{FE}$, a numerical and graphical comparison was performed. The evaluation was based on separability metrics, employing the global separability index S_θ, and on visualizations of the projected feature spaces to assess differences in class discrimination. In these representations, neutral X and Y labels were maintained for the axes to ensure a consistent Cartesian coordinate system across all methods. This choice avoids terminological inaccuracies, as the dimensions represent "Principal Components" in PCA but "Discriminant Dimensions" in the LDA, PCA+LDA, and DE-LDA$_{FE}$ methods. Additionally, small markers were intentionally used to preserve data density, enhance the visualization of cluster boundaries, and maintain the visibility of sensory attribute labels, ensuring that the distribution of individual panelist evaluations remains clear without overplotting.

4. Results

This section presents the experimental results obtained from applying the dimensionality-reduction methods (PCA, LDA, PCA+LDA, and DE-LDA$_{FE}$) to both datasets (Xalapa and Tulancingo) under the two analysis strategies considered: global and attribute-based.

4.1. Projection Performance and Separability Analysis of the Xalapa Dataset

This section presents the experimental results obtained from the Xalapa sensory evaluation dataset. The analysis is structured according to the two evaluation strategies previously defined: global analysis and attribute-based analysis. In both cases, the dimensionality-reduction methods (PCA, LDA, PCA+LDA, and DE-LDA$_{FE}$) are compared using the considered class-separability metrics to assess their discriminative capability in this dataset.

4.1.1. Xalapa Global Analysis

In the global analysis of the Xalapa dataset, all sensory variables were considered simultaneously, resulting in a 300×62 data matrix. This configuration evaluates the overall discriminative capacity of each dimensionality-reduction method using the complete sensory profile. The separability metrics provide a comprehensive assessment of how well the projected spaces preserve and enhance class discrimination across the different tortilla types.

https://doi.org/10.3390/mca31030082

Table 3 reports the minimum, maximum, mean, and standard deviation of the separability metrics obtained from the 10 independent executions of the best parameter configuration $\theta^* = (300, 400, 0.3, 0.7)$. The relatively low standard deviation values observed for DBI (0.1069), SC (0.0114), and KNN-OS (0.0138) indicate stable behavior of the evolutionary optimization process under the selected configuration. Although DE-LDA$_{FE}$ is a stochastic method, the variability across executions remains limited, suggesting robustness of the obtained projection subspace.

Table 3. Results obtained for the DE-LDA$_{FE}$ method with all the variables in the Xalapa dataset considering the parameters $\theta = (300, 400, 0.3, 0.7)$.

Separability Metric	Min	Max	Mean	Standard Deviation
DBI	1.4658	1.7744	1.6202	0.1069
SC	0.1244	0.1603	0.1428	0.0114
KNN-OS	0.4293	0.4800	0.4579	0.0138

Table 4 presents the comparative results between PCA, LDA, PCA+LDA, and the best DE-LDA$_{FE}$ projection. In terms of DBI (to be minimized), DE-LDA$_{FE}$ achieved the lowest values, indicating more compact and better-separated class clusters. Regarding the Silhouette Coefficient, DE-LDA$_{FE}$ obtained the highest value, reflecting improved intra-class cohesion and inter-class separation at the sample level. Similarly, for KNN-OS, DE-LDA$_{FE}$ produced the lowest overlap score, demonstrating reduced local class mixing in the projected feature space.

Table 4. Results obtained with all the variables in the Xalapa dataset.

Separability Metric	PCA	LDA	PCA+LDA	DE-LDA$_{FE}$
DBI	12.5579	1.7611	8.1516	**1.4976**
SC	-0.1013	0.1492	-0.0800	**0.1582**
KNN-OS	0.7453	0.4660	0.7153	**0.4293**
S_θ	0	0.9409	0.1459	**1**

Note: Bold values indicate the best performance obtained among the evaluated dimensionality reduction methods for each separability metric.

Figure 2 shows the graphical representations of the projected feature spaces obtained for the Xalapa global sensory dataset. Visual comparison enables qualitative evaluation of class compactness, inter-class separation, and local overlap generated by each dimensionality-reduction method. In this case, the projections obtained with LDA and DE-LDA$_{FE}$ exhibit a similar structural arrangement of class clusters, differing mainly by an apparent rotation of the discriminant axes. This similarity suggests that the intrinsic between-class structure is well captured by the Fisher criterion, while the evolutionary optimization refines the orientation of the projection space.

4.1.2. Xalapa Attribute-Based Analysis

In the attribute-based analysis, the Xalapa dataset was decomposed into independent subsets corresponding to each sensory attribute (color, smell, texture, taste, mouthfeel, and aftertaste). This approach allows the evaluation of the discriminative power of each sensory dimension separately. By analyzing the separability metrics for each attribute-specific projection, it is possible to identify which sensory attributes contribute most significantly to class differentiation.

https://doi.org/10.3390/mca31030082

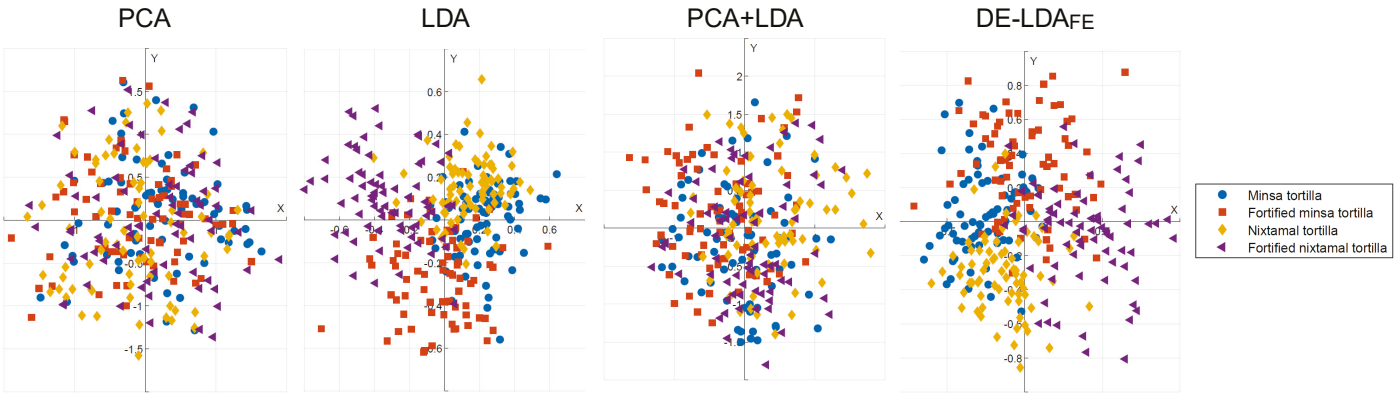


Figure 2. Two-dimensional projections of the Xalapa global sensory dataset generated by PCA, LDA, PCA+LDA, and DE-LDA$_{FE}$.

Table 5 summarizes the separability results obtained for each sensory attribute of the Xalapa dataset. In addition to the individual metrics (DBI, SC, and KNN-OS), the global index S_θ is reported. This index corresponds to the geometric mean of the three normalized metrics (see Equation (33)) and provides a balanced evaluation of global, semi-global, and local discrimination properties within each projected space.

Table 5. Results obtained for each attribute variable in the Xalapa dataset.

Attribute	Separability Metric	PCA	LDA	PCA+LDA	DE-LDA$_{FE}$
Color	DBI	4.3605	2.4262	3.8918	**2.4164**
	SC	−0.0675	0.0124	−0.0901	**0.0132**
	KNN-OS	0.5940	**0.5787**	0.5853	**0.5787**
	S_θ	0	0.9957	0	**1**
Smell	DBI	18.2311	**4.0800**	5.9459	4.1609
	SC	**−0.0873**	−0.1288	−0.1256	−0.1236
	KNN-OS	0.6987	**0.6747**	0.6827	0.6760
	S_θ	0	0	0.2046	**0.2829**
Texture	DBI	14.1031	**4.5378**	8.7000	4.7998
	SC	**−0.0737**	−0.1055	−0.1020	−0.1084
	KNN-OS	0.6753	0.6820	0.7120	**0.6653**
	S_θ	0	**0.3772**	0	0
Taste	DBI	13.9926	25.8575	**8.5632**	15.2000
	SC	**−0.0951**	−0.1300	−0.0959	−0.1139
	KNN-OS	0.6633	0.6527	0.6593	**0.6493**
	S_θ	0	0	0.6536	**0.6575**
Mouthfeel	DBI	10.6261	7.7731	10.5756	**6.4993**
	SC	**−0.0911**	−0.1018	−0.0920	−0.1047
	KNN-OS	0.6700	0.6633	0.6760	**0.6593**
	S_θ	0	**0.4822**	0	0
Aftertaste	DBI	15.8235	**4.9989**	9.0478	5.4470
	SC	−0.0719	−0.1276	**−0.0567**	−0.1306
	KNN-OS	**0.6960**	0.6993	0.7053	0.7020
	S_θ	0	**0.2970**	0	0

Note: Bold values indicate the best performance obtained among the evaluated dimensionality reduction methods for each separability metric.

https://doi.org/10.3390/mca31030082

The results in Table 5 show that the relative performance of the dimensionality-reduction methods varies across sensory attributes, indicating that class separability is strongly attribute-dependent.

For the *Color* attribute, DE-LDA$_{FE}$ achieves the highest global index ($S_\theta = 1$), slightly outperforming LDA. Although both methods produce similar DBI and KNN-OS values, the evolutionary approach yields a more balanced improvement across metrics, suggesting refined adjustment of the projection matrix when moderate local overlap is present. This suggests that evolutionary optimization provides a marginal but consistent improvement in overall separability when color variables are considered independently.

In the *Smell* attribute, although LDA attains the lowest DBI, DE-LDA$_{FE}$ yields the highest global index ($S_\theta = 0.2829$), reflecting a more balanced trade-off among dispersion, cohesion, and local overlap. This suggests that evolutionary optimization can redistribute discriminative emphasis beyond purely centroid-based separation.

For *Texture*, LDA obtains the best global score ($S_\theta = 0.3772$), indicating that, for this attribute, the analytical discriminant structure is already adequate and evolutionary optimization does not provide additional benefit. The comparable performance of DE-LDA$_{FE}$ indicates convergence toward a similar structural solution.

In the *Taste* attribute, DE-LDA$_{FE}$ produces the highest global index ($S_\theta = 0.6575$), mainly driven by improvements in local overlap (KNN-OS), even though PCA+LDA achieves the lowest DBI. The comparable performance of DE-LDA$_{FE}$ indicates convergence toward a similar structural solution.

For *Mouthfeel* and *Aftertaste*, LDA achieves the highest global scores, suggesting that these attributes exhibit a discriminant structure well captured by the classical Fisher-based projection, leaving limited margin for evolutionary refinement.

Overall, the attribute-based analysis demonstrates that while DE-LDA$_{FE}$ dominates in the global analysis, its relative advantage at the attribute level depends on the intrinsic separability structure of each sensory dimension. The method tends to provide greater benefit when class distributions deviate from ideal linear discriminant assumptions or when local mixing effects are significant. Conversely, when the Fisher criterion sufficiently captures the between-class structure, both analytical and evolutionary approaches yield comparable separability.

For visualization purposes, only the projection corresponding to the maximum S_θ value for each attribute is displayed in Figure 3, ensuring that graphical comparisons reflect the most structurally discriminative configuration identified for each sensory dimension.

In contrast to the global projections obtained from all sensory variables, the attribute-based projections for the Xalapa dataset exhibit a more irregular, overlapping structure. This behavior is expected, since each attribute subset contains fewer variables and therefore captures only partial information about the tortillas' sensory profile. Consequently, the discriminative structure becomes less defined in the two-dimensional space, leading to visually intertwined class regions. Nevertheless, despite this increased visual complexity, the displayed projections correspond to the configurations that maximize the global separability index S_θ, meaning they represent the most discriminative transformations achievable under the structural constraints of each individual sensory attribute.

4.2. Projection Performance and Separability Analysis of the Tulancingo Dataset

This section presents results from the Tulancingo sensory evaluation dataset, following the same experimental structure as for the Xalapa dataset. The objective is to assess the consistency and generalization of the dimensionality-reduction methods across different sensory evaluation conditions. Both global and attribute-based analyses are considered.

https://doi.org/10.3390/mca31030082

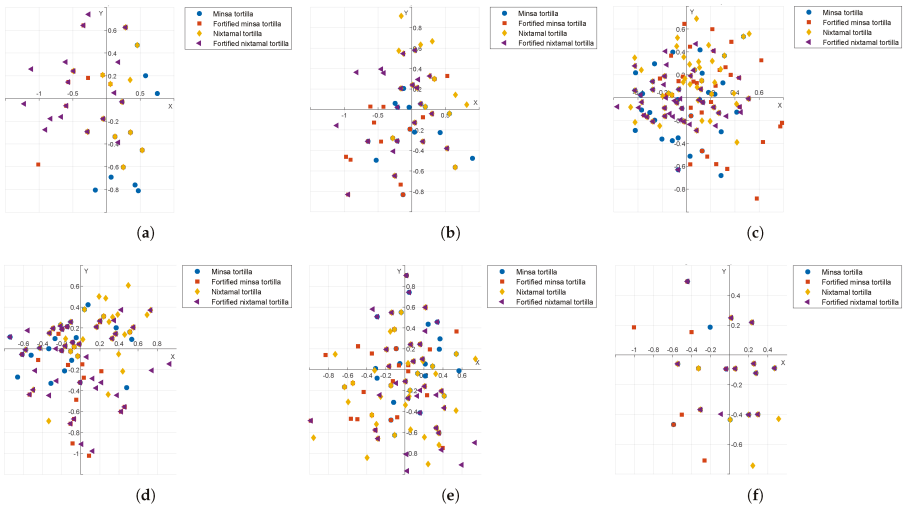


Figure 3. Two-dimensional projections for each sensory attribute in the Xalapa dataset, showing the method that achieved the maximum global separability index S_θ: (**a**) Color (DE-LDA$_{FE}$). (**b**) Smell (DE-LDA$_{FE}$). (**c**) Texture (LDA). (**d**) Taste (DE-LDA$_{FE}$). (**e**) Mouthfeel (LDA). (**f**) Aftertaste (LDA).

4.2.1. Tulancingo Global Analysis

For the global analysis of the Tulancingo dataset, all sensory variables were jointly analyzed, forming a 300×62 data matrix. This evaluation examines the overall separability achieved by each projection method when the complete sensory information is retained. Computed metrics quantify the compactness and discrimination of the resulting projected spaces.

Table 6 summarizes the separability performance obtained from the 10 independent executions of DE-LDA$_{FE}$ using the optimal parameter configuration $\theta = (300, 400, 0.3, 0.7)$ for the Tulancingo global dataset. The table reports the minimum, maximum, mean, and standard deviation of DBI, SC, and KNN-OS, allowing assessment of both projection quality and algorithmic stability. The relatively low standard deviation values across all metrics indicate that the evolutionary optimization process produces consistent projection matrices despite its stochastic nature, while the mean values confirm strong global compactness, positive sample-level cohesion, and reduced local class overlap in the projected space.

Table 7 presents the comparative separability results obtained in the global analysis of the Tulancingo dataset using PCA, LDA, PCA+LDA, and DE-LDA$_{FE}$.

Table 6. Results obtained for the DE-LDA$_{FE}$ method with all the variables in the Tulancingo dataset considering the parameters $\theta = (300, 400, 0.3, 0.7)$.

Separability Metric	Min	Max	Mean	Standard Deviation
DBI	1.5829	1.6973	1.6470	0.0448
SC	0.1835	0.2072	0.1932	0.0072
KNN-OS	0.3847	0.4507	0.4305	0.0191

Table 7. Results obtained with all the variables in the Tulancingo dataset.

Separability Metric	PCA	LDA	PCA+LDA	DE-LDA$_{FE}$
DBI	6.3433	-	3.0408	**1.5829**
SC	-0.0545	-	-0.0530	**0.2072**
KNN-OS	0.7007	-	0.6487	**0.3847**
S_θ	0	-	0.0868	**1**

Note: Bold values indicate the best performance obtained among the evaluated dimensionality reduction methods for each separability metric.

https://doi.org/10.3390/mca31030082

Classical LDA did not yield a valid solution in this case, likely due to the within-class scatter matrix being singular. Among the feasible methods, DE-LDA$_{FE}$ clearly achieves the best overall performance. It obtains the lowest DBI value (1.5829), indicating the best global compactness and inter-class separation. Additionally, it is the only method to produce a positive Silhouette Coefficient (0.2072), indicating improved sample-level cohesion and clearer class boundaries. In terms of local structure, DE-LDA$_{FE}$ also achieves the lowest KNN-OS value (0.3847), demonstrating reduced neighborhood-level class overlap.

These improvements are reflected in the global separability index S_θ, where DE-LDA$_{FE}$ has the maximum normalized value (1), outperforming PCA and PCA+LDA. Overall, the results show that evolutionary optimization provides a more discriminative projection than the analytical dimensionality-reduction approaches in the Tulancingo global sensory space.

Figure 4 presents the graphical representations of the projected feature spaces obtained for the Tulancingo global sensory dataset. The visual comparison highlights the differences in class compactness, inter-class separation, and local overlap produced by each dimensionality-reduction method. Consistent with the quantitative separability metrics, the DE-LDA$_{FE}$ projection exhibits more clearly defined class regions and reduced overlap, supporting its superior discriminative performance in the global analysis.

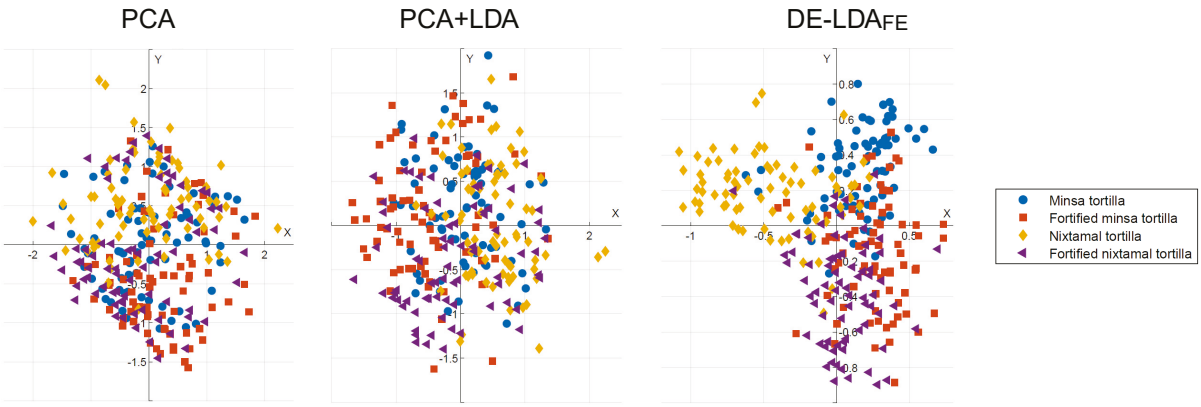


Figure 4. Two-dimensional projections of the Tulancingo global sensory dataset generated by PCA, PCA+LDA, and DE-LDA$_{FE}$.

4.2.2. Tulancingo Attribute-Based Analysis

In the attribute-based analysis, each sensory attribute of the Tulancingo dataset was examined independently. This allows the identification of attribute-specific discriminative patterns and enables a detailed comparison of the projection methods within each sensory dimension. The separability metrics provide insight into the relative contribution of each attribute to class differentiation.

Table 8 presents the separability results for each sensory attribute of the Tulancingo dataset. As in the previous analysis, the global index S_θ corresponds to the geometric mean of the normalized DBI, SC, and KNN-OS values, integrating global cluster structure, semi-global cohesion, and local neighborhood consistency into a single separability indicator.

https://doi.org/10.3390/mca31030082

Table 8. Results obtained for each attribute variable in the Tulancingo dataset.

Attribute	Separability Metric	PCA	LDA	PCA+LDA	DE-LDA$_{FE}$
Color	DBI	4.9046	**2.0018**	4.1952	2.0233
	SC	−0.0034	**0.0357**	−0.0336	0.0321
	KNN-OS	0.5373	0.5260	0.5467	**0.5253**
	S_θ	0	**0.9890**	0	0.9799
Smell	DBI	37.6213	-	22.4539	**8.5606**
	SC	**−0.0861**	-	−0.0883	−0.1407
	KNN-OS	0.6553	-	0.6793	**0.6340**
	S_θ	0	-	**0.9864**	0
Texture	DBI	9.3679	-	7.8502	**4.0530**
	SC	−0.1027	-	**−0.0893**	−0.0946
	KNN-OS	0.6713	-	0.6813	**0.6553**
	S_θ	0	-	0	**0.8455**
Taste	DBI	10.4742	4.8688	10.8196	**4.4773**
	SC	−0.1005	−0.1272	**−0.0891**	−0.1261
	KNN-OS	0.6633	0.6407	0.6407	**0.6320**
	S_θ	0	0	0	**0.3068**
Mouthfeel	DBI	12.7569	4.8261	8.9165	**4.7643**
	SC	−0.0969	−0.0847	−0.0936	**−0.0833**
	KNN-OS	0.6767	0.6687	0.6673	**0.6660**
	S_θ	0	0.8731	0.4679	**1**
Aftertaste	DBI	11.4279	4.7456	6.0643	**4.1373**
	SC	**−0.0763**	−0.1604	−0.0829	−0.1274
	KNN-OS	0.6833	0.6747	0.6767	**0.6740**
	S_θ	0	0	**0.7836**	0.7321

Note: Bold values indicate the best performance obtained among the evaluated dimensionality reduction methods for each separability metric.

The results in Table 8 again demonstrate that discriminative behavior is attribute-dependent, although the structural patterns differ from those observed in the Xalapa dataset.

For the *Color* attribute, LDA achieves the highest global index ($S_\theta = 0.9890$), with DE-LDA$_{FE}$ yielding a very close value ($S_\theta = 0.9799$). Both methods produce similar DBI and KNN-OS values, indicating that the between-class scatter structure is well captured by the analytical Fisher solution. The marginal difference suggests structural alignment between the intrinsic data geometry and the linear discriminant assumption.

In the *Smell* attribute, LDA does not provide a valid solution, highlighting potential singularity or instability in the within-class scatter matrix. In this case, DE-LDA$_{FE}$ becomes particularly relevant, producing the lowest DBI and KNN-OS values among the feasible methods. Although PCA+LDA attains the highest S_θ, the evolutionary approach yields the strongest global compactness and reduced local overlap, emphasizing its robustness when classical LDA is not directly applicable.

For *Texture*, DE-LDA$_{FE}$ achieves the highest global index ($S_\theta = 0.8455$), improving in DBI and reducing local mixing. This indicates that evolutionary optimization effectively adjusts the projection matrix to better accommodate attribute-specific class distributions.

In the *Taste* attribute, DE-LDA$_{FE}$ obtains the highest S_θ value. Although SC remains negative for all methods, suggesting moderate global overlap, the evolutionary approach minimizes DBI and KNN-OS, thereby improving both global compactness and local consistency.

For *Mouthfeel*, DE-LDA$_{FE}$ achieves the maximum global index ($S_\theta = 1$), reflecting simultaneous improvements across global and local separability measures. This suggests that the discriminative structure of this attribute benefits from optimization beyond the closed-form Fisher solution.

Finally, in *Aftertaste*, PCA+LDA obtains the highest S_θ, although DE-LDA$_{FE}$ provides the lowest DBI and competitive KNN-OS values. The difference indicates that, while evolu-

https://doi.org/10.3390/mca31030082

tionary optimization enhances global compactness, the semi-global cohesion measured by SC slightly favors the analytical approach.

Overall, the Tulancingo attribute-based analysis highlights two important aspects:

(1) DE-LDA$_{FE}$ provides clear advantages in attributes where classical LDA is unstable or where local overlap is pronounced (Smell, Texture, Mouthfeel), and
(2) when the Fisher discriminant assumption is structurally adequate (Color), analytical and evolutionary approaches converge toward comparable separability levels.

As in the Xalapa dataset, only the projection corresponding to the maximum S_θ value for each attribute is displayed in Figure 5, ensuring that the visualization reflects the most discriminative configuration identified for each sensory dimension.

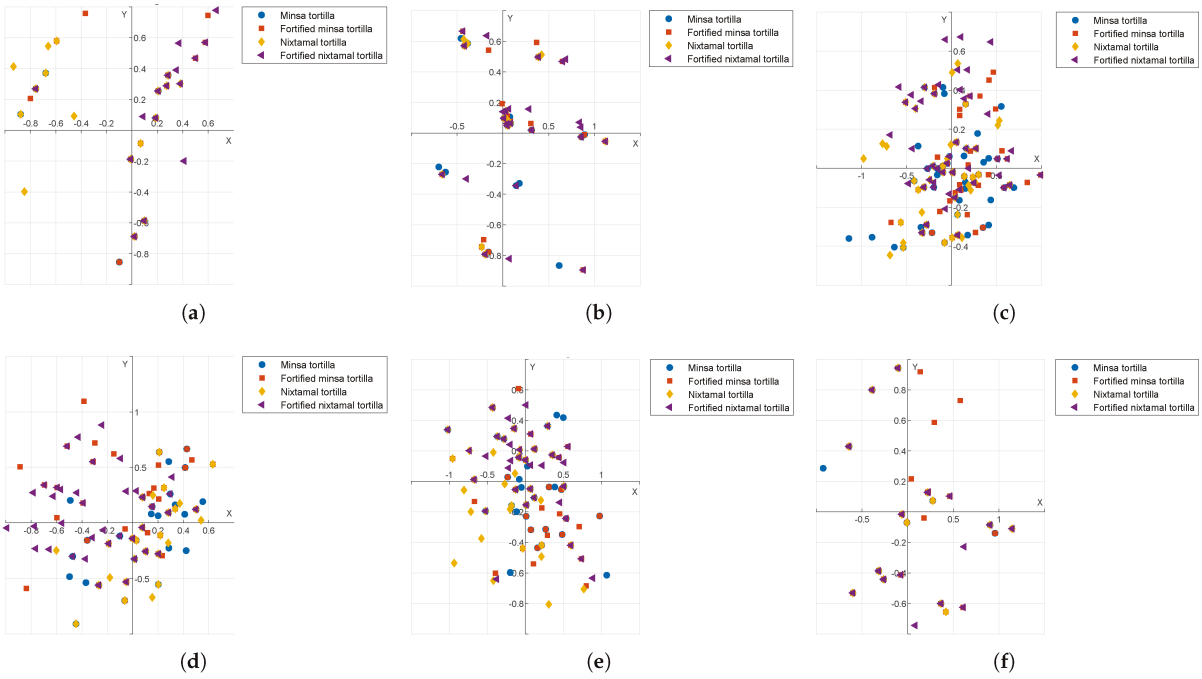


Figure 5. Two-dimensional projections for each sensory attribute in the Tulancingo dataset, showing the method that achieved the maximum global separability index S_θ: (**a**) Color (LDA). (**b**) Smell (PCA+LDA). (**c**) Texture (DE-LDA$_{FE}$). (**d**) Taste (DE-LDA$_{FE}$). (**e**) Mouthfeel (DE-LDA$_{FE}$). (**f**) Aftertaste (PCA+LDA).

As observed in the Xalapa attribute-based projections, the Tulancingo projections derived from individual sensory attributes present a more compact, mixed visual distribution than the global analysis. The reduction in the amount of dimensional information per attribute limits the extent to which inter-class differences can be linearly emphasized in a two-dimensional space. Although the graphical separation appears less pronounced, the selected projections correspond to the highest S_θ values for each attribute. Therefore, they still represent the optimal discriminative configurations available for the given sensory dimension, preserving the maximum achievable balance between global compactness and local separability.

4.3. Comparative Analysis Between Xalapa and Tulancingo Datasets

A cross-dataset comparison shows that separability is not only attribute-dependent but also context-dependent, varying across the Xalapa and Tulancingo sensory information. Although both datasets share an identical dimensional structure, their intrinsic class geometries differ, leading to variations in the relative performance of the projection methods.

In the global analysis, DE-LDA$_{FE}$ consistently achieved superior separability performance in both datasets, indicating that when the complete sensory profile is considered,

https://doi.org/10.3390/mca31030082

evolutionary optimization better balances global compactness (DBI), semi-global cohesion (SC), and local overlap (KNN-OS). This suggests that the joint sensory space contains structural complexities that are not fully captured by closed-form analytical projections.

In the attribute-based analysis, the behavior becomes more heterogeneous. For Xalapa, DE-LDA$_{FE}$ demonstrated clear advantages in Color and Taste, while LDA remained competitive in Texture, Mouthfeel, and Aftertaste. In contrast, for Tulancingo, DE-LDA$_{FE}$ has improvements in Texture, Taste, and Mouthfeel, and showed particular robustness in Smell, where classical LDA failed to produce a solution.

These differences indicate that the discriminant structure of the sensory attributes is influenced by regional variability or product-specific characteristics. In particular, DE-LDA$_{FE}$ proved especially advantageous in scenarios where

- The within-class scatter matrix is ill-conditioned (e.g., Smell in Tulancingo);
- Local class mixing is pronounced;
- The Fisher criterion does not fully capture the discriminant manifold.

Conversely, when the intrinsic data geometry aligns well with the assumptions of linear discriminant analysis, analytical and evolutionary approaches converge toward comparable separability levels.

5. Identification of Sensory Profiles Through Discriminative Projection Analysis

Once the most discriminative projection spaces are determined for each dataset and analysis strategy, the next step is to identify the sensory profiles associated with each tortilla type. To achieve this, the projection matrices obtained from the best-performing models were analyzed. Each projection matrix contains the coefficients that weight the contributions of each sensory variable to the reduced two-dimensional space. These coefficients quantify the relative importance of each attribute in defining the projection's discriminative structure.

By considering the rows of the projection matrix and selecting the first two columns (corresponding to the two-dimensional subspace), each sensory variable can be represented as a point in $\mathbb{R}^2$. Graphically, this representation provides a directional interpretation: variables with similar orientations contribute similarly to class discrimination, while opposing directions indicate contrasting sensory influences. This geometric interpretation enables the identification of dominant sensory attributes associated with specific tortilla groups.

It is important to note that this analysis is not feasible in global projections. When all 62 variables are considered simultaneously, the resulting projection coefficients tend to be very small due to the higher dimensionality and the distributed variance across many attributes. Consequently, individual variable contributions become diluted, preventing a clear directional interpretation. For this reason, sensory profile identification is conducted exclusively using attribute-based projections, where the reduced-dimensional structure allows a more meaningful interpretation of variable influence.

It should be noted that, although PCA+LDA is included for performance benchmarking, it is not considered in the sensory profile interpretation. This is because PCA+LDA involves a two-stage transformation in which the final discriminant dimensions are expressed as linear combinations of principal components, which are themselves linear combinations of the original variables. This nested structure limits the direct interpretability of the results in terms of the original sensory descriptors. In contrast, PCA, LDA, and DE-LDA$_{FE}$ each produce a single projection matrix, where the coefficients explicitly represent the contribution of the original sensory attributes. This enables a more direct, transparent, and geometrically interpretable link between the mathematical representation and the sensory characterization.

https://doi.org/10.3390/mca31030082

5.1. Graphical Representation and Interpretation

To generate the graphical representations used for sensory-feature analysis, the best-discriminative projection for each attribute was employed. Given that class distributions remain partially overlapped even in the optimal projection space, the centroid (mean vector) of each class was first computed in the two-dimensional projected space. These centroids provide a stable representative location for each tortilla type, reducing the influence of intra-class dispersion.

Subsequently, the first two columns of the corresponding projection matrix $\mathbf{W} \in \mathbb{R}^{d\times 2}$ were analyzed. Each row $\boldsymbol{w}_i = (w_{i1}, w_{i2})$ represents the projection weights associated with the i-th sensory variable, so d is the number of sensory variables. These rows were plotted as points in $\mathbb{R}^2$, where each coordinate pair reflects the contribution of a specific attribute feature (e.g., brown within the color dimension) to the discriminative subspace.

The resulting visualization contains two types of elements:

- the four class centroids;
- the projected feature-weight vectors for all variables within the attribute.

Sensory features were then associated with specific tortilla types by evaluating the Euclidean distance between each feature point and the class centroids. Features located closer to a given centroid were interpreted as contributing more strongly to the sensory profile of that tortilla class.

In the following sections, the graphical representations of each sensory attribute are presented to characterize and interpret the sensory profiles associated with the different tortilla classes.

5.1.1. Visualization and Sensory Profile Characterization of the Xalapa Dataset

In the following sections, the graphical representations and corresponding interpretations obtained for each sensory attribute in the Xalapa dataset are presented. The corresponding graphical representations are displayed in Figure 6.

Color

Analysis of the color attribute in the Xalapa dataset shows distinct chromatic patterns associated with each tortilla type. Based on the Euclidean proximity between feature-weight vectors and class centroids in the discriminative projection space, specific color descriptors were linked to each class; see Figure 6a and Table 9. These associations reflect the dominant visual characteristics contributing to the sensory identity of each tortilla type.

Lighter tonalities, such as cream and beige, are predominantly associated with the commercial Minsa tortilla, while yellow is characteristic of the traditional nixtamal tortilla. Fortified variants exhibit darker or more saturated hues: sand and camel are primarily associated with the fortified Minsa tortilla, whereas brown and coffee are more closely related to the fortified nixtamal tortilla. The descriptor 'other' also appears to be associated with the Minsa tortilla, suggesting broader variability in its perceived color profile.

https://doi.org/10.3390/mca31030082

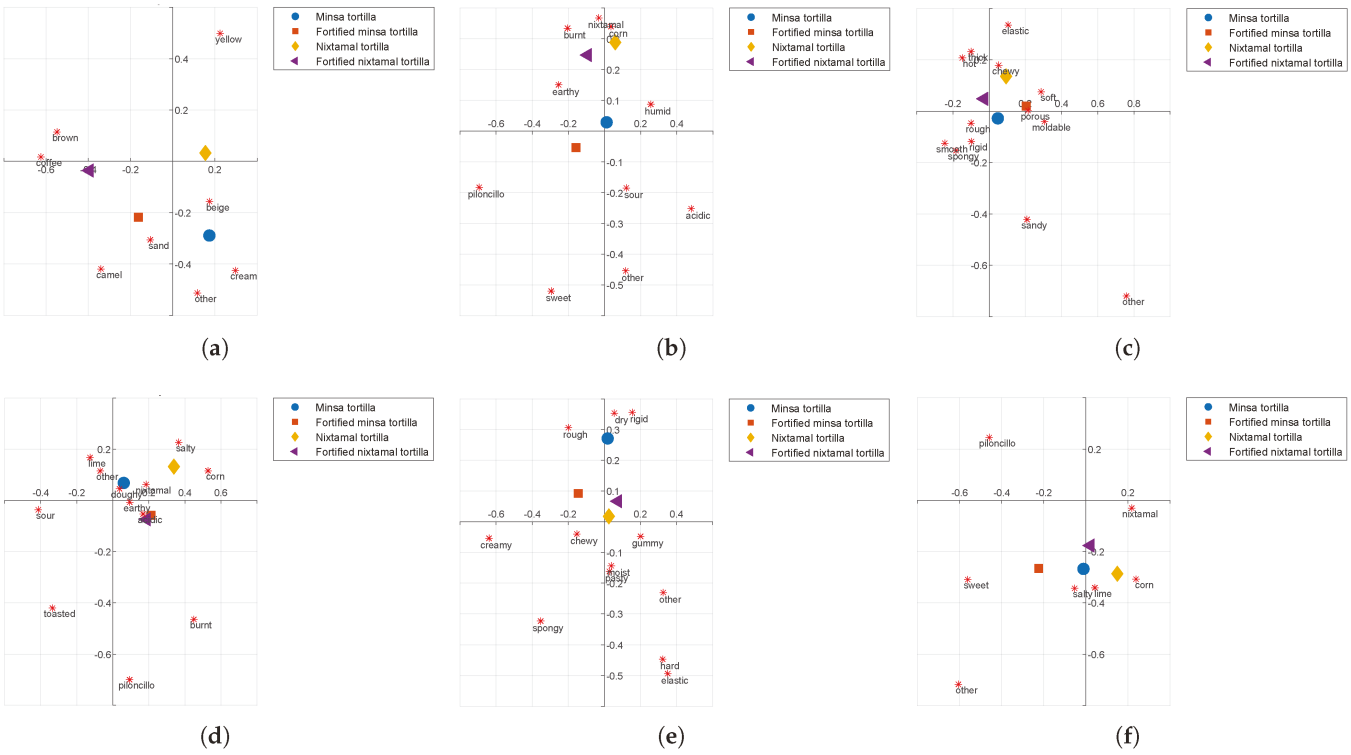


Figure 6. Graphical representation of the attributes in the Xalapa dataset, showing class centroids and projection-weight vectors of individual features in the discriminative two-dimensional space: (**a**) Color (DE-LDA$_{FE}$). (**b**) Smell (DE-LDA$_{FE}$). (**c**) Texture (LDA). (**d**) Taste (DE-LDA$_{FE}$). (**e**) Mouthfeel (LDA). (**f**) Aftertaste (LDA).

Table 9. Structured associations between tortilla classes and color descriptors in the Xalapa dataset. The discriminative projection space was generated with the DE-LDA$_{FE}$ method.

Class	Descriptors
Minsa tortilla	cream, beige, other
Nixtamal tortilla	yellow
Fortified minsa tortilla	sand, camel
Fortified nixtamal tortilla	brown, coffee

Smell

The olfactory profile obtained from the discriminative projection shows distinct aroma patterns across tortilla types in the Xalapa dataset. By analyzing the Euclidean proximity between sensory feature-weight vectors and class centroids, characteristic odor descriptors were identified for each tortilla class; see Figure 6b and Table 10.

The traditional nixtamal tortilla is characterized primarily by nixtamal and corn flavor, reflecting its maize-based processing. In contrast, the commercial Minsa tortilla is associated with acidic, sour, and humid descriptors, suggesting a differentiated fermentation or moisture-related perception. The fortified nixtamal tortilla offers deeper, more intense aromatic nuances, notably burnt and earthy notes. Meanwhile, the fortified Minsa tortilla is distinguished by sweeter tonalities, such as piloncillo and 'other', indicating additional aroma variability introduced by fortification.

These associations highlight how fortification and processing type influence the aromatic sensory structure of the tortillas.

https://doi.org/10.3390/mca31030082

Table 10. Structured associations between tortilla classes and smell descriptors in the Xalapa dataset. The discriminative projection space was generated with the DE-LDA$_{FE}$ method.

Class	Features
Minsa tortilla	acidic, sour, humid
Nixtamal tortilla	nixtamal, corn
Fortified minsa tortilla	sweet, piloncillo, other
Fortified nixtamal tortilla	burnt, earthy

Texture

The texture profile in the Xalapa dataset shows clear differences among tortilla types (see Figure 6c and Table 11), reflecting variations in structural and mechanical perception.

The fortified Minsa tortilla is characterized by porous, soft, moldable, and other descriptors, suggesting a more compliant, aerated texture. In contrast, the fortified nixtamal tortilla is associated with smooth, hot, rough, thick, and spongy attributes, suggesting a more heterogeneous, structurally complex mouthfeel.

The traditional nixtamal tortilla is primarily associated with chewy and elastic properties, consistent with a cohesive dough structure. Meanwhile, the commercial Minsa tortilla is characterized by sandy and rigid descriptors, reflecting a firmer and comparatively less elastic texture.

Table 11. Structured associations between tortilla classes and texture descriptors in the Xalapa dataset. The discriminative projection space was generated with the LDA method.

Class	Features
Minsa tortilla	sandy, rigid
Nixtamal tortilla	chewy, elastic
Fortified minsa tortilla	porous, soft, moldable, other
Fortified nixtamal tortilla	smooth, hot, rough, thick, spongy

Taste

The taste attribute in the Xalapa dataset shows distinctive flavor patterns across tortilla types; see Figure 6d and Table 12.

The traditional nixtamal tortilla is characterized by salty, corn notes, consistent with its maize-based preparation. The commercial Minsa tortilla presents a more heterogeneous flavor profile, with sour, lime, earthy, doughy, and other descriptors, suggesting perceptible acidity and less pronounced toasting.

The fortified nixtamal tortilla exhibits more intense and complex flavor attributes, including acidic, burnt, toasted, and piloncillo, indicating that fortification contributes to deeper roasted and sweet undertones. In contrast, the fortified Minsa tortilla is mainly associated with the descriptor nixtamal, suggesting a shift toward more traditional flavor perception after fortification.

Overall, the taste results highlight how fortification modifies flavor balance and introduces additional sensory nuances across tortilla types.

Table 12. Structured associations between tortilla classes and taste descriptors in the Xalapa dataset. The discriminative projection space was generated with the DE-LDA$_{FE}$ method.

Class	Features
Minsa tortilla	sour, lime, earthy, doughy, other
Nixtamal tortilla	salty, corn
Fortified minsa tortilla	nixtamal
Fortified nixtamal tortilla	acidic, burnt, toasted, piloncillo

https://doi.org/10.3390/mca31030082

Mouthfeel

The mouthfeel attribute reveals differentiated tactile sensations perceived during mastication across the tortilla types in the Xalapa dataset; see Figure 6e and Table 13.

The traditional nixtamal tortilla is mainly associated with elastic, hard, moist, and pasty descriptors, suggesting a cohesive internal structure, perceptible firmness, and humidity. In contrast, the commercial Minsa tortilla is characterized by a rigid, rough, and dry texture, reflecting a firmer, less hydrated perception during consumption.

The fortified Minsa tortilla exhibits a softer, more cohesive mouthfeel, with creamy, spongy, and chewy attributes. Meanwhile, the fortified nixtamal tortilla is linked to gummy and other descriptors, indicating modifications in internal consistency and residual oral texture due to fortification.

These results suggest that fortification influences not only structural texture but also the dynamic oral sensations experienced during chewing.

Table 13. Structured associations between tortilla classes and mouthfeel descriptors in the Xalapa dataset. The discriminative projection space was generated with the LDA method.

Class	Features
Minsa tortilla	rigid, rough, dry
Nixtamal tortilla	elastic, hard, moist, pasty
Fortified minsa tortilla	creamy, spongy, chewy
Fortified nixtamal tortilla	gummy, other

Aftertaste

The aftertaste attribute highlights persistent flavor sensations perceived after consumption, showing distinctive residual profiles across tortilla types; see Figure 6f and Table 14.

The traditional nixtamal tortilla is mainly characterized by a corn aftertaste, reinforcing its maize-based identity. The commercial Minsa tortilla shows salty and lime residual notes, suggesting a more pronounced perception of salinity and acidity after mastication.

The fortified Minsa tortilla exhibits sweeter lingering sensations, linked to sweet, piloncillo, and other descriptors, indicating that fortification enhances sweet tonalities in the residual flavor. In contrast, the fortified nixtamal tortilla is associated with a nixtamal aftertaste, suggesting reinforcement of traditional processing notes in the final sensory perception.

These results indicate that fortification not only modifies immediate taste perception but also influences the persistence and balance of residual flavors.

Table 14. Structured associations between tortilla classes and aftertaste descriptors in the Xalapa dataset. The discriminative projection space was generated with the LDA method.

Class	Features
Minsa tortilla	salty, lime
Nixtamal tortilla	corn
Fortified minsa tortilla	sweet, piloncillo, other
Fortified nixtamal tortilla	nixtamal

Overall, the attribute-based analysis of the Xalapa dataset reveals well-defined sensory tendencies that differentiate the four tortilla types across visual, aromatic, textural, gustatory, and residual dimensions. Traditional nixtamal tortillas are consistently associated with characteristic maize-related descriptors, reflecting preservation of their conventional sensory identity. In contrast, commercial Minsa tortillas exhibit attributes linked to acidity, rigidity, and lighter tonalities, suggesting a distinct processing profile. Fortification

https://doi.org/10.3390/mca31030082

introduces noticeable modifications in several attributes, particularly enhancing sweetness, roasted notes, and structural softness in the fortified variants. Collectively, these patterns confirm that both processing method and fortification significantly shape the tortillas' multidimensional sensory profile in the Xalapa dataset.

5.1.2. Visualization and Sensory Profile Characterization of the Tulancingo Dataset

In the following sections, the graphical representations and corresponding interpretations obtained for each sensory attribute in the Tulancingo dataset are presented. The corresponding graphical representations are displayed in Figure 7. Here, it is important to mention that although PCA+LDA obtained the highest separability index S_θ for smell and aftertaste, it was not used for sensory profile characterization because its projection matrix results from a composition of transformations, limiting direct interpretability in terms of the original sensory variables. Since profile identification requires explicit geometric interpretation of projection weights, the method with the second-highest S_θ value was selected. In both cases, DE-LDA$_{FE}$ offered a suitable alternative, maintaining strong discriminative performance while allowing interpretable sensory characterization.

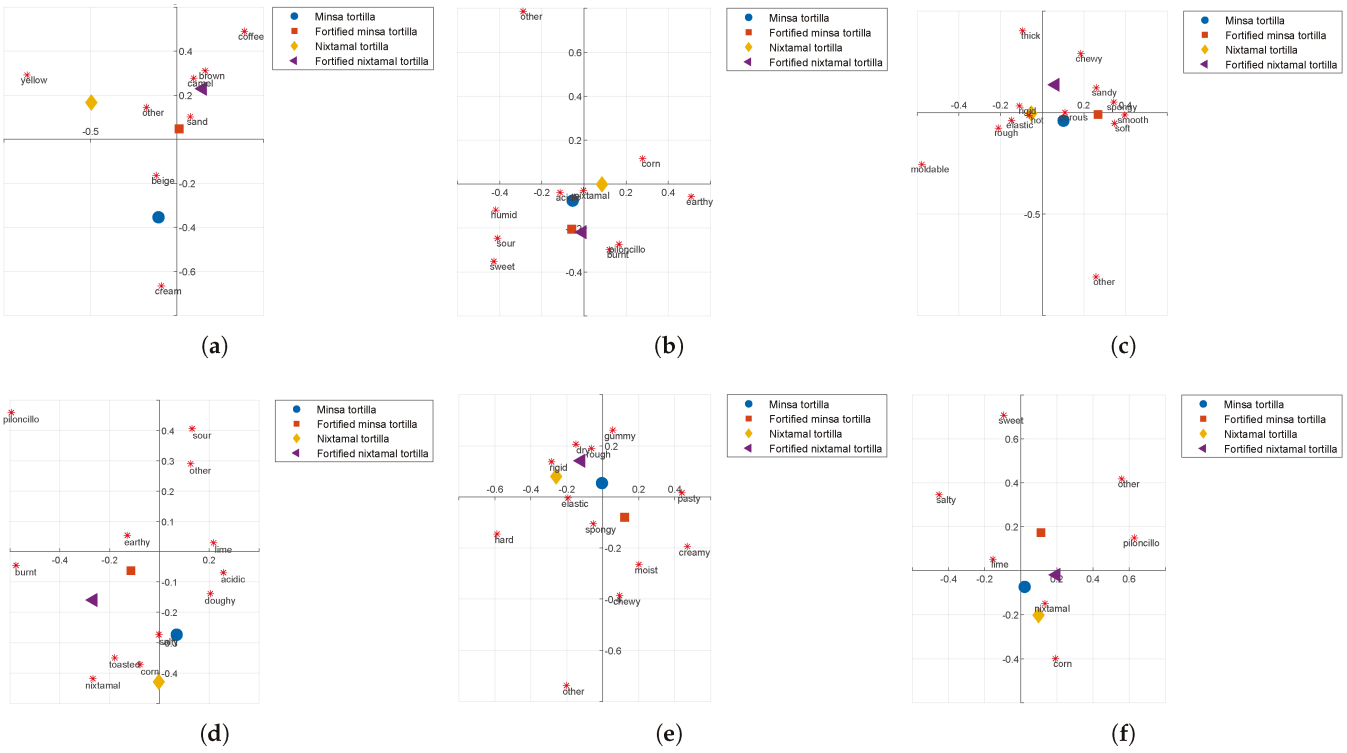


Figure 7. Graphical representation of the attributes in the Xalapa dataset, showing class centroids and projection-weight vectors of individual features in the discriminative two-dimensional space: (**a**) Color (LDA). (**b**) Smell (DE-LDA$_{FE}$ instead of PCA+LDA). (**c**) Texture (DE-LDA$_{FE}$). (**d**) Taste (DE-LDA$_{FE}$). (**e**) Mouthfeel (DE-LDA$_{FE}$). (**f**) Aftertaste (DE-LDA$_{FE}$ instead of PCA+LDA).

Color

The color attribute in the Tulancingo dataset shows differentiated chromatic patterns across tortilla types (see Figure 7a and Table 15), reflecting variations associated with processing method and fortification. The commercial Minsa tortilla is characterized by lighter tonalities, such as cream and beige, suggesting a relatively uniform, pale appearance. The traditional nixtamal tortilla is mainly associated with yellow color, consistent with the natural pigmentation imparted by nixtamalization.

Fortification introduces noticeable chromatic modifications. The fortified Minsa tortilla is linked to sand and other descriptors, indicating subtle tonal shifts compared to its

https://doi.org/10.3390/mca31030082

non-fortified counterpart. Meanwhile, the fortified nixtamal tortilla exhibits stronger associations with darker hues, including brown, camel, and coffee, suggesting increased color depth and intensity.

Overall, the results indicate that fortification affects color perception more markedly in the nixtamal-based tortillas, while commercial Minsa tortillas maintain lighter visual characteristics.

Table 15. Structured associations between tortilla classes and color descriptors in the Tulancingo dataset. The discriminative projection space was generated with the LDA method.

Class	Features
Minsa tortilla	cream, beige
Nixtamal tortilla	yellow
Fortified minsa tortilla	sand, other
Fortified nixtamal tortilla	brown, camel, coffee

Smell

The olfactory profile in the Tulancingo dataset presents distinctive aromatic associations across tortilla types (see Figure 7b and Table 16), revealing differences influenced by both processing and fortification.

The commercial Minsa tortilla is characterized by nixtamal, acidic, and humid descriptors, indicating perceptible fermentation-like and moisture-related notes. The traditional nixtamal tortilla is primarily associated with corn, earthy, and other aromas, reflecting a more natural maize-based and grounded aromatic profile.

The fortified Minsa tortilla exhibits sweeter and slightly fermented characteristics, linked to sour and sweet descriptors. In contrast, the fortified nixtamal tortilla is associated with deeper and more intense aromatic notes, including burnt and piloncillo, suggesting the presence of roasted and sweet undertones introduced through fortification.

Overall, the smell attribute indicates that fortification modifies aromatic balance differently depending on the base tortilla type, enhancing sweetness in the Minsa variant and intensifying roasted–sweet nuances in the nixtamal-based tortilla.

Table 16. Structured associations between tortilla classes and smell descriptors in the Tulancingo dataset. The discriminative projection space was generated with the DE-LDA$_{FE}$ method.

Class	Features
Minsa tortilla	nixtamal, acidic, humid
Nixtamal tortilla	corn, earthy, other
Fortified minsa tortilla	sour, sweet
Fortified nixtamal tortilla	burnt, piloncillo

Texture

The texture attribute in the Tulancingo dataset reflects differentiated structural and mechanical characteristics across tortilla types; see Figure 7c and Table 17.

The commercial Minsa tortilla is associated with porous and other descriptors, suggesting a relatively open structure with less distinctive mechanical features. The traditional nixtamal tortilla exhibits a firmer, more structured profile, characterized by hot, rough, elastic, rigid, and moldable attributes, indicating cohesiveness and perceptible surface irregularity.

The fortified Minsa tortilla presents a softer, more aerated profile, associated with soft, smooth, sandy, and spongy descriptors, suggesting that fortification modifies the internal structure and surface perception. In contrast, the fortified nixtamal tortilla exhibits chewy, thick attributes, reflecting increased density and resistance during mastication.

https://doi.org/10.3390/mca31030082

Overall, the texture results indicate that fortification alters mechanical perception differently depending on the base formulation, softening and aerating the Minsa variant while reinforcing density and chewiness in the nixtamal-based tortilla.

Table 17. Structured associations between tortilla classes and texture descriptors in the Tulancingo dataset. The discriminative projection space was generated with the DE-LDA$_{FE}$ method.

Class	Features
Minsa tortilla	porous, other
Nixtamal tortilla	hot, rough, elastic, rigid, moldable
Fortified minsa tortilla	soft, smooth, sandy, spongy
Fortified nixtamal tortilla	chewy, thick

Taste

The taste attribute in the Tulancingo dataset reveals differentiated flavor profiles shaped by processing method and fortification; see Figure 7d and Table 18.

The commercial Minsa tortilla is characterized by salty, acidic, lime, and doughy descriptors, suggesting a perceptible balance between salinity, acidity, and cereal-based notes. The traditional nixtamal tortilla is associated with corn and toasted flavors, reflecting its characteristic maize identity and thermal processing notes.

The fortified Minsa tortilla presents a more complex flavor structure, linked to sour, earthy, and other descriptors, indicating modifications in acidity and depth after fortification. In contrast, the fortified nixtamal tortilla is associated with more intense and layered notes, including nixtamal, burnt, and piloncillo, suggesting reinforcement of traditional processing characteristics combined with sweeter undertones. Overall, the taste profile in Tulancingo indicates that fortification enhances flavor complexity in both formulations, although the specific sensory shifts differ between Minsa- and nixtamal-based tortillas.

Table 18. Structured associations between tortilla classes and taste descriptors in the Tulancingo dataset. The discriminative projection space was generated with the DE-LDA$_{FE}$ method.

Class	Features
Minsa tortilla	salty, acidic, lime, doughy
Nixtamal tortilla	corn, toasted
Fortified minsa tortilla	sour, earthy, other
Fortified nixtamal tortilla	nixtamal, burnt, piloncillo

Mouthfeel

The mouthfeel attribute in the Tulancingo dataset highlights clear differences in oral texture and structural perception among tortilla types; see Figure 7e and Table 19. The traditional nixtamal tortilla is primarily characterized by elastic, rigid, and hard sensations, indicating a firm and cohesive internal structure during mastication. In contrast, the commercial Minsa tortilla is mainly associated with the descriptor 'spongy,' suggesting a lighter, more aerated oral texture.

Fortification introduces distinct modifications depending on the base formulation. The fortified Minsa tortilla is characterized by moist, creamy, pasty, chewy, and other descriptors, reflecting a softer, more cohesive, and more hydrated mouthfeel. Meanwhile, the fortified nixtamal tortilla is associated with rough, dry, and gummy sensations, indicating increased resistance and residual oral adhesion.

Overall, the mouthfeel results suggest that fortification softens and enhances cohesiveness in the Minsa variant, while it reinforces density and residual texture perception in the nixtamal-based tortilla.

https://doi.org/10.3390/mca31030082

Table 19. Structured associations between tortilla classes and mouthfeel descriptors in the Tulancingo dataset. The discriminative projection space was generated with the DE-LDA$_{FE}$ method.

Class	Features
Minsa tortilla	spongy
Nixtamal tortilla	elastic, rigid, hard
Fortified minsa tortilla	moist, creamy, pasty, chewy, other
Fortified nixtamal tortilla	rough, dry, gummy

Aftertaste

The aftertaste attribute in the Tulancingo dataset reveals differentiated residual flavor patterns across tortilla types (see Figure 7f and Table 20), highlighting how processing and fortification influence persistent sensory perception.

The traditional nixtamal tortilla is characterized by corn and nixtamal descriptors, reinforcing its maize-based identity even after swallowing. The commercial Minsa tortilla shows a lime aftertaste, indicating perceptible residual acidity.

Fortification produces distinct modifications in both formulations. The fortified Minsa tortilla exhibits a combination of sweet, salty, and other residual notes, suggesting a more complex and prolonged flavor persistence. In contrast, the fortified nixtamal tortilla is associated with piloncillo, suggesting a sweet undertone that persists after consumption. Overall, the aftertaste results suggest that fortification enhances residual sweetness and flavor complexity, while traditional nixtamal tortillas maintain a consistent maize-derived sensory persistence.

Table 20. Structured associations between tortilla classes and aftertaste descriptors in the Tulancingo dataset. The discriminative projection space was generated with the DE-LDA$_{FE}$ method.

Class	Features
Minsa tortilla	lime
Nixtamal tortilla	corn, nixtamal
Fortified minsa tortilla	sweet, salty, other
Fortified nixtamal tortilla	piloncillo

Overall, the attribute-based analysis of the Tulancingo dataset reveals consistent and distinct sensory tendencies across tortilla types, though with structural variations compared with the Xalapa results. Traditional nixtamal tortillas maintain a stable maize-centered identity, characterized by yellow color, corn-related aromatic and flavor notes, and firm, cohesive texture. Commercial Minsa tortillas exhibit lighter visual tones, perceptible acidity, and comparatively less structured mechanical characteristics. Fortification introduces notable modifications in both formulations: in the Minsa variant, it enhances sweetness, moisture, and softness, whereas in the nixtamal-based tortilla, it tends to intensify darker coloration, roasted or sweet undertones, and a denser or more adhesive mouthfeel. Collectively, these patterns confirm that fortification alters multiple sensory dimensions while preserving the foundational identity of each base tortilla type in the Tulancingo dataset.

6. Discussion

The experimental results suggest that class separability in sensory tortilla analysis exhibits multi-scale structural behavior. By integrating DBI, SC, and KNN-OS into a unified geometric index S_θ, the evaluation framework indicates that projection quality is assessed from complementary structural perspectives rather than a single criterion.

Across both datasets and analysis types, DE-LDA$_{FE}$ consistently achieved competitive or superior performance compared with PCA, LDA, and PCA+LDA. Its main advantage

https://doi.org/10.3390/mca31030082

lies in its flexibility: unlike classical LDA, which optimizes a closed-form Fisher criterion under strict assumptions, DE-LDA$_{FE}$ directly searches the projection space to optimize separability behavior, enabling adaptation to non-ideal or partially overlapping class distributions. It is important to note that the comparative analysis in this study was intentionally limited to PCA, LDA, and their combination, as these methods remain the most widely adopted in food science due to their interpretability and the clear associations they reveal between variables and sensory attributes. Although alternative approaches such as PLS-DA, Random Forest, and non-linear techniques (e.g., t-SNE and UMAP) offer strong performance, their limited geometric interpretability often hinders their adoption in this field. In this context, DE-LDA$_{FE}$ operates within the same orthonormal projection framework as PCA and LDA, preserving interpretability while optimizing class separability. This positions the proposed method as a natural extension of established techniques, bridging the gap between explainability and discriminative performance. Nonetheless, future work should explore comparisons with non-linear and ensemble methods.

The global analyses indicates that when all sensory attributes are considered simultaneously, evolutionary optimization captures discriminant directions that improve both compactness and local discrimination. The attribute-based analyses show that separability strength is unevenly distributed across sensory dimensions, emphasizing the importance of attribute-level evaluation in food sensory characterization.

Beyond quantitative separability, the projection matrices derived from the best-performing configurations enabled the identification of sensory profiles for each tortilla type. By analyzing the geometric relationship between feature-weight vectors and class centroids in the discriminative subspace, distinct attribute patterns were observed across both datasets. Traditional nixtamal tortillas consistently maintained a maize-centered identity, characterized by yellow color, corn-related aromatic and flavor notes, and firm or elastic texture. Commercial Minsa tortillas tended to exhibit lighter color tones, perceptible acidity, and comparatively weaker structural cohesion. Fortification introduced systematic modifications: in Minsa-based tortillas, it frequently enhanced sweetness, softness, and moisture perception, while in nixtamal-based tortillas, it intensified darker color tones, roasted or burnt notes, and a denser or more adhesive mouthfeel.

It is important to note that, although attribute-based projections exhibit class overlap, the geometric interpretation of projection weights provides a meaningful structural mapping between sensory variables and tortilla types. This suggests that discriminative dimensionality reduction not only improves classification-oriented separability but also facilitates interpretability at the feature level, a crucial aspect in sensory science applications.

Moreover, the robustness of DE-LDA$_{FE}$ when LDA fails or becomes unstable underscores its practical relevance for real-world datasets, where multicollinearity and limited sample structure may compromise analytical solutions.

Regarding the experimental design, it is important to acknowledge the demographic concentration of the sensory panel, which consisted of university students aged 18 to 25. This population was selected to ensure participants' availability and willingness to engage in multiple tasting sessions, thereby enabling the acquisition of a dataset. However, this choice introduces demographic bias. The sensory preferences and perceptions reported herein, therefore, reflect this specific segment and should be generalized to broader populations with caution. Nevertheless, the proposed framework focuses on extracting structural relationships and discriminative patterns within the data rather than on population-level inference, thereby supporting the validity of the methodological findings despite the sample constraint. Future research could benefit from incorporating more diverse panels to evaluate the stability of the identified sensory profiles and confirm the robustness of fortified tortilla characterization across different age groups and socio-economic backgrounds.

https://doi.org/10.3390/mca31030082

Another methodological aspect to consider is that this study does not employ a training/test split or cross-validation. This choice is intentional, as DE-LDA$_{FE}$ is used here as a descriptive and exploratory tool for data characterization rather than for predictive modeling or machine learning tasks. The primary objective is to identify a projection matrix that reveals the intrinsic structural separability of the fortified tortilla samples. In this context, the absence of external validation is consistent with exploratory multivariate techniques such as PCA. Nevertheless, future extensions of the DE-LDA$_{FE}$ framework could incorporate resampling strategies to assess the stability and robustness of the optimized projection across different data subsets.

Overall, the proposed framework indicates that combining evolutionary feature extraction with structured separability evaluation provides improved discriminative projections, greater robustness against ill-conditioned scatter matrices, enhanced local overlap control, interpretable two-dimensional visualizations for sensory analysis, and a structured methodology for identifying attribute-level sensory profiles.

The DE-LDA$_{FE}$ framework is specifically designed to reduce dimensionality to two or three dimensions ($d \in \{2, 3\}$), aligning with its primary goal of enabling intuitive visualization of multivariate sensory data. This restriction reflects the practical limits of human interpretation, as meaningful geometric insights are best obtained in low-dimensional spaces. By concentrating the evolutionary search within these projection spaces, the method enhances class separability while avoiding stability issues commonly encountered by analytical approaches. As a result, it achieves a balance between cluster compactness and visual interpretability, providing clear 2D or 3D representations for the analysis of complex sensory profiles.

The results indicate that DE-LDA$_{FE}$ is a promising and robust alternative for visualizing sensory data, offering significant advantages in class separation in the specific case of fortified tortillas.

7. Conclusions

This study proposed a structured framework for dimensionality reduction in complex sensory datasets, combining analytical and evolutionary projection methods with a multicriteria separability assessment through the unified index S_θ. This approach enabled a balanced evaluation of cluster compactness, cohesion, and neighborhood consistency.

Across both datasets, the DE-LDA$_{FE}$ method demonstrated competitive or superior performance compared to PCA, LDA, and PCA+LDA, particularly in handling overlapping classes and heterogeneous sensory variables. Its ability to directly optimize separability enabled it to overcome limitations of classical linear approaches.

Attribute-level analysis revealed that sensory discrimination is not uniformly distributed, with color, taste, and mouthfeel contributing differently across datasets and formulations. Additionally, the proposed framework facilitated the identification of interpretable sensory profiles, showing that fortification systematically modifies key attributes such as sweetness, color, and texture.

In summary, the proposed methodology provides:

- A structurally grounded separability evaluation scheme;
- A flexible evolutionary projection strategy robust to ill-conditioned scenarios;
- Interpretable low-dimensional representations suitable for visualization;
- A systematic approach for sensory profile identification.

Overall, the methodology provides a robust and interpretable tool for analyzing sensory data on tortilla fortifications with chilacayote powder.

https://doi.org/10.3390/mca31030082

Author Contributions: Conceptualization, A.-L.L.-L., H.-G.A.-M. and E.M.-M.; methodology, A.-L.L.-L., H.-G.A.-M. and E.M.-M.; software, A.-L.L.-L.; validation, H.-G.A.-M., E.M.-M., J.-E.A.-J. and E.-N.A.-B.; formal analysis, A.-L.L.-L.; investigation, A.-L.L.-L.; resources, H.-G.A.-M., E.M.-M., J.-E.A.-J., A.-G.R.-G., E.-N.A.-B. and R.-H.A.-R.; data curation, A.-L.L.-L.; A.-G.R.-G. and R.-H.A.-R.; writing—original draft preparation, A.-L.L.-L.; writing—review and editing, H.-G.A.-M., E.M.-M., J.-E.A.-J. and E.-N.A.-B.; visualization, A.-L.L.-L., H.-G.A.-M. and E.M.-M.; supervision, H.-G.A.-M., E.M.-M., J.-E.A.-J. and E.-N.A.-B.; project administration, H.-G.A.-M., E.M.-M. and J.-E.A.-J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets used in this study are not publicly available, as they are part of ongoing research. Requests for access may be directed to jimalba@uv.mx, eliaquino@uv.mx, and heacosta@uv.mx, provided that participant anonymity is preserved.

Acknowledgments: The first and fifth authors acknowledge the Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI) of Mexico for the financial support provided through scholarships 712182 and 823476, awarded for postdoctoral and doctoral studies at the Artificial Intelligence Research Institute, and at the Center for Food Research and Development of the University of Veracruz, respectively.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

PCA	Principal Components Analysis
LDA	Linear Discriminant Analysis
PCA+LDA	Principal Components Analysis with Linear Discriminant Analysis
DE-LDA$_{FE}$	Differential Evolutionary Linear Discriminant Analysis for Feature Extraction and Visualization
DBI	Davies-Bouldin Index
SC	Silhouette Coefficient
KNN-OS	K-Nearest Neighbours Overlap Score

References

1. Borecka, M.; Karaś, M. A comprehensive review of the nutritional and health-promoting properties of edible parts of selected Cucurbitaceae plants. *Foods* **2025**, *14*, 1200. [CrossRef]
2. Roman-Ramos, R.; Almanza-Perez, J.; Fortis-Barrera, A.; Angeles-Mejia, S.; Banderas-Dorantes, T.; Zamilpa-Alvarez, A.; Diaz-Flores, M.; Jasso, I.; Blancas-Flores, G.; Gomez, J.; et al. Antioxidant and anti-inflammatory effects of a hypoglycemic fraction from Cucurbita ficifolia Bouché in streptozotocin-induced diabetes mice. *Am. J. Chin. Med.* **2012**, *40*, 97–110. [CrossRef]
3. Kruger, J.; Taylor, J.R.; Ferruzzi, M.G.; Debelo, H. What is food-to-food fortification? A working definition and framework for evaluation of efficiency and implementation of best practices. *Compr. Rev. Food Sci. Food Saf.* **2020**, *19*, 3618–3658. [CrossRef]
4. Dwyer, J.T.; Wiemer, K.L.; Dary, O.; Keen, C.L.; King, J.C.; Miller, K.B.; Philbert, M.A.; Tarasuk, V.; Taylor, C.L.; Gaine, P.C.; et al. Fortification and health: Challenges and opportunities. *Adv. Nutr.* **2015**, *6*, 124–131. [CrossRef]
5. Acevedo-Martinez, K.A.; Gonzalez de Mejia, E. Fortification of maize tortilla with an optimized chickpea hydrolysate and its effect on DPPIV inhibition capacity and physicochemical characteristics. *Foods* **2021**, *10*, 1835. [CrossRef]
6. Hassan, S.M.; Forsido, S.F.; Tola, Y.B.; Bikila, A.M. Physicochemical, nutritional, and sensory properties of tortillas prepared from nixtamalized quality protein maize enriched with soybean. *Appl. Food Res.* **2024**, *4*, 100383. [CrossRef]
7. Montemayor-Mora, G.; Hernández-Reyes, K.E.; Heredia-Olea, E.; Pérez-Carrillo, E.; Chew-Guevara, A.A.; Serna-Saldívar, S.O. Rheology, acceptability and texture of wheat flour tortillas supplemented with soybean residue. *J. Food Sci. Technol.* **2018**, *55*, 4964–4972. [CrossRef]
8. Iuga, M.; Ávila Akerberg, V.D.; González Martínez, T.M.; Mironeasa, S. Consumer preferences and sensory profile related to the physico-chemical properties and texture of different maize tortillas types. *Foods* **2019**, *8*, 533. [CrossRef] [PubMed]

https://doi.org/10.3390/mca31030082

9. Rodríguez-Noriega, S.; Buenrostro-Figueroa, J.J.; Rebolloso-Padilla, O.N.; Corona-Flores, J.; Camposeco-Montejo, N.; Flores-Naveda, A.; Ruelas-Chacón, X. Developing a descriptive sensory characterization of flour tortilla applying flash profile. *Foods* **2021**, *10*, 1473. [CrossRef] [PubMed]
10. Salinas-Moreno, Y.; Gálvez-Mariscal, A.; Severiano-Pérez, P.; Vázquez-Carrillo, G.; Trejo-Téllez, L. Flavor and TASTE attributes and nutritional insights of maize tortillas from landraces of Mexican races. *Heliyon* **2024**, *10*, e28314. [CrossRef]
11. López-Lobato, A.L.; Rodríguez-Gómez, A.G.; Aquino-Bolaños, E.N.; Alfaro-Rodríguez, R.H.; Acosta-Mesa, H.G.; Alba-Jiménez, J.E. Using Feature Extraction Methods to Perform a Sensory Analysis of Tortillas Fortified with Chilacayote Powder. In *Proceedings of the Mexican International Conference on Artificial Intelligence*; Springer: Cham, Switzerland, 2025, pp. 125–133.
12. Haider, A.; Iqbal, S.Z.; Bhatti, I.A.; Alim, M.B.; Waseem, M.; Iqbal, M.; Mousavi Khaneghah, A. Food authentication, current issues, analytical techniques, and future challenges: A comprehensive review. *Compr. Rev. Food Sci. Food Saf.* **2024**, *23*, e13360. [CrossRef] [PubMed]
13. Malik, M.; Dewan, A. Machine learning in food authentication: A sustainable solution to global food challenges. *Lett. Food Res.* **2025**, *1*, e25050. [CrossRef]
14. Ares, G.; Jaeger, S.R.; Bava, C.M.; Chheang, S.L.; Jin, D.; Gimenez, A.; Vidal, L.; Fiszman, S.M.; Varela, P. CATA questions for sensory product characterization: Raising awareness of biases. *Food Qual. Prefer.* **2013**, *30*, 114–127. [CrossRef]
15. Kim, M.R.; Heo, J.; Kwak, H.S. Comparison of sensory profiles by two different check-all-that-apply (CATA) terms developed from trained panelists and naïve consumers. *Food Qual. Prefer.* **2023**, *109*, 104902. [CrossRef]
16. Bishop, C.M. *Pattern Recognition and Machine Learning by Christopher M. Bishop*; Springer Science+ Business Media, LLC: Berlin/Heidelberg, Germany, 2006; Volume 350.
17. Guyon, I.; Gunn, S.; Nikravesh, M.; Zadeh, L.A. *Feature Extraction: Foundations and Applications*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006; Volume 207.
18. Tharwat, A. Principal component analysis-a tutorial. *Int. J. Appl. Pattern Recognit.* **2016**, *3*, 197–240. [CrossRef]
19. Scholkopft, B.; Mullert, K.R. Fisher discriminant analysis with kernels. *Neural Netw. Signal Process IX* **1999**, *1*, 1.
20. Tharwat, A.; Gaber, T.; Ibrahim, A.; Hassanien, A.E. Linear discriminant analysis: A detailed tutorial. *AI Commun.* **2017**, *30*, 169–190. [CrossRef]
21. Sharma, A.; Paliwal, K.K. Linear discriminant analysis for the small sample size problem: An overview. *Int. J. Mach. Learn. Cybern.* **2015**, *6*, 443–454. [CrossRef]
22. Storn, R.; Price, K. Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [CrossRef]
23. Eiben, A.E.; Smith, J.E. *Introduction to Evolutionary Computing*; Springer: Berlin/Heidelberg, Germany, 2015.
24. Price, K.; Storn, R.M.; Lampinen, J.A. *Differential Evolution: A Practical Approach to Global Optimization*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006.
25. Ahmad, M.F.; Isa, N.A.M.; Lim, W.H.; Ang, K.M. Differential evolution: A recent review based on state-of-the-art works. *Alex. Eng. J.* **2022**, *61*, 3831–3872. [CrossRef]
26. Pant, M.; Zaheer, H.; Garcia-Hernandez, L.; Abraham, A. Differential Evolution: A review of more than two decades of research. *Eng. Appl. Artif. Intell.* **2020**, *90*, 103479. [CrossRef]
27. Llaguno-Roque, J.L.; López-Lobato, A.L.; Pérez-Arriaga, J.C.; Acosta-Mesa, H.G.; Sánchez-García, Á.J.; Gutiérrez-Ospina, G.; Barranca-Enríquez, A.; Romo-González, T. AI-Driven Biopsychosocial Screening for Breast Cancer: Enhancing Risk Prediction via Differential Evolutionary Linear Discriminant Analysis for Feature Extraction. *Math. Comput. Appl.* **2026**, *31*, 66. [CrossRef]
28. Dl, D. A cluster separation measure. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *1*, 224–227. [CrossRef]
29. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]
30. Han, J.; Kamber, M.; Pei, J. *Data Mining: Concepts and Techniques*, 3rd ed.; Morgan Kaufman Publishers: Waltham, MA, USA, 2012.
31. Ho, T.K.; Basu, M. Complexity measures of supervised classification problems. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 289–300. [CrossRef]
32. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
33. Arfken, G.B.; Weber, H.J.; Harris, F.E. *Mathematical Methods for Physicists: A Comprehensive Guide*; Academic Press: Cambridge, MA, USA, 2011.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

https://doi.org/10.3390/mca31030082

Mathematical and Computational Applications

MDPI

Article

Benchmarking Ethnic Hate Speech Detection in México

Verónica Neri-Mendoza [1,2,*], Yulia Ledeneva [2,*], Jonathan Rojas-Simón [2], Muhammad Tayyab Zamir [3], René Arnulfo García-Hernández [2] and Ángel Hernández-Castañeda [2]

[1] Secretaría de Ciencia, Humanidades, Tecnología e Innovación, Av. Insurgentes Sur 1582, Crédito Constructor, Benito Juárez, Ciudad de México 03940, Mexico
[2] Centro Universitario Tianguistenco, Universidad Autónoma del Estado de México, Instituto Literario 100, Estado de México 50000, Mexico; jrojass@uaemex.mx (J.R.-S.); reagarciah@uaemex.mx (R.A.G.-H.); anhernandezc@uaemex.mx (Á.H.-C.)
[3] Centro de Investigación en Computación, Instituto Politécnico Nacional, Av. Juan de Dios Batiz S/N, Nueva Industrial Vallejo, Gustavo A. Madero, Ciudad de México 07320, Mexico; mzamir2023@cic.ipn.mx
* Correspondence: vnerim001@alumno.uaemex.mx (V.N.-M.); ynledeneva@uaemex.mx (Y.L.)

Abstract

Ethnic hate speech is a form of intersectional violence that affects Indigenous groups in México. Despite the severity of this social phenomenon, there is a lack of computational resources, specifically labeled datasets, to enable the development of automated tools for its detection. This paper presents a methodology for calculating benchmarks of the EthnoHate dataset, designed for the classification of ethnic hate speech in México, and focuses on establishing baselines using machine learning algorithms. Moreover, different levels of linguistic representation are evaluated. The results reveal that hate in the dataset is predominantly explicit, with a strong lexical component, enabling models such as BoW, and TF-IDF to achieve an F1-macro competitive. However, semantic techniques like BERT with a contextual classifier achieves the best overall performance in F1-macro, demonstrating that there is a significant proportion of implicit cases that require deep semantic understanding. Analysis of algorithms reveals that ethnic hate speech in México manifests through a recurring vocabulary with complex combinations, and that, while lexical approaches are highly effective, contextual models are necessary to capture the subtlety and diversity of hate expressions. This work establishes the feasibility of the task, validates the quality of the EthnoHate dataset, and lays the groundwork for future research employing more complex architectures.

Keywords: ethnic; hate speech; benchmarks; México; EthnoHate dataset; machine learning algorithms

1. Introduction

Hate speech refers to any type of communication, whether oral, written, or behavioral, that attacks or uses pejorative or discriminatory language toward a person or group [1]. The most common reasons why people are victims of discrimination include the following: identity and belonging—gender, age, ethnicity, gender identity, sexual preference, spiritual or religious beliefs (or lack thereof), and place of origin; social and personal status—socioeconomic status, level of education, and presence of disabilities; communication and expression—language and/or accent, cultural clothing, and political ideology, among many others [1–3].

Hate speech is commonly spread through social media, a mass communication medium that facilitates its dissemination. Although most hate messages are textual, images

and sounds are also used. Therefore, to address this problem from a computer science perspective, text classification is the most effective method for automated detection [4]. It is not just about text, but rather an amplification system, where users do not use explicit hate speech, but rather encode it through humor or irony to normalize extreme ideas.

Hate speech has serious negative consequences of affecting people, promoting misinformation, and degrading online dialogue. This sheer volume of digital data prevents effective human review, as manually processing and classifying large amounts of text data is slow and difficult. Furthermore, the accuracy of manual text categorization can easily be affected by human factors such as fatigue and competence [4]. Because of this, it is essential to advance research into automated techniques for its detection [5]. That is the reason Natural Language Processing (NLP) is a fundamental tool for mitigating hate speech, as it allows for the analysis of textual linguistic features.

In the state of the art, there have been various efforts to address this problem. NLP has seen a notable increase in academic interest in hate speech detection, ranging from the implementation of traditional machine learning to deep learning. This is reflected in a significant increase in research in this area, with developments allowing social media platforms to identify and block hateful messages before they are published, thereby reducing the risk of harassment [1,6].

Advances in this area have had a significant impact in various fields, leading to the creation of datasets of social media real posts in different regions of the world, mainly in English. These are particularly focused on detecting hate speech against women [7**? ?**], the LGBT community [10**?**], and religious communities such as those of Islam, Christianity, and Judaism [12]; in relation to nationality and racism, there is also work published on the detection of hate speech against immigrants [7].

Hate speech and discrimination are closely related terms. In México, the origins of discrimination can be traced back mainly to the Spanish conquest and the colonial caste system, which established social structures based on ethnic origin and skin color to maintain control and justify inequality. These prejudices have been perpetuated over time and continue to this day [13].

According to the CONAPRED (Consejo Nacional para Prevenir la Discriminación), "Not every act of discrimination is caused by hate speech; however, all hate speech has a discriminatory basis" [14]. Ethnic discrimination is also commonly known as racial discrimination or *racism*, especially in the international area. However, there are no races within the human species. This idea is based on the false premise that human beings can be classified racially, establishing a hierarchy where certain races are considered superior, and therefore, others are seen as morally and intellectually deficient [15]. Ethnic hate speech constitutes a specific form of violence, as it is directed against individuals or groups that are distinguished by their cultural characteristics, traditions, ethnicity, clothing, and skin color, among other factors [16].

This is intersectional violence with other types of violence such as aporophobia or classism, where—according to the special report on discriminatory narratives issued by [14] CONAPRED—the main victims are those living in poverty, who face not only a shortage of economic resources but also marginalization by society and a lack of access to education and job opportunities. Although CONAPRED receives complaints from various motives, it has documented cases related to ethnic origin that demonstrate hate speech put into practice in the form of discrimination. This is reinforced by data obtained in the most recent National Survey on Discrimination (https://www.inegi.org.mx/app/saladeprensa/noticia/8172 (accessed on 15 February 2026)), where indigenous groups have seen an increase in discrimination. The percentage of indigenous people who reported having

https://doi.org/10.3390/mca31030088

suffered an act of discrimination rose from 25% in 2017 to 28% in 2022, making them one of the most vulnerable groups.

However, despite the fact that ethnic hate speech represents a serious and recurring problem within the Mexican social context, there is a lack of labeled datasets that capture the linguistic diversity and contextual particularities of hate speech in the country. This promotes the development of text classification tools, meaning that the severity of the social phenomenon is not supported by the necessary computational resources. Therefore, in this article, we present an effort to address this problem with the creation of the EthnoHate dataset, the purpose of which is to classify ethnic hate speech in México. Specifically, in this work, we focus on the calculation of benchmarks through machine learning algorithms. The importance of this work lies in the fact that it establishes the feasibility and difficulty of addressing this problem and validates the dataset in relation to the labeling performed and the classes established. In addition, it lays the foundation for comparing performance between classification algorithms.

The rest of the paper is structured as follows: The Section 2 reviews the background of hate detection, with emphasis on how benchmarks have been calculated across other datasets. The Section 3 describes the construction of the EthnoHate dataset, its class distribution, the division into training, development, and testing phases, as well as the textual representations and the set of classification algorithms selected to establish the benchmarks. The Section 4 presents the performance of each configuration, including a comparative analysis. Finally, the Section 5 synthesizes the findings, thus validating the quality of the EthnoHate dataset and establishing a solid foundation for future work with more complex architectures.

2. State of the Art

Recently, hate speech detection has gained significant relevance in the field of NLP, reflecting the scientific community's growing interest in social and ethical issues. This boom is directly related to the international commitment to curb extremism and violence on the internet. It is essential to address hate speech, as it stems from intolerance and prejudice and has the potential to trigger acts or even structured policies of violence and discrimination [17]. Hate speech is addressed computationally as a classification problem through NLP, particularly machine learning, which consists of assigning predefined categories to texts, facilitating the management of large volumes of unstructured text data [18].

Automatic detection of hate speech has become a priority for ensuring safe digital spaces. However, the effectiveness of these systems depends critically on the availability of annotated datasets and the definition of robust benchmarks that allow the progress of new algorithms. The effectiveness of text classification models depends on feature selection and engineering, since the quality of the input data conditions the learning of the algorithm. By using units such as words or semantic structures, both accuracy and interpretation capacity can be optimized. Hate speech detection is mainly based on textual feature engineering, using techniques such as TF-IDF and n-grams to represent textual features [19]. However, the landscape has changed with the advent of architectures such as BERT (Bidirectional Encoder Representations from Transformers) [20]. These deep learning models have redefined industry standards by processing bidirectional relationships and long-distance contexts, thus capturing the semantic complexity of text more efficiently [18].

Hate speech becomes especially difficult to distinguish when it involves regional expressions or language used against vulnerable groups, where nuance is crucial. Therefore, in this section, we will review some key initiatives that have laid the methodological

https://doi.org/10.3390/mca31030088

groundwork for establishing benchmarks or baselines for classifying violent content across various social contexts.

As a basis for the study of digital violence, the MEX-A3T dataset [**?**] allows for the detection of aggression in tweets in the Mexican Spanish language. Its relevance lies in the fact that it serves as a starting point for combating complex problems such as hate and harassment, through the analysis of social media posts on topics such as politics, sexism, homophobia, and discrimination. To establish a point of comparison, two traditional baselines were developed: a Bag of Words (BoW) model and another based on character trigrams. In both schemes, preprocessing was applied, consisting of the removal of stop words and special characters from the entire vocabulary of the corpus. The final classification was carried out using a Support Vector Machine (SVM) with a linear kernel.

HOMO-MEX [**?**] is a model designed to detect LGBTQ+phobic content written in Mexican Spanish. This project is crucial because it addresses the duality between aggressive and subtle expressions. Its benchmarks were established by comparing traditional algorithms, such as Naive Bayes (NB), SVM, Logistic Regression (LR) and Random Forest (RF), with cutting-edge models based on BERT. Technically, HOMO-MEX validated that combining frequency features (BoW and TF-IDF) with contextual language models is the most robust strategy for classifying discrimination on social media. On the other hand, In HOMO-LAT, a pretrained BERT database for Spanish was implemented and fitted to the training dataset using the HuggingFace Transformers library [10].

On the other hand, EXIST is a shared task on identifying sexism in social media. EXIST seeks to address sexism in a broad sense, from explicit misogyny to other subtle expressions that imply implicit sexist behaviors. To evaluate the performance of the different approaches proposed for this problem, in its 2021 and 2022 editions [**? ?**], they proposed two different baselines to establish the expected performance of the runs. First, they calculated a benchmark based on an SVM, with input from features extracted with TF-IDF, generated from text BoW. A second benchmark is based on a model that labels each record according to the majority class.

In the field of political and religious hate speech, the third edition of the shared task on detecting hateful content in Italian tweets [12] stands out. In this work, benchmarks were calculated using SVMs trained with BoW representations, employing the XPoliticalHate and HaSpeeDe2 datasets. This approach reaffirms the use of traditional models and lexical representations as a solid basis for benchmarking in hate detection tasks.

Meanwhile, the HatEval shared task was presented [7], focused on the detection of hate speech against immigrants and women, where two benchmarks were calculated: the first is based on the majority class and a representation based on TF-IDF with the SVM classifier.

Table 1 summarizes state-of-the-art approaches to hate speech detection. As observed, this trend is reflected in the predominance of traditional techniques, with the SVMs as the most frequently used algorithm and the quintessential reference model due to its ability to generalize in high-dimensional vector spaces. Naive Bayes and Random Forests are also used, complementing the analysis of digital violence phenomena documented in recent literature. Some studies have established morphological benchmarks, such as MEX-A3T, which established textual representation using character trigrams.

According to the literature review, there is a strong consensus on the representation and classification methods used to calculate reference models, highlighting that TF-IDF and n-gram-based representations have demonstrated consistent performance in capturing the lexicon of sentiment, polarity, and other linguistic features essential for identifying hate speech. In particular, representations such as BoW and TF-IDF remain relevant for detecting offensive language, although their scope is limited to the lexical level, as they

https://doi.org/10.3390/mca31030088

rely exclusively on term frequency and fail to capture complex semantic relationships between words.

Therefore, current research suggests a necessary transition toward integrating advanced linguistic models such as BERT and BETO. This evolution allows us to move beyond the identification of isolated words to understand the semantics and context behind online violence, facilitating the critical distinction between aggressive colloquial language and hate speech intended to offend or marginalize.

Table 1. Summary of datasets.

Dataset	Representation	Algorithm
MEX-A3T: Aggression in Mexican Spanish	BoW and 3-grams characters	SVM
Homo-Mex: LGBTphobia in Mexican Spanish	BERT, BoW and TF-IDF	NB, SVM, LR and RF
Homo-Lat: LGBTphobia in Spanish	BERT	BERT-based model pretrained for Spanish (BETO)
Exist 2021 and 2022: Sexism in English and Spanish	TF-IDF, BoW	SVM, Labels based on majority class
HaSpeeDe3: Political and Religious in Italian	BoW	SVM
HatEval: Immigrants and Women in Spanish and English	TF-IDF	Labels based on majority class and SVM

Despite advances in Natural Language Processing, a significant gap persists in the state of the art regarding the specific detection of ethnic hate speech in the Mexican context. Current efforts in Mexican Spanish have primarily focused on detecting generic aggression using MEX-A3T or forms of violence directed at the LGBT+ community using HOMO-MEX. International databases such as HatEval address hatred toward immigrants and women, while resources like HaSpeeDe3 focus on political and religious hate in other languages. However, there is no specialized corpus analyzing ethnic discrimination against Indigenous peoples in México, an absence that is especially important considering the increased discrimination this group has experienced. This lack represents a crucial opportunity to develop models that, beyond detecting sexism or generic aggression, can decode the historical, cultural, and linguistic nuances of ethnic hate directed at Indigenous communities. Consequently, this paper systematically establishes and presents the initial performance parameters for the EthnoHate dataset; the benchmarks of this dataset are essential not only to validate the new corpus, but also to evaluate the performance of future models and contribute to making visible a topic that is little explored in the literature, thus strengthening research in cultural and linguistic contexts.

3. Proposed Methodology

Hate speech is difficult to identify. Its complex and multifaceted nature presents significant challenges for both human annotators and automated systems. In this context, EthnoHate dataset is presented, wich is composed of tweets specifically collected and annotated for the detection of hate speech directed at Indigenous communities in México. It addresses the need for specialized datasets that allow for the training and evaluation of models capable of learning in this domain. The main objective of the methodology is to establish initial benchmarks for the dataset, which will serve as a reference point for future research. To this end, traditional and modern text representation techniques, as well as widely used classifiers for similar tasks, were employed.

https://doi.org/10.3390/mca31030088

3.1. Data

The instances were collected through a manual search on social network X, using words directed at a specific person or group through terms referring to the 68 indigenous communities of México, the words indio, india, and indigena, as well as their ethnonyms. As a result of the data-collection process, a set of 18,685 publications was obtained. However, after data curation, the resulting number of posts was 17,000. Subsequently, a preprocessing phase was carried out aimed at cleaning and anonymizing the information, in which the usernames of both the sender and the recipient were replaced by a generic label. This procedure made it possible to protect the identity of users and standardize the text for subsequent analysis.

With the participation of experts in social anthropology and indigenous law, the collected publications were analyzed and labeled through guidelines that establish criteria for annotating each post. Subsequently, an annotation team was formed, consisting of eight people selected based on their familiarity with social media, use, and the sociocultural context in which the analyzed interactions take place. Before beginning the labeling process, the annotators were given clear and detailed information about the general objectives of the research, the purpose of the dataset, and the importance of accurate annotation for the development of reliable hate speech detection models.

In order to ensure the consistency and quality of the annotation process, a preliminary annotation training phase was carried out. During this stage, several meetings were held in which the classification categories were presented and explained in detail, along with their formal definitions and inclusion and exclusion criteria. Representative examples of each category were also analysed, and borderline or ambiguous cases were discussed, allowing annotators to compare interpretations, resolve doubts, and harmonize the criteria.

These examples lie on the borderline between hate speech against the Indigenous community and hate speech against another population (or between hate speech and non-discriminatory criticism) for the following reasons:

- *"indio" Dijo el Europeo color cartón mojado.*

This example represents sarcasm. No other population is explicitly attacked, but the use of "European" as a contrast could imply a reverse criticism (a stereotype of the racist European).

- *No mnches! A la India Maria se le respeta, porque siempre portó con orgullo la vestimenta de indígena Mazahua, en cambio la p1nch3 zopilota, solo parece que va a una fiesta de disfraces. Un saludo a todos los indígenas, con mucho cariño.*

The main attack is against a non-Indigenous woman ("Zopilota"), who is accused of cultural appropriation and ridiculing Indigenous clothing. The insult is violent ("p1nch3 zopilota"), but it is not directed at her ethnic identity, but rather at her behavior.

- *Qué increíble ver a una cocinera tradicional cuicateca representando nuestra cultura en este nuevo libro. Oaxaca siempre poniendo el nombre de México en alto con su gastronomía, aunque siendo sinceros me parece Hay chefs con estudios de verdad que sí ponen el nombre de México en alto, no esto, esta es comida taan ordinaria y comun.*

It is not explicit hate speech, but it approaches a microaggression or systemic contempt for Indigenous culture. The ambiguity lies in whether it is a legitimate opinion about the quality of the food or a derogatory judgment based on ethnicity/class.

This iterative process of discussion and consensus was essential to reduce the subjectivity inherent in the task and minimize variability among annotators.

Once the annotation training stage was complete and a common agreement had been reached on the annotation guidelines, the dataset was labeled using the Human Signal®

https://doi.org/10.3390/mca31030088

platform. This tool facilitated the systematic assignment of labels and the monitoring of annotation progress. Each post was classified into one of three defined categories: hate, no hate, and unrelated hate. The hate class includes explicit or implicit expressions that promote discrimination, contempt, or violence towards indigenous communities; the no hate class covers posts that, although related to these communities, do not contain offensive content; meanwhile, the unrelated hate class corresponds to messages with hate unrelated to the population study.

To ensure the reliability and consistency of the manual labeling of the EthnoHate corpus, Fleiss's Kappa coefficient was used, a statistical metric designed to assess the degree of agreement among multiple annotators [22]. The coefficient ranges from -1 to 1, where 1 represents perfect agreement and 0 indicates that the observed agreement is equivalent to that expected by chance. In this study, the Fleiss's Kappa value obtained was 0.9, indicating a substantial level of agreement among the annotators. Furthermore, it validates the consistency of the labeling criteria applied to the categories of ethnic hatred, no hate, and unrelated hate. This high reliability is crucial to ensuring that the subsequently established benchmarks reflect genuine linguistic patterns and not the particular biases of an individual annotator. This annotation scheme made it possible to construct a structured and coherent dataset, suitable for use in automatic classification tasks and for establishing reliable benchmarks in the detection of hate speech towards indigenous communities in México.

Regarding the representativeness of the labels, an imbalance is observed: the predominant category is no hate, followed by hate, while unrelated hate is the least-represented class (as illustrated in Figure 1). The dataset was split as follows: 70% of the instances were allocated to model training, 10% were reserved for hyperparameter tuning (development), and 20% were reserved for the final evaluation (test). This partitioning was performed randomly, maintaining the class distribution.

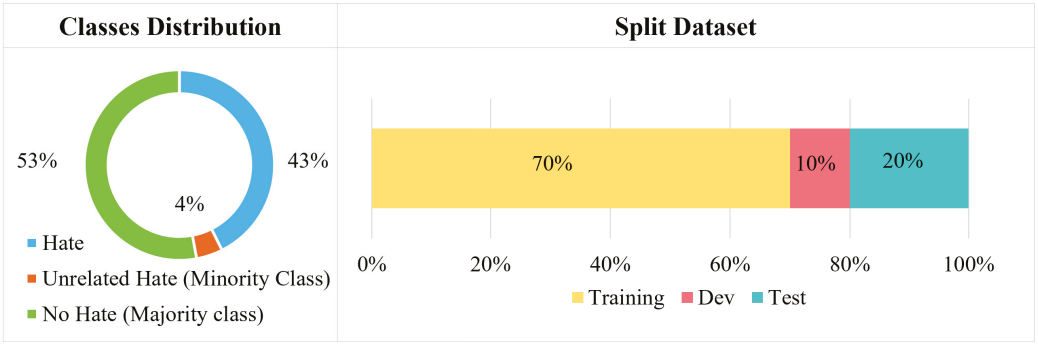


Figure 1. Distribution of classes in the EthnoHate dataset.

A significant limitation of the methodological design is that, despite identifying a significant imbalance in the distribution of classes within the dataset (53% for no hate, 43% for ethnic hate speech, and only 4% for unrelated hate), no strategies to mitigate this imbalance were implemented. This decision was based on preserving the natural distribution of the phenomenon in the collected data. However, it is recognized that this imbalance impacts model performance differently for each class: (a) the majority class (no hate) will tend to obtain the highest metrics; (b) the *ethnic hate* speech class, despite having considerable representation (43%), may be affected by overlap with no hate; and (c) the minority class (unrelated hate, 4%) has a high probability of obtaining low sensitivity (recall) and accuracy values, generating false negatives (undetected hate speech) and false positives (confusion with other classes).

https://doi.org/10.3390/mca31030088

3.2. Text Representation

This section details the various text representation techniques used in the classification task. The selection of these representations sought to cover aspects of different language levels:

- Morphological
  - Probability of ASCII characters: This is based on obtaining the relative frequency of each character by normalizing its individual content by the total length of the text. The final product is a 256-dimensional vector, where each entry reflects the probability of occurrence of a specific character [23,24].
- Lexical
  - BoW: In this technique, the representation is based exclusively on word frequency, without considering syntactic organization.
  - TF-IDF: Calculates a score for each word that reflects how informative it is to a particular text, taking into account it is frequency across the corpus.
- Semantic
  - BERT: Contextual integration is an approach to attribute extraction that interprets the meaning of a word by considering it is environment in a bidirectional way, that is, by examining both the terms that precede it and those that follow it [20].
  - D2V: Allows textual content to be represented in a continuous vector space, enabling the encoding of deep semantics and contextual environment. This capability facilitates the recognition of complex patterns and relational links between various documents in the dataset [25].

We considered the aforementioned representations to compare their performance. At the morphological level, we could identify whether users are employing word distortions (e.g., *1nd14*) or spelling alterations. At the lexical level, we could identify whether hate speech relies on specific vocabulary. Finally, by implementing semantic representations, we could capture the context of irony or sarcasm.

Furthermore, implementing these benchmarks allows to identify annotation biases. If a lexical benchmark performs exceptionally well, it could mean that human annotators relied primarily on the presence of certain words for tagging, rather than on context. Calculating benchmarks with these representations is not just a methodological procedure, but a diagnostic tool. In the case of ethnic hatred, where language is constantly evolving to evade censorship. Understanding the contribution of each linguistic level (morphological, lexical, and semantic) in this task is vital to building more robust systems.

For each classifier (SVM, LR, NB, RF, KNN, DT, MLP), feature vectors were generated from each representation technique. For example, a given text is converted into (i) a 256-dimensional vector with the probability of ASCII characters, (ii) a word frequency vector with BoW, (iii) a TF-IDF vector of the same space, and (iv) a 768-dimensional vector with BERT. Each of these vectors is fed separately into each classifier. The BERT SC classifier, however, directly uses BERT's own internal representations, fine-tuning the entire model.

3.3. Modeling Text Classification

Regarding text classification, we considered traditional and modern approaches, which are based on the state of the art.

- SVM: As the reference algorithm, it is based on its robustness and its inherent ability to handle high-dimensional feature spaces, such as those generated by TF-IDF or BoW representations.

https://doi.org/10.3390/mca31030088

- LR: This is considered a standard for benchmark calculation due to its computational efficiency (fast training) and its ability to generate probabilities, which facilitates the interpretation of model confidence.
- NB: This algorithm is based on *Bayes' theorem*, which addresses the probability theory. It allows the use of subjective information and combines it with empirical information to estimate the probability of events. In any text classification task, it is used as a standard to establish a benchmark for proposed models.
- Random Forest (RF): This is used as a benchmark classifier for hate speech detection. First, text documents are represented by vectorized features (e.g., TF-IDF), and multiple decision trees are trained on random subsets of data and features. Finally, the predictions are obtained through majority voting, providing a robust baseline for performance comparison.
- KNN: We included KNN as a baseline to assess whether vector-space similarity between posts is an effective predictor of hate speech. That is, it seeks semantically or lexically similar texts that share a label, enabling the diagnosis of corpus homogeneity.
- Decision Tree (DT): This approach helps determine whether hate speech can be characterized by straightforward rules, thereby gauging the complexity of the hate speech classification.
- Multilayer Perceptron (MLP): We include this as a baseline to assess whether including nonlinearities in the model improves performance compared to linear classifiers, serving as an intermediate point between classical approaches and more complex deep learning models.
- BERT SC: The contextualized representations produced by the BERT encoder (Section 3.2) were fed into a linear classification head for the Sequence Classification (SC) task.

To ensure transparency and facilitate the replication of this study, both the text representation techniques and the evaluated algorithms are available in the following repository [24]: https://gitlab.com/JohnRojas/NLPClassKit (accessed on 10 February 2026).

We implemented a classification pipeline for detecting ethnic hate speech (see Figure 2). The methodology combines an architecture that compares traditional models (SVM, RF, MLP) with representations such as ASCII, BoW, TF-IDF, and BERT. The evaluation is based on metrics of accuracy, recall, and F1-score.

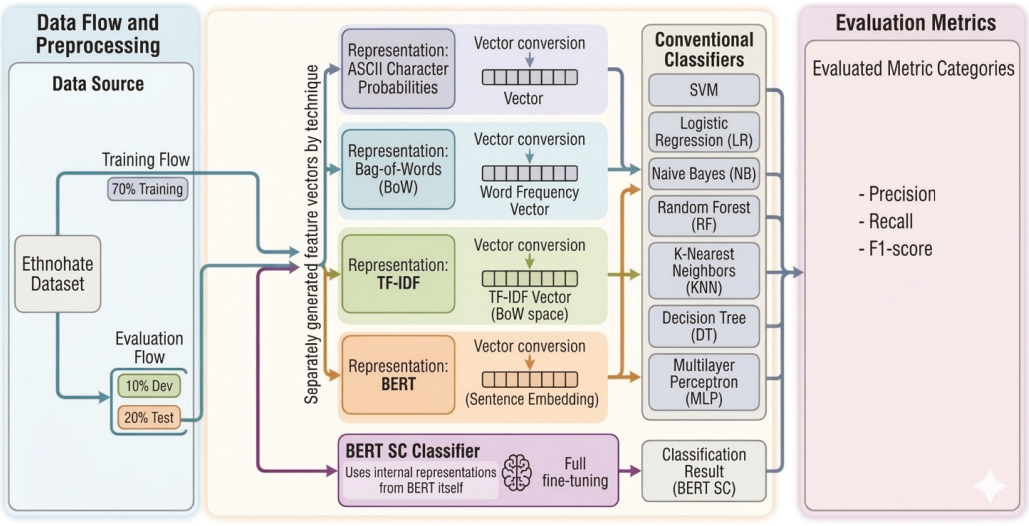


Figure 2. Diagram of the proposed methodology.

https://doi.org/10.3390/mca31030088

3.4. Evaluation Metrics

To evaluate model performance, the following metrics were used: precision, recall, and F1-score. Particular consideration was given to the macro average. The macro average calculates the metric independently for each class and then averages the results, giving equal weight to all classes regardless of their frequency in the dataset. This choice is particularly relevant in this study, as the minority class unrelated hate has the same evaluative importance as the majority class no hate, thus allowing for a fairer and more rigorous assessment of model performance in the face of the natural imbalance in the data.

The following formulas explain the metrics, where TPs is true positives, FPs is false positives, and FNs is false negatives.

Precision: This is the metric that calculates the fraction of true positives out of the total number of items classified as positive.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

Recall: The recall indicates, of all the cases that are positive in reality, how many of them were accurately labeled as positive by the model.

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

F1-score: The F1-score combines precision and recall through the harmonic mean, offering a measure that seeks a balance between these two metrics.

$$F1\text{-}Score = 2 \cdot \frac{Precision \cdot Recall}{Precison + Recall} \tag{3}$$

4. Experimental Results

This section presents and analyzes the results obtained in the evaluation of different classification algorithms for detecting ethnic hate speech, examining the performance of various textual representations: morphological (ASCII character probability), lexical (TF-IDF, BoW), and semantic (BERT, D2V) in capturing linguistic nuances. To establish a robust comparative overview, a diverse set of classification algorithms was employed, from traditional approaches to modern.

Benchmarks

To ensure the robustness and generalizability of our findings, we selected eight machine learning algorithms (SVM, LR, NB, RF, KNN, DT, MLP, and BERT SC). This diverse set of models enables a comprehensive evaluation of performance metrics across different learning approaches. In addition, we considered five text representations (ASCII, TF-IDF, BoW, BERT, and D2V), enabling comparisons among classical, distributed, and contextualized approaches within the benchmark. In addition, these models and representations have been widely adopted as baselines in previous hate speech detection studies [7**? ? ? ?**], supporting fair and interpretable evaluation. Accordingly, we evaluated multiple combinations of text representations and classification models.

The experimental setup used in this study is documented in Table 2, which details the hyperparameters for both text representations and classification algorithms. Regarding text representations, the D2V embeddings were generated using 100-dimensional vectors, a context window of 5, and 5 training epochs, empirically chosen to capture semantic relationships between words and documents. TF-IDF vectors were constructed using the 100 most frequent unigrams and bigrams at the document level. In addition, multilingual BERT (*bert-base-multilingual-cased*) was employed to obtain contextual embeddings of size

https://doi.org/10.3390/mca31030088

256 using *[CLS]* pooling. For the ASCII-based representation, the vector length was fixed at 256, as described in Section 3.2. Finally, the BoW representation yielded a high-dimensional feature space (57,198) derived from the vocabulary of the training set.

The hyperparameters of the classification models were selected to ensure a balanced comparison across both traditional machine learning and deep learning approaches. For transformer-based classification, the BERT model was fine-tuned with a maximum sequence length of 128 tokens, a learning rate of 2×10^{-5}, and 5 training epochs, following common practices in text classification. For traditional classifiers, standard yet effective configurations were adopted. Naive Bayes employed the complement variant (ComplementNB), which is particularly suitable for imbalanced text data. The SVM used a Radial Basis Function (RBF) kernel to capture nonlinear decision boundaries, while KNN was configured with $k = 3$, favoring local neighborhood decisions. Ensemble and tree-based methods included RF with 200 estimators and DT using the Gini impurity criterion for node splitting. The LR was trained for up to 10,000 iterations to ensure convergence. Finally, an MLP with 5 hidden layers, 1160 neurons, and ReLU activation was trained for 50 epochs, enabling the model to capture complex nonlinear relationships in the data.

Table 2. Parameters of each text representation and machine learning classifier.

Representation			**Model**		
Vectorization	**Feature**	**Value**	**Model**	**Feature**	**Value**
D2V	Vector Length	100	BERT SC	Max length	128
	Window	5		Epochs	5
	Epochs	5		learning rate	2×10^{-5}
TF-IDF	Vector Length	100	NB	Algortihm	ComplementNB
	Level	Document	SVM	Kernel	RBF
	Ngram Ranges	1–2	KNN	K	3
BERT	Model	Multilingual	MLP	Neurons	1160
	Vector Length	256		Layers	5
	Pooling	CLS		Activation	Relu
				Epochs	50
ASCII	Vector Length	256	LR	Iterations	10,000
BoW	Vector Length	57,198	RF	Estimators	200
			DT	Criterion	Gini

Table 3 reports the mean performance (±standard deviation) over 10-fold cross-validation for all evaluated models previously described in Table 2. In addition to these models, a baseline was included using the majority class (One-R) to provide a reference point for comparison. As expected, the One-R baseline obtains the lowest F1-score (0.2276). Moreover, the ASCII + MLP model performs similarly to the One-R baseline, indicating that character-based representations capture fewer linguistic patterns of ethnic hate speech.

The results show a wide variability in performance, with F1-scores ranging from 0.2276 to 0.6158. The benchmark based on the majority class (One-R) performed poorly, as it is an extremely simple classification method that selects the most predictive class. This benchmark serves as the basis for this research. Its low performance confirms that the classifiers are learning genuine patterns, while (ASCII + MLP) improves on the classification of One-R, purely morphological character-based information is insufficient to capture hate speech.

https://doi.org/10.3390/mca31030088

Table 3. Obtained results of each model (mean ± standard deviation).

		Macro AVG		
Representation	**Algorithm**	**Precision**	**Recall**	**F1**
-	One-R	0.1728 ± 0.0000	0.3333 ± 0.0000	0.2276 ± 0.0000
ASCII	MLP	0.2050 ± 0.0900	0.3471 ± 0.0435	0.2537 ± 0.0725
D2V	KNN	0.4677 ± 0.0634	0.4338 ± 0.0074	0.4106 ± 0.0095
D2V	LR	0.5717 ± 0.0894	0.5235 ± 0.0103	0.5093 ± 0.0122
TF-IDF	RF	0.5440 ± 0.0072	0.5720 ± 0.0076	0.5575 ± 0.0074
TF-IDF	DT	0.5348 ± 0.0266	0.5282 ± 0.0205	0.5294 ± 0.0229
BERT	SVM	0.5383 ± 0.0051	0.5645 ± 0.0053	0.5507 ± 0.0051
BoW	NB	0.8861 ± 0.0040	0.5857 ± 0.0043	0.5759 ± 0.0039
BERT	SC	0.6930 ± 0.0628	0.6160 ± 0.0293	0.6158 ± 0.0424

On the other hand, there is a group of configurations based on lexical representations, including TF-IDF + DT, TF-IDF + RF, and BoW + NB; these perform very similarly, since the simple frequency-based models are expressed using specific vocabulary, while BERT + SC exhibits the best performance, achieving an F1 value of 0.6158.

To assess whether the performance differences between the models are statistically significant, the non-parametric Friedman test was applied to the 10-fold cross-validation results. According to this test, a Friedman statistic of $x^2 = 67.73$ was obtained. With a p-value = 4.2366×10^{-12} ($p < 0.001$), the differences between all models are significant. Pairwise comparisons showed that the SC + BERT model significantly outperformed all other classifiers. Table 4 shows the pairwise comparisons based on Wilcoxon (BoW + NB **(1)**; D2V + KNN **(2)**; D2V + LR **(3)**; ASCII + MLP **(4)**; BERT + SVM **(5)**; TF-IDF + DT **(6)**; TF-IDF + RF **(7)**; BERT + SC **(8)**) signed-rank test [26].

Table 4. Pairwise comparisons.

	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
(1)	1	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0137
(2)	0.0020	1	0.0020	0.0039	0.0020	0.0020	0.0020	0.0020
(3)	0.0020	0.0020	1.0000	0.0020	0.0020	0.0020	0.0020	0.0020
(4)	0.0020	0.0039	0.0020	1	0.0020	0.0020	0.0020	0.0020
(5)	0.0020	0.0020	0.0020	0.0020	1	0.273	0.488	0.0020
(6)	0.0020	0.0020	0.0020	0.0020	0.273	1	0.0020	0.0020
(7)	0.0020	0.0020	0.0020	0.0020	0.0488	0.0020	1	0.0020
(8)	0.0137	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	1

Figure 3 shows the degree of significance of BERT + SC over the other models by means of (*). Moreover, the BERT+SC model ranks first, significantly outperforming SVM+BERT with an average difference of 0.09 points in F1. On the other hand, the models with traditional representations (BoW + NB, TF-IDF + RF, TF-IDF + DT) show intermediate and consistent performance, with average values between 0.49 and 0.54. D2V embeddings combined with linear (LR) and distance-based (KNN) classifiers performed worse. Finally,

https://doi.org/10.3390/mca31030088

MLP + ASCII was significantly worse (mean = 0.20), probably due to the inability of the ASCII representation to capture relevant semantic information.

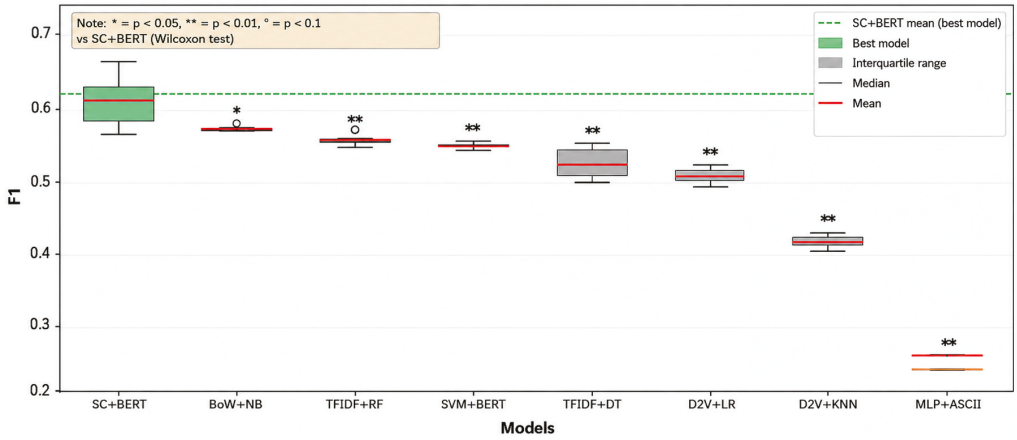


Figure 3. Statistical significance between models.

Finally, Figure 4 shows a comparison between the precision and recall measures, with recall on the Y-axis and precision on the X-axis. The dashed diagonal line represents a balance between both metrics: points below the line indicate higher precision than recall, whereas points above it indicate the opposite. Although BERT + SC and BoW + NB achieve the best F1-score, they show an imbalance between precision and recall, suggesting difficulties in capturing hate speech nuances without incurring false negatives. This behavior may be related to the linguistic characteristics of the dataset, where hate speech is often expressed implicitly through sarcasm, ambiguity, or context-dependent language (see examples of Section 3.1), making it harder to detect.

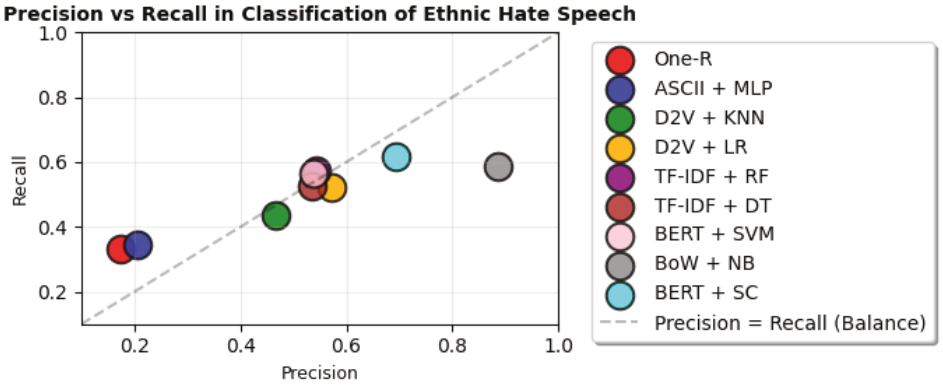


Figure 4. Comparison between precision and recall of generated models.

In contrast, the combinations BERT + SVM, TF-IDF + DT, TF-IDF + RF, D2V + LR, and D2V + KNN are located close to the diagonal line, indicating a balance between precision and recall. Despite their F1-scores being lower than BERT + SC, these models do not favor one type of error over another, suggesting more stable and neutral behavior. On the other hand, although ASCII + MLP and One-R have the lowest performance, their balance indicates that they tend to classify more instances as positive, increasing recall at the expense of precision. This may be influenced by the limited representational capacity of these approaches to capture the lexical and semantic variability of hate speech.

Overall, these results indicate that no single model is universally optimal, highlighting the inherent complexity of ethnic hate speech detection, which is shaped by linguistic variability, implicit expressions, and contextual dependencies. This suggests that selecting a

https://doi.org/10.3390/mca31030088

model should depend on the specific application requirements, particularly the importance of balancing precision and recall.

5. Conclusions

The rise in discrimination against Indigenous groups in México and the persistence of discriminatory structures with colonial roots underscore. The urgent need to develop technological tools that contribute to the detection and mitigation of ethnic hate speech. This work represents an important first step by providing the following:

- A Spanish computational resource designed to represent and analyze the distinctive characteristics of ethnic hate speech within the Mexican context.
- Robust benchmarks that establish the performance of classifiers and allow quantifying the added value of more complex models.
- An analysis of the dataset that reveals a predominantly explicit character, but with important contextual nuances.
- The benchmarks were established using a set of classification algorithms combined with textual representations at different linguistic levels.

The obtained results demonstrate that classifying ethnic hate is achievable using computational techniques. All evaluated models significantly outperform the dummy classifier One-R (F1 = 0.2276). This validates the quality of the labeling and confirms that the established classes ("No *Hate*," "*Hate*," "*Unrelated Hate*") exhibit distinctive linguistic patterns that the algorithms can learn. The algorithms' performance indicates that a substantial portion of the hate is manifested through recurring offensive vocabulary and direct insults. This reflects that, in the Mexican digital context, users frequently resort to explicit terms to express ethnic discrimination. However, BERT's performance suggests that is required a deep contextual understanding. These examples likely include hate, sarcasm, irony, or implicit cultural references that purely lexical models cannot capture.

While BERT obtained the best performance (see Table 3), it is important to evaluate Spanish versions of the model to address the imbalance between recall and precision. Although multilingual transformer models have been implemented, it is essential to explore architectures tailored to Spanish, since the linguistic and cultural nuances of ethnic hate in México may be better represented by these approaches. Thus, we propose evaluating pretrained models such as BERTín, RoBERTa, and BETO, and comparing their performance against their multilingual versions.

The statistical significance test demonstrated that BERT + SC outperforms the other models evaluated in the task of detecting ethnic hate speech in the Mexican context p-value = 4.2366×10^{-12}. This level of significance provides statistical evidence supporting the claim that incorporating deep contextual representations substantially improves discrimination capabilities compared to purely lexical or less complex approaches.

On the other hand, even though the evaluation does not consider accuracy and is based on robust metrics such as precision, recall, and F1-score; the *unrelated hate* minority class (4%) continues to pose a significant challenge. To address this limitation, the following lines of future work are proposed:

- Advanced sampling techniques: Explore synthetic oversampling strategies such as SMOTE (Synthetic Minority Over-sampling Technique), as well as intelligent undersampling methods that preserve the original data distribution.
- Generation of synthetic data with generative models: Employ variants of autoencoders (VAE), conditional Generative Adversarial Networks (GANs), or Large Language Models (LLMs) to generate artificial examples of the Unrelated Hate class under

https://doi.org/10.3390/mca31030088

human supervision, ensuring the semantic consistency of hate content unrelated to ethnicity.
- Evaluation using learning curves by class: Perform a systematic analysis of the model's performance based on the training sample size to determine the saturation point of each class and establish minimum thresholds of required data.

We also propose implementing fine-grained labeling in EnthoHate to identify the motives behind hate speech, such as clothing style, accent, skin tone, and other factors, as well as their intersection with other forms of violence. This would not only allow the classification of hate's presence but also its characterization. This more granular and socially informed approach will not only improve the technical performance of the models but also help generate more useful tools for social researchers, human rights organizations, and government entities dedicated to preventing discrimination in México.

Finally, we seek to extend the models performance analysis using a more number of executions, in order to strengthen the validity of the Wilcoxon signed-rank test. This would allow for the assessment of the stability of statistically significant differences.

Author Contributions: Conceptualization and methodology, V.N.-M., Y.L. and J.R.-S.; software, validation, formal analysis and writing—original draft preparation, V.N.-M., J.R.-S. and M.T.Z.; investigation and data curation, V.N.-M.; writing—review, editing and supervision, Y.L., R.A.G.-H. and Á.H.-C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Informed consent was obtained from all participants regarding the study labeling. Written informed consent was obtained from all participants for the publication of this article.

Data Availability Statement: The dataset presented in this article is not readily available because the data are part of an ongoing study. Requests to access the dataset should be directed to veronica.nerimendoz@gmail.com.

Acknowledgments: We thank Ángel Baron García, for their technical assistance and dedication. We also thank Tania Iliana Santiago and David Eduardo Silva Carmona for their support in the development of labeling guidelines. During the preparation of this manuscript, the authors used [Gemini 3 Flash Image] for the purpose of generating Figure 2. The authors have reviewed and edited the result and assume full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Rawat, A.; Kumar, S.; Samant, S.S. Hate speech detection in social media: Techniques, recent trends, and future challenges. *Wiley Interdiscip. Rev. Comput. Stat.* **2024**, *16*, e1648. [CrossRef]
2. González Gómez, J. La etnofobia en México y sus implicaciones en la educación superior. *Univ. Habana* **2024**, *300*, e8784.
3. Ullah, F.; Zamir, M.; Arif, M.; Ahmad, M.; Felipe-Riveron, E.; Gelbukh, A. Fida@ dravidianlangtech 2024: A novel approach to hate speech detection using distilbert-base-multilingual-cased. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 85–90.
4. Mullah, N.S.; Zainon, W.M.N.W. Advances in Machine Learning Algorithms for Hate Speech Detection in Social Media: A Review. *IEEE Access* **2021**, *9*, 88364–88376. [CrossRef]
5. Gandhi, A.; Ahir, P.; Adhvaryu, K.; Shah, P.; Lohiya, R.; Cambria, E.; Poria, S.; Hussain, A. Hate speech detection: A comprehensive review of recent works. *Expert Syst.* **2024**, *41*, e13562. [CrossRef]
6. Zamir, M.T.; Tash, S.; Ahani, Z.; Gelbukh, A.; Sidorov, G. Lidoma@DravidianLangTech 2024: Identifying Hate Speech in Telugu Code-Mixed: A BERT Multilingual. In *Proceedings of the Fourth Workshop on Speech, Vision, and Language Technologies for Dravidian Languages*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2024; pp. 101–106.
7. Basile, V.; Bosco, C.; Fersini, E.; Nozza, D.; Patti, V.; Rangel, F.; Rosso, P.; Sanguinetti, M. SemEval-2019 Task 5: Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter. In *NAACL HLT 2019—International Workshop on Semantic*

https://doi.org/10.3390/mca31030088

Evaluation, SemEval 2019, Proceedings of the 13th Workshop; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; pp. 54–63. [CrossRef]
8. Rodríguez-Sánchez, F.; Carrillo-De-Albornoz, J.; Plaza, L.; Gonzalo, J.; Rosso, P.; Comet, M.; Donoso, T. Overview of EXIST 2021: SEXism Identification in Social NeTworks. *Proces. Leng. Nat.* **2021**, *67*, 195–207.
9. Rodríguez-Sánchez, F.; Carrillo-De-Albornoz, J.; Plaza, L.; Mendieta-Aragón, A.; Marco-Remón, G.; Makeienko, M.; Plaza, M.; Gonzalo, J.; Spina, D.; Rosso, P. Overview of EXIST 2022: SEXism Identification in Social NeTworks. *Proces. Leng. Nat.* **2022**, *69*, 229–240.
10. Bel-Enguix, G.; Gómez-Adorno, H.; Ojeda-Trueba, S.; Sierra, G.; Barco, J.; Lee-Romero, E.; Dunstan, J.; Manrique, R. Overview of HOMO-LAT at IberLEF 2025: Human-Centric Polarity Detection in Online Messages Oriented to the Latin American-Speaking LGBTQ+ PopulaTion. *Proces. Leng. Nat.* **2025**, *75*, 413–424.
11. Vásquez, J.; Andersen, S.T.; Bel-Enguix, G.; Ojeda-Trueba, S.L.; Gómez-Adorno, H. HOMO-MEX: A Mexican Spanish Annotated Corpus for LGBT+phobia Detection on Twitter. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2023; pp. 202–214.
12. Lai, M.; Celli, F.; Ramponi, A.; Tonelli, S.; Bosco, C.; Patti, V. HaSpeeDe3 at EVALITA 2023: Overview of the Political and Religious Hate Speech Detection Task. In *CEUR Workshop Proceedings*; CEUR-WS: Bonn, Germany, 2023.
13. Neri-Mendoza, V.; Ledeneva, Y.; Rojas-Simón, J.; Santos-Bobadilla, Y.A.; García-Hernández, R.A. Pilot Test of Ethnic Hate Speech Detection in Mexico. In *Mexican International Conference on Artificial Intelligence*; Springer: Cham, Switzerland, 2026; pp. 168–176. [CrossRef]
14. Consejo Nacional para Prevenir la Discriminación (CONAPRED). Informe Especial 01/2024 Sobre Narrativas Discriminatorias. CONAPRED: Mexico City, Mexico, 2024. Available online: https://www.conapred.gob.mx/wp-content/uploads/2024/09/Inform-Especial-NArrativas-Discriminatorias-04.09.2024.rev_.juridico-ok.pdf (accessed on 15 February 2026).
15. Mato, D. Violencia verbal y otras prácticas de discriminación étnico-racial hacia estudiantes indígenas y afrodescendientes en universidades de América Latina. *Rev. Latinoam. Estud. Educ.* **2025**, *55*, 197–234. [CrossRef]
16. Escalante, Y. *Derechos de los Pueblos Indígenas y Discriminación Étnica o Racial*; Consejo Nacional para Prevenir la Discriminación: Mexico City, Mexico, 2009. Available online: https://teams.microsoft.com/l/message/19:3e7c9f3d-60fb-45ad-9fff-6ec53fe194d6_8c1b6d40-148c-490c-8457-0083e8490793@unq.gbl.spaces/1780280191939?context=%7B%22contextType%22%3A%22chat%22%7D (accessed on 15 February 2026).
17. Poletto, F.; Basile, V.; Sanguinetti, M.; Bosco, C.; Patti, V. Resources and benchmark corpora for hate speech detection: A systematic review. *Lang. Resour. Eval.* **2021**, *55*, 477–523. [CrossRef]
18. Allam, H.; Makubvure, L.; Gyamfi, B.; Graham, K.N.; Akinwolere, K. Text Classification: How Machine Learning Is Revolutionizing Text Categorization. *Information* **2025**, *16*, 130. [CrossRef]
19. Mao, J.; Shi, H.; Li, X. Research on multimodal hate speech detection based on self-attention mechanism feature fusion. *J. Supercomput.* **2025**, *81*, 28. [CrossRef]
20. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 4171–4186.
21. Aragón, M.E.; Álvarez-Carmona, M.; Montes-Y-Gómez, M.; Escalante, H.J.; Villaseñor-Pineda, L.; Moctezuma, D. Overview of MEX-A3T at IberLEF 2019: Authorship and Aggressiveness Analysis in Mexican Spanish Tweets. *CEUR Workshop Proc.* **2019**, *2421*, 478–494.
22. Fleiss, J.L.; Cohen, J. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educ. Psychol. Meas.* **1973**, *33*, 613–619. [CrossRef]
23. Rojas-Simón, J.; Ledeneva, Y.; García-Hernández, R.A. Classification of human and machine-generated texts using lexical features and supervised/unsupervised machine learning algorithms. In *Mexican Conference on Pattern Recognition*; Springer: Cham, Switzerland, 2024; pp. 331–341.
24. Rojas-Simón, J., Ledeneva, Y., García-Hernández, R. A. A Dimensionality Reduction Approach for Text Vectorization in Detecting Human and Machine-generated Texts. *Comput. Sist.* **2024**, *28*, 1919–1929. [CrossRef]
25. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on International Conference on Machine Learning—Volume 32 (ICML'14)*; Proceedings of Machine Learning Research: Cambridge, MA, USA, 2014.
26. Wilcoxon, F. Individual comparisons by ranking methods. *Biom. Bull.* **1945**, *1*, 80–83. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

https://doi.org/10.3390/mca31030088

Mathematical and Computational Applications

MDPI

Article

Learning Bi-Objective Bayesian Network Structure from Data Using Particle Swarm Optimization

Vicente-Josué Aguilera-Rueda [1,*], Nicandro Cruz-Ramírez [2], Efrén Mezura-Montes [2] and Ricardo Vilalta [3]

[1] Faculty of Accounting and Administration, Universidad Veracruzana, Xalapa 91090, Mexico
[2] Artificial Intelligence Research Institute, Universidad Veracruzana, Xalapa 91097, Mexico; ncruz@uv.mx (N.C.-R.); emezura@uv.mx (E.M.-M.)
[3] Center for Science, Technology, Engineering, and Mathematics, University of Austin, Austin, TX 78712, USA; rvilalta@uaustin.org
* Correspondence: vaguilera@uv.mx

Abstract

This paper proposes a bi-objective approach to address the data-driven Bayesian network structure learning problem. The objectives considered for optimization are minimum description length (MDL) and misclassification. An algorithm based on the well-known multi-objective particle swarm optimization (MOPSO), called MOPSO-BN, is used to tackle the bi-objective learning problem. Furthermore, a strategy for preference handling from the Pareto front that selects the nearest model to a reference point is proposed. Finally, this bi-objective approach is compared against a single-objective approach. Numerical results show how this multi-objective approach is highly efficient at competitive Bayesian networks with a balanced trade-off between MDL and misclassification.

Keywords: Bayesian network; minimum description length; misclassification; structure learning problem

1. Introduction

Bayesian networks (BNs) are probabilistic graphical models that represent dependencies among random variables via a directed acyclic graph (DAG) [1]. Their ability to simultaneously support prediction, classification, inference, and visual interpretation of causal relationships has made them a widely adopted tool in many applications, including medicine, fault diagnosis, and decision-making [2].

Learning a BN structure from data is a combinatorial optimization problem of considerable difficulty. The number of possible DAG structures grows super-exponentially with the number of variables n according to Robinson's formula [3]: with $n = 10$ variables, there are already more than 4×10^{18} candidate structures, and exhaustive evaluation becomes completely infeasible beyond $n \approx 6$. This explosive growth of the search space is the central computational challenge motivating the use of metaheuristics in BN structure learning. Existing approaches can be broadly divided into constraint-based methods [2], which use conditional independence tests to prune the search space, and search-and-score methods [4–6], which combine a scoring function with a search strategy to navigate the DAG space. This paper focuses on the second category.

Among scoring functions, the minimum description length (MDL) principle [7] has been widely adopted for BN structure learning. MDL balances model fit against structural complexity through a penalized log-likelihood, favoring compact models that avoid overfitting. However, a fundamental limitation of MDL as a sole optimization criterion is that it

measures the global reconstruction error of the BN over all variables in the domain, rather than the local prediction error for the class variable [8]. As a consequence, the BN structure that minimizes MDL is not necessarily the one that minimizes misclassification rate, and vice versa. This discrepancy defines a clear technical gap: existing search-and-score methods that optimize MDL alone cannot simultaneously guarantee good classification performance, yet no prior work has explicitly and jointly optimized these two conflicting objectives in a principled bi-objective framework.

This paper addresses this gap by proposing MOPSO-BN, a multi-objective particle swarm optimization algorithm that simultaneously minimizes MDL and misclassification rate as two conflicting objectives. The key contributions of this work are as follows: (1) a graphical demonstration of the discrepancy between MDL and misclassification through exhaustive evaluation of all possible DAG structures in low-dimensional datasets; (2) the MOPSO-BN algorithm, which produces a Pareto front of non-dominated BN structures that explicitly trade off data compression against classification accuracy; (3) a preference-handling strategy that selects the solution nearest to a reference point from the Pareto front; and (4) a comparative evaluation against Naïve Bayes, TAN, and hill-climbing BNs on 20 real-world datasets. Unlike existing single-objective methods, MOPSO-BN provides the end-user with a set of structurally interpretable BN classifiers that balance both objectives, preserving the representational and inferential advantages of BNs alongside competitive classification performance.

The remainder of this paper is organized as follows: Section 2 describes the proposed method. Section 3 presents experiments and results. Section 4 summarizes the findings and proposes future work.

1.1. Literature Survey

Many methods have been proposed to learn the structure of BNs for classification purposes. One approach is based on the Naïve Bayes (NB) classifier. In [8], for instance, the authors propose the tree-augmented NB (TAN) approach, which approximates the relationship between attributes using a tree structure. In a TAN-based model, the class variable has no parents, and each attribute has as parents the class variable and at most another attribute; this approach does not consider the possibility of variables correlated with several variables. The work of Cerquides and Mántaras [9] improves TAN-style classifiers by defining a conjugate distribution called a decomposable distribution. Other authors have proposed methods to alleviate shortcomings of TAN-based classifiers [10–14].

The work of Grossman and Domingos [15] proposes a method for learning BN classifiers by maximizing the conditional likelihood of the class variable; they conclude that their approach produces better class probability estimates than maximum-likelihood approaches such as NB and TAN. Nevertheless, since the learned structures are not shown, it is unclear what the models' complexity is (i.e., the number of arcs). The work of Madden [16] compares NB, TAN, and BNs learned using the K2 algorithm [17] as a search procedure with the BDeu metric [4] (that is, a refinement of the K2 metric) as a scoring function, versus a BN learned using hill-climbing search with MDL as a score function. The results show that BN-K2 classifiers significantly outperform NB and are comparable to TAN's performance; the advantages of BNs over TAN point to representational power and complexity. Previous work shows that even if a BN exhibits good classification performance, other algorithms can outperform it. Nevertheless, improving the performance of BN classifiers is a topic with clear potential for research.

The work of Gheisari et al. [18] introduces two automata-based algorithms for BN learning. First, to measure the quality of their constructed networks, they use the BIC metric, which is considered equivalent to MDL [19]. Additionally, they analyze the classification

https://doi.org/10.3390/mca31030094

accuracy of their proposed algorithm on 25 datasets by using a scoring function called classification rate. However, they do not show the models obtained, and the learning process does not integrate searching for optimal structure with assessing classification performance.

Regarding the learning problem of a BN structure using evolutionary algorithms (EAs), we list some representative examples. The work of Fang et al. [20] presents an algorithm for learning the structure of Bayesian networks using a genetic algorithm, which employs concepts of Markov blankets and equivalence in Bayesian network structures, and the Bayesian Dirichlet equivalent uniform (BDeu) scoring metric. The authors claim that their proposed method outperforms on datasets with many variables, but they do not show the induced structures.

In [21], the performance of a genetic algorithm (GA) is compared against two novel population-based, stochastic search algorithms; three different scoring functions are used, and the results show that their proposed algorithm can find structures similar to the original one when the order of the nodes is taken into account. In [22], a novel algorithm based on immune binary particle swarm optimization (PSO) and MDL as fitness function is proposed. Experiments show that the fitness function quality is superior in PSO compared to a GA. The work of Kuo et al. [23] applies MDL in a PSO-based algorithm to learn BN structures. They conclude that their proposed method (PSO-MDL) outperforms support vector machines (SVMs) and Weka's BayesNet procedures [24] (in terms of classification accuracy). In [25], a hybrid algorithm combining the maximal information coefficient with a binary PSO is proposed. The experimental results show that the algorithm performs better than five other state-of-the-art algorithms without a given node ordering.

In [26], the authors proposed a bi-objective evolutionary Bayesian network structure-learning method. The authors argue that the score-based approach of learning is bi-objective due to the two terms of the scores, log-likelihood and complexity, so they proposed a structure learning method via skeleton constraints (BBS) for medium-scale networks. However, BBS and related methods focus on recovering the dependency structure among variables rather than on optimizing classification performance, a key distinction from our proposed MOPSO-BN.

Our own previous work [27,28], presents a bi-objective method, named the nondominated sorting genetic algorithm for learning Bayesian networks (NS2BN), which is based on the well-known NSGA-II algorithm. To avoid complex models that overfit the data, NS2BN employs two MDL terms as objectives: the log-likelihood, which measures goodness-of-fit, and a term that penalizes complex models. That problem is known as the bias–variance dilemma. The works of [26–28] are focused on learning structures that reflect the relationships between variables but do not necessarily focus on BNs for classification tasks.

The work of Ross and Zuviria [29] uses a multi-objective genetic approach to learn dynamic BNs from data, balancing likelihood and complexity. This work focuses on modeling biological phenomena that typically involve low-connectivity networks with at most four parents. They propose a learning methodology that allows an intuitive, user-defined structure for solution networks. In general, works based on evolutionary algorithms (EAs) and MDL do not report the learned network and are mainly designed to recover the original network, except for the approach of Kuo et al. [23], which uses classification accuracy as the performance metric.

Recent work has continued to explore diverse strategies for BN structure learning, combining heuristic search, regularization techniques, and hyper-heuristic frameworks. The work of Sun et al. [30] proposes OP-PSO-DE, a hybrid algorithm that combines Particle Swarm Optimization (PSO) and Differential Evolution (DE) with opposition-based learning to accelerate convergence in BN structure learning. The key idea is to generate opposite solutions alongside regular solutions at each iteration, effectively halving the search space

https://doi.org/10.3390/mca31030094

and efficiently guiding particles toward the global optimum. The authors apply a binary mutation with a probability difference operator to handle the discrete nature of the DAG search space, and adopt an adaptive inertia-weight strategy based on the swarm's success rate. The experimental results on CANCER, ASIA, and INSURANCE networks show that OP-PSO-DE converges significantly faster than PC-PSO and BNC-PSO, achieving competitive BIC scores, particularly in small-sample regimes. However, the authors acknowledge that premature convergence occurs as network scale increases, limiting its applicability to large networks.

The work of Fallahnejad et al. [31] introduces an Elastic Net-based K2 algorithm (EN-K2) for BN structure learning that addresses the K2 algorithm's sensitivity to node ordering. Their approach first estimates the Markov Blanket (MB) of each variable using Elastic Net-penalized regression, which balances L1 and L2 regularization to improve stability in the presence of correlated predictors. A novel dependency-based scoring function is then applied to identify candidate parents within each estimated MB through pairwise comparisons, and the resulting candidate sets are passed as input to the K2 algorithm. Experiments on ASIA, ALARM, CAR, and MIDWAY datasets demonstrate that EN-K2 detects more correct edges and substantially reduces reversed and additional edges compared to the Lasso-based and Ko and Kim methods, particularly as sample size increases.

On the heuristic and metaheuristic side, recent work has proposed novel search strategies for discrete BN structure learning. Yang et al. [32] proposed to use scatter search to learn BN structures by combining diversification and intensification strategies in the DAG space, showing competitive results against hill-climbing and genetic algorithm baselines. He et al. [33] introduced a neighboring complete-node-ordering search algorithm that determines the topological ordering of variables and uses hill-climbing to find the best network structure within each ordering. Liu et al. [34] proposed an improved Harris Hawks Optimization algorithm with genetic operators for BN structure learning, demonstrating competitive BIC scores on standard benchmark networks. Wang et al. [35] proposed a novel skeleton learning method based on dynamic thresholds and triangle-breaking combined with hill-climbing, achieving better structural accuracy than standard hybrid approaches. These reports confirm that heuristic search remains an active and productive research direction for BN structure learning, and that population-based methods continue to offer advantages over purely greedy approaches in terms of solution quality and diversity.

The work of Dang et al. [36] proposes a hyper-heuristic algorithm for BN structure learning that integrates both soft and hard structural constraints. Soft constraints are derived from a feature-selection-based local learning method that mines conditional independence information to define restricted search spaces, while hard constraints are provided by expert domain knowledge. These constraints are incorporated into an Exponential Monte Carlo with Counter (EMCQ) hyper-heuristic framework that manages a library of low-level operators, including mutation, neighborhood hill-climbing, learning, restart, and expert-knowledge operators. The experimental results on six standard BN benchmarks from the BNLEARN repository—ranging from 37 to 441 nodes—show that EMCQ-HH achieves superior BIC scores and F1-scores compared to PC-Stable, GS, F2SL, MMHC, and BNC-PSO, demonstrating stronger robustness in large-scale and limited-data scenarios.

Beyond algorithmic advances in BN structure learning, recent work has demonstrated the practical value of Bayesian probabilistic modeling in real-world engineering applications, further motivating the dual objectives pursued in MOPSO-BN.

Yan et al. [37] propose a probabilistic fault diagnosis framework for bridge expansion joints in small and medium bridges using a convolutional neural network with Bayesian deep ensemble (CNN-BDE). Their method quantifies predictive uncertainty via a weighted

https://doi.org/10.3390/mca31030094

ensemble of independently trained CNNs, with each model's contribution determined by its negative log-likelihood on a validation set. An adaptive inter-class variance regularization term is introduced to enhance feature discrimination under noisy, small-sample conditions. Experiments on real-world acoustic data show that CNN-BDE achieves higher diagnostic accuracy and better uncertainty calibration than MC dropout, standard deep ensembles, and Gaussian process classifiers. This work illustrates how Bayesian ensemble strategies can yield robust and interpretable probabilistic outputs in safety-critical classification tasks with limited labeled data—a challenge structurally analogous to BN structure learning in small or medium-dimensional datasets.

Li et al. [38] address the problem of predicting typhoon-induced dynamic responses on long-span bridges by proposing CNN-BiLSTM-EDE, an explainable deep ensemble model that integrates convolutional and bidirectional LSTM layers within a probabilistic framework. The deep ensemble scheme produces a conditional predictive distribution over bridge responses, while Shapley additive explanations (SHAP) are incorporated to reveal the marginal contributions of input features, identifying the mean wind speed and wind direction angle as the most influential factors. Validated on decade-long typhoon datasets from a kilometer-scale bridge, the model outperforms four probabilistic benchmark methods in both prediction accuracy and uncertainty quantification. This study highlights the growing importance of combining predictive accuracy with structural interpretability and uncertainty quantification in data-driven engineering models—properties that are also central to the motivation of MOPSO-BN, which seeks BN structures that simultaneously optimize classification performance and model compactness.

In general, the recent literature confirms two converging trends: (1) the increasing use of population-based and metaheuristic algorithms for BN structure learning in discrete domains, where gradient-based continuous methods are not directly applicable; and (2) a growing interest in multi-objective formulations, as evidenced by works such as [26] and our own previous contributions [27,28], that go beyond single-score optimization. Our proposed MOPSO-BN is positioned at the intersection of both trends, contributing a principled bi-objective PSO-based framework that, unlike existing multi-objective BN learning methods, explicitly optimizes classification performance as a second objective alongside MDL, exploiting both the representational and classification capacities of BNs simultaneously.

1.2. Main Contribution

Even though there are multiple approaches to tackle the data-driven Bayesian network structure learning problem, existing solutions have not deeply explored the capacities of BNs, their graphical power for explaining the phenomenon under investigation, the relations among the random variables, or their ability to be used for classification purposes.

The core objective of this work is to propose an algorithm, MOPSO-BN, based on the well-known multi-objective particle swarm optimization (MOPSO), to identify a set of competitive Bayesian networks by minimizing both objectives: data compression using MDL and misclassification.

The choice of PSO as the underlying metaheuristic is motivated by complementary reasons. First, PSO has demonstrated competitive performance on BN structure-learning problems [22,23,30], demonstrating its ability to efficiently explore large combinatorial search spaces while avoiding premature convergence. Second, PSO is particularly well-suited for population-based search in binary spaces since its velocity-position update mechanism can be naturally adapted to handle adjacency matrix representations of BN candidates through a sigmoid transformation [39], as described in Section 2.2.3. Third, compared to other metaheuristics such as genetic algorithms (GAs), PSO requires fewer operator-specific parameters and is simpler to implement while maintaining competitive

https://doi.org/10.3390/mca31030094

search performance [23]. Fourth, and most critically for our bi-objective formulation, PSO extends naturally to multi-objective settings through the MOPSO framework [40,41], where the concepts of personal best (*pbest*) and global best (*gbest*) can be redefined using Pareto dominance and hypervolume contribution, respectively, enabling effective approximation of the Pareto front. This last property distinguishes PSO from simpler single-objective heuristics and makes it a principled choice for simultaneously optimizing MDL and misclassification. Finally, the work of Sun et al. [30] further confirms that PSO-based algorithms for BN structure learning remain competitive against more recent alternatives, reinforcing the suitability of PSO as the backbone of our proposed method.

The main contributions of this paper are summarized as follows:

- The discrepancy between the MDL and the classification error was graphically demonstrated through the exhaustive evaluation of the set of possible BN candidates in low-dimensional instances.
- A multi-objective bio-inspired algorithm called MOPSO-BN was proposed to exploit the advantages of the shortest description of the original data, using MDL and expected classification.
- A set of resultant networks was obtained, which presents a trade-off between MDL and misclassification. They can be helpful in inference, classification, and explaining the relationships among the variables under investigation.
- A strategy to select a BN from a set of solutions in a Pareto front was explored, and the obtained solution was compared with a single objective algorithm.

In summary, our work suggests a bi-objective approach to addressing the BN structure learning problem, with objectives including MDL and misclassification rate.

The metaheuristic used in this work is based on PSO due to its well-documented effectiveness in multi-objective combinatorial optimization, its natural compatibility with binary solution representations for DAG-based BN structures, and its straightforward extension to the MOPSO framework, which provides a principled mechanism for Pareto front approximation through hypervolume-based leader selection [40].

2. Proposed Method

This section describes the proposed method and summarizes key fundamentals of BNs, including the multi-objective and non-dominated concepts.

2.1. Bayesian Networks

Formally, a BN [1] is represented as a pair (G, Θ), where G is a directed acyclic graph (DAG), which can be expressed by (U, E_G); U is the set of nodes or random variables (denoted by $\{X_1, \ldots, X_n\}$), and E_G is the set of arcs that represents the probabilistic relationships among these variables. The parents of X_i are denoted by PA_i; X_i is independent of its non-descendant variables given its parents. We denote as Θ the conditional probability distributions $P(X_i|PA_i)$ for each X_i. The joint probability distribution can be recovered from local conditional probability distributions as $P(X_1, \ldots, X_n) = \prod_{i=1}^{n} P(X_i|PA_i)$.

2.2. Bi-Objective Bayesian Network Structure Learning

We divide the proposed method into three levels, shown in Figure 1, and described in detail next.

https://doi.org/10.3390/mca31030094

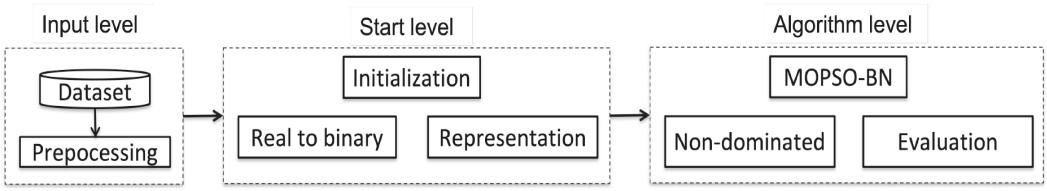


Figure 1. Diagram of the proposed method.

2.2.1. Input Level

As input to our proposed algorithm, we use a dataset represented as an $m \times n$ matrix, where m is the number of cases (or examples) and n is the number of attributes. As a preprocessing step, continuous variables must be discretized; for this, we use the Class-Attribute Interdependence Maximization (CAIM) algorithm [42], implemented in the Weka data mining software [24]. We delete records with missing data.

2.2.2. Start Level

Our method is a population-based algorithm that is part of swarm intelligence (SI). The population is called a *swarm*: a set of N particles or solutions, each randomly generated. In our case, particles represent BN candidates. We use an adjacency matrix to represent each candidate structure (particle), as shown in Figure 2.

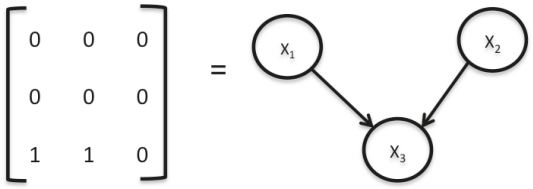


Figure 2. Example of an adjacency matrix and its corresponding BN according to [28].

Every BN candidate must be a DAG, meaning it cannot contain any directed cycles. If the optimization process generates a network with one or more cycles, it must be repaired before evaluation. We use a repair operator that modifies the adjacency matrix by replacing specific values when a cycle is identified. Three types of cycles can occur:

- **Self-cycle:** This happens when a node has a directed arc pointing to itself, i.e., a 1 in the diagonal of the adjacency matrix. The repair strategy simply replaces that 1 with a 0, eliminating the self-loop.
- **Bi-directional cycle:** This occurs when two nodes mutually influence each other, meaning there is an arc from node v_i to node v_j and simultaneously an arc from v_j to v_i. Since this creates a cycle of length 2, the repair operator randomly removes one of the two conflicting arcs by setting the corresponding matrix entry to 0.
- **General cycle:** This involves a directed path of length greater than 2, where a sequence of nodes $v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_k \rightarrow v_1$ forms a closed loop. To resolve it, the repair operator first identifies the path responsible for the cycle and then randomly removes one of the offending arcs, breaking the cycle while preserving as much of the network structure as possible.

The cycle repair mechanism acts as a constraint projection operator: it maps each infeasible position back into the feasible DAG space by removing the minimum number of arcs needed to eliminate the detected cycle [43]. Although the repair step creates a transient discrepancy between the stored velocity $\vec{v}_i$ and the repaired position $\vec{x}_i$, this discrepancy is self-correcting: in the next velocity update (Equation (3)), both attraction terms $(\vec{pbest}_i - \vec{x}_i)$ and $(\vec{gbest} - \vec{x}_i)$ are recomputed relative to the repaired position, so the particle automatically re-aligns its trajectory toward the personal and global best solutions

https://doi.org/10.3390/mca31030094

from its new feasible structure [39]. The random selection of which arc to remove when multiple arcs contribute to a cycle is a deliberate choice that introduces controlled structural diversity into the repair process, preventing all particles from following identical repair trajectories and thus reducing the risk of premature convergence. As the swarm converges and velocity magnitudes decrease—driven by the inertia weight $w = 0.4$, which favors exploitation over exploration in later stages—cycle violations become less frequent, and the influence of the repair operator on the search trajectory diminishes naturally. The repair mechanism, therefore, preserves the fundamental PSO dynamics while ensuring that each evaluated candidate corresponds to a valid DAG.

2.2.3. Algorithm Level

Inspired by [40,41], we propose a bi-objective particle swarm optimization algorithm, MOPSO-BN. In the next sections, we provide multi-objective concepts relevant to our proposed algorithm. Section 2.4 presents the objectives to be optimized, followed by an explanation of MOPSO-BN.

2.3. The Multi-Objective and Non-Dominated Concepts

According to Deb and Kalyanmoy [44], a multi-objective optimization problem (MOOP) can be seen as a search problem that aims to minimize or maximize two (or more) objectives that are in conflict. A MOOP can be formalized as $\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_l(\vec{x})]$, where $\vec{x} = [x_1, \ldots, x_n]$ is a n-variable decision vector, $\vec{f}$ is the set of objective functions to be minimized or maximized, and l is the number of objectives.

We base our proposed algorithm on the concept of dominance [40]. Before moving forward, we provide the following definition. A solution x_1 dominates a solution x_2 (denoted by $x_1 \preceq x_2$) if the solution x_1 is not worse than x_2 in all objectives and it is better than x_2 in at least one objective. When we are dealing with MOOPs, there is no single optimal solution; instead, there is a set of alternative solutions that have no solution that dominates them all when all objectives are considered. The set of non-dominated solutions is called a *Pareto optimal* set, and its corresponding objective function values are known as the *Pareto front* [44].

2.4. Scoring Metrics

As we pointed out before, a BN is a graphical model representing probabilistic relationships among variables of interest; its qualitative part, or structural model, provides a visual representation of the phenomenon under investigation. These are some of its main advantages. Therefore, one of our main interests is to exploit the BN's representational power and classifier capacity using a search-and-score approach. However, it has been demonstrated that other models, such as NB and TAN, outperform BNs in classification, as we discuss in Section 1.1.

The MDL metric has been widely used for learning the structure of a BN from data. However, it has been discussed that a BN with an acceptable MDL score is not always a good classifier [45] due to MDL globally measuring the error of the learned BN across all variables in the domain, rather than the error in predicting the variable class (Section 3.2 shows this discrepancy graphically).

This subsection describes the metrics used in our bi-objective approach. MOPSO-BN seeks a competitive set of solutions that balance MDL (f_1) and misclassification (f_2). Solutions that entail this trade-off can support inference and classification. The scoring metrics are described below:

https://doi.org/10.3390/mca31030094

MDL is defined by [46] as follows:

$$f_1 = -logP(D|\Theta) + \frac{k}{2} log\, n \tag{1}$$

where D is the dataset, Θ represents the parameters of the model, k is the dimension of the model, and n is the sample size. Parameter Θ is the corresponding local probability distribution for each node in the network, determined by each BN structure.

The first term of Equation (1), $-\log P(D|\Theta)$, corresponds to the negative log-likelihood of the data given the model parameters Θ. These parameters are estimated using the simple estimator procedure implemented in Weka [24], which computes the conditional probability tables (CPTs) of each node in the network from the data using relative frequencies with a small Laplace-style smoothing value ($\alpha = 0.5$ by default) to avoid zero probabilities in cases where a particular combination of variable values does not appear in the dataset. Once Θ is estimated, the log-likelihood term is computed as

$$-\log P(D|\Theta) = -\sum_{j=1}^{m}\sum_{i=1}^{n} \log P\left(x_i^{(j)} \mid pa_i^{(j)}, \Theta\right) \tag{1a}$$

where $x_i^{(j)}$ denotes the value of variable X_i in instance j, and $pa_i^{(j)}$ represents the corresponding values of the parent variables of X_i in that same instance. This decomposition follows directly from the conditional independence assumptions encoded in the BN structure, which allow the joint likelihood to be factored into local contributions from each variable, given its parents.

The model dimension (k) is given as follows.

$$k = \sum_{i=1}^{m} q_i(r_i - 1) \tag{1b}$$

where m is the number of attributes, q_i is the number of possible configurations of the parents of variable X_i, and r_i is the number of values of that variable.

The first term of Equation (1) measures the accuracy of the model (in terms of the likelihood of the data), and the second term measures its complexity. The complexity is proportional to the number of arcs, as shown in Equation (1b).

Equation (2) describes f_2, where a_i represents the number of instances incorrectly classified in the i-th fold of the testing set, b is the total number of instances in the fold, and s is the number of folds ($s = 10$ in our case). The misclassification rate is thus computed as the average classification error across all folds of an s-fold cross-validation procedure.

$$f_2 = \frac{1}{s}\sum_{i=1}^{s} \frac{a_i}{b} \tag{2}$$

The model parameters are computed using the simple estimator procedure implemented in Weka [24].

2.5. MOPSO-BN

The original PSO was designed to handle continuous search problems. In our approach, each candidate network solution is represented as a binary vector. We use a sigmoid function to map the vector of each particle in the swarm to 0 and 1 after updating the velocity vector [39], as shown in Algorithm 1 at lines 3 and 16.

The evaluation of the network candidate is based on two objectives. (i) MDL (Equation (1)) and (ii) misclassification (Equation (2)). Section 2.2.2 explains the repre-

https://doi.org/10.3390/mca31030094

sentation and cycling-repair process. Our methodology is described in Algorithm 1. The initial $\vec{pbest}_i$ corresponds to the first particle, while the hypervolume contribution of each particle serves to update $\vec{pbest}$. The hypervolume is a performance metric used in evolutionary multi-objective optimization [41]. The non-dominated particles are stored in a specified file (file A) beforehand. The maximum number of iterations (max_{gen}) must be specified as a stopping criterion for the algorithm.

To select the leader $\vec{gbest}$, we calculate the hypervolume contribution of the particles in file A, as can be seen in Section 2.5.1. We compute the new velocity and position of each particle in the population according to Equations (3) and (4) (see Algorithm 1), where three parameters determine the particle's position: the current position $\vec{x}_i$, the previous best position $\vec{pbest}_i$, and the velocity $\vec{v}_i$. w represents the inertia, which controls the impact through the velocity history and the current velocity. C_1 and C_2 are considered the learning factors. C_1 is the cognitive factor and represents the attraction that a particle has towards its success. C_2 is the social factor and represents the attraction that a particle has toward the success of its neighbors. Both factors are usually defined as constants.

Algorithm 1 MOPSO-BN

1: Initialize the generation counter with $t = 0$
2: Initialize randomly the position and velocity of each particle in swarm S
3: Transform the real position vector into a binary vector for each particle in swarm S through a sigmoid function
4: Repair the position vector if cycles are detected
5: Evaluate each particle in swarm S in the two objective functions
6: Select the initial $\vec{pbest}$ of each particle in swarm S
7: Insert non-dominated particles of the in swarm S in the file A
8: **while** $t < max_{gen}$ **do**
9: **for** $i = 1$ to $|S|$ **do**
10: Select $gbest$, compute velocity and position
11: **end for**
12: Compute the contribution of each particle to the hypervolume value of swarm S
13: Compute the new velocity and position of each particle in the population. According following:

$$\vec{v}_i = w\vec{v}_i + C_i \otimes r_1(\vec{pbest}_i - \vec{x}_i) + C_2 r_2\ (\vec{gbest} - \vec{x}_i) \tag{3}$$

$$\vec{x}_i \leftarrow \vec{x}_i + \vec{v}_i \tag{4}$$

14: Repair cycles
15: Evaluate each particle in swarm S in the two objective functions
16: Using the sigmoid function to transform the real position vector into a binary vector
17: Update $pbest$ particle
18: Update non-dominated particles in file A
19: $t = t + 1$
20: **end while**
21: Return A

It is important to mention that the work of [47] suggests that one of the critical issues in BN structure learning is the search procedure. This is why several proposed heuristics can address such a challenge. Although there are different ways to search for BN structures, we focus on the space of DAGs since their representation as adjacency matrices can facilitate the implementation of operators in a PSO-based algorithm, such as our MOPSO-BN. A problem with this search space is the global acyclicity constraint [48]; to overcome this obstacle, we use an operator that avoids cycles while maintaining the randomness of the particle when adding arcs that were eliminated due to cyclicity.

https://doi.org/10.3390/mca31030094

Another significant issue in the BN learning problem is parameter optimization; we use a simple estimator for classification, but the parameters depend on the sample size [24]. We defer learning BN parameters to future work; after running our proposed MOPSO-BN, we obtain the Pareto front, and a local search procedure further optimizes the parameters, improving our final results.

The computational cost of MOPSO-BN per dataset is determined by the number of fitness evaluations, $E_{total} = R \times I \times P = 10 \times 200 \times 100 = 200{,}000$, where each evaluation involves computing MDL (Equation (1)) and misclassification via 10-fold cross-validation (Equation (2)), with a cost of $O(s \cdot N \cdot \sum_{i=1}^{n} q_i \cdot r_i)$ per particle. Although this represents a higher computational investment than deterministic methods such as NB, TAN, and HC-BN, this cost is justified by the population-based global search capability of MOPSO-BN and, most importantly, by the fact that it simultaneously optimizes two conflicting objectives, producing a Pareto front of non-dominated solutions that cannot be obtained by any single-objective deterministic method. Furthermore, since BN structure learning is typically performed offline prior to deployment, this one-time computational cost is amortized over the operational lifetime of the learned model.

2.5.1. Selection of *pbest* and *gbest* via Hypervolume Contribution

In a multi-objective setting, the classical PSO definitions of personal best (*pbest*) and global best (*gbest*) must be adapted since there is no longer a single scalar fitness value to compare solutions. In MOPSO-BN, both selections are based on the concept of hypervolume contribution, which measures the volume of the objective space dominated exclusively by a given solution with respect to a reference point $\vec{r} = (r_1, r_2)$, typically set to a point that is dominated by all solutions in the current archive (file A).

2.5.2. Personal Best (*pbest*) Update

The personal best $\vec{pbest}_i$ of particle i is updated by comparing its current position $\vec{x}_i$ against its previous best position using the concept of Pareto dominance. Specifically:

- If $\vec{x}_i$ dominates $\vec{pbest}_i$, then $\vec{pbest}_i$ is replaced by $\vec{x}_i$.
- If $\vec{pbest}_i$ dominates $\vec{x}_i$, then $\vec{pbest}_i$ remains unchanged.
- If neither solution dominates the other, the hypervolume contribution of each is computed with respect to the current archive A, and the solution with the greater contribution is selected as the new $\vec{pbest}_i$. This criterion favors solutions that contribute more to the diversity and spread of the Pareto front approximation.

2.5.3. Global Best (*gbest*) Selection

The global best $\vec{gbest}$ is selected from the external archive A, which stores the non-dominated solutions found so far across all particles and iterations. To select $\vec{gbest}$, the hypervolume contribution $\Delta_h(s)$ of each solution $s \in A$ is computed as

$$\Delta_h(s) = \mathcal{H}(A, \vec{r}) - \mathcal{H}(A \setminus \{s\}, \vec{r}) \tag{5}$$

where $\mathcal{H}(A, \vec{r})$ denotes the hypervolume of the archive A with respect to the reference point $\vec{r}$, and $A \setminus \{s\}$ is the archive without solution s. Thus, $\Delta_h(s)$ measures the exclusive hypervolume region dominated only by s. The solution with the highest hypervolume contribution is selected as $\vec{gbest}$ as it represents the most influential individual in shaping the current Pareto front approximation. This selection strategy encourages particles to move toward regions of the objective space that are both well-performing and diverse, balancing convergence and spread of the Pareto front throughout the optimization process [40,41].

https://doi.org/10.3390/mca31030094

3. Experimental Design and Results

We categorize our experiments into two steps. First, to show the discrepancy between MDL and misclassification, we evaluated all DAGs in low-dimensional domains. The second experiment tested our proposed algorithm using real-world datasets. The experiments and results are described below.

3.1. Datasets

For the experiments, we evaluated MDL and misclassification rate across seven low-dimensional datasets, thereby exhaustively considering all possible Bayesian network structures. Three of them were taken from the literature, each with $n = 5$ variables, including the class variable: the Iris dataset with 150 instances, the Balance dataset with 625 instances, and the Sewell and Shah dataset with 10,318 instances [2,49]. The remaining four are synthetic datasets with $n = 6$ and 10,000 instances, where all variables are binary; two were generated using a random probability distribution (RDP-1 and RDP-2) and the other two using a low-entropy distribution (LED-1 and LED-2) [50] (see details in Table 1).

Table 1. Characteristics of datasets in low-dimensionality domains.

Dataset	No. Classes	No. Features	No. Instances
Balance Scale	5	5	625
Iris	3	5	150
Sewell and Shah	2	5	10,318
RPD-1	2	6	10,000
RPD-2	2	6	10,000
LED-1	2	6	10,000
LED-2	2	6	10,000

For the second experiment, a set of 20 datasets from the UCI repository [49] is used. Continuous variables are discretized using Weka's CAIM algorithm [24], and cases with missing values are removed. Table 2 describes the datasets.

Table 2. Characteristics of datasets of real-world domain.

Dataset	No. Classes	No. Features	No. Instances
Australian	2	14	690
Chess	2	36	3296
Cleve	2	11	296
CorrAl	2	6	128
crx	2	15	653
Diabetes	2	8	768
Flare	8	10	1389
German	2	20	1000
Glass2	2	9	163
Heart	2	13	270
Hepatitis	2	19	80
Lymphograph	4	18	148
Mofn-3-10	2	10	1324
Pima	2	8	768
Segment	7	19	2310
SoyBean-large	19	35	316
Tic-tac-toe	2	9	958
Vehicle	4	18	958
Vote	2	17	436
Waveform-21	3	21	301

https://doi.org/10.3390/mca31030094

3.2. MDL and Misclassification

In general, it is not possible to exhaustively evaluate all DAGs since the number of possible structures increases exponentially with the number of variables. It is possible to compute the number of DAGs as a function of the number of attributes n according to Equation (6); e.g., when $r(5)$, the number of possible structures, is 29,281 [3].

$$r(n) = \sum_{i=1}^{n} (-1)^{i+1} \binom{n}{i} 2^{i(n-1)} r(n-i) \tag{6}$$

We evaluate MDL and misclassification rate in the seven datasets described in Section 3.1. We apply the 95% non-parametric Mann–Whitney U test to compare the misclassification of the network with the minimum MDL value against the model with the best misclassification score.

The following briefly explains how we present our results. Figures 3–5 show an exhaustive evaluation of each BN structure. We plot misclassification (x-axis) vs. MDL (y-axis). The blue dots represent BN; the black dot represents the minimum value of MDL, while the red dots represent the minimum value of misclassification. It is possible to have more than one structure with the same value of MDL or misclassification, in which case dots may overlap, as is the case in Figure 4, where it is possible to see several BN structures with the same value of misclassification in contrast with different values of MDL, and in Figure 5, where the whole of BN structures has the same value of misclassification and different value of MDL.

Table 3 compares performance among the seven tested datasets. We first identified the BN structure with the smallest MDL value; then, we compared it to the networks with the smallest misclassification rate. Since we have several networks with the same value of misclassification, we present the average structural difference (the difference regarding the arcs: reversed, missing, and extra, between the model with the smallest MDL value and the models with the minimum misclassification) and the average Kullback–Leibler (KL) divergence (computed as the $\log_2$ of the ratio of the smallest MDL value network/the model with the minimum misclassification) to distinguish the distance and the difference among the networks. Figures 3 and 6a,b present a visual comparison of the BN structures for two datasets: Balance Scale and RPD-1.

Our experiment aims at showing the degree of discrepancy between MDL and misclassification graphically (see Figures 3–5). We know that MDL measures the error of the BN across all variables in the dataset; thus, minimizing this error does not entail minimizing the local error in predicting the class variable given the attributes [8]. Some authors have explored the use of MDL to learn BN classifiers [15,16,23,51,52]. In this sense, our results suggest that since the BN with the smallest MDL value could have interactive effects with classification error, we do not need to observe statistically significant differences regarding misclassification (see Table 3). Other ways to learn BN classifiers include: (1) approximating the true joint probability distribution over the attributes and (2) classifying the class variable given the attributes. A standard approach to achieving the first method is to use a quality measure based on log-likelihood, as in MDL. For the second method, one can use prediction error as misclassification or as the conditional log-likelihood; this is typically not used directly in practice due to its computational complexity [15,53]. Our results show that it is possible to find several BNs with the same misclassification rate but different MDL values; hence, selecting one of them could lead to overfitting [16].

https://doi.org/10.3390/mca31030094

Table 3. Average structural difference and average Kullback–Leibler (KL) divergence between the model with the smallest MDL value and the networks with the minimum misclassification. Values in parentheses represent the standard deviation, and values in boldface mean the significant best value found.

Dataset	Reversed	Missing	Extra	Average KL Divergence	Minimum MDL Value		Minimum Misclassification Value	
					MDL	Misclassification	MDL	Misclassification
1. Balance Scale	7.6889	0	3.9004	0.3891	4573.7254	7.83 (±0.46)	9599.5836	**0.0 (±0.0)**
2. Iris	3.2195	0.6293	3.9102	0.3558	529.3484	4.0 (±4.422)	655.1409	3.33 (±3.33)
3. Sewell and Shah	5.5323	0.9835	3.1405	0.0117	45,636.9539	19.94 (±1.18)	47,574.7018	19.84 (±1.33)
4. RPD-1	5.1296	1.4168	5.0337	0.0254	32,093.7447	29.17 (±0.85)	34,072.1691	29.17 (±0.85)
5. RPD-2	8.1888	1.2158	3.6749	0.0114	35,368.8459	15.24 (±0.55)	36,139.7766	15.24 (±0.55)
6. LED-1	0	0	8.8368	0.0010	19,277.8087	9.72 (±0.04)	19,277.8087	9.72 (±0.04)
7. LED-2	0	0	8.8368	0.0010	19,553.7187	9.79 (±0.03)	19,553.7187	9.79 (±0.03)

One of our primary goals is to learn BN structures that capture dependence, conditional independence, and even causal relationships among attributes; we wish to use this learned model for classification. Using the metrics mentioned above, this is in general not possible; e.g., we graph the networks at the extremes, that is, the network with the smallest value of misclassification but the worst value of MDL (see Figure 6c) and the network with the smallest MDL value (see Figure 6d). Although both models are equally valid regarding accuracy, Figure 6c could have less representational power in comparison with Figure 6d. This behavior highlights the conflict between both objectives [8,15,52].

Regarding the synthetic low-entropy datasets, as noted in [50], MDL tends to select simpler models and, hence, does not exploit the advantages of BNs (see Figure 5). Although several studies have exhaustively evaluated MDL in low-dimensional domains [2,16,47,52], they do not compare the learned models with respect to their structure. To address this discrepancy, we propose a bi-objective approach, evaluated in the next section.

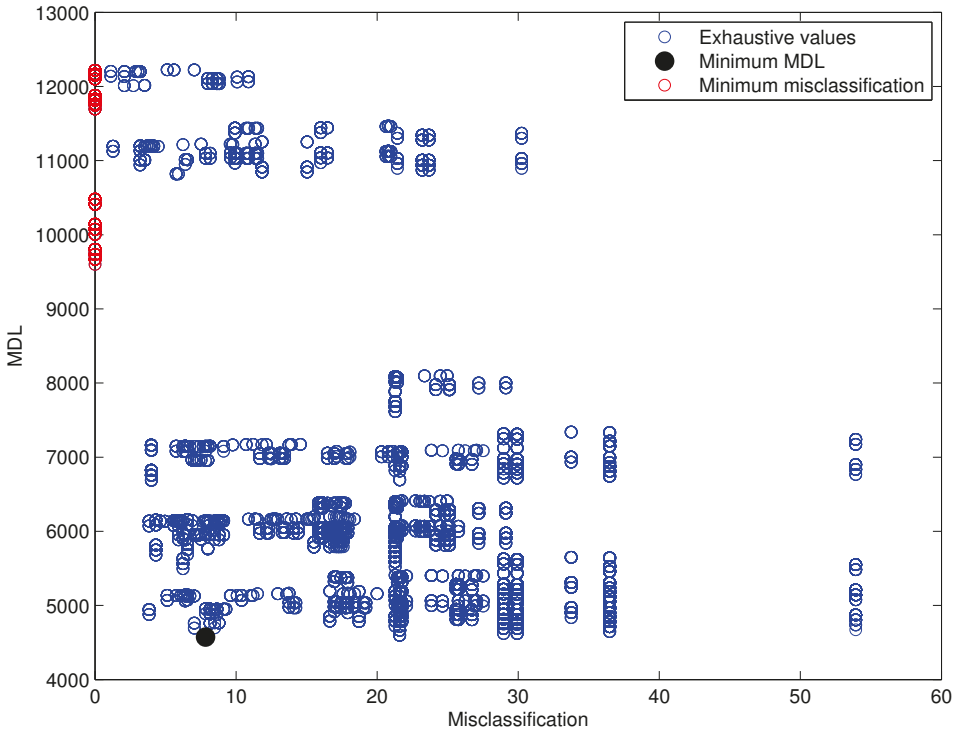


Figure 3. Balance scale.

https://doi.org/10.3390/mca31030094

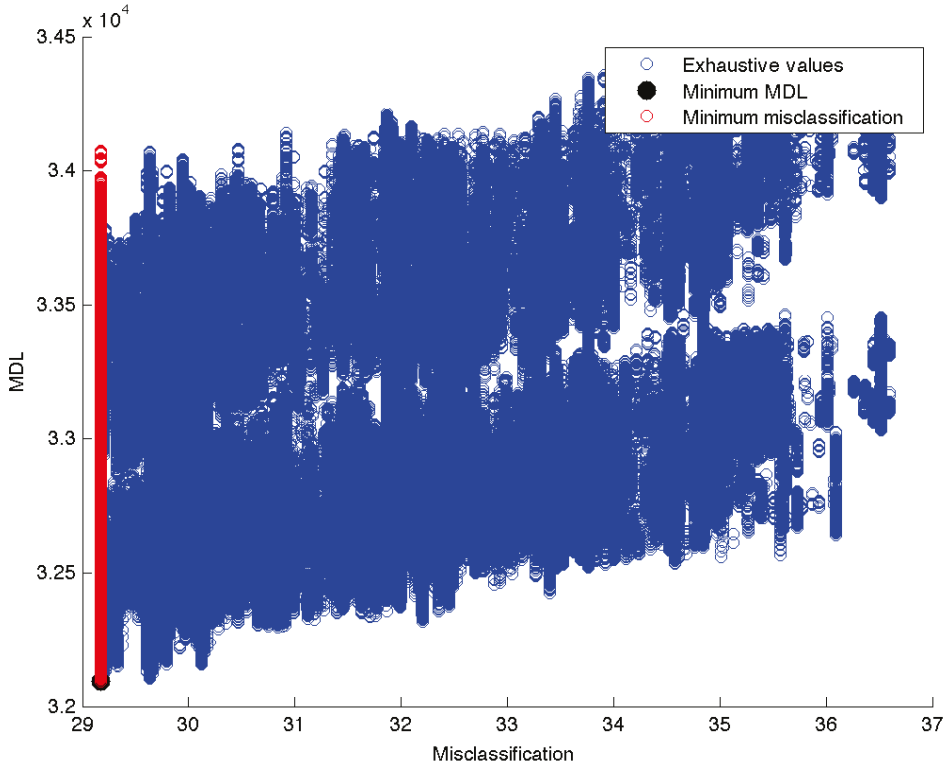


Figure 4. Random probability distribution 1 (RPD-1).

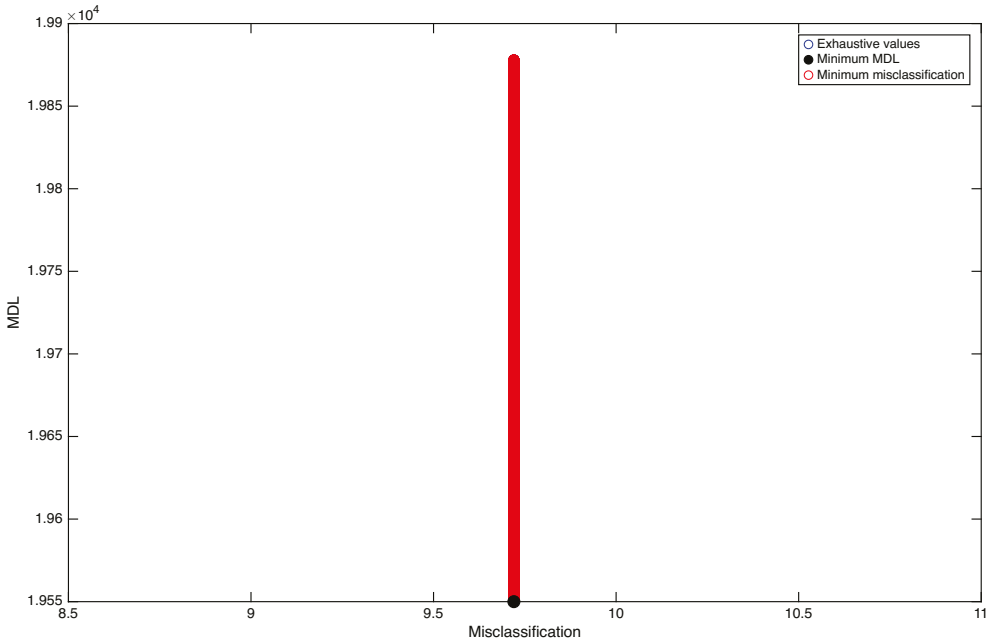


Figure 5. Low-entropy probability distribution 1 (LED-1). Red circles overlap the blue circles.

3.3. The Trade-Off Between MDL and Misclassification

This section describes a set of experiments to test our proposed algorithm MOPSO-BN. The details of the dataset can be seen in Table 2.

For this experiment, which involves 20 real-world datasets of higher dimensionality where an exhaustive search is computationally infeasible, MOPSO-BN requires a parameter configuration. Given the high computational cost of running the algorithm—each configuration requires 10 independent runs with 20,000 evaluations across 20 datasets—a full factorial or grid search parameter analysis was outside the scope of this work. Instead, we

https://doi.org/10.3390/mca31030094

followed the common practice in the evolutionary computation literature by starting from values reported in foundational PSO and MOPSO works [39,40,54–57] and then performing a manual sensitivity analysis by varying one parameter at a time while keeping the others fixed. The final values, $C_1 = 2.1$, $C_2 = 2.0$, $w = 0.4$, particles $= 100$, and cycles $= 200$, were selected because they consistently yielded stable convergence and competitive Pareto fronts across the tested datasets. The inertia weight $w = 0.4$ was chosen to favor exploitation over exploration in the later stages of the search, while C_1 and C_2 were kept close to theoretically recommended values that ensure particle convergence [39]. The swarm size and number of cycles were set to balance solution quality and computational feasibility. More systematic parameter tuning, such as using automatic configuration tools like *irace*, is part of our future work.

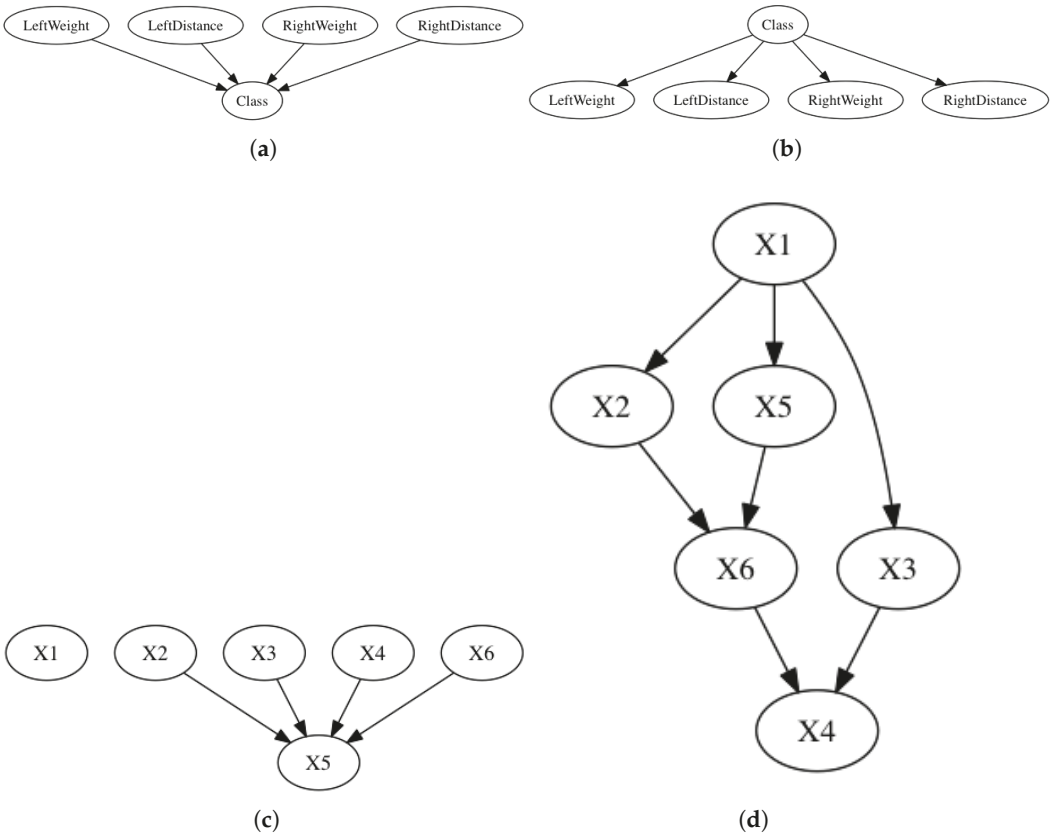


Figure 6. Graphs with the smallest values. (**a**) Network with the smallest value of misclassification and MDL from the Balance Scale dataset. (**b**) Network with the smallest value of MDL from the Balance Scale dataset. (**c**) Network with the smallest value of misclassification and MDL from the RPD-1 dataset. (**d**) Network with the smallest value of MDL from RPD-1.

The algorithm performs 10 independent runs per database, for a total of 20,000 evaluations. Each run produces a set of trade-off solutions. The process of selecting the most suitable among them is known as preference handling. This work compares the bi-objective approach with other Bayesian network models, including NB, TAN, BNs learned using hill-climbing (HC-BN), and our selected BNs from the Pareto front. To find the latter, we propose a subsequent preference-handling approach: after accumulating the Pareto front across all ten independent runs, a single solution is selected using a reference-point strategy [40]. Let $\mathcal{F} = \{(\hat{f}_1^{(k)}, \hat{f}_2^{(k)})\}_{k=1}^{M}$ be the set of M non-dominated solutions, where $\hat{f}_1^{(k)}$ and $\hat{f}_2^{(k)}$ are the MDL and misclassification values of solution k. Since both objectives

https://doi.org/10.3390/mca31030094

differ substantially in magnitude, each is independently normalized to $[0,1]$ via min-max scaling over the accumulated Pareto front:

$$\tilde{f}_j^{(k)} = \frac{\hat{f}_j^{(k)} - \min_l \hat{f}_j^{(l)}}{\max_l \hat{f}_j^{(l)} - \min_l \hat{f}_j^{(l)}}, \quad j \in \{1,2\}, \quad k = 1,\ldots,M \tag{7}$$

The selected solution k^* minimizes the Euclidean distance to the ideal reference point $\vec{r} = (0,0)$ in the normalized space:

$$k^* = \arg\min_k \sqrt{\left(\tilde{f}_1^{(k)}\right)^2 + \left(\tilde{f}_2^{(k)}\right)^2} \tag{8}$$

This criterion assigns equal weight to both objectives after normalization, favoring solutions that balance data compression and classification accuracy without requiring any dataset-specific parameter tuning. The normalization bounds are computed from the accumulated Pareto front of each dataset, making the procedure fully self-contained and reproducible. The selected solution for each dataset is reported in Table 4 under the MOPSO-BN column.

To learn BNs using HC-BN, we use Weka's hill-climber algorithm [24] with the following parameter values: the initial structure NB: false, number of parents: 100,000, score type: MDL, and arc reversal: true.

Table 4 shows misclassification and standard deviation of each algorithm (NB, TAN, HC-BN, and our chosen solution from the Pareto front). We highlight the classifier with the fewest misclassifications in boldface. To conduct a statistical comparison, we applied the non-parametric 95%-confidence Kruskal–Wallis multi-comparison test. According to this test, there is a significant difference between NB, TAN, HC-BN, and MOPSO-BN but none between MOPSO-BN and HC-BN. To further analyze those results and the statistical validation, we applied the post hoc Bonferroni test. Figure 7 shows the results, where it is possible to observe how our obtained solution has a better performance than TAN and NB. However, we see a similar performance between our proposed MOPSO-BN and HC-BN. Additionally, we applied the 95% non-parametric Mann–Whitney U test between HC-BN and MOPSO-BN; the results show our proposed approach outperforms others in 6 datasets, ties in 12 datasets, and has worse performance in just two datasets.

Results published by [16], where twenty-six datasets from UCI are tested, show that in 13 datasets TAN displays a significant difference between BNs using K2 and HC as the search procedure. Our results show better accuracy. Unfortunately, Madden's results do not show the MDL metric for each model or the models learned, making it impossible to compare MDL. Similarly, the work of [52] shows poor accuracy when MDL is used as the driving metric. They propose a new score, risk minimization by cross-validation (RMVC), for an HC-based algorithm. One of their conclusions is that the best results are obtained when the initial structure is an NB rather than an empty network. Although EAs can learn robust structures compared with standard methods such as the K2 algorithm or simple deterministic methods, such as HC [58], the initialization of the search space can improve results [59]. MOPSO-BN employs random particle initialization, which helps the search process, but it is also possible to incorporate information about conditional independence between variables to further improve results. The work of [52] does not show the learned model using MDL; similarly, refs. [15,60] do not.

Table 4 shows that the chosen solution has the biggest value of MDL in 19 cases. These results agree with [8,15,52], who assert that a relatively good MDL score does not necessarily yield suitable classifiers. Note that we are empirically selecting an overall solution from a set of solutions and that we consider this solution to offer a good trade-off

https://doi.org/10.3390/mca31030094

between misclassification and MDL, with the advantages that entail; nevertheless, our solution leaves open the possibility for expert input. Regarding Pareto fronts, they have advantages and have been used extensively before (see Figures 8–10).

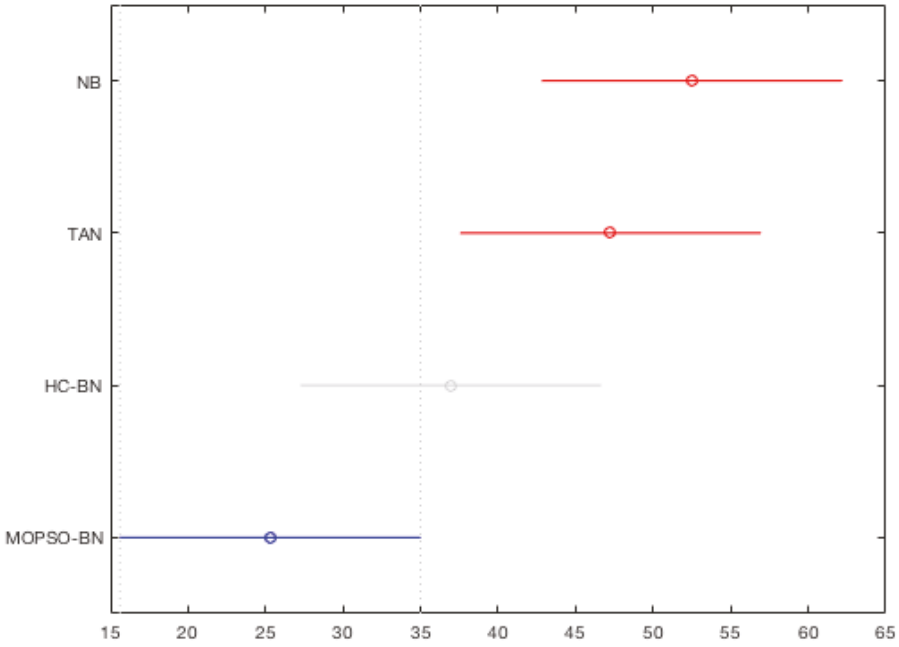


Figure 7. Post-hoc Bonferroni test to compare the performance of the four classifiers considering all datasets. The *x*-axis shows the confidence intervals of the midrange, and the *y*-axis shows the names of each classifier.

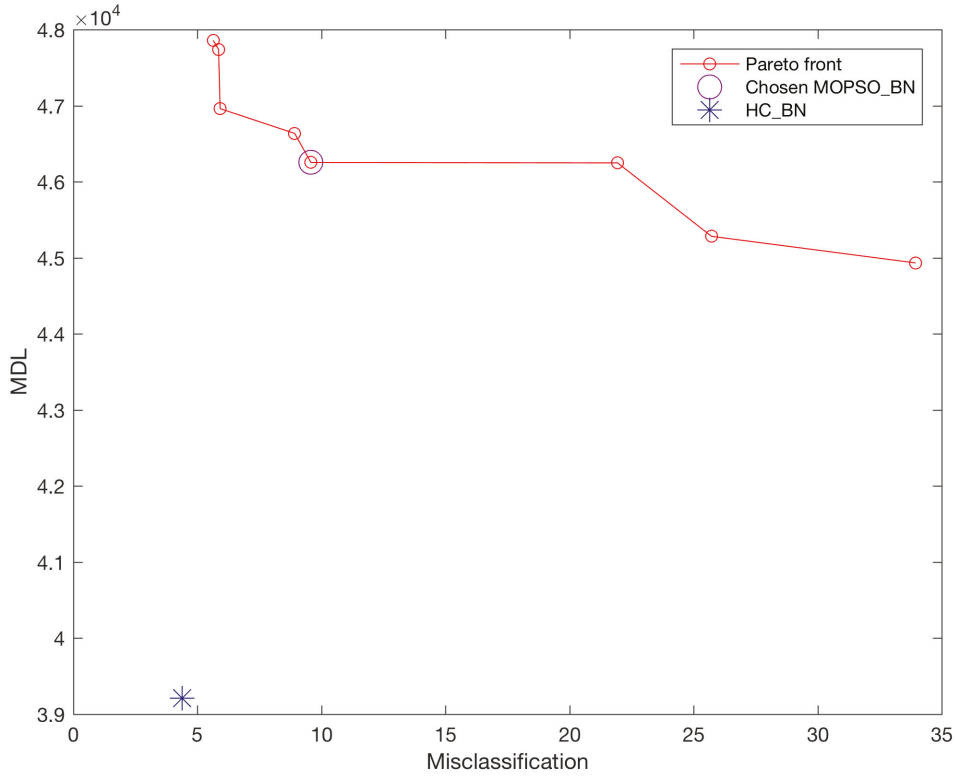


Figure 8. Accumulated Pareto front of Chess dataset obtained by ten runs of MOPSO-BN.

https://doi.org/10.3390/mca31030094

Table 4. Mean of misclassification and standard deviation. Values in boldface mean the best values found. The *p*-value corresponds to the Mann–Whitney U test between HC-BN and MOPSO-BN. (+) means significance difference, (-) means significance difference in favor of HC-BN, and (=) means that there is no significant difference.

No.	Dataset	NB	TAN	HC-BN		MOPSO-BN		HC-BN vs. MOPSO-BN *p*-Value
				Misc	MDL	Misc	MDL	
1	Australia	14.2 ± 4.03	14.94 ± 3.90	11.73 ± 0.00	5767.22	**11.44 ± 0.04 (=)**	6123.65	0.88076
2	Chess	12.15 ± 1.70	7.91 ± 1.39	**4.38 ± 0.0**	39,208.69	9.57 ± 0.35 (-)	46,253.37	0.00034
3	Cleve	17.13 ± 6.20	18.96 ± 6.77	26.24 ± 0.03	3900.49	**10.75 ± 0.13 (+)**	13,583.83	0.00116
4	Corral	12.95 ± 9.46	0.77 ± 3.19	0.0 ± 0.0	594.61	0.0 ± 0.0 (=)	590.47	0.9681
5	crx	14.39 ± 2.30	16.11 ± 2.70	12.40 ± 0.0	7705.44	**7.96 ± 0.35 (+)**	10,797.20	0.00168
6	Diabetes	26.61 ± 2.8	27.41 ± 3.3	22.39 ± 0.01	4768.37	**19.00 ± 0.35 (=)**	5599.08	0.09692
7	Flare	19.88 ± 3.47	17.35 ± 3.47	15.00 ± 0.01	6824.10	**12.75 ± 0.45 (=)**	11,204.16	0.1031
8	German	25.39 ± 4.31	27.93 ± 4.04	20.29 ± 0.01	18,002.210	**13.40 ± 0.44 (+)**	29,057.68	0.00318
9	Glass2	18.84 ± 8.68	20.63 ± 8.95	7.35 ± 0.0	1522.69	**2.42 ± 0.76 (=)**	3147.29	0.08186
10	Heart	17.26 ± 6.70	16.89 ± 7.30	14.81 ± 0.02	2357.38	**9.62 ± 0.46 (=)**	2695.59	0.07508
11	Hepatitis	13.96 ± 10.97	12.0 ± 11.64	2.5 ± 0.00	950.34	2.5 ± 0.79 (=)	959.19	0.79486
12	Lymphography	17.84 ± 10.61	18.93 ± 9.57	6.71 ± 0.02	3595.33	**2.04 ± 1.71 (=)**	11,832.99	0.19706
13	Mofn-3-10	14.66 ± 3.43	8.04 ± 2.63	9.66 ± 0.0	2228.48	**2.33 ± 0.31(+)**	2372.03	0.00018
14	Pima	24.31 ± 4.42	23.63 ± 3.94	22.38 ± 0.01	4768.37	**19.39 ± 1.14 (=)**	5325.07	0.28914
15	Segment	8.73 ± 1.70	**4.73 ± 1.49**	5.38 ± 0.0	57,877.24	**4.22 ± 0.51 (=)**	49,573.03	0.22628
16	Soybean-large	8.17 ± 3.50	7.65 ± 3.08	**5.69 ± 0.0**	18,642.39	10.32 ± 0.44 (-)	26,106.89	0.01552
17	Tic tac toe	30.24 ± 4.45	23.68 ± 3.82	22.44 ± 0.0	9830.84	**2.19 ± 0.30 (+)**	14,393.29	0.00018
18	Vehicle	39.38 ± 4.88	29.64 ± 4.58	23.27 ± 0.02	14,457.84	**18.32 ± 1.65 (+)**	279,541.11	0.0139
19	Vote	9.73 ± 4.30	6.16 ± 3.26	4.80 ± 0.0	5820.75	**2.05 ± 0.65 (=)**	6015.33	0.0536
20	Waveform	19.1 ± 1.64	18.04 ± 1.70	9.0 ± 0.01	5198.34	**6.0 ± 0.21 (=)**	14,549.53	0.25848

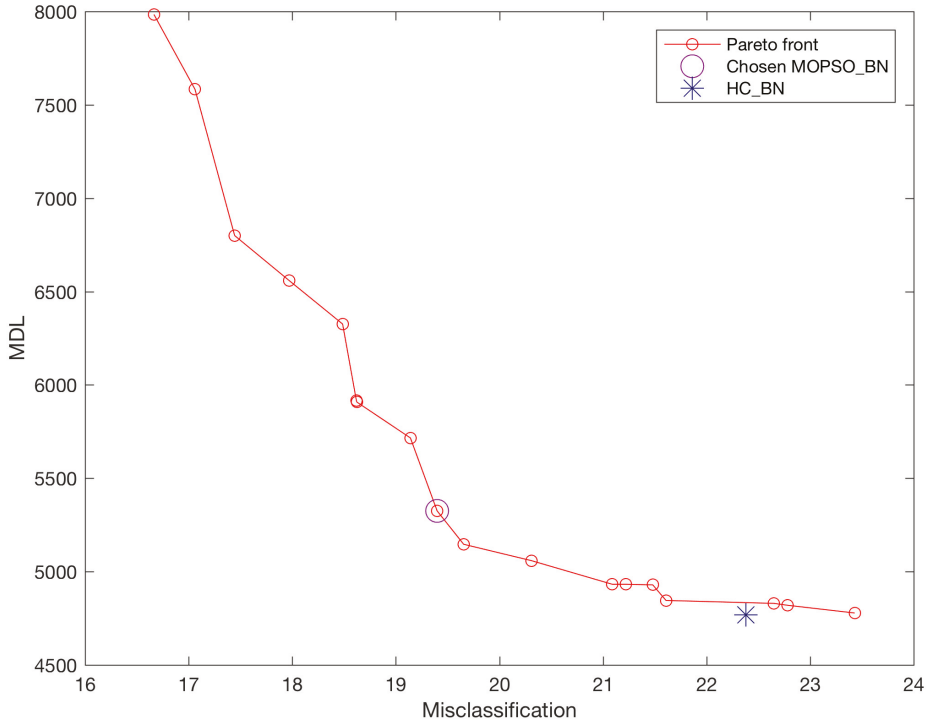


Figure 9. Accumulated Pareto front of Pima dataset obtained by ten runs of MOPSO-BN.

https://doi.org/10.3390/mca31030094

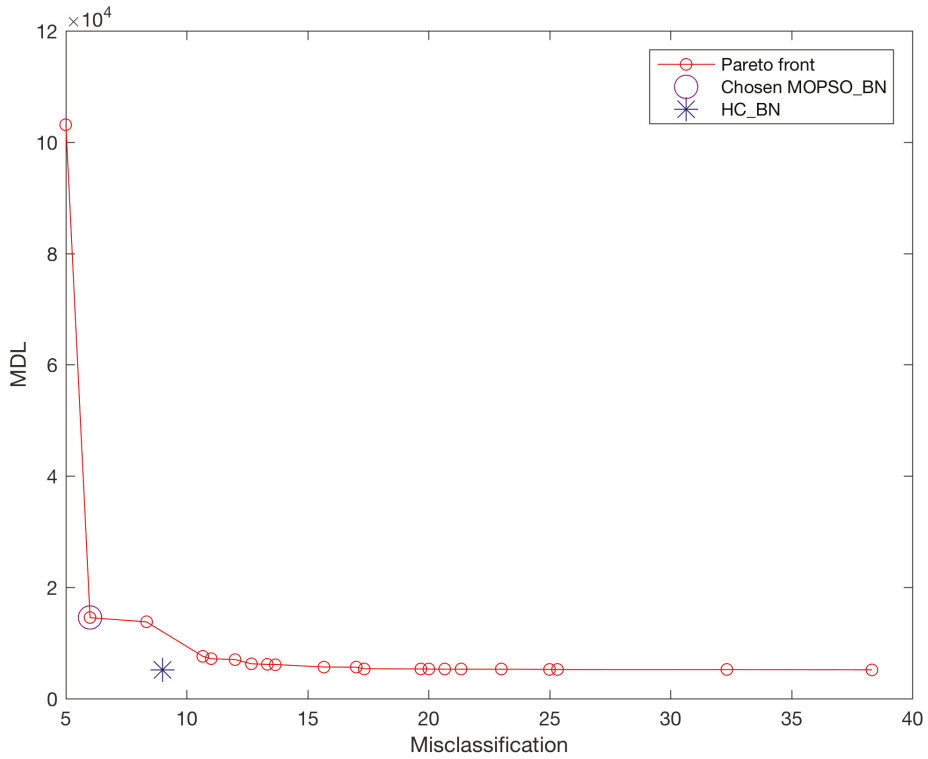


Figure 10. Accumulated Pareto front of Waveform dataset obtained by ten runs of MOPSO-BN.

3.4. Computational Cost Analysis

A natural concern when proposing a population-based metaheuristic such as MOPSO-BN is whether the additional computational investment over deterministic methods is justified by the gains in solution quality. In this section, we provide a theoretical analysis of the computational cost of each method and argue that the trade-off is favorable for MOPSO-BN in the context of BN structure learning for classification.

3.4.1. Cost of Deterministic Baseline Methods

The compared deterministic methods—NB, TAN, and HC-BN—have well-known computational properties. NB requires only a single pass over the data to estimate marginal and conditional frequency tables, with a cost of $O(N \cdot n)$, where N is the number of instances and n is the number of variables. Its structural assumption (all attributes are conditionally independent, given the class) requires no search whatsoever. TAN augments NB by computing a maximum spanning tree over the $\binom{n}{2}$ pairwise conditional mutual information values, which adds a cost of $O(n^2 \cdot N)$ for the mutual information estimation and $O(n^2 \log n)$ for Prim's or Kruskal's algorithm [51]. Both methods are therefore extremely fast in practice, but they impose rigid structural constraints on the BN topology that severely limit their representational power: NB assumes no dependencies among attributes, and TAN allows at most one additional parent per attribute beyond the class node.

HC-BN performs a greedy local search in the space of DAGs, iteratively evaluating arc additions, deletions, and reversals and accepting those that improve the MDL score. Its cost per iteration is $O(n^2)$ score evaluations, each requiring $O(N \cdot q_i \cdot r_i)$ operations to update the sufficient statistics, where q_i is the number of parent configurations of variable X_i and r_i is the number of its possible values. While HC-BN is significantly more flexible than NB and TAN in terms of the structures it can represent, it is a deterministic single-start greedy algorithm that is well known to be highly sensitive to the initial structure and prone to converging to local optima [58]. Its output is always a single network, providing

https://doi.org/10.3390/mca31030094

no information about the landscape of alternative solutions or the trade-offs between competing objectives.

3.4.2. Cost of MOPSO-BN

The computational cost of MOPSO-BN is determined by four components that operate at each iteration:

1. **MDL evaluation** (f_1): Given a candidate BN structure represented as an adjacency matrix, the parameters Θ are estimated using the simple estimator [24], which requires a single pass over the dataset to compute frequency tables. The MDL score is then computed according to Equation (1), with a cost of $O(N \cdot \sum_{i=1}^{n} q_i \cdot r_i)$ per particle.
2. **Misclassification evaluation** (f_2): The misclassification rate is computed via $s = 10$-fold cross-validation (Equation (2)), which requires training and testing the BN classifier s times. The dominant cost is $O(s \cdot N \cdot \sum_{i=1}^{n} q_i \cdot r_i)$ per particle, making this the most expensive component of each evaluation.
3. **Cycle repair**: The repair operator that ensures DAG validity operates directly on the adjacency matrix of each candidate BN and has a cost of $O(n^2)$ per particle. It handles three types of cycles: (i) self-cycles, detected by inspecting the diagonal of the adjacency matrix in $O(n)$ and resolved by setting the offending entry to 0; (ii) bi-directional cycles, detected by checking for simultaneous non-zero entries at positions (i, j) and (j, i) in $O(n^2)$ and resolved by randomly removing one of the two conflicting arcs; and (iii) general cycles of length greater than 2, identified by inspecting directed paths in the adjacency matrix and resolved by randomly removing one of the arcs responsible for closing the loop [28]. By applying this repair mechanism at each iteration, every candidate solution evaluated by MOPSO-BN is guaranteed to correspond to a valid DAG.
4. **Hypervolume contribution**: The selection of $\vec{gbest}$ requires computing the hypervolume contribution $\Delta_h(s)$ of each solution in the archive A (Equation (5)). For a bi-objective problem, this can be computed exactly in $O(|A| \log |A|)$ time, where $|A|$ is the archive size [40].

The total number of fitness evaluations per dataset is

$$E_{total} = R \times I \times P = 10 \times 200 \times 100 = 200{,}000 \tag{9}$$

where $R = 10$ is the number of independent runs, $I = 200$ is the number of iterations per run, and $P = 100$ is the swarm size. This represents a substantially larger computational investment than any of the deterministic baselines.

3.4.3. Justification of the Computational Trade-Off

Despite its higher computational cost, the investment in MOPSO-BN is justified on both theoretical and practical grounds.

Search Space Coverage

The number of possible DAG structures over n variables grows super-exponentially according to Robinson's formula [3] (Equation (6)). For example, with $n = 10$ variables, the number of possible DAGs exceeds 4×10^{18}, making any exhaustive evaluation completely infeasible. Deterministic methods such as HC-BN navigate this space via a single greedy trajectory, which is computationally cheap but sacrifices global search. In contrast, MOPSO-BN maintains a population of $P = 100$ particles that simultaneously explore different regions of the DAG space, guided by both individual experience (*pbest*) and collective knowledge (*gbest*). This population-based mechanism provides a substantially broader

https://doi.org/10.3390/mca31030094

coverage of the search space, reducing the risk of premature convergence to poor local optima—a well-documented limitation of greedy methods in combinatorial optimization problems [58].

Bi-Objective Optimization and Pareto Front

A fundamental limitation of all deterministic baselines is that they optimize a single objective and return a single solution. HC-BN, for instance, uses MDL as its scoring function and outputs the network with the smallest MDL value. As demonstrated graphically in Figures 4 and 5 and quantitatively in Table 3, the network with the minimum MDL value is not necessarily the one with the best misclassification rate, and vice versa. This discrepancy is the central motivation of our bi-objective approach. MOPSO-BN simultaneously minimizes both f_1 (MDL) and f_2 (misclassification), producing a Pareto front of non-dominated solutions that explicitly trade off these two conflicting objectives. This output richness cannot be obtained by running NB, TAN, or HC-BN multiple times since these methods do not explore the objective space in a structured way. The additional computational cost of MOPSO-BN is therefore the price to be paid for obtaining this richer, more informative output.

Offline Learning Context

In the majority of practical applications of BNs—including medical diagnosis [2], fault detection, and risk assessment—the structure learning phase is performed offline, prior to deployment. Once the model is learned, inference is carried out online at negligible cost. In this context, the higher training cost of MOPSO-BN is a one-time investment that is amortized over the entire operational lifetime of the deployed model. Moreover, since MOPSO-BN returns a set of Pareto-optimal solutions rather than a single network, the end-user or domain expert can select the most appropriate model for a specific deployment scenario—for instance, prioritizing misclassification in safety-critical settings or MDL in resource-constrained environments—without the need to rerun the algorithm.

Comparison with HC-BN as the Closest Baseline

Among the compared methods, HC-BN is the most relevant computational baseline as it also performs search-based MDL optimization. The key difference is that HC-BN performs a single deterministic greedy trajectory, while MOPSO-BN runs 10 independent multi-objective searches with a population of 100 particles. Although this makes MOPSO-BN considerably more expensive, the Mann–Whitney U test results in Table 4 show that MOPSO-BN significantly outperforms HC-BN in 6 out of 20 datasets and achieves comparable performance in 12 others, with inferior results in only 2 datasets. Furthermore, the MDL values of the selected MOPSO-BN solution are higher than those of HC-BN in 19 out of 20 cases, confirming that MOPSO-BN deliberately trades a higher MDL value for a better misclassification rate—a behavior structurally impossible for HC-BN, which only optimizes MDL. This confirms that the additional computational cost of MOPSO-BN yields a qualitatively different, richer type of solution that single-objective deterministic methods cannot produce by construction.

Scalability Considerations

We acknowledge that the fixed budget of 200,000 function evaluations may become a limiting factor for datasets with a very large number of variables n since the cost of each evaluation grows with the complexity of the BN structure. For the 20 datasets used in our experiments, which range from 6 to 36 variables (Table 2), this budget was found to be sufficient to produce competitive Pareto fronts across all 10 independent runs, as evidenced by the consistency of the results shown in Figures 8–10. For larger networks, strategies

https://doi.org/10.3390/mca31030094

such as automatic algorithm configuration [61] or adaptive budget allocation could be used to dynamically adjust the number of evaluations based on network complexity, and are considered part of our future work.

3.5. Ablation Analysis

To clarify the individual contribution of each component of MOPSO-BN, we present a structured ablation analysis decomposing the proposed method into four configurations: (1) **PSO-MDL**, which uses PSO with only the MDL objective f_1; (2) **PSO-Misc**, which uses PSO with only the misclassification objective f_2; (3) **MOPSO (no selection)**, which runs the full bi-objective search but selects a random solution from the Pareto front; and (4) **MOPSO-BN (proposed)**, the complete method with reference-point preference handling. Table 5 summarizes the expected behavior of each configuration.

Table 5. Summary of the ablation analysis. Each configuration isolates one component of MOPSO-BN and its expected effect on MDL and misclassification performance.

Configuration	Objectives	Search	Selection	Expected Behavior
PSO-MDL	f_1 only	PSO	Best f_1	Low MDL but high misclassification risk, analogous to HC-BN
PSO-Misc	f_2 only	PSO	Best f_2	Low misclassification but high MDL, overfitting risk
MOPSO (no sel.)	$f_1 + f_2$	MOPSO	Random	Pareto front available but inconsistent solution quality across runs
MOPSO-BN (proposed)	$f_1 + f_2$	MOPSO	Ref. point	Balanced trade-off, competitive and reproducible on both objectives

The contribution of each component is supported by evidence already present in the paper. First, regarding the MDL objective: the exhaustive evaluation in Section 3.2 shows that the network with the minimum MDL value does not necessarily achieve the best misclassification rate (Table 3), with differences exceeding 100% in MDL between the minimum-MDL and minimum-misclassification networks in datasets such as Balance Scale (4573.73 vs. 9599.58). This directly demonstrates that PSO-MDL alone—like HC-BN—cannot guarantee good classification performance, motivating the inclusion of f_2. Second, regarding the misclassification objective: optimizing f_2 alone would remove the structural regularization provided by MDL, leading to unnecessarily complex networks that overfit the training data, as evidenced by the consistently high MDL values of the minimum-misclassification networks in Table 4.

Third, regarding the Pareto front: Figures 8–10 show that the Pareto front spans a wide range of trade-off solutions that are collectively inaccessible to any single-objective method, including regions where MOPSO-BN significantly outperforms HC-BN in misclassification (e.g., Cleve, crx, German, Tic-tac-toe, and Vehicle in Table 4).

Finally, regarding the reference-point selection, the chosen solution achieves a higher MDL than HC-BN in 19 out of 20 datasets (Table 4), confirming that the selection strategy explicitly balances both objectives rather than minimizing either in isolation, a behavior that random selection cannot reproduce consistently. Taken together, these observations demonstrate that each component of MOPSO-BN plays a distinct, non-redundant role: the MDL objective regularizes the structure, the misclassification objective drives classification performance, the MOPSO framework enables simultaneous Pareto-based optimization, and the reference-point strategy ensures a principled, reproducible solution selection. We acknowledge that a full experimental ablation with separate algorithm runs for each configuration would provide additional quantitative evidence, and we consider this a valuable direction for future work.

https://doi.org/10.3390/mca31030094

4. Conclusions and Future Work

A BN is a powerful tool for representing relationships between variables; although there are methods for learning BNs from data, their value depends on the particular application. In some cases, it is possible to exploit only some of the advantages of BNs, as NB or TAN do. The path to better utilize these benefits (by leveraging classification and representational power) could be costly since the learning process exhibits exponential growth in the number of parameters. Our proposed method explores the problem of automatically learning BN structures using MDL and misclassification as scoring metrics. A set of trade-off solutions is obtained for each dataset; we choose the solution closest to the origin as the competitive solution. We compare this chosen solution with NB, TAN, and BNs learned using HC. From this comparison, we conclude that the chosen solution achieves competitive results, particularly with respect to the misclassification objective. It is important to note that one of the main advantages of this approach is the set of trade-off solutions; the selection of a model can ultimately be performed by a domain expert. Additional advantages include that our proposed method can be applied to datasets from different domains and extended to other models, such as Support Vector Machines (SVMs) and Neural Networks.

As future work, we propose exploring the automatic configuration of parameters and the number of evaluations. Also, we plan to study alternative solutions to reduce the computational cost of the algorithm and extend the evaluation framework to include class-sensitive metrics such as F1-score, Precision, Recall, and AUC, which provide a more complete characterization of classification performance in datasets with imbalanced class distributions. In particular, AUC represents a promising alternative second objective for the bi-objective framework, given its threshold-independence and its natural compatibility with the probabilistic outputs of Bayesian network classifiers. The whole set of our results can be found at the following link: https://sites.google.com/view/vaguilerar/thesis (accessed on 26 February 2026).

Author Contributions: V.-J.A.-R. and N.C.-R. conceptualized and designed the analysis, methodology, and wrote the paper; V.-J.A.-R. and E.M.-M. reviewed and analyzed the results and designed the algorithm; R.V. supervised, wrote the paper, and analyzed the results. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: All the datasets used for experimentation are publicly available at https://archive.ics.uci.edu/ (accessed on 7 January 2025).

Acknowledgments: The first author acknowledges support from the Secretaría de Ciencia, Humanidades, Tecnología e Innovación (Secihti) through a scholarship 376521 to pursue PhD studies at the Universidad Veracruzana, México.

Conflicts of Interest: All authors declare that they have no conflicts of interest.

References

1. Pearl, J. Bayesian networks: A model of self-activated memory for evidential reasoning. In Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine, CA, USA, 15–17 August 1985; pp. 329–334.
2. Heckerman, D. A Tutorial on Learning with Bayesian Networks. In *Learning in Graphical Models*; Jordan, M.I., Ed.; MIT Press: Cambridge, MA, USA, 1999; pp. 301–354.
3. Robinson, R.W. Counting unlabeled acyclic digraphs. In *Combinatorial Mathematics V*; Lecture Notes in Mathematics; Little, C., Ed.; Springer: Berlin/Heidelberg, Germany, 1977; Volume 622, Chapter 2, pp. 28–43. [CrossRef]
4. Buntine, W. A Guide to the Literature on Learning Probabilistic Networks from Data. *IEEE Trans. Knowl. Data Eng.* **1996**, *8*, 195–210. [CrossRef]
5. Neapolitan, R.E. *Learning Bayesian Networks*; Prentice-Hall, Inc.: Upper Saddle River, NJ, USA, 2003.

https://doi.org/10.3390/mca31030094

6. Rónán, D.; Qiang, D.; Stuart, A. Learning Bayesian networks: Approaches and issues. *Knowl. Eng. Rev.* **2011**, *26*, 99–157. [CrossRef]
7. Hansen, M.H.; Yu, B. Model Selection and the Principle of Minimum Description Length. *J. Am. Stat. Assoc.* **2001**, *96*, 746–774. [CrossRef]
8. Friedman, J.H. On Bias, Variance, 0′/1 Loss, and the Curse-of-Dimensionality. *Data Min. Knowl. Discov.* **1997**, *1*, 55–77. [CrossRef]
9. Cerquides, J.; Mántaras, R.L. TAN Classifiers Based on Decomposable Distributions. *Mach. Learn.* **2005**, *59*, 323–354. [CrossRef]
10. Keogh, E.J.; Pazzani, M.J. Learning the structure of augmented Bayesian classifiers. *Int. J. Artif. Intell. Tools* **2002**, *11*, 587–601. [CrossRef]
11. Hamine, V.; Helman, P. Learning optimal augmented Bayes networks. *arXiv* **2005**, arXiv:cs/0509055. [CrossRef]
12. Jing, Y.; Pavlovic, V.; Rehg, J.M. Efficient discriminative learning of Bayesian network classifier via boosted augmented naive Bayes. In *Proceedings of the 22nd International Conference on Machine Learning (ICML-05)*; Raedt, L.D., Wrobel, S., Eds.; ACM: New York, NY, USA, 2005; pp. 369–376.
13. Carvalho, A.M.; Oliveira, A.L.; Sagot, M.F. Efficient Learning of Bayesian Network Classifiers. In *Proceedings of the AI 2007: Advances in Artificial Intelligence, Gold Coast, Australia, 2–6 December 2007*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 16–25. [CrossRef]
14. Bouhamed, H.; Masmoudi, A.; Lecroq, T.; Rebai, A. A New Approach for Bayesian Classifier Learning Structure via K2 Algorithm. In *Proceedings of the Emerging Intelligent Computing Technology and Applications, 8th International Conference, ICIC 2012, Huangshan, China, 25–29 July 2012*; Communications in Computer and Information Science; Huang, D., Gupta, P., Zhang, X., Premaratne, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 304, pp. 387–393. [CrossRef]
15. Grossman, D.; Domingos, P. Learning Bayesian Network Classifiers by Maximizing Conditional Likelihood. In Proceedings of the Twenty-First International Conference on Machine Learning (ICML'04), New York, NY, USA, 4–8 July 2004; p. 46. [CrossRef]
16. Madden, M.G. On the Classification Performance of TAN and General Bayesian Networks. In *Research and Development in Intelligent Systems XXV*; Bramer, M., Petridis, M., Coenen, F., Eds.; Springer: London, UK, 2009; pp. 3–16.
17. Cooper, G.F.; Herskovits, E. A Bayesian Method for the Induction of Probabilistic Networks from Data. *Mach. Learn.* **1992**, *9*, 309–347. [CrossRef]
18. Gheisari, S.; Meybodi, M.R.; Dehghan, M.; Ebadzadeh, M.M. Bayesian network structure training based on a game of learning automata. *Int. J. Mach. Learn. Cybern.* **2017**, *8*, 1093–1105. [CrossRef]
19. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*; Springer Series in Statistics; Springer: New York, NY, USA, 2001.
20. Fang, W.; Zhang, W.; Ma, L.; Wu, Y.; Yan, K.; Lu, H.; Sun, J.; Wu, X.; Yuan, B. An efficient Bayesian network structure learning algorithm based on structural information. *Swarm Evol. Comput.* **2023**, *76*, 101224. [CrossRef]
21. Blanco, R.; Inza, I.; Larrañaga, P. Learning Bayesian networks in the space of structures by estimation of distribution algorithms. *Int. J. Intell. Syst.* **2003**, *18*, 205–220. [CrossRef]
22. Li, X.; He, X.; Chen, C. A Method for Learning Bayesian Networks by Using Immune Binary Particle Swarm Optimization. In *Proceedings of the Database Theory and Application*; Communications in Computer and Information Science; Slezak, D., Kim, T., Zhang, Y., Ma, J., Chung, K., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; Volume 64, pp. 115–121. [CrossRef]
23. Kuo, S.C.; Wang, H.J.; Wei, H.Y.; Chen, C.C.; Li, S.T. Applying MDL in PSO for learning Bayesian networks. In Proceedings of the 2011 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2011), Taipei, Taiwan, 27–30 June 2011; pp. 1587–1592. [CrossRef]
24. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I.H. The WEKA Data Mining Software: An Update. *SIGKDD Explor. Newsl.* **2009**, *11*, 10–18. [CrossRef]
25. Li, G.; Xing, L.; Chen, Y. A New BN Structure Learning Mechanism Based on Decomposability of Scoring Functions. In *Bio-Inspired Computing–Theories and Applications: 10th International Conference, BIC-TA 2015 Hefei, China, 25–28 September 2015, Proceedings*; Springer: Berlin/Heidelberg, Germany, 2015; Chapter 19, pp. 212–224. [CrossRef]
26. Wu, T.; Qian, H.; Liu, Z.; Zhou, J.; Zhou, A. Bi-objective evolutionary Bayesian network structure learning via skeleton constraint. *Front. Comput. Sci.* **2023** *17*, 176350. [CrossRef]
27. Aguilera-Rueda, V.J.; Cruz-Ramírez, N.; Mezura-Montes, E. Data-Driven Bayesian Network Learning: Towards a Bi-Objective Approach to Address the Bias-Variance Decomposition. *Res. Comput. Sci.* **2020**, *149*, 9–17. [CrossRef]
28. Aguilera-Rueda, V.J.; Cruz-Ramírez, N.; Mezura-Montes, E. Data-driven Bayesian Network learning: A bi-objective approach to address the bias-variance decomposition. *Math. Comput. Appl.* **2020**, *25*, 37. [CrossRef]
29. Ross, B.J.; Zuviria, E. Evolving dynamic Bayesian networks with Multi-objective genetic algorithms. *Appl. Intell.* **2007**, *26*, 13–23. [CrossRef]
30. Sun, B.; Zhang, X.; Jiang, J.; Gong, J.; Lin, D. Bayesian network structure learning by opposition-based learning. *Sci. Rep.* **2025**, *15*, 18447. [CrossRef]

https://doi.org/10.3390/mca31030094

31. Fallahnejad, M.; Rezaeitabar, V.; Kazemi, M. Elastic net–based K2 algorithm for Bayesian network structure learning. *Statistics* **2025**, 1–19. [CrossRef]
32. Yang, W.T.; Tamssaouet, K.; Dauzere-Peres, S. Bayesian network structure learning using scatter search. *Knowl.-Based Syst.* **2024**, *300*, 112149. [CrossRef]
33. He, C.; Wang, P.; Tian, L.; Di, R.; Wang, Z.; Yang, Y. A novel structure learning method of Bayesian networks based on the neighboring complete node ordering search. *Neurocomputing* **2024**, *585*, 127620. [CrossRef]
34. Liu, H.; Liu, H.; Cai, Y.; Shi, Q.;Wang, N.; Zhang, L.; Li, S.; Cui, S. An improved Harris hawks optimization for Bayesian network structure learning via genetic operators. *Soft Comput.* **2023**, *27*, 14659–14672. [CrossRef]
35. Wang, N.; Liu, H.; Zhang, L.; Cai, Y.; Shi, Q. An efficient skeleton learning approach-based hybrid algorithm for identifying Bayesian network structure. *Eng. Appl. Artif. Intell.* **2024**, *133*, 108105. [CrossRef]
36. Dang, Y.; Gao, X.; Wang, Z. A Novel Hyper-Heuristic Algorithm for Bayesian Network Structure Learning Based on Feature Selection. *Axioms* **2025**, *14*, 538. [CrossRef]
37. Yan, Y.; Zhang, Y.; Zhou, L.; Zhang, F.; Wang, H. Acoustic signals-based probabilistic fault diagnosis for expansion joints of small and medium bridges using Bayesian ensemble learning. *Eng. Struct.* **2026**, *354*, 122379. [CrossRef]
38. Li, H.; Zhang, Y.; Wang, H.; Xu, Y.; Li, D. An explainable deep ensemble model for probabilistic prediction of typhoon effects on a long-span bridge. *J. Wind Eng. Ind. Aerodyn.* **2026**, *269*, 106330. [CrossRef]
39. Engelbrecht, A.P. *Fundamentals of Computational Swarm Intelligence*; John Wiley & Sons: Hoboken, NJ, USA, 2006.
40. Nebro, A.J.; Durillo, J.J.; García-Nieto, J.; Coello, C.A.; Luna, F.; Alba, E. SMPSO: A New PSO-based Metaheuristic for Multi-objective Optimization. In *Proceedings of the 2009 IEEE Symposium on Computational Intelligence in Multicriteria Decision-Making (MCDM 2009)*; IEEE Press: New York, NY, USA, 2009; pp. 66–73.
41. Nebro, A.J.; Durillo, J.J.; Coello, C.A. Analysis of leader selection strategies in a multi-objective Particle Swarm Optimizer. In Proceedings of the IEEE Congress on Evolutionary Computation, Cancun, Mexico, 20–23 June 2013; pp. 3153–3160.
42. Kurgan, L.A.; Cios, K.J. CAIM discretization algorithm. *IEEE Trans. Knowl. Data Eng.* **2004**, *16*, 145–153. [CrossRef]
43. Coello Coello, C.A.; Van Veldhuizen, D.A.; Lamont, G.B. *Evolutionary Algorithms for Solving Multi-Objective Problems*; Kluwer Academic Publishers: New York, NY, USA, 2002. [CrossRef]
44. Deb, K.; Kalyanmoy, D. *Multi-Objective Optimization Using Evolutionary Algorithms*; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2001.
45. Friedman, N.; Goldszmidt, M. *Learning Bayesian Networks from Data*; Morgan Kaufmann: Burlington, MA, USA, 1999.
46. Grünwald, P.D. *The Minimum Description Length Principle (Adaptive Computation and Machine Learning)*; The MIT Press: Cambridge, MA, USA, 2007.
47. Cruz-Ramírez, N.; Acosta-Mesa, H.G.; Mezura-Montes, E.; Guerra-Hernández, A.; Hoyos-Rivera, G.d.J.; Barrientos-Martínez, R.E.; Gutiérrez-Fragoso, K.; Nava-Fernández, L.A.; González-Gaspar, P.; Novoa-del Toro, E.M.; et al. How good is crude MDL for solving the bias-variance dilemma? An empirical investigation based on Bayesian networks. *PLoS ONE* **2014**, *9*, e92866. [CrossRef]
48. Teyssier, M. Ordering-based search: A simple and effective algorithm for learning Bayesian networks. In Proceedings of the UAI, Edinburgh, UK, 26–29 July 2005; pp. 584–590.
49. Kelly, M.; Longjohn, R.; Nottingham, K. UCI Machine Learning Repository. 2023. Available online: https://archive.ics.uci.edu/ (accessed on 7 January 2025).
50. Allen, T.V.; Greiner, R. Model Selection Criteria for Learning Belief Nets: An Empirical Comparison. In Proceedings of the International Conference on Machine Learning, Stanford, CA, USA, 29 June–2 July 2000; pp. 1047–1054.
51. Friedman, N.; Geiger, D.; Goldszmidt, M.; Provan, G.; Langley, P.; Smyth, P. Bayesian Network Classifiers. *Mach. Learn.* **1997**, *29*, 131–163. [CrossRef]
52. Kelner, R.; Lerner, B. Learning Bayesian network classifiers by risk minimization. *Int. J. Approx. Reason.* **2012**, *53*, 248–272. [CrossRef]
53. Drugan, M.M.; Wiering, M.A. Feature selection for Bayesian network classifiers using the MDL-FS score. *Int. J. Approx. Reason.* **2010**, *51*, 695–717. [CrossRef]
54. Shi, Y.; Eberhart, R. A modified particle swarm optimizer. In *Proceedings of the 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No. 98TH8360), Anchorage, AK, USA, 4–9 May 1998*; IEEE: New York, NY, USA, 1998; pp. 69–73. [CrossRef]
55. Clerc, M.; Kennedy, J. The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58–73. [CrossRef]
56. Fan, S.K.S.; Chang, J.M. A Modified Particle Swarm Optimizer Using an Adaptive Dynamic Weight Scheme. In *Proceedings of the Digital Human Modeling*; Duffy, V.G., Ed.; Spring: Berlin/Heidelberg, Germany, 2007; pp. 56–65.
57. Gasse, M.; Aussem, A.; Elghazel, H. A hybrid algorithm for Bayesian network structure learning with application to multi-label learning. *Expert Syst. Appl.* **2014**, *41*, 6755–6772. [CrossRef]

https://doi.org/10.3390/mca31030094

58. Larrañaga, P.; Karshenas, H.; Bielza, C.; Santana, R. A Review on Evolutionary Algorithms in Bayesian Network Learning and Inference Tasks. *Inf. Sci.* **2013**, *233*, 109–125. [CrossRef]
59. Wong, M.L.; Leung, K.S. An efficient data mining method for learning Bayesian networks using an evolutionary algorithm-based hybrid approach. *IEEE Trans. Evol. Comput.* **2004**, *8*, 378–404. [CrossRef]
60. Acid, S.; de Campos, L.M.; Castellano, J.G. Learning Bayesian Network Classifiers: Searching in a Space of Partially Directed Acyclic Graphs. *Mach. Learn.* **2005**, *59*, 213–235. [CrossRef]
61. López-Ibáñez, M.; Dubois-Lacoste, J.; Pérez Cáceres, L.; Birattari, M.; Stützle, T. The irace package: Iterated racing for automatic algorithm configuration. *Oper. Res. Perspect.* **2016**, *3*, 43–58. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.

https://doi.org/10.3390/mca31030094

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

Mathematical and Computational Applications Editorial Office
E-mail: mca@mdpi.com
www.mdpi.com/journal/mca



Disclaimer/Publisher's Note: The title and front matter of this reprint are at the discretion of the Guest Editors. The publisher is not responsible for their content or any associated concerns. The statements, opinions and data contained in all individual articles are solely those of the individual Editors and contributors and not of MDPI. MDPI disclaims responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.





Academic Open
Access Publishing

mdpi.com

ISBN 978-3-7258-9132-0