algorithms

Special Issue Reprint

Algorithmic Innovations

Bridging Theoretical Foundations and Practical Applications

Edited by
Carolina Del-Valle-Soto and Ramiro Velazquez

mdpi.com/journal/algorithms

MDPI

Algorithmic Innovations: Bridging Theoretical Foundations and Practical Applications

Algorithmic Innovations: Bridging Theoretical Foundations and Practical Applications

Guest Editors

Carolina Del-Valle-Soto
Ramiro Velazquez

MDPI

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

Guest Editors

Carolina Del-Valle-Soto
School of Engineering
Universidad Panamericana
Zapopan
Mexico

Ramiro Velazquez
School of Engineering
Universidad Panamericana
Aguascalientes
Mexico

Editorial Office
MDPI AG
Grosspeteranlage 5
4052 Basel, Switzerland

This is a reprint of the Special Issue, published open access by the journal *Algorithms* (ISSN 1999-4893), freely accessible at: https://www.mdpi.com/journal/algorithms/special_issues/6D6P401X96.

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 978-3-7258-9207-5 (Hbk)
ISBN 978-3-7258-9208-2 (PDF)
https://doi.org/10.3390/books978-3-7258-9208-2

© 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**

Qingyang Yuan and Bo Zhang
A Clustering-Based Dimensionality Reduction Method Guided by POD Structures and Its Application to Convective Flow Problems
Reprinted from: *Algorithms* **2025**, *18*, 366, https://doi.org/10.3390/a18060366 . . . . . **1**

Rubén Ruiz-Torrubiano, Deepak Dhungana, Sarita Paudel and Himanshu Buckchash
Modeling Local Search Metaheuristics Using Markov Decision Processes
Reprinted from: *Algorithms* **2025**, *18*, 512, https://doi.org/10.3390/a18080512 . . . . . **30**

Xiaochuan Wang, Na Li and Xingchen Jin
A Reinforcement Learning Hyper-Heuristic with Cumulative Rewards for Dual-Peak Time-Varying Network Optimization in Heterogeneous Multi-Trip Vehicle Routing
Reprinted from: *Algorithms* **2025**, *18*, 536, https://doi.org/10.3390/a18090536 . . . . . **44**

Kun Jiang and Zhihong Fu
Backward Signal Propagation: A Symmetry-Based Training Method for Neural Networks
Reprinted from: *Algorithms* **2025**, *18*, 594, https://doi.org/10.3390/a18100594 . . . . . **72**

Carolina Del-Valle-Soto, José A. Del-Puerto-Flores, Leonardo J. Valdivia, Aimé Lay-Ekuakille and Paolo Visconti
A Resilient Energy-Efficient Framework for Jamming Mitigation in Cluster-Based Wireless Sensor Networks
Reprinted from: *Algorithms* **2025**, *18*, 614, https://doi.org/10.3390/a18100614 . . . . . **92**

Luis Angel Iturralde Carrera, Omar Rodríguez-Abreo, Jose Manuel Álvarez-Alvarado, Gerardo I. Pérez-Soto, Carlos Gustavo Manriquez-Padilla and Juvenal Rodríguez-Reséndiz
A New Genetic Algorithm-Based Optimization Methodology for Energy Efficiency in Buildings
Reprinted from: *Algorithms* **2026**, *19*, 27, https://doi.org/10.3390/a19010027 . . . . . **121**

Manuel Cedillo-Hernandez, Antonio Cedillo-Hernandez, Francisco Javier Garcia-Ugalde and Juan Carlos Sanchez-Garcia
Semi-Fragile Watermarking Scheme for High-Resolution Color Images: Tamper Identification, Ownership Authentication, and Self-Recovery
Reprinted from: *Algorithms* **2026**, *19*, 28, https://doi.org/10.3390/a19010028 . . . . . **151**

Panagiotis Korkidis and Anastasios Dounis
Bayesian Optimisation and Adaptive Evolutionary Algorithms for Higher-Order Fuzzy Models with Application on Wind Speed Prediction
Reprinted from: *Algorithms* **2026**, *19*, 46, https://doi.org/10.3390/a19010046 . . . . . **177**

Miguel-Angel Gil-Rios, Ivan Cruz-Aceves and Arturo Hernandez-Aguirre
Hybrid Evolutionary Multi-Objective Method for Automatic Design of a Lightweight CNN Architecture Applied to Coronary Stenosis Classification
Reprinted from: *Algorithms* **2026**, *19*, 47, https://doi.org/10.3390/a19010047 . . . . . **209**

Ivan Cruz-Aceves, Erick G.G. de Paz, Arturo Hernandez-Aguirre and Miguel-Angel Gil-Rios
Optimizing Facial Muscle Activation Features for Emotion Recognition: A Metaheuristic Approach Using Inner Triangle Points
Reprinted from: *Algorithms* **2026**, *19*, 57, https://doi.org/10.3390/a19010057 . . . . . **235**

Luis E. Tonix-Gleason, José A. Del-Puerto-Flores, Fernando Peña-Campos, Dunstano del Puerto-Flores, Juan-Carlos López-Pimentel, Carolina Del-Valle-Soto and Luis René Vela-Garcia
Efficient Deep Learning-Based M-PSK Detection for OFDM V2V Systems Using MobileNetV3
Reprinted from: *Algorithms* **2026**, *19*, 210, https://doi.org/10.3390/a19030210 . . . . . **256**

About the Editors

Carolina Del-Valle-Soto

Carolina Del-Valle-Soto is a Research Professor at the School of Engineering of Universidad Panamericana, Campus Guadalajara, Mexico, and a member of the Mexican National System of Researchers. Her research addresses wireless sensor networks, routing protocols, energy-aware and resilient communications, the Internet of Things, and applied machine learning for network security and health monitoring. She has published extensively in international peer-reviewed journals, leads collaborative projects with academic and industrial partners across Latin America and Europe, and serves as an editor and reviewer for several indexed journals in computing and engineering.

Ramiro Velazquez

Ramiro Velazquez is a Professor at the School of Engineering of Universidad Panamericana, Campus Aguascalientes, Mexico. His work centers on robotics and mechatronics, haptic interfaces, assistive technologies for people with visual and motor impairments, smart actuators, and embedded systems. He has coordinated international research projects, published widely in journals and conferences on robotics and human–machine interaction, and collaborates regularly with research groups in Europe and the Americas.

Preface

Algorithms sit at the meeting point of mathematics and engineering practice. A result that is elegant on paper is only useful once it survives noisy data, limited energy budgets, and the messy constraints of a deployed system. This Reprint was assembled around that tension. Our aim was to gather work in which a clear theoretical formulation and a convincing empirical validation appear side by side, and to make the connection between the two visible to the reader.

The eleven contributions collected here cover algorithm design and optimization, metaheuristics and evolutionary computation, reinforcement learning, deep learning architectures, dimensionality reduction, and the design of resilient, energy-aware communication networks. They are applied to problems as varied as coronary stenosis classification, emotion recognition, jamming mitigation in wireless sensor networks, tamper detection in digital images, vehicle routing under time-varying demand, energy efficiency in buildings, wind speed forecasting, and vehicle-to-vehicle communication. The diversity is deliberate: it shows that a shared methodological core supports very different domains.

We are grateful to every author who trusted this Special Issue with their work, to the reviewers whose careful assessment raised the quality of each manuscript, and to the editorial team at *Algorithms* for their steady support throughout the process. We hope this Reprint is useful to researchers, graduate students, and practitioners who want to see how algorithmic theory becomes engineering practice, and that it encourages further interdisciplinary collaboration.

Carolina Del-Valle-Soto and Ramiro Velazquez
Guest Editors

algorithms

MDPI

Article

A Clustering-Based Dimensionality Reduction Method Guided by POD Structures and Its Application to Convective Flow Problems

Qingyang Yuan and Bo Zhang *

School of Energy and Power Engineering, Dalian University of Technology, Dalian 116081, China; yuan_qingy@mail.dlut.edu.cn

* Correspondence: zhangbo@dlut.edu.cn

Abstract

Proper orthogonal decomposition (POD) is a widely used linear dimensionality reduction technique, but it often fails to capture critical features in complex nonlinear flows. In contrast, clustering methods are effective for nonlinear feature extraction, yet their application in dimensionality reduction methods is hindered by unstable cluster initialization and inefficient mode sorting. To address these issues, we propose a clustering-based dimensionality reduction method guided by POD structures (C-POD), which uses POD preprocessing to stabilize the selection of cluster centers. Additionally, we introduce an entropy-controlled Euclidean-to-probability mapping (ECEPM) method to improve modal sorting and assess mode importance. The C-POD approach is evaluated using the one-dimensional Burgers' equation and a two-dimensional cylinder wake flow. Results show that C-POD achieves higher accuracy in dimensionality reduction than POD. Its dominant modes capture more temporal dynamics, while higher-order modes offer better physical interpretability. When solving an inverse problem using sparse sensor data, the Gappy C-POD method improves reconstruction accuracy by 19.75% and enhances the lower bound of reconstruction capability by 13.4% compared to Gappy POD. Overall, C-POD demonstrates strong potential for modeling and reconstructing complex nonlinear flow fields, providing a valuable tool for dimensionality reduction methods in fluid dynamics.

Keywords: POD (proper orthogonal decomposition); clustering-based dimensionality reduction method guided by POD structures (C-POD); dimensionality reduction method; nonlinear model order reduction (NMOR)

1. Introduction

Efficient representation of high-dimensional spatiotemporal flow fields and accurate capture of their dynamic evolution have long posed significant challenges in computational fluid dynamics (CFD). This is especially true for flow systems, such as cylinder wake flows [1,2], which exhibit both classical and complex behaviors. Although traditional CFD methods deliver high computational accuracy, their substantial computational costs and storage requirements severely limit their applicability in real-time prediction and optimization.

To overcome these limitations, a wide range of model reduction strategies—broadly referred to as reduced-order models (ROMs) [3,4]—have been used to construct low-dimensional representations that capture dominant flow features. The fundamental strength of such approaches lies in establishing a robust mapping between high-dimensional dynamical systems

and their low-dimensional surrogates—a task that involves not only compressing data but also uncovering the essential physical mechanisms underlying fluid motion.

Traditional ROM methods can be broadly categorized into three types:

(1) Modal decomposition projection methods (e.g., POD-Galerkin [5]) construct low-dimensional subspaces by extracting dominant modes and projecting governing equations onto these subspaces.
(2) Balanced truncation methods [6] utilize controllability and observability analysis to retain state variables most significant to input–output behavior.
(3) Harmonic balance methods [7] address periodic flows by approximating steady-state solutions in the frequency domain using a finite set of Fourier basis functions.

These approaches are physically grounded and interpretable, offering energy-optimal modal selection and high-fidelity reconstruction. However, they often rely on linear subspace assumptions and require precise numerical integration, which can be limiting for nonlinear or geometrically complex flows. Beyond traditional projection-based approaches, data-driven techniques have enabled a broader exploration of flow structures without reliance on governing equations. These can be classified into:

(1) Data-driven modal methods, like snapshot POD [8] and dynamic mode decomposition (DMD) [9], which extract coherent structures directly from simulation or experimental data. They are equation-free but often sensitive to noise and require extensive datasets.
(2) Neural-network-based approaches, such as autoencoders [10] and their derivatives [11], which learn nonlinear latent representations but may suffer from overfitting and lack interpretability.
(3) Hybrid strategies combining ROM and machine learning, such as POD-RBF, POD-LSTM [12], and physics-informed neural networks (PINNs) [13,14], aiming to balance physical relevance and learning flexibility.

In addition to standard POD, various studies have expanded its framework:

- Colanera [15] proposed a robust spectral POD (SPOD) integrating robust principal component analysis for improved noise resilience.
- Gu [16] introduced frequency-domain POD (FD-POD) to reconstruct unsteady flows using embedded frequency information.
- Bui [17] developed POD-ISAT, combining POD with in situ adaptive tabulation for efficient steady-state PDE approximation.

Snapshot POD remains a popular tool due to its algorithmic simplicity and modal resolution, but its linear nature limits its ability to capture nonlinear interactions, such as turbulence. This motivates the search for more expressive and adaptive dimensionality reduction techniques.

Among these, clustering-based methods have emerged as powerful alternatives for representing nonlinear, multi-regime flow behaviors. Instead of relying on orthogonal basis functions, they group data points by similarity in feature space, making them well-suited to reflect local state transitions and complex dynamics. This is particularly useful in transitional flows, where traditional modal decompositions fall short.

While many clustering-based methods are inspired by ROM principles, in the strictest sense, ROMs refer to low-dimensional dynamical systems derived via projection (e.g., Galerkin or Petrov-Galerkin) from governing equations. The method proposed in this study does not involve such a projection and does not construct time-evolving models. Instead, it focuses on extracting and organizing spatial flow structures directly from data. Therefore, our approach is best categorized as a data-driven dimensionality reduction method, guided by physical insights from POD.

Building on these foundations, clustering-based dimensionality reduction methods have shown increasing promise. Burkardt et al. [18] proposed a CVT-based approach but did not address mode importance. Iqbal et al. [19] introduced modified pole clustering for improved initialization, with emphasis on clustering rather than reduction. Huera-Huarte and D'Adamo [20,21] combined clustering and POD to analyze vortex dynamics, yet lacked an effective mode-ranking strategy. Wei et al. [22] presented CROM, which models transitions via a Markov matrix, but its dimensionality and sparsity hinder application to complex systems.

To tackle the dual challenges of unstable clustering center initialization and the absence of mode ranking, we propose a clustering-based dimensionality reduction method guided by POD structures (C-POD). This framework leverages POD to enhance initialization robustness and introduces a novel entropy-controlled Euclidean-to-probability mapping (ECEPM) to provide probabilistic, interpretable mode ranking.

We validate the C-POD framework on the one-dimensional Burgers' equation and a two-dimensional cylinder wake flow. We further test its performance in an inverse reconstruction task using sparse sensor data.

To enhance conceptual clarity, Figure 1 summarizes the taxonomy of the methods discussed.

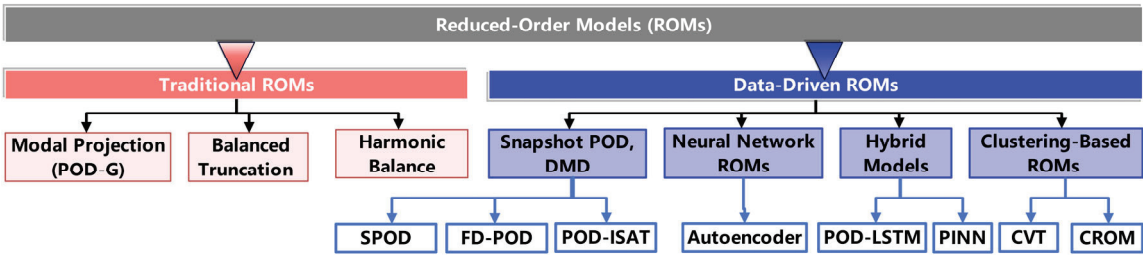


Figure 1. Taxonomy of ROM.

In summary, this study offers the following contributions:

- A clustering-based dimensionality reduction framework guided by POD, improving stability in nonlinear mode extraction.
- A novel ECEPM-based probabilistic mode-ranking method.
- Empirical validation demonstrating improved interpretability and reconstruction accuracy over traditional POD methods.

2. C-POD Method and Modal Sorting Method

2.1. C-POD Method

The proper orthogonal decomposition (POD) method, based on singular value decomposition (SVD), is a well-established, data-driven model reduction technique. POD generates a set of orthogonal basis functions that encapsulate the dominant features of a system. By employing linear combinations of these selected basis functions, one can effectively approximate the original system, thereby transforming a high-dimensional problem into a low-dimensional representation. To preserve computational resources, the snapshot POD method proposed by Sirovich is frequently used for dimensionality reduction of the dataset [23]. However, POD is inherently a linear reduction technique. Although it captures the primary dynamic features of linear systems, it may not fully represent the complexities of nonlinear flows, particularly when nonlinear behaviors vary across multiple time steps and spatial domains. In such cases, the low-rank approximation afforded by POD might not sufficiently capture these nonlinear effects.

Clustering methods exhibit robust capabilities in extracting nonlinear features, effectively capturing patterns and regularities in data, especially within complex nonlinear systems. Moreover, these methods can autonomously adapt to diverse data patterns, mapping both linear and nonlinear datasets into a low-dimensional space that elucidates underlying dynamic characteristics. Consequently, this paper integrates clustering concepts with model reduction techniques by proposing a clustering-based dimensionality reduction framework, referred to as C-POD, to enhance the effectiveness of proper orthogonal decomposition (POD) in addressing nonlinear model reduction challenges. The construction process is detailed as follows:

(1) Arrange the responses of m data points under n different operating conditions as columns to form the database matrix $\boldsymbol{D}$:

$$\boldsymbol{D} = \{d_1, d_2, \ldots, d_n\} \in R^{m \times n} \tag{1}$$

Here, $d_i \in R^m$ (i = 1, 2, …, n) represents the data snapshot under the i-th operating condition.

(2) Determining initial cluster centers based on the snapshot POD method [23]. First, the correlation matrix of the snapshot matrix $\boldsymbol{D}$ is constructed:

$$\boldsymbol{C} = \boldsymbol{D}\boldsymbol{D}^T \in R^{m \times m} \tag{2}$$

Solving the eigenvalue problem of the correlation matrix $\boldsymbol{C}$:

$$\boldsymbol{C}\boldsymbol{\varphi}_i = \boldsymbol{\lambda}_i \boldsymbol{\varphi}_i \ (i = 1, 2, \ldots m) \tag{3}$$

Here, $\boldsymbol{\lambda}_i$ represents the eigenvalue, and $\boldsymbol{\varphi}_i$ is the corresponding eigenvector. The eigenvalues are arranged in descending order:

$$\boldsymbol{\lambda}_1 \geq \boldsymbol{\lambda}_2 \geq \ldots \geq \boldsymbol{\lambda}_m \geq 0 \tag{4}$$

The corresponding eigenvectors $\boldsymbol{\varphi}_i$ form an orthogonal basis, and the first K eigenvectors are selected as the initial cluster centers:

$$\boldsymbol{\mu}_i{}^{(0)} = \boldsymbol{\varphi}_i (i = 1, 2, \ldots K) \tag{5}$$

(3) Sample assignment. Each column vector d_j (j = 1, 2, …, n) from the data matrix $\boldsymbol{D}$ is assigned to the nearest cluster center, forming K clusters:

$$\boldsymbol{S}_i^{(t)} = \left\{ \boldsymbol{d} \in \boldsymbol{D} \middle| \left\| \boldsymbol{d} - \boldsymbol{\mu}_i^{(t-1)} \right\|^2 \leq \left\| \boldsymbol{d} - \boldsymbol{\mu}_j^{(t-1)} \right\|^2, \forall j \neq i \right\} \tag{6}$$

Here, $\boldsymbol{S}_i{}^{(t)}$ represents the number of vectors in the i cluster, $\boldsymbol{\mu}_\mathrm{i}$ denotes the i-th cluster centroid, t is the current iteration count, and $||\cdot||$ represents the Euclidean distance norm.

(4) Updating the cluster centers. The cluster centers are updated by calculating the mean of all vectors in each cluster $\boldsymbol{S}_i$, and this mean becomes the new cluster centroid:

$$\boldsymbol{\mu}_i^{(t)} = \left(\frac{1}{\left|\boldsymbol{S}_i^{(t)}\right|}\right) \sum_{x \in \boldsymbol{S}_i^{(t)}} x (i = 1, 2, \ldots, K) \tag{7}$$

where $\boldsymbol{S}_i^{(t)}$ denotes the set of snapshots assigned to cluster i. Each $\boldsymbol{\mu}_i^{(t)}$ can be interpreted as a local average attractor in the trajectory space of the flow field—analogous to a mean coherent structure in physical space.

(5) Repeat steps (6) and (7) until the following convergence criterion is met:

$$\left\| \boldsymbol{\mu}_i^{(t)} - \boldsymbol{\mu}_i^{(t-1)} \right\|^2 < \varepsilon, \forall i \in \{1, 2, \ldots, K\} \tag{8}$$

(6) To obtain the modal coefficients, project the original data matrix $\boldsymbol{D}$ onto the cluster bases $\boldsymbol{\mu}$ by solving the following least squares problem [24]:

$$\boldsymbol{a}_r = (\boldsymbol{\mu}^T \boldsymbol{\mu})^{-1} \boldsymbol{\mu}^T \boldsymbol{d}_r \ (r = 1, 2, \ldots, n) \tag{9}$$

Here, $\boldsymbol{a}_r \in R^k$ is the modal coefficient vector corresponding to the r-th snapshot. Physically, solving the least squares problem in Equation (9) provides a projection of each snapshot $\boldsymbol{d}_r$ onto the low-dimensional basis formed by cluster centers $\{\boldsymbol{\mu_1}, \boldsymbol{\mu_2}, \ldots, \boldsymbol{\mu}_K\}$. These cluster centers are not arbitrary vectors but statistically representative patterns extracted from the dataset. As such, the projection yields modal coefficients $\boldsymbol{a}_r$ that quantify how much each typical flow pattern (i.e., cluster mode) contributes to the reconstruction of the current state. The full data reconstruction expression is:

$$\hat{\boldsymbol{D}} = \boldsymbol{\mu} \boldsymbol{A} \tag{10}$$

where $\boldsymbol{A} = [a_1, a_2, \ldots, a_n] \in R^{k \times n}$ is the modal coefficient matrix, and $\hat{\boldsymbol{D}}$ is the reconstructed data matrix. Through the C-POD method, the database matrix $\boldsymbol{D}$ is decomposed into low-dimensional clustering bases $\boldsymbol{\mu}$ and the modal coefficient matrix $\boldsymbol{A}$. The construction of the C-POD method is illustrated in Algorithm 1, which presents the pseudocode of the entire procedure.

Algorithm 1: C-POD Method

Input: $\boldsymbol{D} = \{d_1, d_2, \ldots, d_n\} \in R^{m \times n}$ //Snapshot matrix

K //Number of clusters

Output:

$\boldsymbol{\mu} = \{\boldsymbol{\mu_1}, \boldsymbol{\mu_2}, \ldots, \boldsymbol{\mu}_K\}$ //Cluster basis

$\boldsymbol{A} = [\boldsymbol{a_1}\ \boldsymbol{a_2} \ldots \boldsymbol{a_n}]$ //Modal coefficients

Steps:

1. POD Initialization

   - Solving the eigenvalue problem: $\boldsymbol{C\varphi}_i = \boldsymbol{\lambda}_i \boldsymbol{\varphi}_i \ (i = 1, 2, \ldots m)$

   - Select first K eigenvectors as the initial cluster centers: $\boldsymbol{\mu}_i{}^{(0)} = \boldsymbol{\varphi}_i (i = 1, 2, \ldots K)$

2. Clustering

   - Assign each snapshot vector to its closest cluster center

   - Compute cluster centroids: $\boldsymbol{\mu}_i^{(t)} = (\frac{1}{\left|\boldsymbol{S}_i^{(t)}\right|}) \sum_{x \in \boldsymbol{S}_i^{(t)}} x (i = 1, 2, \ldots, K)$

3. Compute modal coefficients

   - Modal coefficient vector $\boldsymbol{a}_r \in R^k$: $\boldsymbol{a}_r = (\boldsymbol{\mu}^T \boldsymbol{\mu})^{-1} \boldsymbol{\mu}^T \boldsymbol{d}_r \ (r = 1, 2, \ldots, n)$

4. Return $\boldsymbol{\mu}$, $\boldsymbol{A}$

2.2. Entropy-Controlled Euclidean-to-Probability Mapping Modal Sorting Method

One notable advantage of the POD method is its capacity to quantify the contribution of each mode to the reconstruction of the original data. This contribution is indicated by the singular values, which represent the projection strength of the original data matrix onto the orthogonal modes. Larger singular values imply a more significant contribution from the corresponding mode in reconstructing the data, thereby identifying it as a dominant mode that captures the primary operational characteristics of the system. In contrast, the C-POD method does not involve singular value decomposition during its construction, rendering it infeasible to assess mode contributions based on singular value magnitudes. However, in the initialization phase, we solve the eigenvalue problem of the snapshot correlation matrix—whose eigenvalues are mathematically related to the singular values of the original data matrix. These eigenvalues could theoretically be used to rank modes by importance. Yet, because C-POD replaces orthogonal POD modes with clustering centroids—which are neither orthogonal nor ordered—the original eigenvalue-based ranking becomes inconsistent and inapplicable. To address this, we propose a novel probabilistic mode sorting method: the entropy-controlled Euclidean-to-probability mapping (ECEPM), which offers a robust and interpretable way to evaluate mode significance within the C-POD framework. This probabilistic mode sorting framework incorporates both the spatial proximity between modes and snapshots, and the consistency of modal contributions. It effectively replaces energy-based sorting in POD by offering a statistical interpretation of mode importance, thereby enabling robust feature extraction and mode separation in the clustering-based dimensionality reduction method.

Euclidean distance: The Euclidean distance quantifies the average proximity between each clustering mode and the dataset samples. A smaller distance suggests that the corresponding mode plays a significant role across a broader range of system states, thus representing a dominant dynamic feature. This measure offers clear geometric and physical interpretability.

(1) Euclidean distance matrix construction: Assume the database matrix $\boldsymbol{D} \in R^{m\times n}$, clustering bases $\boldsymbol{\mu} \in R^{m\times b}$, and the column vectors are $\boldsymbol{D} = \{d_1, d_2, \ldots, d_n\} \in R^{m\times n}$ and $\boldsymbol{\mu} \in \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_n\} \in R^{m\times b}$, where $\boldsymbol{C}_{x,y} = \left\|d_x - \boldsymbol{\mu}_y\right\|$.

(2) Compute the column-wise sum of squares for $\boldsymbol{D}$ and $\boldsymbol{\mu}$, as shown in Equations (11) and (12):

$$\boldsymbol{S}_{\boldsymbol{D}} = [\|d_1\|^2, \|d_2\|^2, \ldots, \|d_n\|^2] \in R^{1\times n} \tag{11}$$

$$\boldsymbol{S}_{\boldsymbol{\mu}} = [\|\boldsymbol{\mu}_1\|^2, \|\boldsymbol{\mu}_2\|^2, \ldots, \|\boldsymbol{\mu}_n\|^2] \in R^{1\times b} \tag{12}$$

(3) Construct the inner product matrix $\boldsymbol{G}$:

$$\boldsymbol{G} = \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{D} \tag{13}$$

(4) Construct the squared Euclidean distance matrix:

$$\boldsymbol{C}^2 = \boldsymbol{S}_{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{1}_{1\times n} + \mathbf{1}_{1\times n}\boldsymbol{S}_{\boldsymbol{D}} - 2\boldsymbol{G} \tag{14}$$

where $\mathbf{1}$ is a matrix with all elements equal to 1, so the Euclidean distance matrix is:

$$\boldsymbol{C} = \sqrt{\boldsymbol{C}^2} \tag{15}$$

Probability mapping function construction: Relying solely on distance may lead to overestimating the importance of modes with sporadic proximity. To enhance robustness,

we introduce information entropy as a regulatory factor, penalizing modes with dispersed or unstable contributions. A lower entropy value indicates that the mode has a focused and consistent influence on specific samples, making it more reliable for mode ranking. After obtaining the Euclidean distance matrix $\boldsymbol{C} \in R^{b\times n}$, the goal is to transform it into a weight probability mapping matrix $\boldsymbol{W} \in R^{b\times n}$, where $\boldsymbol{W}_{ij}$ represents the probability value of the *i*-th clustering base for the *j*-th sample. A higher probability value indicates a greater contribution of that clustering base to the current sample value, while satisfying $\sum_{i=1}^{b} \boldsymbol{W}_{ij} = 1$.

(1) Take the negative of the distance matrix $\boldsymbol{C}$ and shift each column, to avoid overflow in exponential calculations, resulting in the matrix $\boldsymbol{Z}$:

$$\boldsymbol{Z} = -\boldsymbol{C} - \boldsymbol{M} \odot \mathbf{1}_b^{\mathrm{T}} \tag{16}$$

where $\boldsymbol{M} = [\boldsymbol{M}_1, \boldsymbol{M}_2, \ldots, \boldsymbol{M}_k]$, $\boldsymbol{M}_j = \max_i(-\boldsymbol{D})_{ij}$, and $\odot$ represents the outer product.

(2) Perform the exponential operation on the shifted matrix and introduce the temperature parameter $\tau > 0$, resulting in:

$$\boldsymbol{E} = \exp(\frac{\boldsymbol{Z}}{\tau}) \tag{17}$$

(3) Normalize the exponential matrix column-wise to obtain the final probability weight matrix $\boldsymbol{W}$:

$$\boldsymbol{W} = \boldsymbol{E} \oslash (\mathbf{1}_n \mathbf{1}_n^{\mathrm{T}} \boldsymbol{E}) \tag{18}$$

where $\oslash$ represents element-wise division, $\mathbf{1}_n^{\mathrm{T}}\boldsymbol{E}$ is the sum of each column, and $\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}\boldsymbol{E}$ extends the column sum to a matrix where all columns are identical. In the construction of the probability mapping function, the temperature parameter $\tau > 0$ is used to adjust the entropy of the probability distribution. The temperature parameter τ plays a critical role in shaping the entropy level of the probability mapping. A smaller τ (e.g., $\tau \approx 0.1$) leads to a more peaked distribution, favoring one dominant mode per snapshot, which increases mode separability but may result in instability. In contrast, a larger τ (e.g., $\tau > 5$) produces a flatter distribution, balancing contributions across multiple modes and improving robustness but potentially reducing interpretability.

To balance these trade-offs, we recommend choosing τ based on the desired sharpness of mode assignment. In this study, τ was selected via grid search in the range [0.1, 5], and the optimal value was determined by evaluating the reconstruction performance using the Cr index. This approach ensures that the mode-ranking mechanism remains both physically meaningful and numerically stable.

Through the above entropy-controlled Euclidean-to-probability mapping method, the probability weight matrix $\boldsymbol{W}$ was obtained. For different samples in the database matrix, $\boldsymbol{W}_{\mathrm{ij}}$ represents the probability value of the *i*-th clustering base for the *j*-th sample. A higher probability value indicates a greater contribution of that clustering base to the current sample value. Therefore, by examining the values in $\boldsymbol{W}$, the importance of each clustering base in the C-POD method was distinguished. The construction process of the C-POD method and the entropy-controlled Euclidean-to-probability mapping method is shown in Figure 2. In Sections 3 and 4, to validate the effectiveness of the C-POD method, we applied it to two classic fluid dynamics problems: the one-dimensional Burgers' equation and the two-dimensional cylinder wake flow. These problems exhibited distinct nonlinear characteristics, making them ideal for testing the practical applicability of the method.

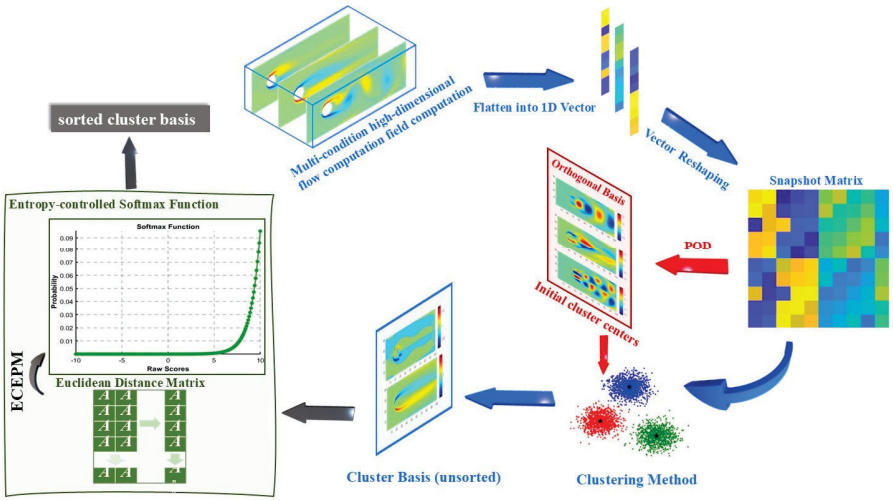


Figure 2. Schematic representation of the C-POD method construction process.

The full procedure of the ECEPM is summarized in Algorithm 2.

Algorithm 2: ECEPM

Input: $\boldsymbol{D} \in R^{m\times n}$ //Snapshot data matrix

$\boldsymbol{\mu} \in R^{m\times b}$ //Clustering bases (b modes)

$\tau > 0$ //Temperature parameter for entropy control

Output:

$\boldsymbol{W} \in R^{b\times n}$ //Probability weight matrix

Steps:

1. Compute the squared norms of $\boldsymbol{D}$ and μ:

-$\boldsymbol{S_D} = [\|d_1\|^2, \|d_2\|^2, \ldots, \|d_n\|^2] \in R^{1\times n}$

-$\boldsymbol{S_\mu} = [\|\boldsymbol{\mu}_1\|^2, \|\boldsymbol{\mu}_2\|^2, \ldots, \|\boldsymbol{\mu}_n\|^2] \in R^{1\times b}$

2. Compute the inner product matrix $\boldsymbol{G} = \boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{D}$

3. Compute the squared Euclidean distance matrix:

-$\boldsymbol{C}^2 = \boldsymbol{S}_{\boldsymbol{\mu}}^{\mathrm{T}}\mathbf{1}_{1\times n} + \mathbf{1}_{1\times n}\boldsymbol{S_D} - 2\boldsymbol{G}$

-$\boldsymbol{C} = \sqrt{\boldsymbol{C}^2}$

4. Shift the distance matrix column-wise to prevent numerical overflow:

$\boldsymbol{Z} = -\boldsymbol{C} - \boldsymbol{M} \odot \mathbf{1}_b^{\mathrm{T}}$ //Broadcasting to each row

5. Compute exponentials with entropy scaling:

$\boldsymbol{E} = \exp(\frac{\boldsymbol{Z}}{\tau})$

6. Normalize E column-wise to obtain the probability matrix:

$\boldsymbol{W} = \boldsymbol{E} \oslash (\mathbf{1}_n\mathbf{1}_n^{\mathrm{T}}\boldsymbol{E})$ //Element-wise division, columns sum to 1

Return:

$\boldsymbol{W}$ //Each column $\boldsymbol{W}$_j contains probabilities of modes for sample j

3. Burgers' Equation and C-POD Method

3.1. Introduction to the One-Dimensional Burgers' Equation

As a nonlinear partial differential equation, the Burgers' equation holds significant research value across various domains, including fluid mechanics [25] and meteorological forecasting [26]. In addition to characterizing the propagation of nonlinear waves, it encapsulates essential physical phenomena, such as viscosity and diffusion. Its inherent nonlinearity facilitates the simulation of complex flow phenomena, particularly in convection-dominated fields where highly nonlinear behaviors are prevalent. Dimensionality reduction methods aim to minimize computational costs while preserving the fundamental characteristics of the system. Consequently, using the Burgers' equation as a test case provides an effective means to evaluate the efficacy and accuracy of these models in addressing nonlinear and transient flow processes. In this study, the accuracy of the POD and C-POD methods was comparatively assessed using a one-dimensional Burgers' equation as the benchmark case. The mathematical formulation of the Burgers' equation is presented as follows:

$$\frac{\partial u(x,t)}{\partial t} + c\frac{\partial u(x,t)}{\partial x} = \beta\frac{\partial^2 u(x,t)}{\partial x^2} \tag{19}$$

Let $u(x, t)$ denote the wave amplitude, c represent the wave speed (i.e., the coefficient of the convection term), and β denote the dissipation coefficient. The one-dimensional Burgers' equation incorporates both convection and diffusion characteristics, reflecting aspects of the heat conduction equation and the Navier–Stokes equations. Consequently, it establishes a conceptual link to both heat conduction and convection phenomena.

Consider the initial condition in the form of a Gaussian pulse, as shown in Equation (20):

$$u(x,0) = \exp(-\frac{(x-x_0)^2}{2\sigma^2}) \tag{20}$$

Under this initial condition, the analytical solution to the one-dimensional Burgers' equation is given by [27]:

$$u(x,t) = \frac{1}{\sqrt{1+\frac{2\beta t}{\sigma^2}}}\exp(-\frac{(x-x_0-ct)^2}{2\sigma^2}) \tag{21}$$

Here, x_0 denotes the initial peak position of the wave, σ represents the width of the wave packet, and c is the propagation speed (consistent with the meaning and value specified in Equation (19)). The analytical solution describes a Gaussian wave packet propagating in the x-direction with velocity c. Notably, the width of the wave packet increases gradually over time, indicative of a diffusion phenomenon, while its amplitude decreases, reflecting a dissipation phenomenon that leads to gradual attenuation during propagation. A schematic illustration of the wave evolution in the one-dimensional Burgers' equation is presented in Figure 3. It is worth noting that although no artificial noise was explicitly added to the test data, numerical disturbances may naturally arise during simulation and data preprocessing. In light of this, the C-POD method framework incorporated proper orthogonal decomposition (POD) as the initialization step for clustering. By extracting dominant energy modes, POD inherently suppressed high-frequency perturbations and served as a natural denoising mechanism. Consequently, C-POD demonstrated a degree of robustness to noise throughout the training and reconstruction processes, enhancing its reliability in unsupervised reduced-order modeling.

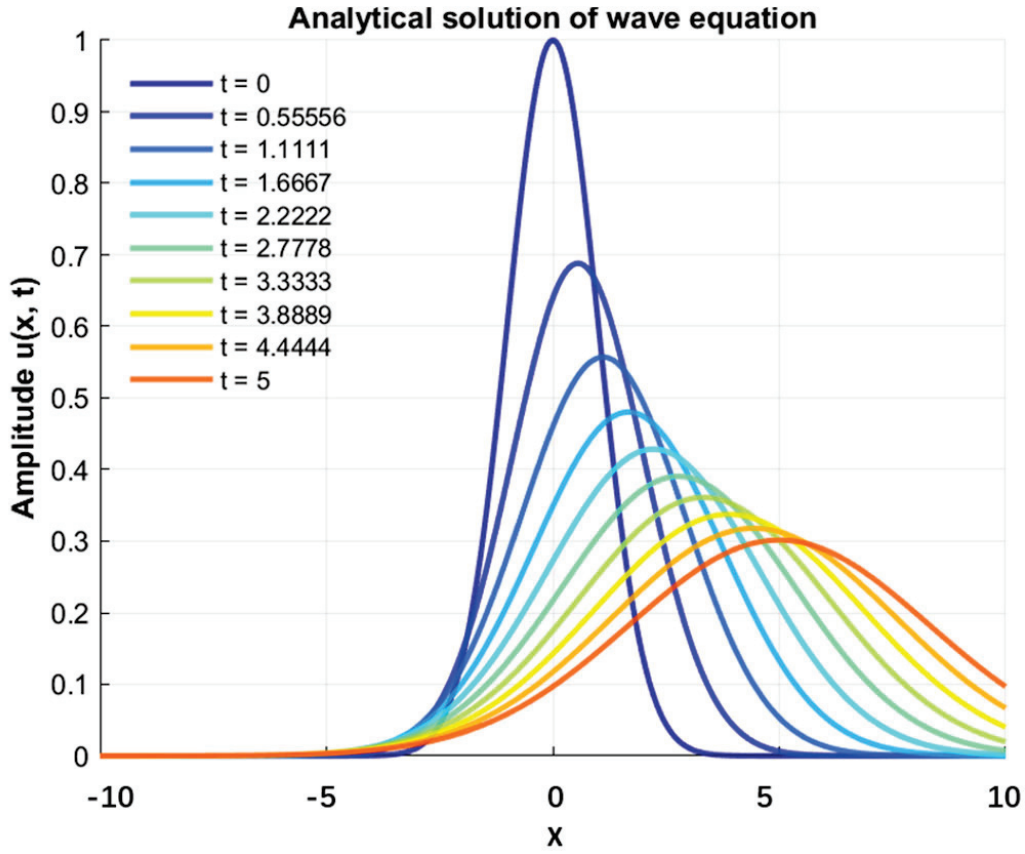


Figure 3. Schematic of wave evolution in the one-dimensional Burgers' equation.

3.2. Database Construction

To evaluate the performance of the POD and C-POD methods, a dataset based on the Burgers' equation was first constructed. The initial conditions and parameters were set as follows:

Initial Conditions:

Initial wave peak position: $x_0 = 0$
Initial wave packet width: $\sigma = 1$

Spatial and Temporal Discretization:

Spatial range: $x \in [0, 10]$, with $Nx = 1000$ uniformly distributed sampling points
Temporal range: $t \in [0, 5]$, with waveforms sampled at intervals of $\Delta t = 0.5$

Independent Variable Settings:

Wave speed c range: $c \in [0.1, 2]$
Dissipation coefficient β range: $\beta \in [0, 1]$

For each pair of (c, β), the Burgers' equation was solved at all time steps within the defined temporal range. At each time step, the spatial field $u(x,t)$ was recorded. Each such spatial field—corresponding to a unique combination of (c, β, t)—was treated as an independent snapshot. As a result, the snapshot matrix was constructed by stacking spatial fields obtained from different (c, β, t) combinations as column vectors. This resulted in a final snapshot matrix $\boldsymbol{D}_1$ of size 1000×303, where 1000 is the number of spatial points and 303 is the total number of snapshot instances across all parameters and time steps. This construction strictly followed the traditional POD framework, ensuring that each snapshot

represented a single spatial field, with variations across snapshots arising from changes in either parameters or time—but not a mixture within a snapshot.

3.3. C-POD Method Accuracy Testing and Modal Decomposition

To evaluate the performance of the POD and C-POD methods, we conducted order reduction on the dataset and reconstructed the original data using various modal orders. The correlation coefficient and root mean square error (RMSE) were employed as evaluation metrics. While these two indicators, respectively, capture different aspects of reconstruction quality—trend similarity and numerical deviation—their combined use is not only less intuitive but also somewhat cumbersome in comparative analysis. Therefore, we introduced a new metric, Cr, defined in Equation (22), which integrates both correlation and error information to provide a more concise and comprehensive assessment of model performance:

$$\mathrm{Cr} = \frac{\mathrm{RMSE}}{|\mathrm{r}|} \tag{22}$$

where the root mean square error (RMSE) and the Pearson correlation coefficient r are defined as shown in Equations (23) and (24):

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{23}$$

$$r = \frac{\sum_{i=1}^{n}(y_i - \overline{y})(\hat{y}_i - \overline{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})\sum_{i=1}^{n}(\hat{y}_i - \overline{\hat{y}})}} \tag{24}$$

where n is the number of data points, y_i is the actual value, $\hat{y}_i$ is the predicted value, $\overline{y}$ is the mean of the actual data, and $\overline{\hat{y}}$ is the mean of the predicted data. A higher Cr value indicates an increasing trend in RMSE or a decreasing trend in the correlation coefficient, suggesting poorer reduced-order modeling performance.

Under varying levels of modal truncation, the datasets were reduced using both the POD and C-POD methods. The resulting modal coefficients and orthogonal bases were then employed to reconstruct the original datasets. Figures 4 and 5 present the average and maximum values of Cr, computed using Equation (22). The average Cr value reflects the overall reduction capability, while the maximum value indicates the lower bound of this capability. As illustrated in Figure 4, both methods exhibited enhanced reduction performance with increasing modal truncation, stabilizing at higher modal orders. For modal orders below 6, the reduction capabilities of the two methods were comparable; however, for order values above this threshold, the C-POD method significantly outperformed POD. Figure 4 further shows that, regarding the lower bound of reduction capabilities, C-POD was superior at modal orders 7, 8, and 9, whereas POD maintained an advantage at other modal orders. These findings preliminarily highlight the superior reduction accuracy of the C-POD method.

To compare the modal extraction capabilities of the C-POD and POD methods, we analyzed the reduced-order modeling of the Burgers' equation under the following conditions: $c = 0.1$, $\beta = 0$, and $c = 2$, $\beta = 1$. At times $t = 0$ and 2 s, both methods were applied to extract reduced-order modes, and the solutions reconstructed using only the first mode were analyzed. The temporal evolution of these first-mode-reconstructed solutions was tracked using the ECEPM method, as illustrated in Figures 6 and 7.

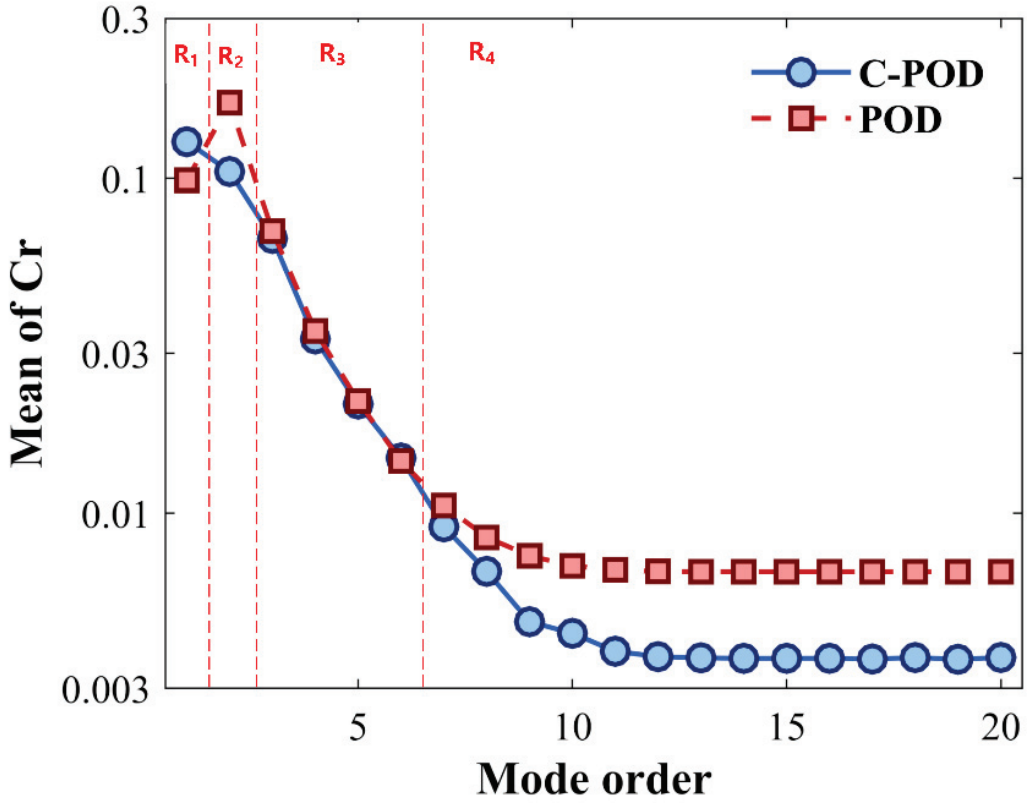


Figure 4. Comparison of average Cr values under different modal orders: illustrating the overall reconstruction accuracy differences between POD and C-POD methods.

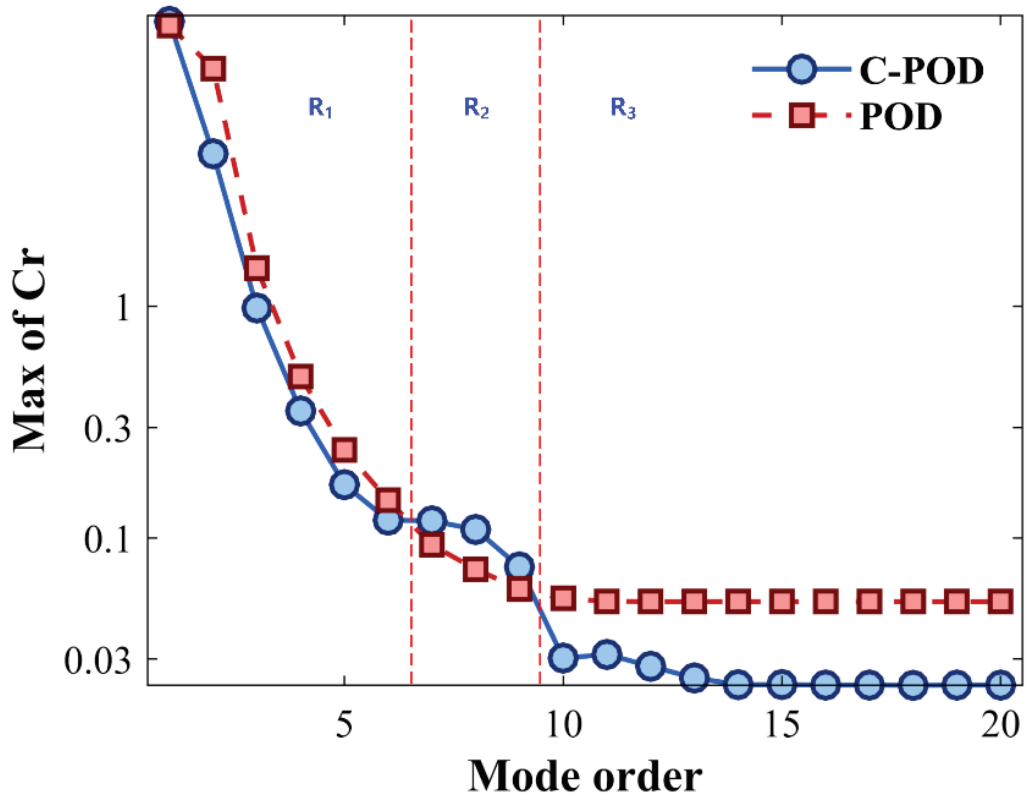


Figure 5. Comparison of maximum Cr values under different modal orders: illustrating the minimum reconstruction accuracy differences between POD and C-POD methods.

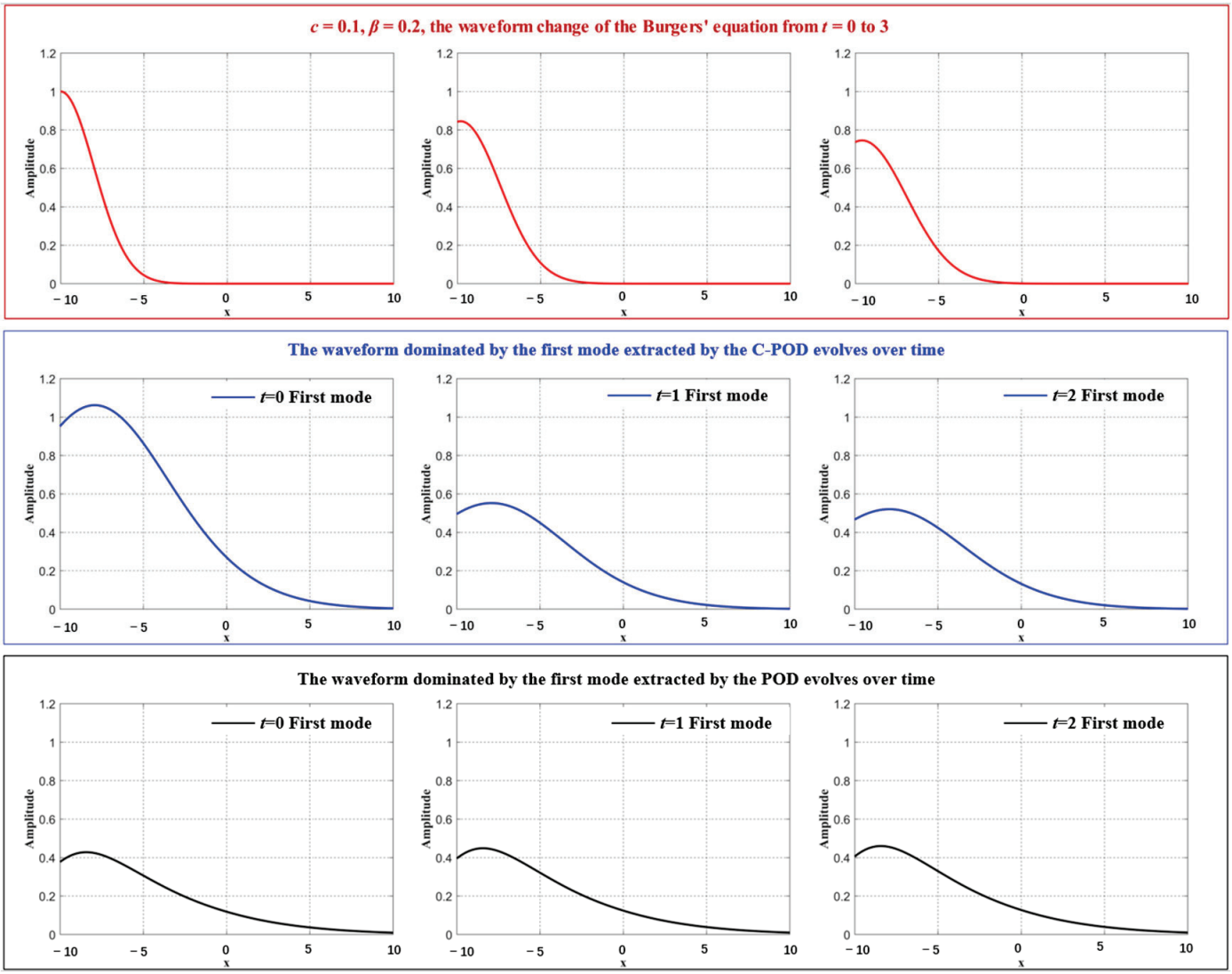


Figure 6. Time evolution of the waveform reconstructed using only the first mode extracted by the C-POD and POD methods, for $c = 0.1$ and $\beta = 0.2$, from $t = 0$ to $t = 2$.

As observed from Figures 6 and 7, the first-mode-reconstructed solutions based on the C-POD method clearly exhibited time-dependent characteristics, including a gradual reduction in peak amplitude and forward propagation along the positive x-axis. In contrast, the first-mode reconstruction using the POD method showed minimal temporal variation, failing to capture the evolving features of the waveform.

Further comparison revealed that, in the case of C-POD, the rate of peak amplitude decay was positively correlated with the parameter β, while the propagation distance along the x-axis increased with wave speed c. These results suggested that, compared to POD, the C-POD method provided more physically meaningful first-mode reconstructions that better reflected the temporal evolution of the system.

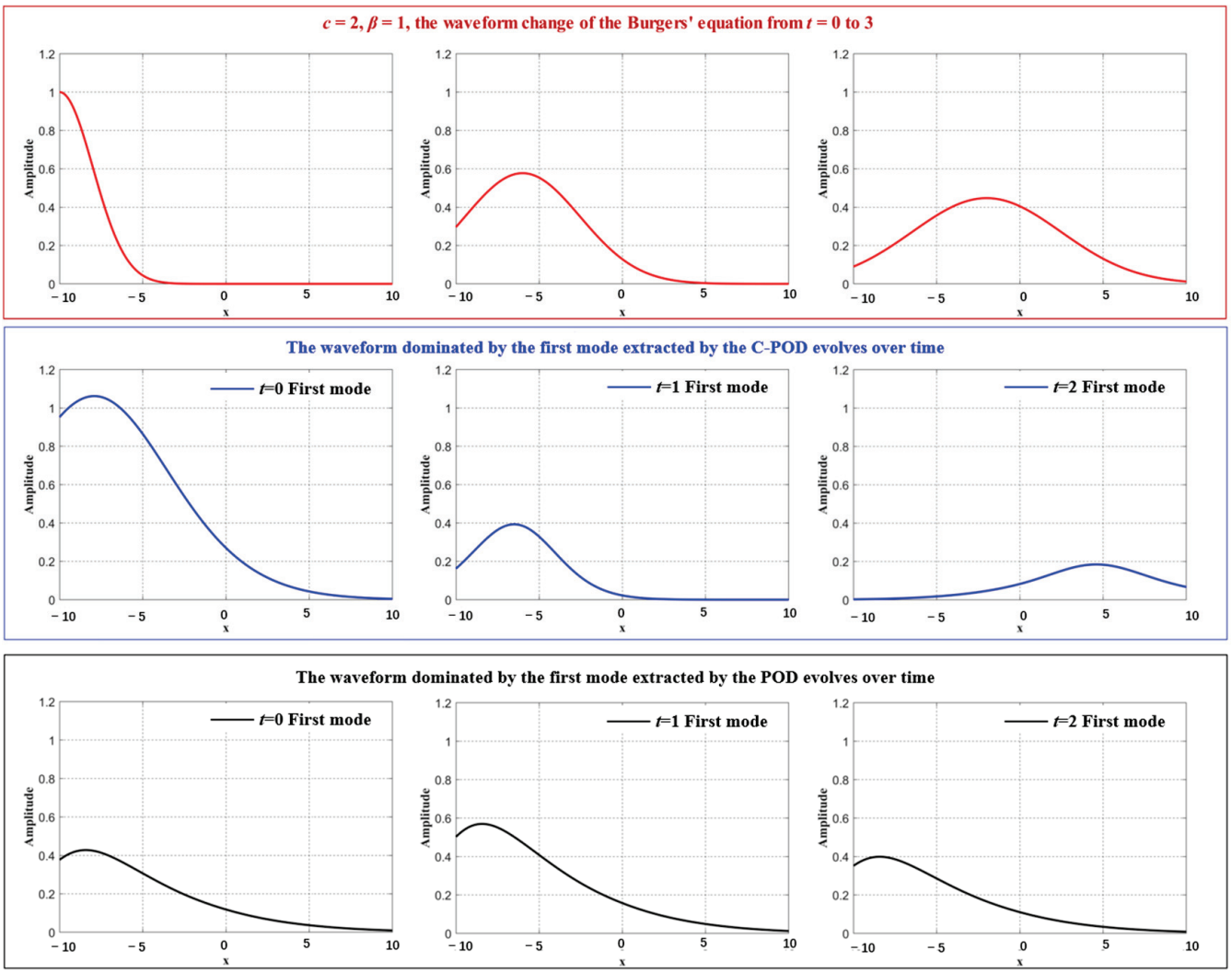


Figure 7. Time evolution of the waveform reconstructed using only the first mode extracted by the C-POD and POD methods, for $c = 2$ and $\beta = 2$, from $t = 0$ to $t = 2$.

4. Cylinder Wake Flow Case Introduction

Two-dimensional flow around a cylinder represents a classical complex flow problem in fluid dynamics, characterized by phenomena such as vortex shedding, flow separation, and turbulence. In contrast to the one-dimensional Burgers' equation, this flow configuration exhibits more intricate turbulent features, revealing a rich spectrum of physical phenomena. Consequently, it serves as an excellent benchmark for validating the efficacy of reduced-order models in capturing complex flow dynamics.

4.1. Model Introduction

To simulate two-dimensional fluid flow around a cylinder, the computational domain was configured as a rectangular region with the cylinder positioned near the mid-left boundary. The cylinder had a radius of 5 mm, and the domain dimensions were $L_x = 1$ mm (length) and $L_y = 40$ mm (width). Inflow velocities at the left boundary were varied to adjust the Reynolds number (Re), thereby capturing a range of flow phenomena. Figure 8 illustrates the geometric configuration of the computational domain.

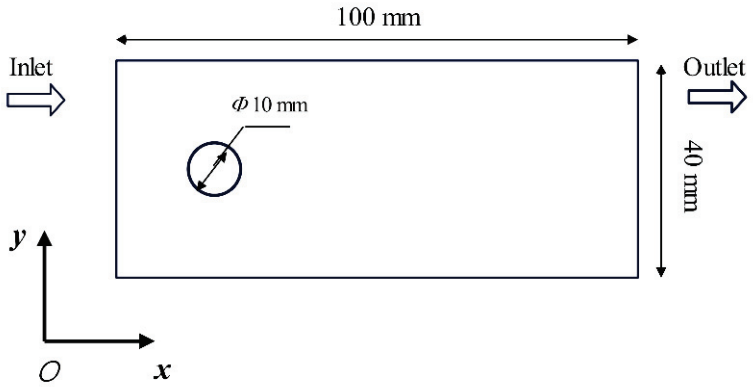


Figure 8. Geometric dimensions of the computational domain for the two-dimensional cylinder wake flow.

For the two-dimensional incompressible Navier–Stokes equations, the motion of the fluid is described by the following equations:

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla \boldsymbol{p} + \nu \nabla^2 \boldsymbol{u} + \boldsymbol{f} \tag{25}$$

$$\nabla \cdot \boldsymbol{u} = 0 \tag{26}$$

where $\boldsymbol{u} = (u, v)$ represents the velocity field components, with u being the velocity in the x direction and v is the velocity in the y direction, $\boldsymbol{p}$ is the pressure field, ν is the fluid's kinematic viscosity, and $\boldsymbol{f}$ represents the body force (in this case, the body force field). For this two-dimensional flow problem, the governing equations were discretized using the finite difference method, and the implicit method combined with the alternating direction implicit (ADI) method was used for time-stepping solutions. The specific solution process is briefly described below:

(1) Using the alternating direction implicit (ADI) method, the equation was split into two one-dimensional equations. The implicit solution was first performed in the x direction, followed by the implicit solution in the y direction:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\nabla \boldsymbol{p} + \nu \nabla^2 u \tag{27}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\nabla \boldsymbol{p} + \nu \nabla^2 v \tag{28}$$

(2) The pressure field was obtained by solving the Poisson equation using the fast Fourier transform (FFT):

$$\nabla^2 \boldsymbol{p} = \frac{\rho}{\Delta t}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right) \tag{29}$$

(3) We set a fixed flow velocity at the left boundary $x = 0$ and applied a no-slip boundary condition on the surface of the cylinder.

(4) To quantify the rotational behavior in the flow field around the cylinder, the calculation of vorticity was introduced in the two-dimensional incompressible flow field. The vorticity is defined as:

$$w = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \tag{30}$$

The vorticity field distribution was obtained through the differential method, as follows:

$$w_{ij} = \frac{v_{i+1,j} - v_{i-1,j}}{2\Delta x} - \frac{u_{i,j+1} - u_{i,j-1}}{2\Delta y} \tag{31}$$

4.2. Grid Independence and Numerical Method Validation

To validate the reliability of the numerical method and the grid used in this study, we first performed a computational model verification for a fixed single cylinder with Re = 100. The effects of the total grid number and dimensionless time step on the results were considered, and the results were compared with the Stanton number and Lift coefficient values obtained from the literature [28–30]. The comparison results are presented in Table 1. The dimensionless time step $\Delta t*$ is defined as shown in Equation (32):

$$\Delta t^* = \frac{\Delta t \cdot U_0}{D_c} \tag{32}$$

where Δt represents the time step, U_0 is the upstream inlet velocity of the cylinder, and D_c is the diameter of the cylinder.

Table 1. Comparison of numerical solutions with literature results for varying total grid numbers and dimensionless time steps.

Operating Condition Name	Total Number of Grids	Δ*t**	CD	St
A1	100 × 40	0.01	0.215	0.16
A2	200 × 80	0.01	0.225	0.164
A3	400 × 160	0.01	0.229	0.165
A4	600 × 240	0.01	0.229	0.165
A5	800 × 320	0.01	0.229	0.165
A6	400 × 160	0.005	0.229	0.165
A7	400 × 160	0.02	0.231	0.167
Experiment from [28]				0.165
CFD from [29]			0.226	0.165
CFD from [30]			0.233	0.166

From Table 1, it can be observed that the simulation results remained consistent as the grid number increased from 400 × 160 to 800 × 320. When the dimensionless time step was less than or equal to 0.01, the results were in close agreement with those in the literature. Therefore, in this study, we chose the A_3 condition, setting the total grid number to 400 × 160 and the dimensionless time step to 0.01.

4.3. Presentation of Computational Results

Figure 9 presents the vorticity distribution corresponding to several typical flow state transitions in the two-dimensional flow around a cylinder at Re = 150. The specific flow states are detailed as follows:

(1) From t = 1 to 9 s, a pair of side vortices formed on either side of the cylinder, gradually creating a pair of vortices with opposite rotational directions behind the cylinder.
(2) From t = 9 to 33 s, the counter-rotating vortices behind the cylinder gradually moved and diffused in the direction of the flow.
(3) From t = 33 to 110 s, the counter-rotating vortices behind the cylinder disappeared, and the flow became more stable.
(4) For times exceeding 110 s, the flow pattern underwent notable changes, with vortices detaching periodically to form a stable vortex street. Multiple regions of alternating

positive and negative vorticity emerged, indicating pronounced shear and recirculation phenomena within the flow field—hallmarks of the classic Kármán vortex street.

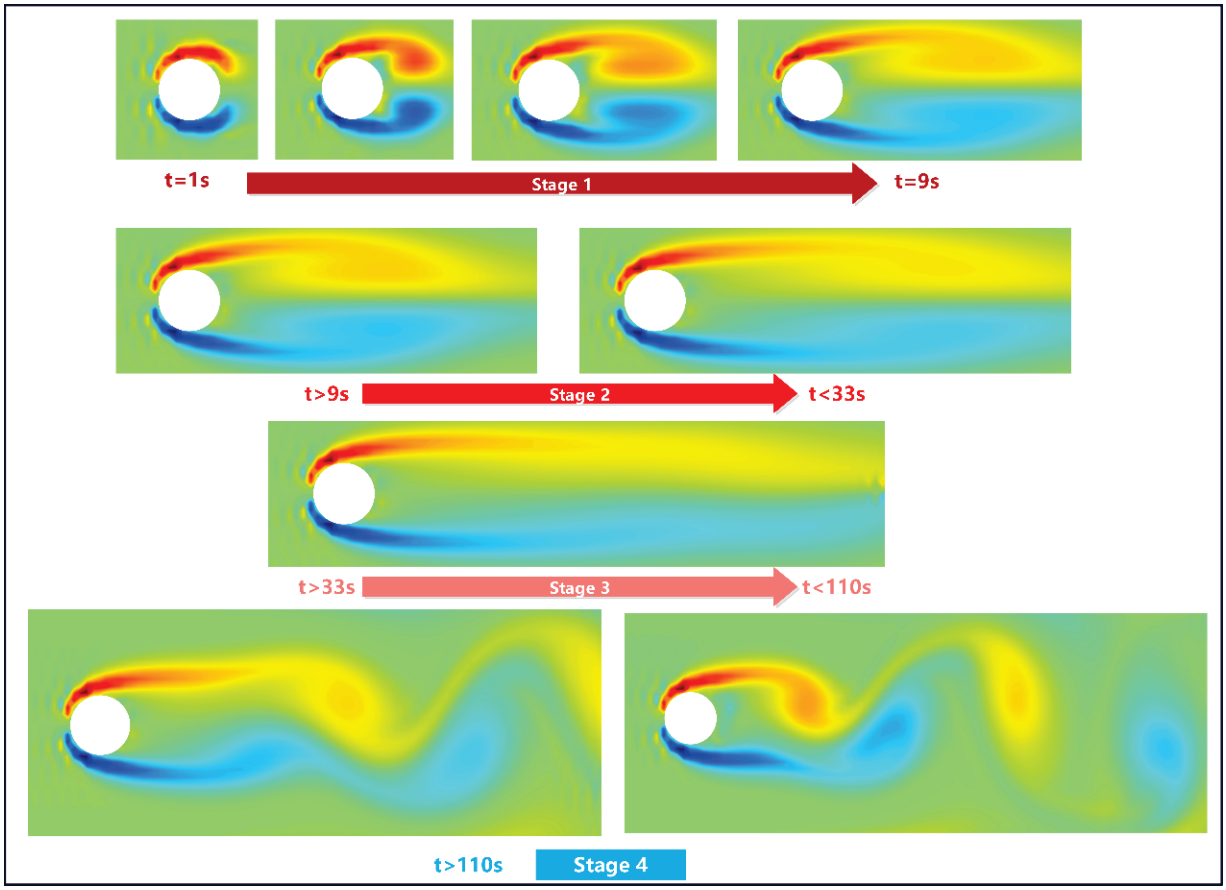


Figure 9. Typical flow patterns in the wake of a cylinder at Re = 150.

4.4. Cylinder Wake Flow Field Dimensionality Reduction and Reconstruction Accuracy, and Modal Decomposition

4.4.1. Dataset and Test Set Construction

To test the accuracy variations of POD and C-POD in the reduced-order modeling and reconstruction of vorticity fields in flow around a cylinder, as well as their modal decomposition capabilities for the flow behavior of the cylinder, a database was constructed over the entire evolution of the cylinder's flow pattern at Re = 150. Specifically, the time range was set from t = 0 to 145 s, with data collected every 0.1 s. The vorticity field distribution data were then written into the dataset, resulting in a total database size of 64,000 × 1450.

4.4.2. Dimensionality Reduction and Reconstruction Accuracy of Cylinder Wake Flow Field

This section assesses the accuracy of the POD and C-POD methods in both the dimensionality reduction and reconstruction of the cylinder wake vorticity field, as well as their modal decomposition capabilities for characterizing the cylinder's wake flow behavior. As illustrated in Figure 10, the average Cr value of C-POD was higher at lower modal orders; however, as the mode number increased, the average Cr value of C-POD decreased rapidly and eventually stabilized at values consistently lower than those observed for POD. At higher modal orders, C-POD maintained a lower average Cr value, indicating improved

reconstruction accuracy across multiple modes relative to POD. Furthermore, as shown in Figure 11, C-POD exhibited a lower maximum Cr value when managing higher modal orders compared to POD, suggesting a distinct advantage in minimizing the worst-case reconstruction error. Overall, C-POD demonstrated superior performance in terms of dimensionality reduction model accuracy—both in the lower bound of capability (maximum Cr) and the overall average accuracy (mean Cr)—thereby confirming its enhanced dimensionality reduction capability over that of POD.

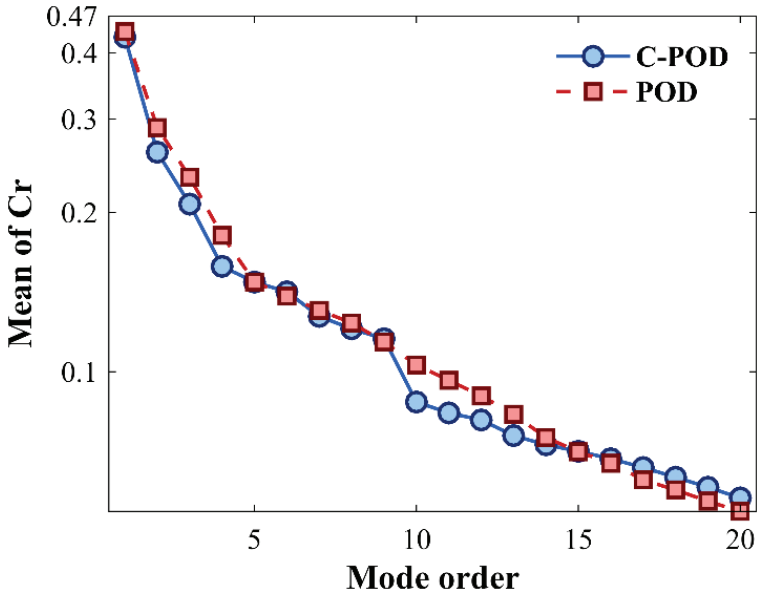


Figure 10. Variation in average dimensionality reduction and reconstruction capability of POD and C-POD methods at different modal orders.

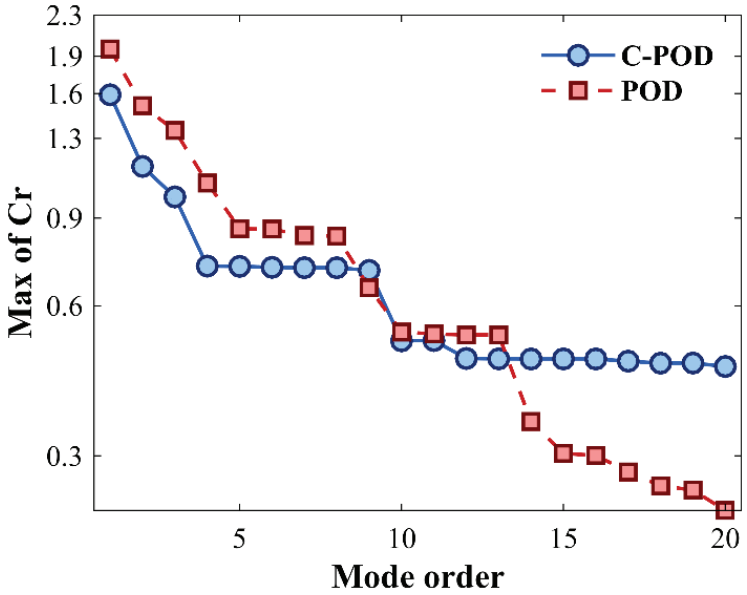


Figure 11. Variation in the lower bound of dimensionality reduction and reconstruction capability of POD and C-POD methods at different modal orders.

4.4.3. Comparison of Modal Decomposition Results

Based on Figure 9, the first ten modes were selected, and the POD and C-POD methods were used to decompose the vorticity field of the cylinder wake at t = 236 s and t = 240 s

under the condition of Re = 150, when the flow transitioned into a stable vortex street. The vorticity field dominated by the first six modes is shown in Figures 12 and 13. From the comparison of Figures 12 and 13, it can be observed that the vorticity field dominated by the first six modes extracted by C-POD mainly contained four types of vortical structures, as shown in Figure 14. These are named as:

(1) Type 1 vortex: Symmetrical side vortices extending downstream from both sides of the cylinder into the flow region.
(2) Type 2 vortex: Counter-rotating pair of vortices formed at the rear of the cylinder due to recirculation, resembling a "horseshoe" shape.
(3) Type 3 vortex: Developing horseshoe-shaped vortices.
(4) Type 4 vortex: Asymmetric side vortices on both sides of the cylinder.

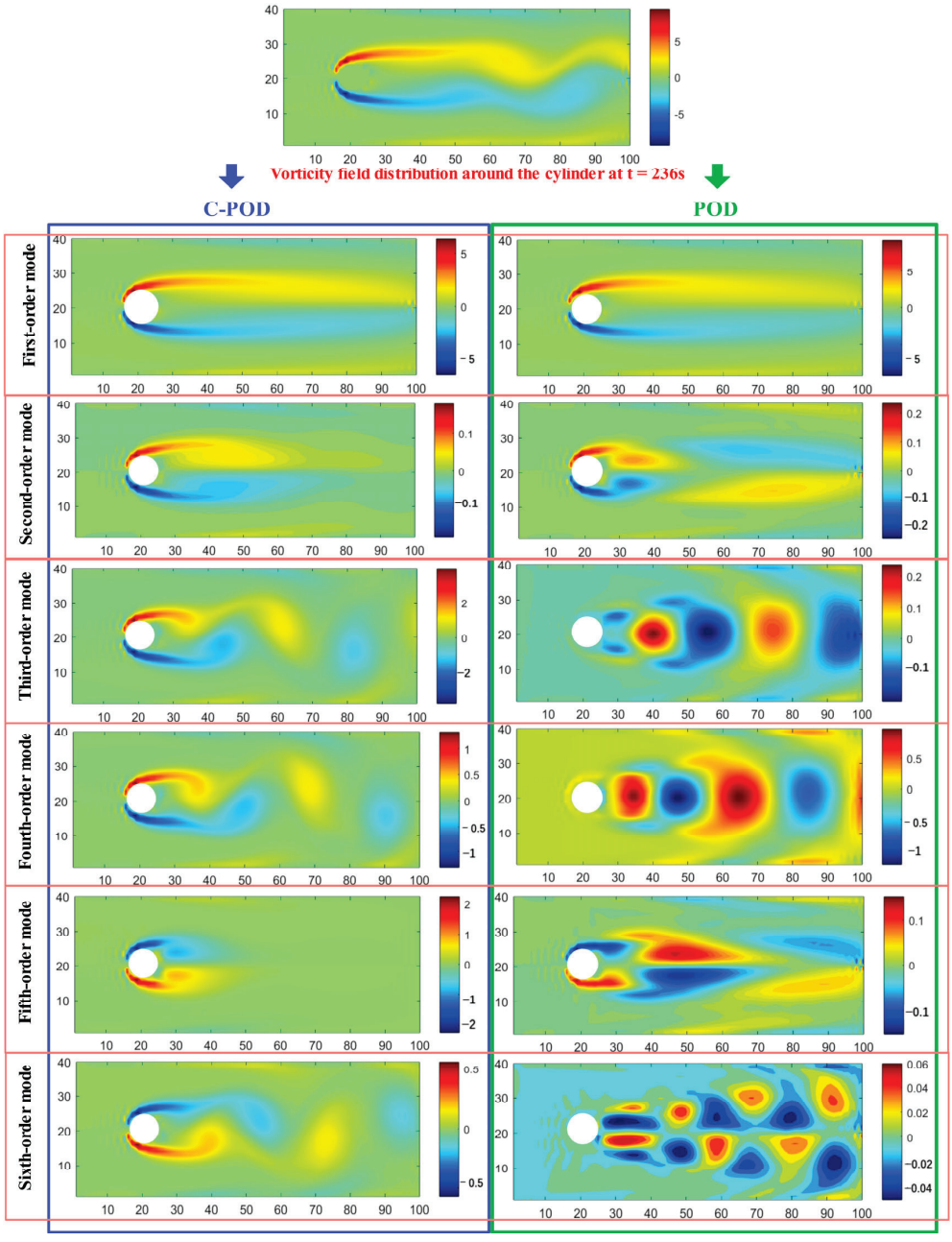


Figure 12. Distribution of the vorticity field dominated by the first six modes extracted by C-POD and POD at t = 236 s.

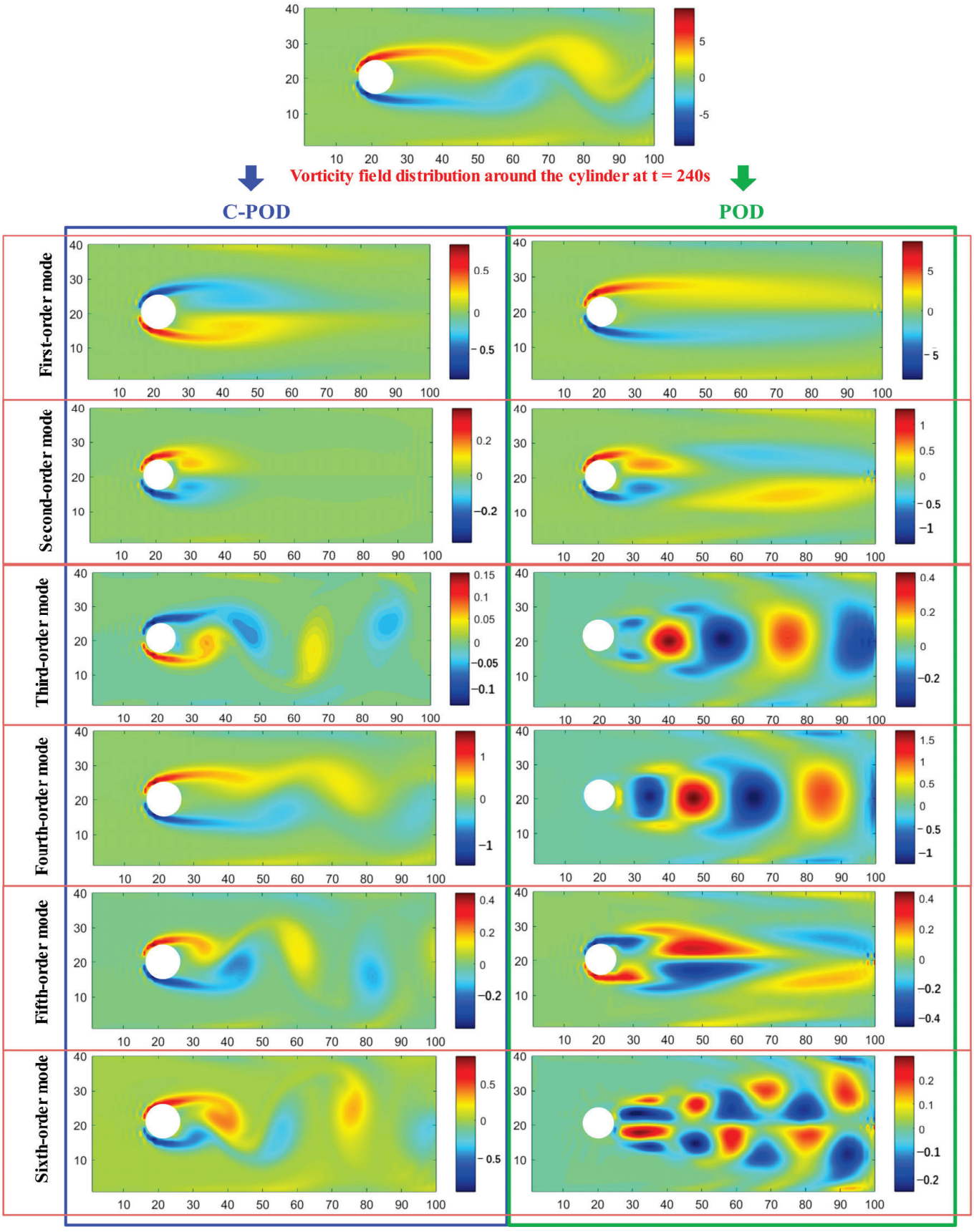


Figure 13. Distribution of the vorticity field dominated by the first six modes extracted by C-POD and POD at t = 240 s.

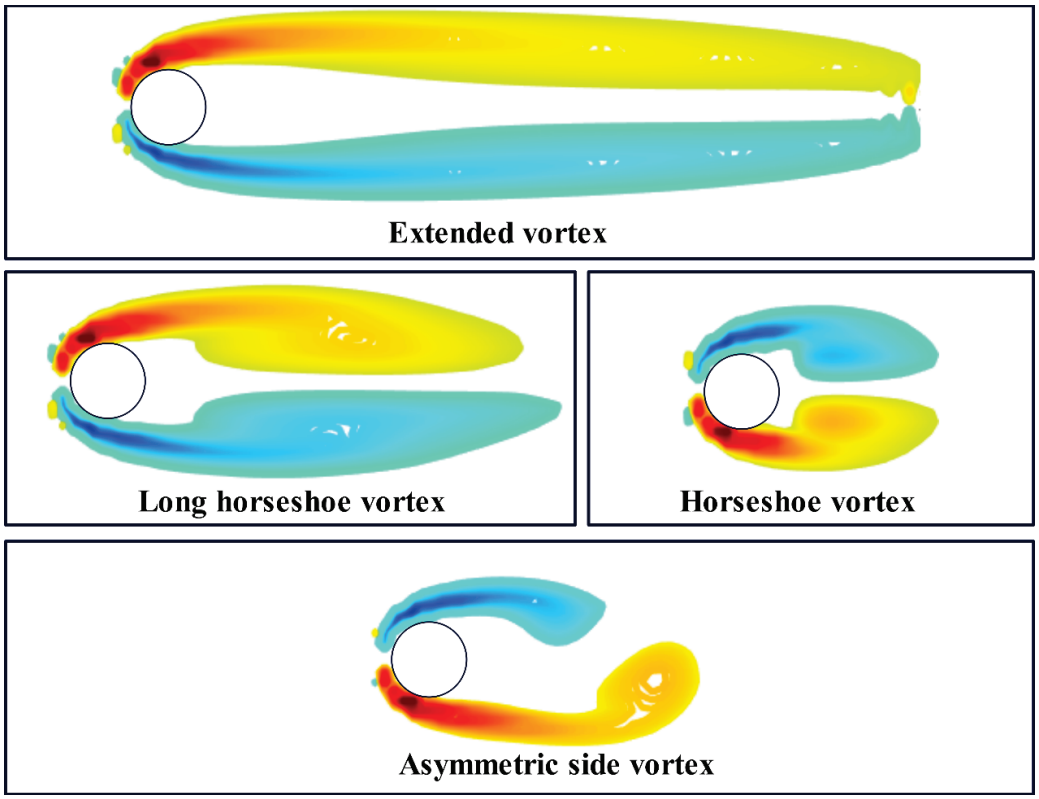


Figure 14. Various vortex patterns in the vorticity field dominated by C-POD modes.

Between t = 236 s and t = 240 s, vortex formation in the vorticity field was characterized by changes in the 1st, 2nd, and 5th modes, with all vortices exhibiting a contraction trend toward the rear of the cylinder. Concurrently, the rotational direction of vortices in the 1st, 3rd, 5th, and 6th modes reversed. Lower-order POD modes corresponded to the primary flow structures and were thus more intuitively interpretable. However, as the modal order increased, modes such as the 2nd and 3rd captured more complex flow characteristics, including primary vortices, secondary vortices, and localized vortices. Despite this, the physical interpretation of vortex evolution—particularly the temporal dynamics of higher-order modes—remains a challenging endeavor.

To further validate the reliability of the ECEPM method in ranking the significance of the basis for C-POD method clustering, the total number of modes was set to 10. The ECEPM method selected the first n modes, reconstructing the vorticity field distributions at t = 236 s and t = 240 s while computing the corresponding Cr values. These results were then compared with those obtained using the POD method, as shown in Figure 15. Figure 15 illustrates that when the modal order was less than 3, the reconstruction accuracy of the C-POD method increased rapidly with additional modes, exhibiting a significantly greater improvement than the POD method. This finding suggests that the first n modes selected by the ECEPM method promptly captured the system's primary features, and the C-POD method demonstrated robust reconstruction capabilities at lower modal orders. However, as the modal order increased further, the reconstruction accuracies of both methods tended to converge.

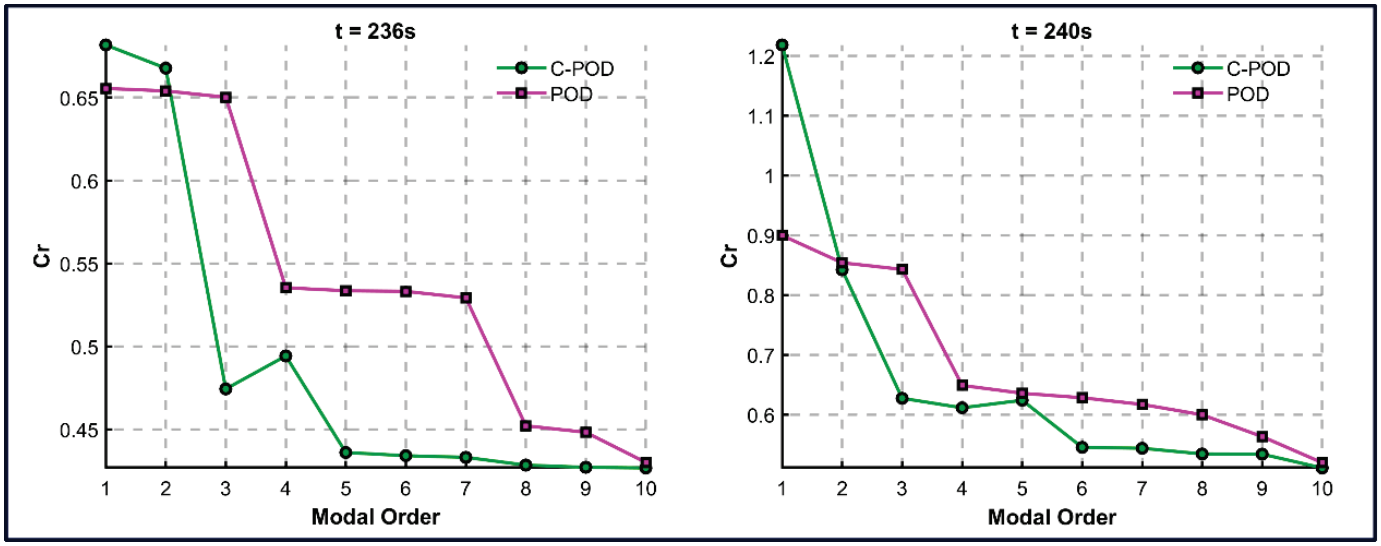


Figure 15. Comparison of reconstruction accuracy for the first n modes extracted by POD and C-POD methods.

4.5. Evolutionary Characteristics of Modes Extracted by C-POD over Time

To further investigate the temporal evolution of modes extracted by C-POD, Figure 16 plots the variation of the first and second mode coefficients over time. The vorticity values in the cloud plot were restricted to the range $[-2, 2]$ to facilitate comparison. It can be observed that the modal coefficients for both the first and second modes exhibited inflection points not only at the primary transition moments of the flow states (i.e., t = 1 s, 9 s, 33 s, and 110 s) but also at secondary fluctuation moments: t = 5 s, 13.5 s, 22 s, and 119.5 s.

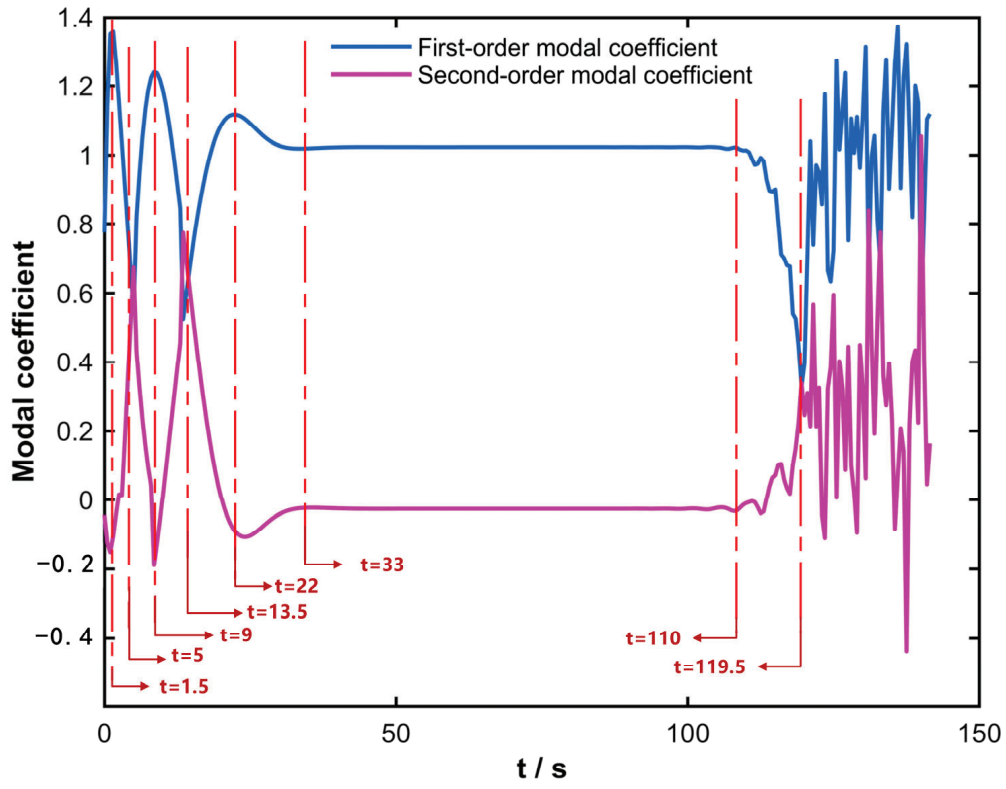


Figure 16. Temporal variation of the first and second modal coefficients extracted by the C-POD method using the ECEPM method.

This indicates that C-POD's time-varying modal coefficients effectively captured both dominant state transitions and subtle flow variations in the cylinder wake. The evolution of vorticity fields dominated by the first and second modes is illustrated at key time points: as shown in Figures 17–19, the time-resolved vorticity structures reflected the dynamic flow features governed by these leading modes.

1. For $t < 9$ s:
    - First mode: Dominated by third-type vortex, which weakened over time.
    - Second mode: Transitioned from irregular fourth-type vortex to first-type vortex, with rotation changes observed at $t = 1$ s, 5 s, and 9 s.
2. For 9 s $< t <$ 14.5 s:
    - First mode: Evolved from third-type to first-type vortex, extending its influence across the field.
    - Second mode: Underwent gradual rotational transition.
3. For 13.5 s $< t <$ 33 s:
    - The vorticity distribution became more stable.
    - Second-mode rotation direction changed.
4. For 33 s $< t <$ 110 s:
    - System entered a quasi-stable regime, with negligible changes in vortex structures and modal vorticity values.
5. For 110 s $< t <$ 119.5 s:
    - Flow transitioned toward vortex street formation.
    - Side vortices began shedding at the tail, and second-mode rotation altered again.
6. For $t >$ 119.5 s:
    - The system reached a state of periodic vortex street generation.
    - Mode coefficients showed periodic oscillation (Figure 15).

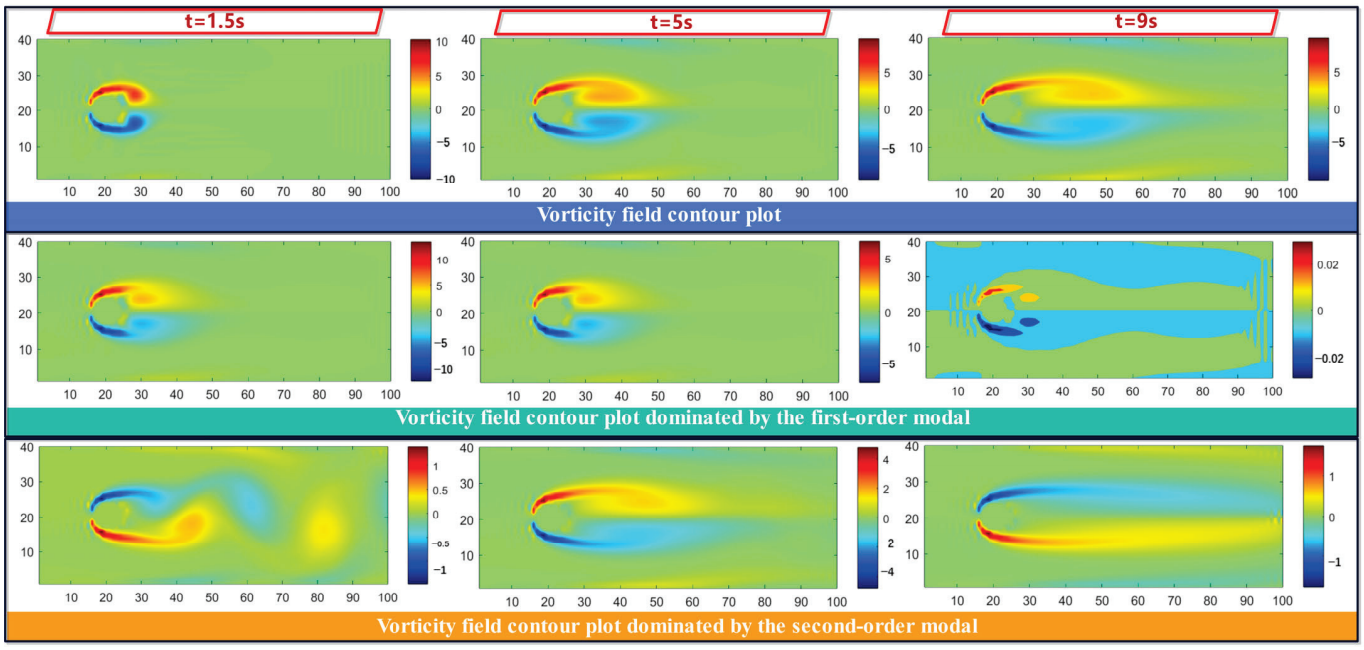


Figure 17. Temporal changes in the vorticity field dominated by the first and second modes of the C-POD method at $t = 1.5$, 5, and 9 s.

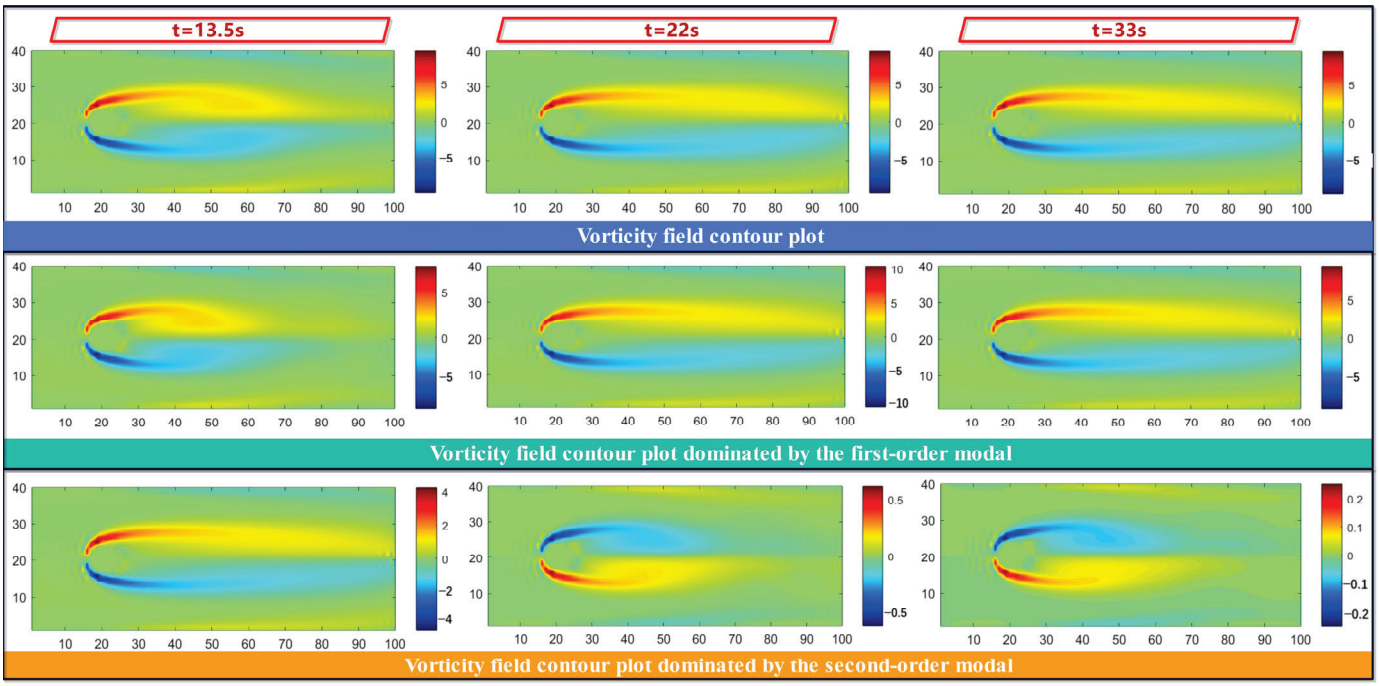


Figure 18. Temporal changes in the vorticity field dominated by the first and second modes of the C-POD method at t = 13.5, 22, and 33 s.

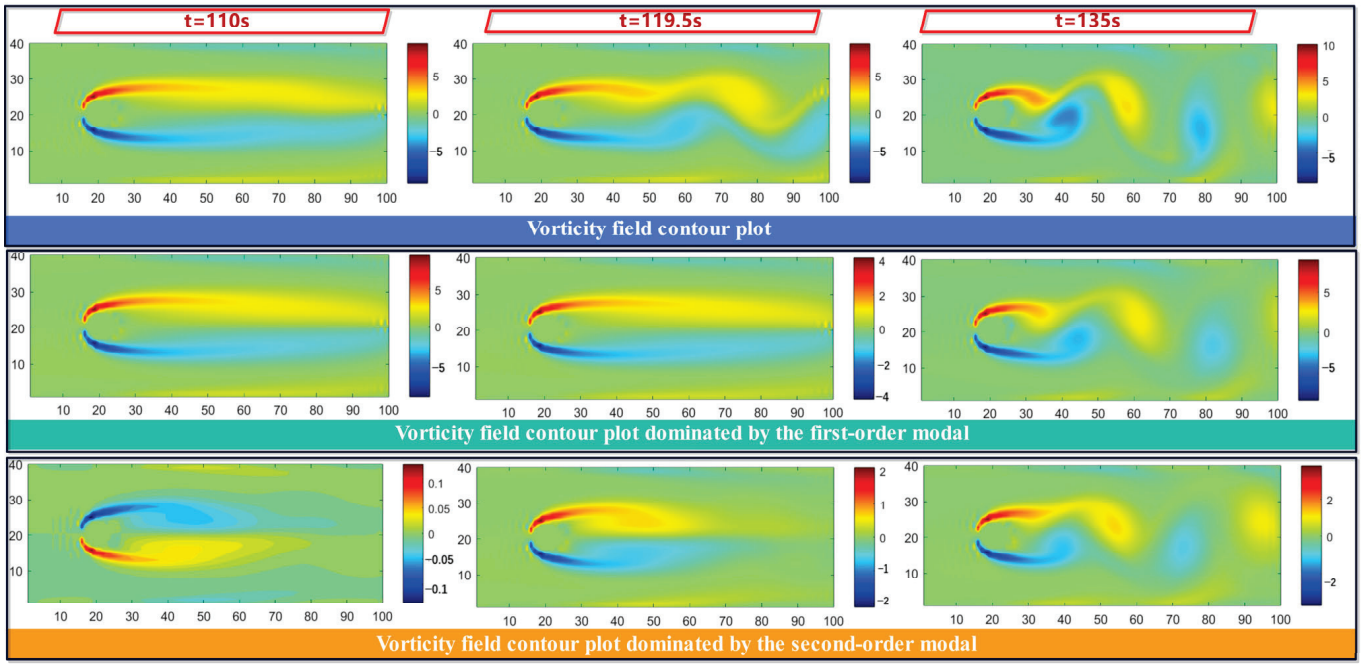


Figure 19. Temporal changes in the vorticity field dominated by the first and second modes of the C-POD method at t = 110, 119.5, and 135 s.

5. Application Value of C-POD ROM in Inverse Problems

The Gappy POD method [31,32] is a derivative of the POD reduced-order model, primarily used to solve the full-field reconstruction inverse problem when only sparse sensor data are available. The accuracy of this method heavily depends on the precision of the reduced-order model under low-order conditions. In both POD and C-POD methods, low-order modes represent the primary dynamics of the system, typically carrying most of

the energy or important information, while high-order modes often represent finer details or noise. For a system, low-order modes are typically associated with the main frequencies and significant physical behaviors, making them more accurate for data reconstruction. Through the analysis in Section 5, we found that, in the cylinder wake problem, the C-POD method had a greater advantage over POD at lower modal orders. This low-order advantage should be better reflected in inverse problems based on reduced-order methods. Continuing with the cylinder wake case described in Section 4.4.3, we describe the performance of this inverse problem in the cylinder wake scenario as follows: placing a specified number of sparse sensors in the vorticity field of the cylinder wake and reconstructing the vorticity field in real time based only on the sparse sensor data, to capture the complete vortex evolution of the cylinder wake.

Here, 80% of the original dataset was randomly selected as the training set, and the remaining 20% was used as the validation set. Using the correlation coefficient filtering method proposed by Qingyang Yuan [33], seven optimal sparse sensor locations were selected in the vorticity field behind the cylinder. Specifically, this method computed the Pearson correlation coefficient between each candidate sensor point and the low-order POD modal coefficients obtained from the training set, then ranked and selected sensor locations with the highest absolute correlations. This ensured that the selected sensors were most sensitive to the dominant dynamic features of the flow field. These sparse sensors can output real-time vorticity information at their respective positions. Based on the sparse sensor data, the missing reconstruction methods based on POD (Gappy POD) and C-POD (Gappy C-POD) were used to reconstruct the cylinder wake flow field under different modal orders, and the reconstruction capabilities were measured, as shown in Figures 20 and 21.

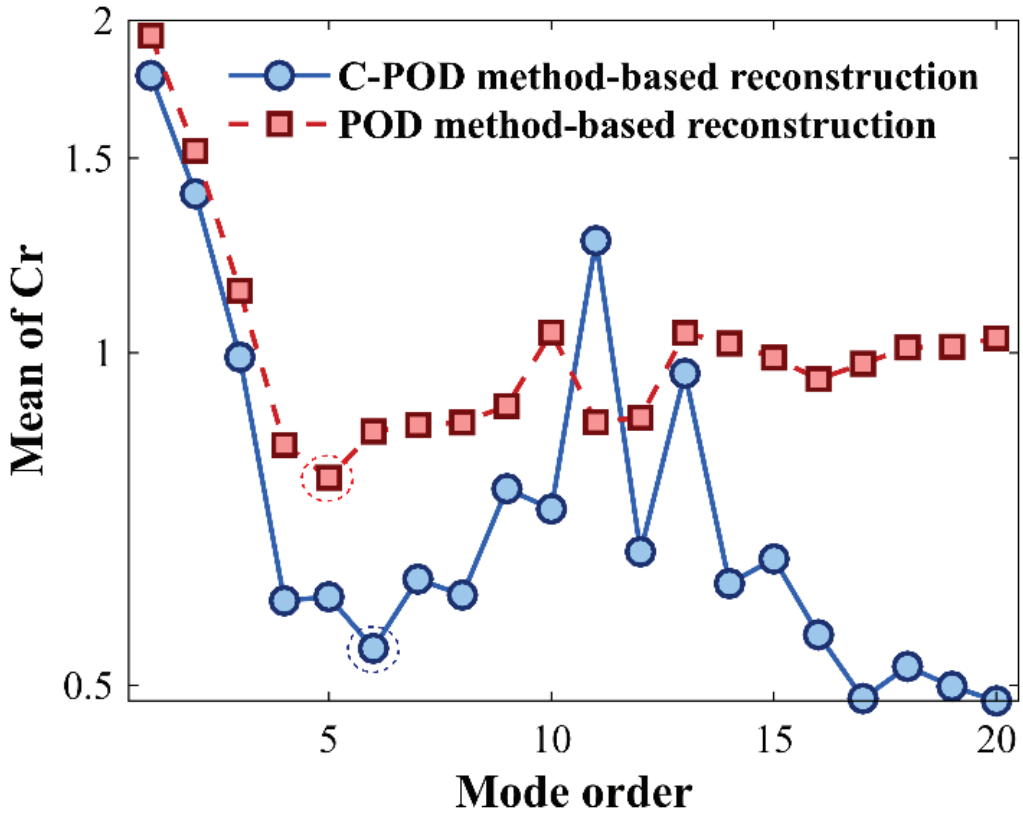


Figure 20. Variation in the average field reconstruction capability of Gappy POD and Gappy C-POD methods at different modal orders.

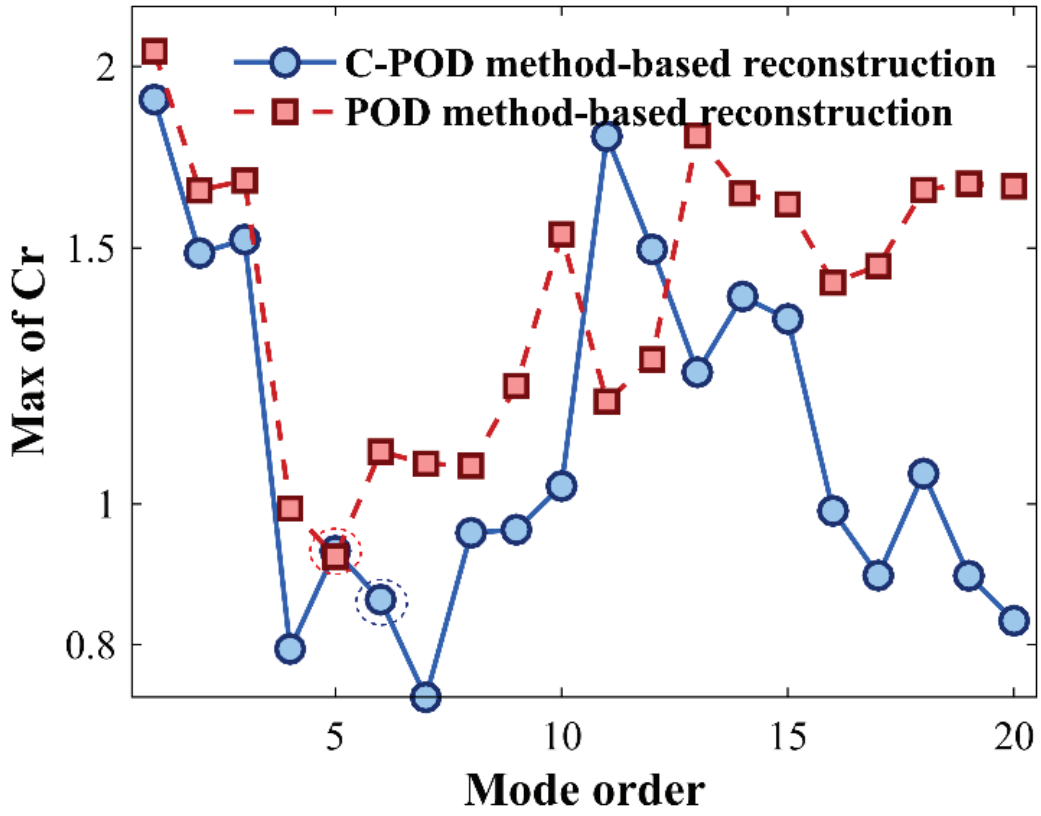


Figure 21. Variation in the lower bound of field reconstruction capability for Gappy POD and Gappy C-POD methods at different modal orders.

From Figures 20 and 21, it is observed that the reconstruction accuracy of the Gappy POD method increased initially and then decreased as the modal order increased. When the modal order was 5, the Gappy POD method achieved its optimal reconstruction accuracy, with an average Cr value of 0.81 and a maximum Cr value of 0.97. When the modal order was less than 12, the reconstruction accuracy of the Gappy C-POD method followed a similar trend, increasing initially and then decreasing. When the modal order was 6, the average Cr value was 0.65 and the maximum Cr value was 0.84. However, when the modal order exceeded 12, the reconstruction accuracy of Gappy C-POD decreased as the modal order increased. When the modal order reached 20, the average Cr value dropped to 0.49, lower than the average value of 0.65 at modal order 6. This demonstrated the significant advantage of the Gappy C-POD method over the Gappy POD method in reconstructing inverse problems. The average reconstruction accuracy of Gappy C-POD was improved by 19.75% compared to Gappy POD, and the lower bound of reconstruction capability was improved by 13.4%. Furthermore, the Gappy C-POD method can achieve higher field reconstruction accuracy under higher modal order conditions, as higher modal orders can capture more complex flow behaviors. Therefore, the Gappy C-POD method is more suitable for field reconstruction in complex nonlinear systems.

Based on the above analysis, a modal order of 5 was chosen for the Gappy POD method and a modal order of 6 for the Gappy C-POD method. The reconstruction of the cylinder wake vorticity field at t = 141 s is shown in Figures 22 and 23. From left to right, the three images in Figures 22 and 23 represent the cylinder wake vorticity field at t = 141 s, the reconstructed field with sparse sensor layout, and the distribution of reconstruction relative error. From Figures 21 and 22, it can be seen that the maximum error in the

field reconstruction using the Gappy C-POD method was 25%, which is smaller than the maximum error of 35% in the field reconstruction using the Gappy POD method.

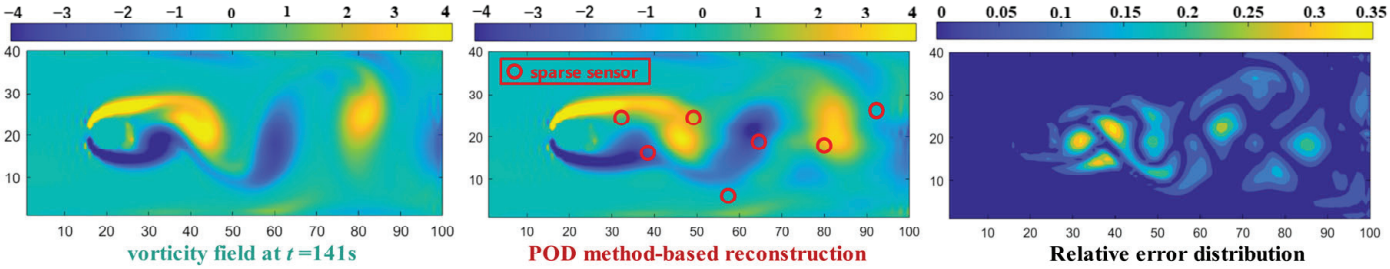


Figure 22. Gappy POD reconstruction of the cylinder wake vorticity field at t = 141 s.

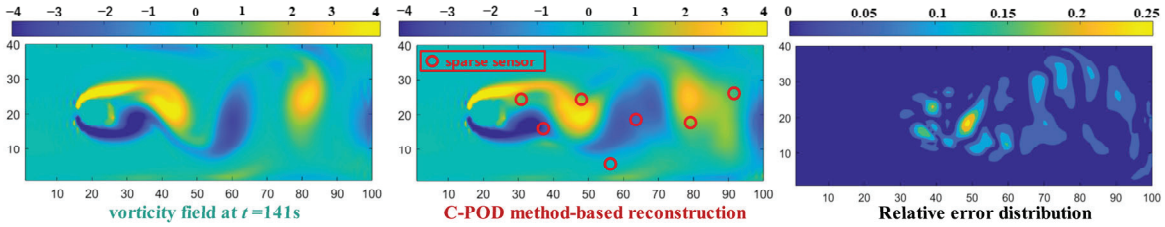


Figure 23. Gappy C-POD reconstruction of the cylinder wake vorticity field at t = 141 s.

Limitations and Future Improvements

While the proposed C-POD ROM achieved improved reconstruction accuracy and enhanced interpretability compared to traditional POD-based methods, certain limitations remain, particularly in practical applications.

- Computational Cost

The integration of clustering and entropy-based ranking introduces additional computational burden. When dealing with large-scale datasets or increasing the number of clusters, both memory usage and computation time grow considerably.

- Sensitivity to Sensor Layout in Inverse Reconstruction

For tasks involving sparse sensor reconstruction, the spatial placement of sensors significantly affects the performance. If the sensors fail to cover dynamically important regions, the model may suffer from decreased reconstruction fidelity. This highlights the necessity of coupling reduced-order modeling with sensor placement optimization.

- Hyperparameter Dependency

The performance of clustering and entropy-controlled ranking is sensitive to several hyperparameters, such as the number of clusters and entropy scaling coefficients. Inappropriate parameter settings can degrade the model's effectiveness or lead to unstable results.

To address these limitations, future work may focus on:

(1) Developing lightweight clustering algorithms or incremental learning schemes to reduce computational overhead.
(2) Exploring adaptive sensor deployment strategies.
(3) Investigating robust and data-driven hyperparameter tuning frameworks.

These improvements will enhance the scalability and robustness of C-POD ROM in more complex and data-limited fluid dynamics scenarios.

6. Conclusions

This work introduced a clustering-based dimensionality reduction method guided by POD structures (C-POD), aiming to address the instability of clustering center initialization and the lack of effective mode ranking in conventional clustering-based approaches. By integrating the orthogonality and interpretability of POD with the flexibility of unsupervised clustering, and by incorporating an entropy-controlled Euclidean-to-probability mapping (ECEPM) for robust mode evaluation, the proposed method enhanced both the stability and transparency of dimensionality reduction.

The C-POD method was validated through applications to convective flow problems, including the one-dimensional Burgers' equation and the two-dimensional cylinder wake flow. The results demonstrated that C-POD effectively identified dominant flow structures and preserved essential dynamics under reduced-dimensional settings. Moreover, it showed strong resilience in sparse reconstruction tasks, delivering better performance when only limited sensor data or a small number of modes were available.

Beyond improved numerical accuracy, the method provided meaningful insights into flow physics, making it well-suited for real-time prediction, control, and inverse analysis of complex unsteady systems.

Future work will extend this framework to three-dimensional turbulent convective flows and explore adaptive clustering mechanisms that dynamically optimize cluster configurations, further enhancing model scalability and generalization.

Author Contributions: Conceptualization, Q.Y. and B.Z.; methodology, Q.Y.; validation, Q.Y. and B.Z.; formal analysis, Q.Y.; investigation, Q.Y.; data curation, Q.Y.; writing—original draft preparation, Q.Y.; writing—review and editing, B.Z.; visualization, Q.Y.; supervision, B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

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

References

1. Guo, T.; Wu, W.-L.; Luo, Z.-M.; Zhang, Y.-N. Mode Transition and Drag Characteristics of Non-Circular Cylinders in a Uniform Flow. *Ocean Eng.* **2024**, *297*, 117025. [CrossRef]
2. Santana, G.M.; Fabro, A.T.; Miserda, R.F.B. Analysis of the Dynamic Modes of the Transonic Flow around a Cylinder. *J. Braz. Soc. Mech. Sci. Eng.* **2024**, *46*, 414. [CrossRef]
3. Copeland, D.M.; Cheung, S.W.; Huynh, K.; Choi, Y. Reduced Order Models for Lagrangian Hydrodynamics. *Comput. Methods Appl. Mech. Eng.* **2022**, *388*, 114259. [CrossRef]
4. Knockaert, L.; De Zutter, D. Stable Laguerre-SVD Reduced-Order Modeling. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **2003**, *50*, 576–579. [CrossRef]
5. Couplet, M.; Basdevant, C.; Sagaut, P. Calibrated Reduced-Order POD-Galerkin System for Fluid Flow Modelling. *J. Comput. Phys.* **2005**, *207*, 192–220. [CrossRef]
6. Besselink, B.; van de Wouw, N.; Scherpen, J.M.A.; Nijmeijer, H. Model Reduction for Nonlinear Systems by Incremental Balanced Truncation. *IEEE Trans. Autom. Control* **2014**, *59*, 2739–2753. [CrossRef]
7. Grolet, A.; Thouverez, F. On a New Harmonic Selection Technique for Harmonic Balance Method. *Mech. Syst. Signal Process.* **2012**, *30*, 43–60. [CrossRef]
8. Bagheri, M.H.; Esmailpour, K.; Hoseinalipour, S.M.; Mujumdar, A.S. Numerical Study and POD Snapshot Analysis of Flow Characteristics for Pulsating Turbulent Opposing Jets. *Int. J. Numer. Methods Heat Fluid Flow* **2019**, *29*, 2009–2031. [CrossRef]
9. Bistrian, D.A.; Navon, I.M. Randomized Dynamic Mode Decomposition for Nonintrusive Reduced Order Modelling. *Int. J. Numer. Methods Eng.* **2017**, *112*, 3–25. [CrossRef]
10. Phillips, T.R.F.; Heaney, C.E.; Smith, P.N.; Pain, C.C. An Autoencoder-Based Reduced-Order Model for Eigenvalue Problems with Application to Neutron Diffusion. *Int. J. Numer. Methods Eng.* **2021**, *122*, 3780–3811. [CrossRef]

11. Zhang, X.; Ji, T.; Xie, F.; Zheng, C.; Zheng, Y. Data-Driven Nonlinear Reduced-Order Modeling of Unsteady Fluid–Structure Interactions. *Phys. Fluids* **2022**, *34*, 053608. [CrossRef]
12. Shen, X.; Du, C.; Jiang, S.; Sun, L.; Chen, L. Enhancing Deep Neural Networks for Multivariate Uncertainty Analysis of Cracked Structures by POD-RBF. *Theor. Appl. Fract. Mech.* **2023**, *125*, 103925. [CrossRef]
13. Yuan, L.; Ni, Y.-Q.; Deng, X.-Y.; Hao, S. A-PINN: Auxiliary Physics Informed Neural Networks for Forward and Inverse Problems of Nonlinear Integro-Differential Equations. *J. Comput. Phys.* **2022**, *462*, 111260. [CrossRef]
14. Taassob, A.; Kumar, A.; Gitushi, K.M.; Ranade, R.; Echekki, T. A PINN-DeepONet Framework for Extracting Turbulent Combustion Closure from Multiscalar Measurements. *Comput. Methods Appl. Mech. Eng.* **2024**, *429*, 117163. [CrossRef]
15. Colanera, A.; Schmidt, O.T.; Chiatto, M. Robust Spectral Proper Orthogonal Decomposition. *Comput. Phys. Commun.* **2025**, *307*, 110056. [CrossRef]
16. Gu, X.; Xu, C.; Liu, M.; Mao, Y. Frequency-Domain Proper Orthogonal Decomposition for Efficient Reconstruction of Unsteady Flows. *Phys. Fluids* **2025**, *37*, 025109. [CrossRef]
17. Bui, D.; Hamdaoui, M.; De Vuyst, F. POD-ISAT: An Efficient POD-Based Surrogate Approach with Adaptive Tabulation and Fidelity Regions for Parametrized Steady-State PDE Discrete Solutions. *Int. J. Numer. Methods Eng.* **2013**, *94*, 648–671. [CrossRef]
18. Burkardt, J.; Gunzburger, M.; Lee, H.-C. Centroidal Voronoi Tessellation-Based Reduced-Order Modeling of Complex Systems. *SIAM J. Sci. Comput.* **2006**, *28*, 459–484. [CrossRef]
19. Iqbal, M.M.; Xavier, R.J. Development of Optimal Reduced-Order Model for Gas Turbine Power Plants Using Particle Swarm Optimization Technique. *Int. Trans. Electr. Energy Syst.* **2020**, *30*, e12278. [CrossRef]
20. Huera-Huarte, F.J.; Vernet, A. Vortex Modes in the Wake of an Oscillating Long Flexible Cylinder Combining POD and Fuzzy Clustering. *Exp. Fluids* **2010**, *48*, 999–1013. [CrossRef]
21. D'aDamo, J.; Collaud, M.; Sosa, R.; Godoy-Diana, R. Wake and Aeroelasticity of a Flexible Pitching Foil. *Bioinspiration Biomim.* **2022**, *17*, 045002. [CrossRef]
22. Wei, Z.; Yang, Z.; Xia, C.; Li, Q. Cluster-Based Reduced-Order Modelling of the Wake Stabilization Mechanism behind a Twisted Cylinder. *J. Wind. Eng. Ind. Aerodyn.* **2017**, *171*, 288–303. [CrossRef]
23. Sirovich, L.; Kirby, M. Turbulence and the Dynamics of Coherent Structures. Part1: Coherent Structures. *Q. Appl. Math.* **1987**, *45*, 561–571. [CrossRef]
24. Guo, H.B.; Renaut, R.A. A Regularized Total Least Squares Algorithm. In *Total Least Squares and Errors-in-Variables Modeling: Analysis, Algorithms and Applications*; Van Huffel, S., Lemmerling, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2002; pp. 57–66.
25. Yuan, S.; Blömker, D.; Duan, J. Stochastic Turbulence for Burgers Equation Driven by Cylindrical Lévy Process. *Stoch. Dyn.* **2022**, *22*, 2240004. [CrossRef]
26. Li, L.; Ong, K.W. Dynamic Transitions of Generalized Burgers Equation. *J. Math. Fluid Mech.* **2016**, *18*, 89–102. [CrossRef]
27. Senechal, M. Spatial Tessellations: Concepts and Applications of Voronoi Diagrams. *Science* **1993**, *260*, 1170. [CrossRef]
28. Williamson, C.H.K. Oblique and Parallel Modes of Vortex Shedding in the Wake of a Circular Cylinder at Low Reynolds Numbers. *J. Fluid Mech.* **1989**, *206*, 579–627. [CrossRef]
29. Singha, S.; Sinhamahapatra, K.P. High-Resolution Numerical Simulation of Low Reynolds Number Incompressible Flow About Two Cylinders in Tandem. *J. Fluids Eng.* **2010**, *132*, 011101. [CrossRef]
30. Qu, L.; Norberg, C.; Davidson, L.; Peng, S.-H.; Wang, F. Quantitative Numerical Analysis of Flow Past a Circular Cylinder at Reynolds Number Between 50 and 200. *J. Fluids Struct.* **2013**, *39*, 347–370. [CrossRef]
31. Everson, R.; Sirovich, L. Karhunen–Loève Procedure for Gappy Data. *J. Opt. Soc. Am. A* **1995**, *12*, 1657–1664. [CrossRef]
32. Deus, J.; Martin, E. Efficient Cardiovascular Parameters Estimation for Fluid-Structure Simulations Using Gappy Proper Orthogonal Decomposition. *Ann. Biomed. Eng.* **2024**, *52*, 3037–3052. [CrossRef] [PubMed]
33. Yuan, Q.; Guo, X.; Han, J.; Han, B.; Zhang, B.; Lan, T.; Ariyur, K. Correlation Coefficient Filtering Method for Optimal Sensor Layout Strategies in Inverse Problems. *J. Eng.* **2025**, *2025*, 5142118. [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.

algorithms

MDPI

Article

Modeling Local Search Metaheuristics Using Markov Decision Processes

Rubén Ruiz-Torrubiano *,†, Deepak Dhungana *,†, Sarita Paudel † and Himanshu Buckchash †

IMC University of Applied Sciences Krems, 3500 Krems, Austria; sarita.paudel@imc.ac.at (S.P.); himanshu.buckchash@imc.ac.at (H.B.)

* Correspondence: ruben.ruiz@imc.ac.at (R.R.-T.); deepak.dhungana@imc.ac.at (D.D.)

† These authors contributed equally to this work.

Abstract

Local search metaheuristics like tabu search or simulated annealing are popular heuristic optimization algorithms for finding near-optimal solutions for combinatorial optimization problems. However, it is still challenging for researchers and practitioners to analyze their behavior and systematically choose one over a vast set of possible metaheuristics for the particular problem at hand. In this paper, we introduce a theoretical framework based on Markov Decision Processes (MDPs) for analyzing local search metaheuristics. This framework not only helps in providing convergence results for individual algorithms but also provides an explicit characterization of the exploration–exploitation tradeoff and a theory-grounded guidance for practitioners for choosing an appropriate metaheuristic for the problem at hand. We present this framework in detail and show how to apply it in the case of hill climbing and the simulated annealing algorithm, including computational experiments.

Keywords: metaheuristics; combinatorial optimization; Markov Decision Process; stochastic control problems

1. Introduction

Metaheuristics are high-level algorithmic frameworks that provide guidance or strategies for developing heuristic optimization algorithms [1]. Frequently, these strategies use concepts from other disciplines as their underlying inspiration, like the evolution process in biology, the process of cooling a metallic alloy, the cooperative behavior of flocks of birds, and many more [2]. Despite being heuristic in nature, so that in general there is no general theoretical guarantee that the algorithm will find the global optimum, metaheuristics have demonstrated a tremendous success in practice, being effectively used in applications like logistics (e.g., vehicle routing problem [3]), network design [4], scheduling [5–8], finance (e.g., portfolio optimization [9]) and many others.

In general, it can be distinguished between local search (or single-solution) metaheuristics like simulated annealing (SA) [10], tabu search (TS) [11], variable neighborhood search (VNS) [12] and large neighborhood search (LNS) [13] on the one hand, and population-based metaheuristics on the other, like genetic algorithms (GAs) [14] particle swarm optimization (PSO) [15] or estimation of distribution algorithms (EDAs) [16]. Local search metaheuristics typically start with a candidate solution x_0 (not necessarily feasible) and apply stochastic operators to select the next candidate solution from a suitable chosen

neighborhood or pool of individuals. By iteratively repeating this process, local search metaheuristics are able to find near-optimal solutions to the problem at hand efficiently, even if the search space is very large. By contrast, population-based metaheuristics start with a group of candidate solutions (often referred to as a population) $P = \{x_1, x_2, \ldots, x_n\}$ and apply different types of operators to generate new candidate solutions, updating the population in the process.

It is well acknowledged that there is a lack of strong theoretical frameworks for understanding and analyzing the behavior of metaheuristic algorithms [17,18]. While theoretical results and convergence analysis do exist for individual algorithms like genetic algorithms and simulated annealing [19,20], existing frameworks are either focused on one type of analysis (like convergence analysis [17]), or they provide only weak results [18]. Moreover, they provide no theory-grounded guidance for practitioners on which algorithm might be most appropriate for a particular task.

In this paper, we fill this gap by proposing a novel framework based on Markov Decision Processes (MDPs) as the underlying model for metaheuristic algorithms. Specifically, in this study we focus on local search metaheuristics, leaving population-based metaheuristics for future work. We show that particular instances of local search metaheuristics can be modeled as policies of a suitably defined MDP, and based on that, we derive different coefficients to model their convergence and exploration–exploitation behavior. By introducing different parameters into these bounds and assigning them a practical meaning, we show which properties have to be taken into account for a given metaheuristic (policy) to perform well on a particular class of problems.

This paper is structured as follows. In Section 2, we review previous relevant work. Section 3 presents the MDP framework for local search metaheuristics and applies it to the hill climbing algorithm as a first example. In Section 4, we apply this framework to simulated annealing, showing how the relevant measures of convergence and exploration–exploitation can be explicitly calculated. We present some computational experiments in Section 5 that provide empirical evidence for our theoretical results. Section 6 concludes the paper and outlines future work.

2. Previous Work

Mathematical analysis of optimization metaheuristics has been done previously in the literature in a variety of ways. The work on the schema theorem by Holland [21] can be considered one of the first attempts at mathematical formalization of metaheuristic algorithms. However, this result applied only to simple GAs with one-point crossover and binary encoding. One of the most popular variants of mathematical analysis was spearheaded by the Markov chain analysis of genetic algorithms (GAs), which was mostly done in the 1990s. For instance, in early work, the authors of [22] proved that GAs with an elitist selection scheme converge to the optimum with probability 1. Suzuki [19,20] proved theoretical results based on Markov chain analysis for GAs, focusing on mutation probabilities, and developed an early comparison with simulated annealing (SA). In [23], the authors applied a non-stationary Markov model for the convergence analysis of genetic algorithms. One of the main drawbacks of using Markov chain theory for metaheuristic analysis is that Markov chains are not expressive enough for handling important aspects of the search process in metaheuristics. Critically, the reward structure and the different operators used in the algorithm can only be expressed implicitly via the transition matrix.

More recent approaches in the literature use other models, like the evolutionary Turing machine [24] and the nested partition algorithm [18]. However, most of these results focus on convergence analysis and do not handle the exploration–exploitation dilemma

explicitly. This question was addressed in [25] by experimental means, but no theoretical analysis was provided. Similarly, a case study based on elitist evolutionary algorithms (concretely, random univariate search and evolutionary programming) was presented in [26] using a specially-tailored probabilistic framework. An extensive survey on the exploration–exploitation tradeoff in metaheuristics is given in [27]. In [28], a mathematical characterization of the exploration–exploitation tradeoff is given in the form of the optimal contraction theorem. However, their definition of exploration leaves out the possibility that information collected during the search can be used for exploring other areas of the search space.

In contrast to previous approaches, our framework applies Markov decision processes (MDPs) to the problem of modeling the convergence behavior and the exploration–exploitation tradeoff of local search metaheuristics. We consider MDP to be a more suitable framework than the previously presented approaches since the reward structure of the general search problem can be represented explicitly. Moreover, transformations and sequences of candidate solutions can be modeled explicitly by means of actions, which in MDP are naturally linked to rewards. We can therefore model a particular metaheuristic algorithm as a stochastic agent that makes decisions about which candidate solution to visit next based on the information obtained so far and the reward obtained. We explore the implications of this modeling choice and we show in the following sections that the convergence and the exploration–exploitation analysis of metaheuristic algorithms can be calculated explicitly.

3. Markov Decision Processes for Local Search Metaheuristics

In this section, we develop the mathematical machinery needed to define an MDP model for local search metaheuristics. Our main insight is that we can model any local search metaheuristic as a stochastic agent that executes some policy on a suitably defined MDP. Critically, we note that even if the goal of a specific metaheuristic is to find the optimum of an objective function, the algorithm itself proceeds by maximizing the total reward over a (possibly) infinite time horizon. Clearly, the rewards are directly connected with the best values found so far of the objective function, but the stochastic agent needs to proceed by gathering information about promising directions in the search space (exploration–exploitation tradeoff). Therefore, the exploration of the search space is guided mainly by the maximization of the total reward.

3.1. Definitions

We start by introducing some preliminaries in the form of notation and definitions. Then we specify the model in detail and describe how to define metaheuristic policies using this framework. In the following, we will mainly follow the conventions and definitions used in [29].

Definition 1 (Markov Decision Process)**.** *A Markov Decision Process (MDP) is a tuple* $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \alpha \rangle$*, where:*

- $\mathcal{S}$ *is a set of states.*
- $\mathcal{A}(i)$ *is a set of actions available in state* $i \in \mathcal{S}$*.*
- $\mathcal{P}(a) = \{p_{ij}(a)\}_{ij}$ *is the state transition matrix, given action* a*, where* $p_{ij}(a) = \mathbb{P}\{S_{t+1} = j | S_t = i, A_t = a\}$*.*
- $\mathcal{R}(a)$ *is a reward function,* $\mathcal{R}_s(a) = \mathbb{E}[r_{t+1} | S_i = s, A_t = a]$*.*
- $\alpha \in [0, 1]$ *is a discount factor.*

In words, an MDP is a stochastic decision model where an agent makes decisions on each time step about which action a to take from a predefined action set $\mathcal{A}(i)$, which might be different for each state $i \in \mathcal{A}(i)$. Most importantly, the transition probabilities to a new state S_{t+1} depend only on the last state S_t and not on the previous history (i.e., the previously visited states). This is known as the *Markov property*.

Definition 2. *A* policy *R is a decision rule $R = (\pi^1, \pi^2, \dots, \pi^t, \dots)$ that can be expressed as a probability distribution of the actions, given states:*

$$\pi_{ia}^t = \mathbb{P}\{A_t = a | S_t = s\}$$

If the policy R is such that there exists an action a such that $\pi_{ia}^t = 1$ for every state $i \in \mathcal{S}$, we say the policy is deterministic*; otherwise, we call the policy* stochastic*. If the policy does not depend on time, we call this policy* stationary *and simply write π_{ia}.*

The natural problem that now arises is how to find policies that maximize the rewards obtained in some precise sense. This is captured by means of a utility function, which might be, e.g., maximize the total expected or average reward. An MDP can be classified according to its time horizon in finite or infinite time MDP. Likewise, we can also distinguish between discrete and continuous time MDPs. For local search metaheuristics, we focus on the MDP formulation where:

1. The time is discrete and the time horizon is infinite.
2. The utility function is the total expected reward.

Regarding condition (1), we note that clearly the execution time of a metaheuristic is always finite, but this finite time cannot be fixed beforehand (except for the class of metaheuristics that are run for a fixed number of iterations). This is modeled by having terminal states with no outgoing transitions (i.e., the probability of transitioning to any other state is 0, and $\mathcal{A}(i) = \emptyset$ for all terminal states $i \in \mathcal{S}$). We argue for condition (2) that, in general, the main strategy used by all metaheuristics is to search for the optimum *indirectly* by exploring the search space $\mathcal{X}$ and exploiting promising regions where an optimum $x^* \in \mathcal{X}$ is supposed to be. However, this is done by *sampling* from the search space in some specific sense and evaluating the objective function f on that sample, thus collecting information about the target function value. If we interpret this sampling and evaluating procedure as collecting *rewards*, the agent (i.e., the metaheuristic) can be interpreted as trying to maximize the total reward obtained.

3.2. Optimal Policies

In general, we model local search metaheuristic algorithms by means of policies R as defined before. We define the reward of such a policy as the total (discounted) expected reward:

$$v_i^\alpha(R) = \sum_{t=1}^{\infty} \mathbb{E}_{i,R}\{\alpha^{t-1} r_t\} \tag{1}$$

where r_t denotes the collected actual reward at time step t, and the expectation is over policy R, given initial state $i \in \mathcal{S}$, and α is a discount factor that regulates the value of future rewards. In the following, we will assume $\alpha = 1$ so we can write:

$$v_i(R) = \sum_{t=1}^{\infty} \sum_{j,a} \mathbb{P}\{S_{t+1} = j | S_t = i, A_t = a\} r_j(a) \tag{2}$$

where $r_j(a)$ denotes the reward obtained from state j after action a. Let $\{P(\pi^t)\}_{ij}$ denote the transition matrix of policy R, and $\{r(\pi^t)\}_i$ its reward vector. We have

$$\{P(\pi^t)\}_{ij} = \sum_{a \in \mathcal{A}} p_{ij}^t(a)\pi_{ia}^t \text{ for all } i, j \in \mathcal{S} \tag{3}$$

$$\{r(\pi^r)\}_i = \sum_{a \in \mathcal{A}} r_i^t(a)\pi_{ia}^t \text{ for all } i \in \mathcal{S} \tag{4}$$

The reward of policy R can be now expressed based on the transition probabilities as follows:

$$v^\alpha(R) = \sum_{t=1}^{\infty} \alpha^{t-1} P(\pi^1)P(\pi^2)\dots P(\pi^{t-1})r(\pi^t) \tag{5}$$

Definition 3. *We call a policy R_** optimal *if it is the best achievable policy considering the reward function, i.e.,*

$$v^\alpha(R_*) = \sup_R v^\alpha(R)$$

3.3. Local Search Metaheuristics as Policies

We now formally represent an arbitrary local search metaheuristic as a policy in an MDP as follows. In this study, we focus on binary strings of length n as candidate solutions, therefore $\mathcal{X} \subseteq \{0,1\}^n$. From now on, we denote by $\mathbf{x} \in \{0,1\}^n$ a binary vector that is a candidate solution for the problem at hand, which might be feasible or not (i.e., it can be that, during the search process, there is an $\mathbf{x}_t$ such that $\mathbf{x}_t \notin \mathcal{X}$). In the following, we will assume (without loss of generality) a maximization problem, i.e., find $\mathbf{x}^*$ such that $\mathbf{x}^* = \text{argmax}_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x})$.

Definition 4. *A criterium H is a function $H : \mathcal{X} \times \mathcal{X} \to \{0,1\}$ such that $H(\mathbf{x}';\mathbf{x}) = 1$ if and only if $\mathbf{x}'$ is a neighbour of $\mathbf{x}$ according to a metaheuristic-specific strategy, and 0 otherwise.*

Typically, instantiations of different local search metaheuristics provide one or several criteria H to determine which candidate solutions $\mathbf{x}'$ can be considered neighbors of the current solution $\mathbf{x}$. For instance, in hill-climbing there is typically only one criterium H that defines the possible candidate solutions starting from the current one (for instance, all solutions that can be obtained from the current one by flipping a random bit). However, other metaheuristics like VNS typically require a set of criteria $\mathcal{H} = \{H_1, \dots, H_n\}$ to be specified, and the algorithm iterates in some form over $\mathcal{H}$.

Definition 5. *We define a* neighborhood $\mathcal{N}_H(\mathbf{x})$ *as the subset of $\mathcal{X}$ induced by H, i.e., all candidate solutions $\mathbf{x}' \in \mathcal{X}$ that are neighbors of $\mathbf{x}$ according to criterium H; therefore, $\mathcal{N}_H(\mathbf{x}) = \{x' \in \mathcal{X} \mid H(\mathbf{x}';\mathbf{x}) = 1\}$.*

Let $\mathbf{x} \in \mathcal{X}$ and $\mathbf{x}' \in \mathcal{N}_H(\mathbf{x})$. We denote by $\mathbf{x} \mapsto \mathbf{x}'$ the operator that transforms $\mathbf{x}$ into $\mathbf{x}'$. For example, the neighborhood of all binary strings located at Hamming distance one from $\mathbf{x}$ would be defined by $\mathcal{N}_{H_D}(\mathbf{x}) = \{x' \in \mathcal{X} \mid H_D(\mathbf{x}';\mathbf{x}) = 1\}$. In this case, $\mathbf{x} \mapsto \mathbf{x}'$ denotes the operator that flips the one bit necessary to turn $\mathbf{x}$ into $\mathbf{x}'$.

Definition 6 (Local search MDP)**.** *Let our local search MDP be defined by:*

- $\mathcal{S} = \{0, 1, \dots, 2^{n-1}\}$. *Any state $i \in \mathcal{S}$ can trivially be converted into a candidate solution as the binary representation of the integer i.*
- $\mathcal{A}(i) = \{i \mapsto j \mid j \in \bigcup_k \mathcal{N}_k\}$, *the set of all possible transformations in all neighborhoods applicable to the current state $i \in \mathcal{S}$.*

- *The transition probabilities, for all $i, j \in \mathcal{S}$, $a \in \mathcal{A}(i)$:*

$$p_{ij}(a) = \begin{cases} 1/|\mathcal{A}(i)|, \text{ if } a = i \mapsto j \\ 0, \text{ otherwise} \end{cases} \tag{6}$$

- *The reward for state j given action $a = i \mapsto j$ is defined by $r_j(a) = f(j) - f(i)$, where f is the objective function (note the slight abuse of notation here since f is defined on $\mathcal{X}$ rather than $\mathcal{S}$, but the conversion is trivial).*

An instantiation of this MDP therefore starts at an arbitrary state $i \in \mathcal{S}$ and visits a number of states according to the transition probabilities before halting at time T. The algorithm terminates as soon as it reaches a *terminal* or *absorbing* state, i.e., a state $i \in \mathcal{S}$ where the transition probability of reaching any other state is 0 ($p_{ij}(a) = 0$ for all $i \neq j$, and $p_{ii}(a) = 1$ otherwise). Note that the rewards are chosen in such a way that, in general, they incentivize making progress towards any optimum $\mathbf{x}^*$, setting the stage for the optimization process to advance towards an optimal solution.

In this setting, we identify a policy R with a particular local search metaheuristic.

Example 1 (Hill climbing)**.** *We define a policy R_{HC} for the standard hill climbing algorithm as follows:*

- $\mathcal{A}(i) = \{i \mapsto j \mid j \in \mathcal{N}_{H_D}\}$ *as defined before (we flip one bit), and*
- *Let $\mathcal{M}(i) = \{j \in \mathcal{S} \mid j = \operatorname{argmax} f(j'), a = i \mapsto j, a \in \mathcal{A}(i)\}$, then*

$$\pi_{ia} = \begin{cases} 1/|\mathcal{M}(i)|, \text{ if } j \in \mathcal{M}(i) \\ 0, \text{ otherwise} \end{cases} \tag{7}$$

Note that R_{HC} is a deterministic policy except when $|\mathcal{M}(i)| > 1$, in which case ties are broken randomly.

In general, the MDP formulation is defined in such a way that the balance between exploration and exploitation becomes explicit in the definition of the policy. Informally, the agent can either maximize the reward in the short run (thus going into an exploitation phase), or temporarily accept negative rewards and therefore going into an exploration phase, which will depend on the particular trajectory taken from an initial solution $\mathbf{x_0}$. Using the previous notions, we define exploration and exploitation explicitly as follows:

Definition 7 (Exploration and exploitation)**.** *Let $a \in \mathcal{A}(i)$ be an action such that $a = i \mapsto j$. We say that a is an* exploration *action if and only if $r_j(a) \leq 0$. Otherwise, we call a an* exploitation *action.*

Definition 8 (Exploration–exploitation function)**.** *Let $\sigma : \mathcal{A}(i) \to \{0, 1\}$ be a function such that $\sigma(a) = 1$ if and only if a is an exploration action, and $\sigma(a) = 0$ otherwise. We call σ the* exploration–exploitation *function.*

We now state our main result, which shows that any local search metaheuristic can be represented by a policy where the balance between exploration and exploitation becomes explicit.

Theorem 1 (Local search exploration-exploitation theorem)**.** *Let M be a local search MDP. For any local search metaheuristic A, there exists a policy R_A such that*

$$\pi_{ia}^t = \alpha_i^A(t)\mathbb{P}\{\sigma(a) = 1 \mid S_t = i\} + \beta_i^A(t)\mathbb{P}\{\sigma(a) = 0 \mid S_t = i\} \tag{8}$$

Theorem 1 shows that for any metaheuristic A, we can explicitly construct the corresponding policy as a mixture of exploration and exploitation actions.

Proof. Let $\mathcal{M}_t^+(i) = \{j \in \mathcal{S} \mid f(j) > f(i), a = i \mapsto j, a \in \mathcal{A}(i), S_t = i\}$ and $\mathcal{M}_t^-(i) = \{j \in \mathcal{S} \mid f(j) \leq f(i), a = i \mapsto j, a \in \mathcal{A}(i), S_t = i\}$ be the sets of improving and non-improving states given action a, respectively. We have:

$$\begin{aligned}
\pi_{ia}^t &= \mathbb{P}\{A_t = a \mid S_t = i\} \\
&= \mathbb{P}\{A_t = a, j \in \mathcal{M}_t^+(i) \mid S_t = i\} + \mathbb{P}\{A_t = a, j \in \mathcal{M}_t^-(i) \mid S_t = i\} \\
&= \frac{|\mathcal{M}_t(i)^+|}{|\mathcal{M}_t(i)^+| + |\mathcal{M}_t(i)^-|}\mathbb{P}\{\sigma(a) = 0 \mid S_t = i\} + \\
&+ \frac{|\mathcal{M}_t(i)^-|}{|\mathcal{M}_t(i)^+| + |\mathcal{M}_t(i)^-|}\mathbb{P}\{\sigma(a) = 1 \mid S_t = i\}
\end{aligned} \tag{9}$$

Therefore,

$$\alpha_i^A(t) = \frac{|\mathcal{M}_t(i)^-|}{|\mathcal{M}_t(i)^+| + |\mathcal{M}_t(i)^-|}, \text{ and } \beta_i^A(t) = \frac{|\mathcal{M}_t(i)^+|}{|\mathcal{M}_t(i)^+| + |\mathcal{M}_t(i)^-|}$$

□

Theorem 1 and its proof provide hints to several measures of interest. The first one is a measure that can be used for convergence analysis and defined as follows:

$$\gamma_i^A(t) := \frac{\beta_i^A(t)}{\alpha_i^A(t)} = \frac{|\mathcal{M}_t(i)^+|}{|\mathcal{M}_t(i)^-|} \tag{10}$$

We call $\gamma_i^A(t)$ the *convergence coefficient* of local search metaheuristic A (conditioned on initial state $i \in \mathcal{S}$) at time t, and denote $\gamma^A(t)$ as the vector of all $\{\gamma_i^A(t)\}_{i\in\mathcal{S}}$ (that we call *convergence vector*). We can now study the convergence behavior of local search metaheuristics according to the limit of the components $\gamma_i^A(t)$ as $t \to \infty$. For instance, if $\lim_{t\to\infty} \gamma_i^A(t) = 0$ (for some state i), this means that algorithm A has converged, since the improving set becomes zero faster than the non-improving set. Note that $\gamma_i^A(t) = 0$ only implies convergence and not optimality (i.e., it can be a local optimum).

The second relevant measure represents an explicit way of evaluating the balance between exploration and exploitation that a specific local search metaheuristic provides. We define

$$\delta_{ia}^A(t) := \frac{\mathbb{P}\{\sigma(a) = 1 \mid S_t = i\}}{\mathbb{P}\{\sigma(a) = 0 \mid S_t = i\}} \tag{11}$$

Averaging over all actions, we get

$$\delta_i^A = \sum_{t=0}^{\infty} \delta_i^A(t) = \sum_{t=0}^{\infty} \sum_{j\in\mathcal{A}(i)} p_{ij}(a)\delta_{ja}^A(t) = \frac{1}{|\mathcal{A}(i)|} \sum_{j\in\mathcal{A}(i)} \sum_{t=0}^{\infty} \delta_{ja}^A(t) \tag{12}$$

We call δ_i^A the *exploration–exploitation coefficient* of local search metaheuristic A conditioned on initial state $i \in \mathcal{S}$. In vector notation, we write δ^A and call it the *exploration–*

exploitation coefficient vector. In general, there can be components of δ^A that converge and others that diverge. Therefore, one possibility would be to assert the general exploration–exploitation behavior of A based on $\delta^A_* = \max_i \delta^A_i$ and investigate the individual components of δ^A in case a more detailed analysis is needed.

We can now define for each metaheuristic a measure of its exploration–exploitation balance as follows:

Definition 9. *We say that A is a balanced local search metaheuristic if $\delta^A_* = C$ for a constant $C > 0, C \in \mathbb{R}$. Otherwise, if δ^A_* diverges to ∞, we say that A is exploration-oriented. Finally, if $\delta^A_* = 0$, we say that A is exploitation-oriented.*

Using our local search MDP, we have transformed the problem of convergence analysis and determining the exploration–exploitation balance of any metaheuristic into the quantitative problem of calculating δ^A_i, δ^A_* and the limits of $\gamma^A_i(t)$ as $t \to \infty$ for $i \in \mathcal{S}$. We now apply these notions to the hill climbing heuristic already mentioned before.

Example 2. *In the case of hill climbing, for every $i \in \mathcal{S}$ we have:*

$$\begin{aligned} \alpha^A_i(t) &= 0 \\ \beta^A_i(t) &= 1/|\mathcal{M}(i)| \\ \gamma^A_i(t) &= 0 \end{aligned} \tag{13}$$

Since $\lim_{t\to\infty} \gamma^A_i(t) = 0, \forall i \in \mathcal{S}$, we have that hill climbing always converges to a (possibly local) optimum. Regarding exploration–exploitation behavior, we have $\delta_{ia}(t) = 0, \forall a \in \mathcal{A}(i), \forall i \in \mathcal{S}$. We therefore obtain that $\delta^A_* = 0$ and hill climbing is exploitation-oriented.

3.4. Summary

For the sake of clarity, we provide a graphical summary of our approach in Figure 1. In this figure, we depict the analysis described in the text as a sequence of five steps and break down each step according to the measures defined.

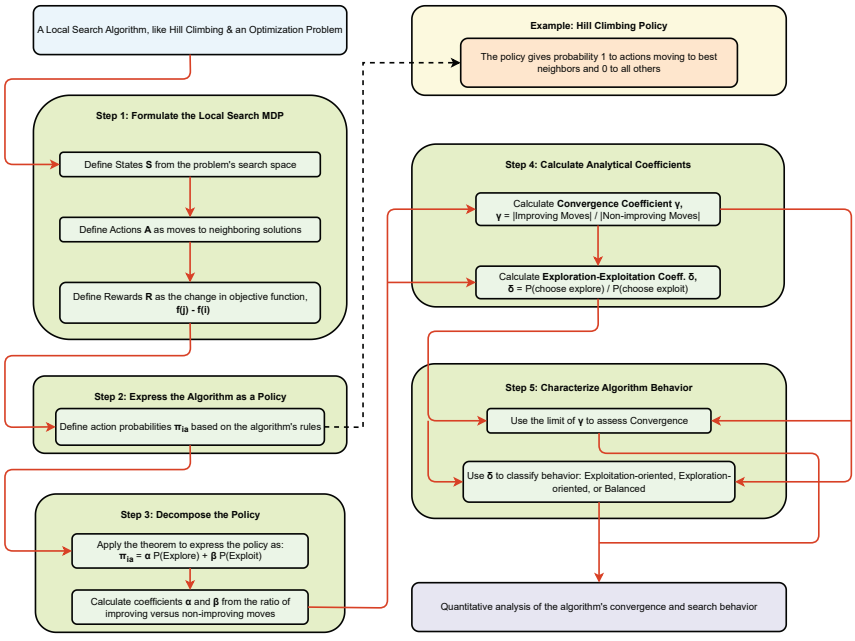


Figure 1. Graphical summary of the MDP approach outlined in this paper.

In the next section, we apply our MDP model to the simulated annealing (SA) metaheuristic and derive some of its convergence and exploration–exploitation properties.

4. Simulated Annealing

Simulated annealing (SA) [10] is a local search metaheuristic inspired by the procedure of cooling a molten solid according to a predefined schedule. The main idea is to start with a high temperature T and decrease this temperature in subsequent iterations with the goal of reaching the minimum value of an energy function E. Transitions from the current candidate solution that improve the objective value are accepted with probability 1. Otherwise, the solution is accepted with a probability that depends on the magnitude of the difference between the current best objective value and the value of the new candidate solution. In the initial, high-temperature phase, the probability of accepting worse solutions is high (emphasizes exploration), whereas in later stages lower temperatures also decrease this probability (emphasizes exploitation). More formally, this probability can be written as:

$$\mathbb{P}\{\mathbf{x}_{t+1} = \mathbf{x}' \,|\, \mathbf{x}_t = \mathbf{x}\} = \begin{cases} 1, \text{ if } E(\mathbf{x}') < E(\mathbf{x}) \\ \exp\left(\dfrac{-\left(E(\mathbf{x}) - E(\mathbf{x}')\right)}{T}\right), \text{ otherwise} \end{cases} \tag{14}$$

In our case, since we are solving by convention maximization problems, without loss of generality we set $E = -f$. Usually, a geometric cooling scheme is used: $T_{new} = \alpha T_{current}$, where $\alpha \in [0, 1)$. Let T_0 denote the initial temperature (which is a free parameter of the algorithm). Then the temperature at time t is given by $T = \alpha^t T_0$.

We now define a policy R_{SA} to model the behavior of SA as a stochastic decision agent as follows:

- Let $\mathcal{M}_t^+(i)$ and $\mathcal{M}_t^-(i)$ be defined as in Theorem 1.
- Let us consider action $a = i \mapsto j$ and state $i \in \mathcal{S}$. Using Theorem 1 we obtain

$$\begin{aligned} \pi_{ia}^t &= \mathbb{P}\{A_t = a \,|\, S_t = i\} \\ &= \frac{|\mathcal{M}_t(i)^+|}{|\mathcal{M}_t(i)^+| + |\mathcal{M}_t(i)^-|} + \frac{|\mathcal{M}_t(i)^-|}{|\mathcal{M}_t(i)^+| + |\mathcal{M}_t(i)^-|} \exp\left(-\frac{f(j) - f(i)}{\alpha^t T_0}\right) \end{aligned} \tag{15}$$

Now, the convergence coefficients can be calculated as follows:

$$\lim_{t\to\infty} \gamma_i^A(t) = \lim_{t\to\infty} \frac{|\mathcal{M}_t(i)^+|}{|\mathcal{M}_t(i)^-|} \xrightarrow[t\to\infty]{} 0, \forall i \in \mathcal{S} \tag{16}$$

Since when time approaches infinity, assuming a geometric cooling scheme, the algorithm will only accept improving actions and in the limit there will be more non-improving than improving actions (note that $\alpha < 1$ by definition).

Let us now calculate the exploration–exploitation coefficient:

$$\mathbb{P}\{\sigma(a) = 1 \,|\, S_t = i\} = \begin{cases} 0, & \text{if } f(j) > f(i) \\ \exp\left(-\frac{f(j)-f(i)}{\alpha^t T_0}\right), & \text{otherwise} \end{cases} \tag{17}$$

Using Equations (11) and (12) we get:

$$\delta_i^A = \frac{1}{|\mathcal{A}(i)|} \sum_{j\in\mathcal{A}(i)} \sum_{t=0}^{\infty} \delta_{ja}^A(t) = \frac{1}{|\mathcal{A}(i)|} \sum_{\substack{j\in\mathcal{A}(i) \\ f(j)\geq f(i)}} \sum_{t=0}^{\infty} \exp\left(-\frac{f(j) - f(i)}{\alpha^t T_0}\right) \tag{18}$$

which for all $i \in \mathcal{S}$ and $\alpha \in [0,1)$ converges to a positive constant $C > 0$, since:

$$\frac{\exp\left(-\frac{f(j)-f(i)}{\alpha^{t+1}T_0}\right)}{\exp\left(-\frac{f(j)-f(i)}{\alpha^{t}T_0}\right)} = \exp\left(-\frac{(1-\alpha)(f(j)-f(i))}{\alpha^{t+1}T_0}\right) \xrightarrow[t\to\infty]{} 0 \tag{19}$$

Therefore, simulated annealing is a *balanced* local search metaheuristic according to our definition.

5. Computational Experiments

In this section, we present some preliminary computational experiments (the code for the experiments is publicly available under https://github.com/IMC-UAS-Krems/mdp_local_search_experiments, accessed on 12 August 2025) that validate the theoretical results obtained in the previous section. We calculate the convergence and exploration–exploitation coefficients $\gamma^A(t)$ and $\delta^A(t)$ explicitly for hill climbing and simulated annealing and visualize their temporal evolution using a random initial state across sample runs with the `OneMax` problem. In `OneMax`, the objective is to maximize the number of 1's of an n bit string, starting from a randomly generated solution. It represents a widely used and simple benchmark for preliminary tests applied to search heuristics.

5.1. Hill Climbing

Figure 2 shows a visualization of the temporal evolution of the convergence coefficient across a sample run of the algorithm using $n = 30$ as string size. As can be seen, the convergence coefficient decreases monotonically until it reaches 0 on the last iteration (red curve) as theoretically predicted. Since there is always an increment of 1 in the objective function after each move, the objective function also increases monotonically and linearly until reaching the optimal objective value (blue curve).

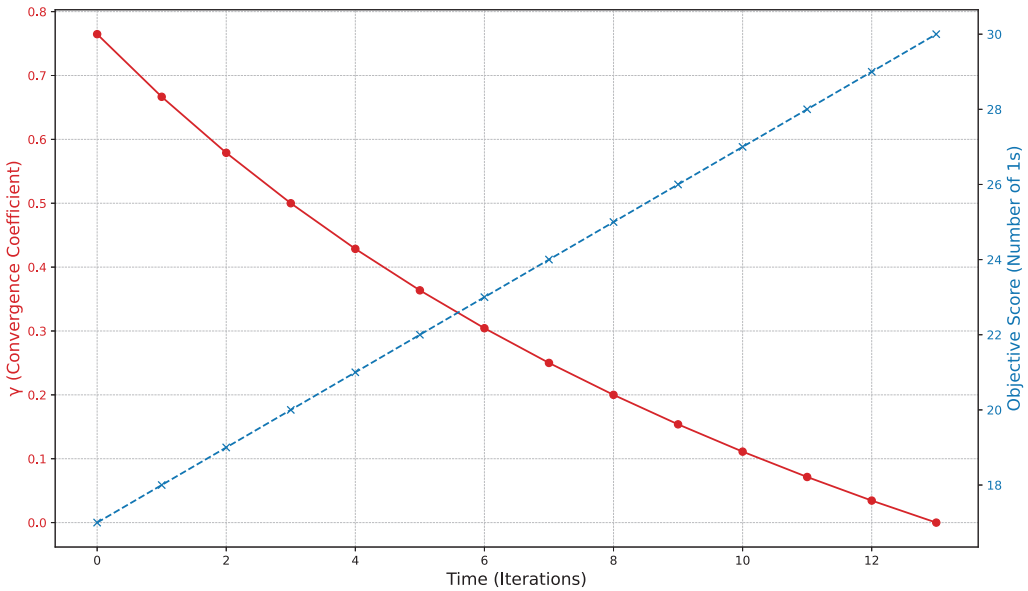


Figure 2. Convergence Coefficient (γ^A) for hill climbing on `OneMax` ($n = 30$).

In Figure 3, we plot the evolution of both the convergence and the exploration–exploitation coefficient in another run of hill climbing (starting from a new random string). As can be seen, the exploration–exploitation coefficient stays (by definition) at 0, indicating

that only exploitation-oriented moves, and no exploration-oriented, are performed by the algorithm.

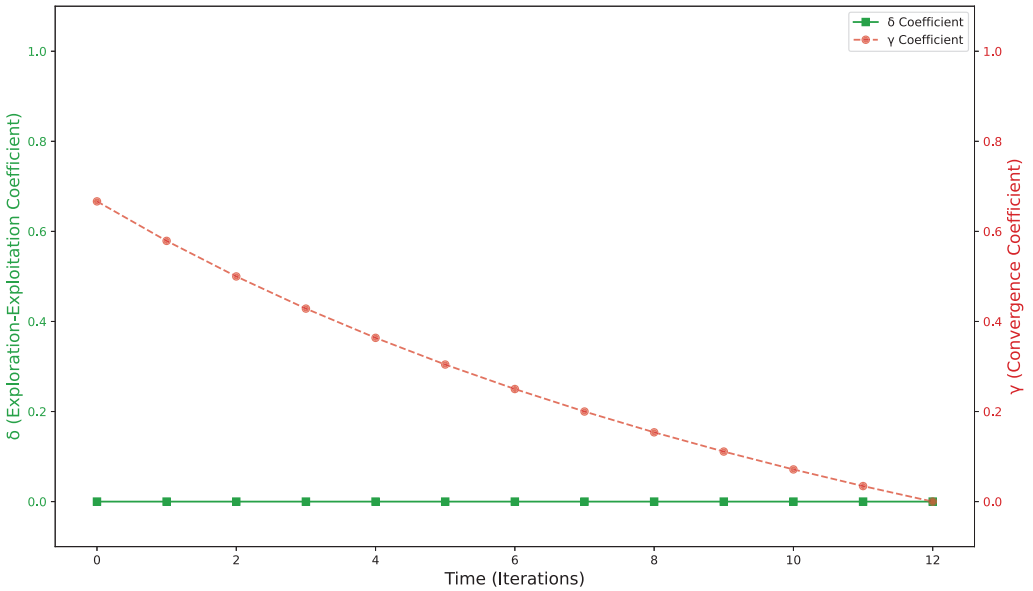


Figure 3. Exploration–exploitation (δ^A) and convergence coefficient (γ^A) for hill climbing on `OneMax` ($n = 30$).

5.2. Simulated Annealing

In the case of SA, we provide the evolution of the convergence coefficient in Figure 4. In this case, the evolution of both γ^A and the objective value differ substantially from hill climbing. In the initial stages of the algorithm (where T is still high), γ^A fluctuates as expected, alternating between improving and non-improving moves. However, as the temperature decreases and the algorithm focuses on promising regions of the search space, the fluctuations are also reduced and γ^A reaches its final value of 0, indicating that the algorithm has converged.

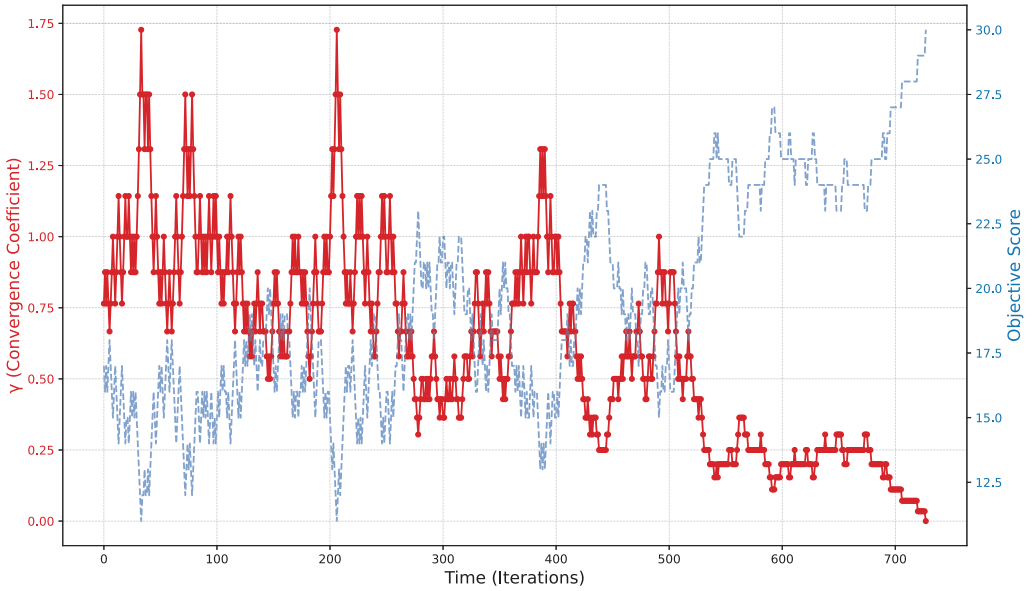


Figure 4. Convergence coefficient (γ^A) for SA on `OneMax` ($n = 30$), using $\alpha = 0.995$ and an initial temperature of $T_0 = 10$.

Figure 5 provides a visualization of the evolution of the exploration–exploitation coefficient for SA. As expected from the theoretical discussion, the evolution of the coefficient closely matches the current temperature: in the initial stages (high temperature), δ^A is also high, while in later stages with decreasing temperature, the coefficient also decreases, signaling the transition from a highly exploration-oriented to exploitation-oriented behavior. In sum, we can conclude that SA has a *balanced* exploration–exploitation behavior.

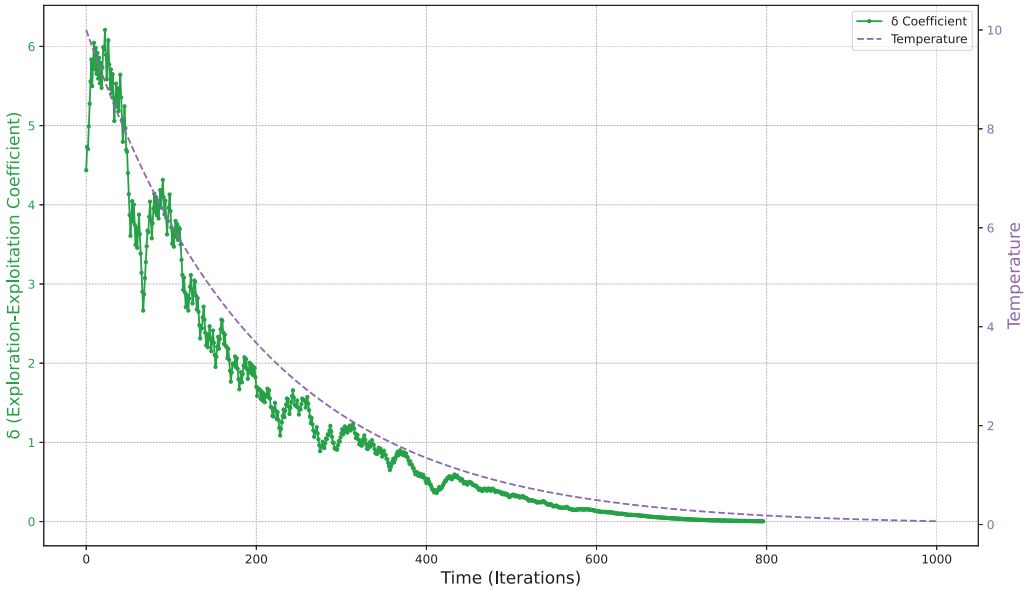


Figure 5. Exploration–exploitation (δ^A) for SA `OneMax` ($n = 30$) along with temperature in a sample run with maximum 1000 iterations. Convergence is achieved around iteration 800.

6. Conclusions

In this paper, we have presented a new framework for modeling the convergence and exploration–exploitation behavior of local search metaheuristics based on Markov decision processes. This framework provides a novel and intuitive way of characterizing essential properties of local search metaheuristics in a principled way. We proved that any local search metaheuristic can be analyzed using the tools provided and showed how to apply the framework in two cases: the hill climbing and the simulated annealing metaheuristics. In the case of hill climbing, we showed that the framework correctly classifies it as exploitation-oriented, whereas simulated annealing was classified as balanced, which is consistent with the general consensus in the literature. Our study goes beyond the state-of-the-art in metaheuristic analysis, since the tools we develop provide a more expressive methodology than methods based on Markov chains and other probabilistic approaches. As a consequence, the behavior of local search metaheuristics can be modeled in a more realistic way and the results obtained are consistent with practice.

As future work, we plan to extend our results to other local search metaheuristics like variable neighborhood search (VNS) [12] and large neighborhood search (LNS) [30]. Additionally, the framework will be extended to be able to handle population-based metaheuristics (PBMHs) like genetic algorithms (GA) [14], particle swarm optimization (PSO) [15] and estimation of distribution algorithms (EDAs) [16]. Additionally, more empirical studies will be conducted with the goal of explicitly calculating the measures described in this paper for different stages in the optimization process and different instances, potentially leading to new tools for instance space analysis.

Author Contributions: Conceptualization, R.R.-T., D.D., S.P. and H.B.; methodology, R.R.-T., D.D., S.P. and H.B.; validation, R.R.-T., D.D., S.P. and H.B.; formal analysis, R.R.-T.; writing—original draft preparation, R.R.-T.; writing—review and editing, R.R.-T., D.D., S.P. and H.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is contained within the article. Further inquiries can be directed to the corresponding authors.

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

References

1. Sörensen, K.; Glover, F. Metaheuristics. In *Encyclopedia of Operations Research and Management Science*; Springer: Boston, MA, USA, 2013; Volume 62, pp. 960–970.
2. Silberholz, J.; Golden, B. Comparison of Metaheuristics. In *Handbook of Metaheuristics*; Gendreau, M., Potvin, J.Y., Eds.; Springer: Boston, MA, USA, 2010; pp. 625–640. [CrossRef]
3. Lin, S.W.; Lee, Z.J.; Ying, K.C.; Lee, C.Y. Applying hybrid meta-heuristics for capacitated vehicle routing problem. *Expert Syst. Appl.* **2009**, *36*, 1505–1512. [CrossRef]
4. Fernandez, S.A.; Juan, A.A.; de Armas Adrián, J.; Silva, D.G.e.; Terrén, D.R. Metaheuristics in Telecommunication Systems: Network Design, Routing, and Allocation Problems. *IEEE Syst. J.* **2018**, *12*, 3948–3957. [CrossRef]
5. Pillay, N. A survey of school timetabling research. *Ann. Oper. Res.* **2014**, *218*, 261–293. [CrossRef]
6. Kaur, M.; Saini, S. A Review of Metaheuristic Techniques for Solving University Course Timetabling Problem. In *Advances in Information Communication Technology and Computing*; Goar, V., Kuri, M., Kumar, R., Senjyu, T., Eds.; Lecture Notes in Networks and Systems; Springer: Singapore, 2021; pp. 19–25. [CrossRef]
7. Ecoretti, A.; Ceschia, S.; Schaerf, A. Local search for integrated predictive maintenance and scheduling in flow-shop. In Proceedings of the 14th Metaheuristics International Conference, Syracuse, Italy, 11–14 July 2022.
8. Ceschia, S.; Guido, R.; Schaerf, A. Solving the static INRC-II nurse rostering problem by simulated annealing based on large neighborhoods. *Ann. Oper. Res.* **2020**, *288*, 95–113. [CrossRef]
9. Doering, J.; Kizys, R.; Juan, A.A.; Fitó, À.; Polat, O. Metaheuristics for rich portfolio optimisation and risk management: Current state and future trends. *Oper. Res. Perspect.* **2019**, *6*, 100121. [CrossRef]
10. Kirkpatrick, S.; Gelatt, C.D., Jr.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [CrossRef] [PubMed]
11. Glover, F.; Laguna, M. *Tabu Search*; Springer: Berlin/Heidelberg, Germany, 1998.
12. Mladenović, N.; Hansen, P. Variable neighborhood search. *Comput. Oper. Res.* **1997**, *24*, 1097–1100. [CrossRef]
13. Shaw, P. Using constraint programming and local search methods to solve vehicle routing problems. In Proceedings of the International Conference on Principles and Practice of Constraint Programming, Pisa, Italy, 26–30 October 1998; pp. 417–431.
14. Holland, J.H. Genetic Algorithms. *Sci. Am.* **1992**, *267*, 66–73. [CrossRef]
15. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
16. Larrañaga, P.; Lozano, J.A. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*; Springer Science & Business Media: Berlin, Germany, 2001; Volume 2.
17. Liu, B.; Wang, L.; Liu, Y.; Wang, S. A unified framework for population-based metaheuristics. *Ann. Oper. Res.* **2011**, *186*, 231–262. [CrossRef]
18. Chauhdry, M.H.M. A framework using nested partitions algorithm for convergence analysis of population distribution-based methods. *EURO J. Comput. Optim.* **2023**, *11*, 100067. [CrossRef]
19. Suzuki, J. A Markov chain analysis on simple genetic algorithms. *IEEE Trans. Syst. Man Cybern.* **1995**, *25*, 655–659. [CrossRef]
20. Suzuki, J. A further result on the Markov chain model of genetic algorithms and its application to a simulated annealing-like strategy. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **1998**, *28*, 95–102. [CrossRef] [PubMed]
21. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; U Michigan Press: Ann Arbor, MI, USA, 1975.
22. Eiben, A.E.; Aarts, E.H.L.; Van Hee, K.M. Global convergence of genetic algorithms: A markov chain analysis. In Proceedings of the Parallel Problem Solving from Nature, Dortmund, Germany, 1–3 October 1991; Schwefel, H.P., Männer, R., Eds.; Springer: Berlin/Heidelberg, Germany, 1991; pp. 3–12. [CrossRef]

23. Cao, Y.; Wu, Q. Convergence analysis of adaptive genetic algorithms. In Proceedings of the Second International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, Glasgow, UK, 2–4 September 1997; pp. 85–89. ISSN: 0537-9989. [CrossRef]
24. Eberbach, E. Toward a theory of evolutionary computation. *Biosystems* **2005**, *82*, 1–19. [CrossRef] [PubMed]
25. Cuevas, E.; Diaz, P.; Camarena, O. Experimental Analysis Between Exploration and Exploitation. In *Metaheuristic Computation: A Performance Perspective*; Cuevas, E., Diaz, P., Camarena, O., Eds.; Intelligent Systems Reference Library, Springer International Publishing: Cham, Switzerland, 2021; pp. 249–269. [CrossRef]
26. Chen, Y.; He, J. Exploitation and Exploration Analysis of Elitist Evolutionary Algorithms: A Case Study. *arXiv* **2020**, arXiv:2001.10932. [CrossRef]
27. Xu, J.; Zhang, J. Exploration-exploitation tradeoffs in metaheuristics: Survey and analysis. In Proceedings of the 33rd Chinese Control Conference, Nanjing, China, 28–30 July 2014; pp. 8633–8638. ISSN: 1934-1768. [CrossRef]
28. Chen, J.; Xin, B.; Peng, Z.; Dou, L.; Zhang, J. Optimal Contraction Theorem for Exploration–Exploitation Tradeoff in Search and Optimization. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2009**, *39*, 680–691. [CrossRef]
29. Kallenberg, L.C. *Lecture Notes on Markov Decision Processes*; University of Leiden: Leiden, The Netherlands, 2022.
30. Pisinger, D.; Ropke, S. Large Neighborhood Search. In *Handbook of Metaheuristics*; Gendreau, M., Potvin, J.Y., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 99–127. [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.

algorithms

MDPI

Article

A Reinforcement Learning Hyper-Heuristic with Cumulative Rewards for Dual-Peak Time-Varying Network Optimization in Heterogeneous Multi-Trip Vehicle Routing

Xiaochuan Wang, Na Li * and Xingchen Jin

College of Transportation Engineering, Dalian Maritime University, Dalian 116026, China; wxc508181034@dlmu.edu.cn (X.W.); jxc_925777264@dlmu.edu.cn (X.J.)
* Correspondence: lina@dlmu.edu.cn

Abstract

Urban logistics face complexity due to traffic congestion, fleet heterogeneity, warehouse constraints, and driver workload balancing, especially in the Heterogeneous Multi-Trip Vehicle Routing Problem with Time Windows and Time-Varying Networks (HMTVRPTW-TVN). We develop a mixed-integer linear programming (MILP) model with dual-peak time discretization and exact linearization for heterogeneous fleet coordination. Given the NP-hard nature, we propose a Hyper-Heuristic based on Cumulative Reward Q-Learning (HHCRQL), integrating reinforcement learning with heuristic operators in a Markov Decision Process (MDP). The algorithm dynamically selects operators using a four-dimensional state space and a cumulative reward function combining timestep and fitness. Experiments show that, for small instances, HHCRQL achieves solutions within 3% of Gurobi's optimum when customer nodes exceed 15, outperforming Large Neighborhood Search (LNS) and LNS with Simulated Annealing (LNSSA) with stable, shorter runtime. For large-scale instances, HHCRQL reduces gaps by up to 9.17% versus Iterated Local Search (ILS), 6.74% versus LNS, and 5.95% versus LNSSA, while maintaining relatively stable runtime. Real-world validation using Shanghai logistics data reduces waiting times by 35.36% and total transportation times by 24.68%, confirming HHCRQL's effectiveness, robustness, and scalability.

Keywords: routing; multi-trip; time-varying road networks; heterogeneous fleets; Q-learning; hyper-heuristic

1. Introduction

1.1. Background and Motivation

The rapid progress in the modern logistics industry presents both opportunities and challenges for logistics and supply chain management companies. With fluctuating customer demands, swift technological shifts, fierce competition, and the uncertainties of globalization, logistics has become increasingly challenging [1]. Brands and e-commerce retailers have significant daily inventory and return needs, imposing a substantial load on third-party transportation companies managing vehicle scheduling and task allocation. These companies must keep up with real-time inventory and return demands to make appropriate vehicle dispatches, meeting the requirements of every agent. Advances in transportation systems have enabled logistics companies to better manage their planning.

The impact of service quality is enormous [2], so logistics companies need efficient delivery services and good service quality to prevent late delivery, which can lead to customer dissatisfaction and losses for the company.

To better serve their clients, supply chain management companies typically establish their own logistics platforms to facilitate transportation services. Transportation is a critical function within the supply chain, encompassing the process of delivering finished goods from distribution centers to customers. For transportation companies, effectively planning vehicle routes to deliver goods to customers while reducing the associated costs of total investment and operational expenses is a challenging issue known as the Vehicle Routing Problem (VRP) [3]. Today, the classic VRP model has been extensively studied and no longer fully meets the evolving needs of transportation, leading to the development of numerous variants that correspond to real-world issues, such as the Capacitated Vehicle Routing Problem (CVRP) [4], the Vehicle Routing Problem with Time Windows (VRPTW) [5], the Heterogeneous Fleet Vehicle Routing Problem (HVRP) [6], the Split Delivery Vehicle Routing Problem (SDVRP) [7], and the Multi-Trip Vehicle Routing Problem (MTVRP) [8], among others. Scholars have developed numerous algorithms to address the VRP and its variants. For large-scale problems, classic metaheuristic algorithms include Genetic Algorithms (GA) [9], Ant Colony Optimization (ACO) [10], Tabu Search (TS) [11], Iterated Local Search Algorithm (ILS) [12], Particle Swarm Optimization (PSO) [13], and more. For small-scale problems, precise algorithms have been proposed by, for example, Su E [14], who introduced an exact branch-and-price-and-cut algorithm to solve the pickup and delivery problem in crowd-sourced last-mile delivery (PDPCBT), and Zhou H [15], who proposed an exact branch-and-price algorithm for a two-level vehicle routing problem involving drones. The motivation for our research is a route design challenge faced by a company in Shanghai, China, aimed at fulfilling the pickup and delivery demands of each warehouse, thereby facilitating the timely distribution of goods needed by customers across the country. It is known that each warehouse is located around Shanghai, with the furthest one in Kunshan, China. For this multi-trip vehicle routing problem, we have formulated it as an NP-hard problem and incorporated significant real-world characteristics faced by the company into the model. To address this, we designed a deep learning-based hyper-heuristic algorithm and applied it to real-case scenarios to verify its effectiveness.

1.2. Related Work

The focus of this paper is a novel variant of the multi-trip vehicle routing problem, specifically the complex real-world constraints vehicle scheduling problem based on the logistics platforms of business retailers or supply chain management companies. We have considered real-world constraints such as heterogeneous fleets, time windows, warehouse loading and unloading efficiency, and peak traffic on road segments, and designed a hyper-heuristic algorithm based on reinforcement learning to address this issue. The structure of this paper is as follows: Section 2 discusses the hyperheuristic algorithm based on reinforcement learning, and proposes a hyperheuristic algorithm with cumulative reward mechanism. Section 3 defines the parameters and establishes the MILP model. Section 4 reports the relevant numerical experiments and numerical analysis. Section 5 provides conclusions and future research directions.

1.2.1. Research on Multi-Trip Vehicle Routing Problems (MTVRPs)

In this paper, we focus on the topic or keywords, restrict our search to the fields of Operations Research and Management Science, Transportation, or Traffic Science and

Technology, and set the publication date between 2013 and 2024. We conducted our search on ScienceDirect and Web of Science, with the results shown in Figure 1.

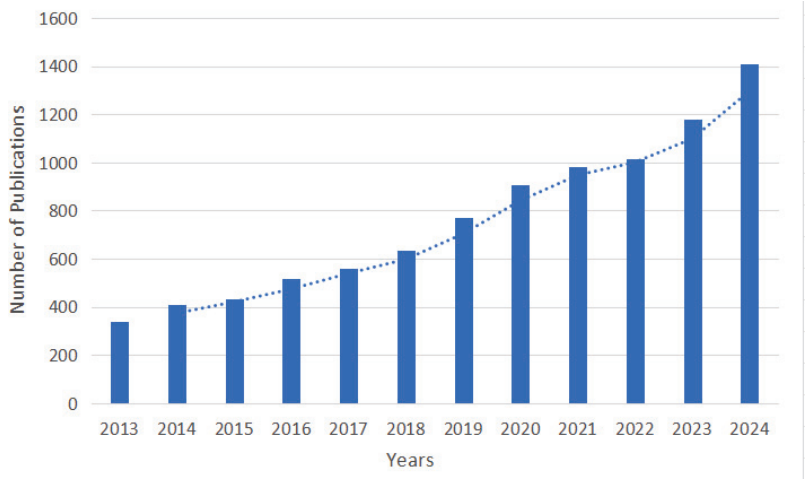


Figure 1. Annual number of MTVRP literature publications.

From the figure, it is evident that the MTVRP has become an increasingly hot topic, and indeed, in reality, the MTVRP is more common and complex. The MTVRP was first proposed in 1990, involving a vehicle making several journeys during a day. Compared to the VRP, the MTVRP is undoubtedly more realistic, prompting more scholars to focus on this issue. Real-world problems are generally more complex than those found in academic research, making these issues more practically relevant. Pan et al. [16] studied the restocking problem faced by an automated coffee vending company for its branch locations, each with varying capacities, demands, and daytime restocking time windows, and employed a hybrid metaheuristic algorithm to address the problem. Moon et al. [17] investigated the location-routing problem with multi-compartment and multi-trip aspects and designed a corresponding hybrid genetic algorithm to solve it. Yu et al. [18] examined the Home Replenishment Delivery System (HRDS), introduced a new variant of the profitable tour problem known as the Multi-Vehicle Profitable Tour Problem with Flexible Compartments and Mandatory Customers (MVPTPFC-MC), and proposed an Adaptive Large Neighborhood Search (ALNS) algorithm to tackle this issue. You et al. [19] investigated a novel Truck Platooning Container Drayage Problem with Multiple Trips (MT-CDP-TP), developing a branch-and-price-and-cut (BPC) algorithm to solve the set partitioning model. They proposed a tailored pulse propagation algorithm based on the pricing problem of BPC. Sethanan et al. [20] introduced an integer linear programming formulation and a novel hybrid differential evolution algorithm to address the Multi-Trip Vehicle Routing Problem with backhauls and heterogeneous fleet. Neira et al. [21] presented a new compact integer programming formulation for the Multi-Trip Vehicle Routing Problem with Time Windows, highlighting the ease of implementation in any available optimization software. Liu et al. [22] proposed a dual adaptive general variable neighborhood search to tackle the Unmanned Electric Vehicle Routing and Scheduling Problem in green manufacturing systems.

1.2.2. Reinforcement Learning Approaches for MTVRP

The integration of reinforcement learning (RL) with hyper-heuristic frameworks has become an emerging trend in solving complex operational research problems. This combination leverages RL to adaptively select among multiple low-level heuristics, thereby handling large-scale instances and multiple constraints more effectively. It has been widely explored

in various domains, such as heterogeneous vehicle routing problems (Qin et al. [23]), automated container terminal scheduling (Xu et al. [24]), batch flow shop scheduling (Duan et al. [25]), distributed flexible job shop scheduling (Zhang et al. [26]), and manufacturing project scheduling (Li et al. [27]), demonstrating its versatility and practical efficiency.

Building on this methodological trend, the application of RL-based hyper-heuristic algorithms to Multi-Trip Vehicle Routing Problems (MTVRPs) and related VRP variants has emerged as a promising research direction. For example, Li et al. [28] developed a novel RL-based hyper-heuristic for heterogeneous VRPs, while Zhang et al. [29] proposed a deep RL approach for heterogeneous capacitated VRPs, both demonstrating the ability of RL to manage complex vehicle constraints. Shang et al. [30] designed a Q-learning and multi-strategy hyper-heuristic for multi-compartment green routing, addressing multi-trip and multi-time-window constraints. Xu et al. [31] introduced a dynamic embedding-based RL method for VRP with multiple time windows. Other works further explore RL in combinatorial action spaces, column generation frameworks, and graph-based VRP representations (Dastpak et al. [32]). These studies illustrate that RL combined with hyper-heuristics can effectively coordinate multiple decision levels, handle complex constraints, and achieve high-quality solutions for large-scale MTVRP instances. Table 1 summarizes selected related studies.

Table 1. The related literature.

Author	Problem Characteristics				Method
	HF	Multi-Trip	LC	TVN	
Nguyen et al. [3]		✔			AHA
Pan et al. [16]		✔			HMA
Yu et al. [18]	✔	✔			ALNS
You et al. [19]		✔			BPC + PPC
Sethanan et al. [20]	✔	✔			DE + FLC
Liu et al. [22]		✔			ALNS
Wang et al. [33]	✔				TDRL
Huang et al. [34]		✔	✔		BPC + CG
Fragkogios et al. [35]		✔	✔	✔	BD
Blauth et al. [36]				✔	Local Search
Sun et al. [37]		✔		✔	RHCG
Gutierrez et al. [38]		✔		✔	HH
Lu et al. [39]				✔	BP
Coelho et al. [40]	✔	✔			ILS
Zhao et al. [41]		✔			HH
He et al. [42]		✔	✔		HH
Vieira et al. [43]	✔	✔			AVNR
Our work	✔	✔	✔	✔	HHCRQL

HF: heterogeneous fleet; TVN: time-varying network; LC: loading capacity; FLC: fuzzy logic controller; DE: differential evolution; BPC: branch-and-price-and-cut algorithm; PPC: pulse propagation algorithm; HMA: hybrid metaheuristic algorithm; ALNS: Adaptive Large Neighborhood Search; NLNS: Neural Large Neighborhood Search; AHA: Adaptive Heuristic Algorithm; HH: hybrid heuristics; HA: hybrid approaches; AVNR: Adaptive Variable Neighborhood Race; RHCG: rolling-horizon-based column generation; BP: branch-and-price: BD: benders decomposition; ILS: Iterated Local Search.

The main contributions of this paper are summarized as follows:

1. In response to the warehouse transfer issues faced by the company's B2C operations, we have simultaneously considered the balance of driver workload, time windows, heterogeneous fleet, loading capacity, and time-varying road networks (morning and evening peaks), which are significant real-world characteristics with substantial practical value for logistics enterprises. To date, we have not found any MTVRP studies that comprehensively account for all these realistic features.
2. To address the problem, we define this problem as the Heterogeneous Multi-Trip Vehicle Routing Problem with Time Windows and Time-Varying Networks (HMTVRPTW-

TVN) and develop a mixed-integer linear programming (MILP) formulation. A dual-peak time discretization method is proposed to efficiently model time-varying networks by segmenting continuous peak periods, alongside an exact linearization technique for handling times to enable coordinated scheduling of heterogeneous fleets.

3. Based on the characteristics of this real-world problem, we propose a novel HHCRQL algorithm that incorporates reinforcement learning (RL) into MTVRP. The algorithm effectively reduces the total time required for scheduling multiple trips, with its performance advantages becoming more pronounced when handling large-scale instances.
4. We established an appropriate Markov Decision Process (MDP) model and proposed a cumulative reward assignment method based on time steps, applying the concept of cumulative rewards throughout the training process. By solving the MDP model, we effectively reduced the number of iterations and improved overall computational efficiency.

2. Materials and Methods

This section introduces the solution algorithm for the aforementioned problem. The algorithm discussed in this paper is a hyper-heuristic approach that combines Q-learning with heuristic operators (HO). The main content includes the establishment of an MDP model, the definition of specific state sets, action sets, and so on.

2.1. Heuristic Operators–Reinforcement Learning Framework

Due to the combinatorial optimization nature of the logistics platform's multi-reality constraint scheduling problem, it is challenging to enumerate all solutions through exhaustive methods. Therefore, optimization algorithms require high computational efficiency, and hyper-heuristic algorithms are well suited for this scenario. A typical hyper-heuristic (HH) structure includes two levels: low-level heuristics (LLHs) and high-level heuristics (HLHs) [44]. Low-level heuristics encompass solution representation, computation functions, and a set of LLHs, which are specific problem operators directly applied to solutions, also known as heuristic operators (HOs). High-level heuristics are responsible for two main tasks: selecting HOs to generate new solutions and evaluating their acceptability. Both significantly impact the performance of the hyper-heuristic model, making the development of advanced and low-level strategies for specific problems a significant challenge. To address this challenge, we have developed a reinforcement learning-based hyper-heuristic (HO-RL) that leverages the strengths of different HOs. The RL method is a high-level strategy that chooses the appropriate LLH based on the environment. Specifically, the HO-RL framework starts with an initial solution and then iteratively improves it using HOs. At each iteration, RL first invokes a selection mechanism to choose the most suitable HO based on performance metrics. Then, it applies the HO to the current solution and generates a new solution. Finally, RL invokes acceptance criteria to determine whether to accept the new solution. The goal is to automatically select effective HOs at each decision point, as illustrated in Figure 2.

The process of selecting a heuristic operator (HO) can be viewed as a Markov Decision Process (MDP), where the agent must choose from a set of actions in a set of states with unknown reward models. Q-learning is a model-free reinforcement learning algorithm that learns to maximize cumulative rewards by learning a policy. In Q-learning, the agent learns an action–value function (Q-function) through interaction with the environment, which estimates the expected utility of taking a specific action in a given state. The core of Q-learning is the trade-off between exploration and exploitation. In the following sections,

we introduce the construction of the specific MDP model, including the definition of state sets, action sets, selection strategies, and the reward function.

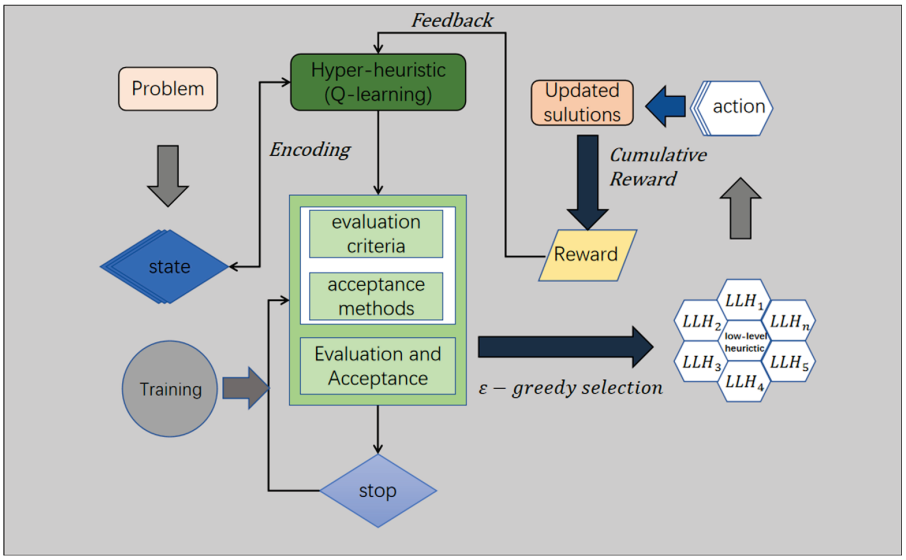


Figure 2. HO-RL framework.

2.1.1. Initial Solution Generation

This paper begins by employing a greedy algorithm to obtain an initial near-optimal solution. This initial solution serves to narrow the solution space and reduce the algorithm's overall running time. The solution is structured by separating the number of vehicle uses (available), vehicle sequence number (car), operating routes (task), and cargo quantity (carry) into distinct lists. These four lists collectively define the initial solution state. A key advantage of this representation is its facilitation of heuristic operator application. A critical design decision within the HO-RL framework involves the representation of solutions in the search space. To clarify this representation, a simple numerical illustration is provided. Figure 3 depicts a candidate solution involving 5 warehouses and 2 vehicles, each assigned between 3 and 5 tasks. The left side shows the four vehicle routes, the center displays the associated vehicles and their available times, and the right side indicates the loading quantity.

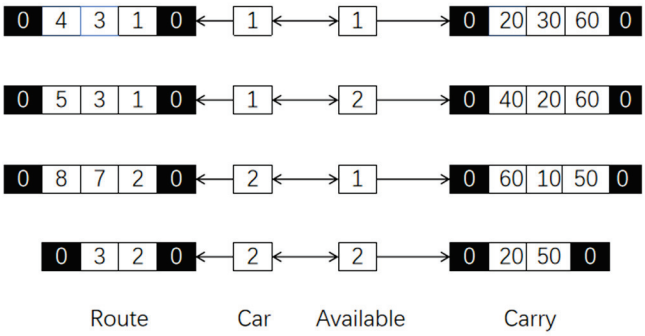


Figure 3. Encoding of the solution.

2.1.2. Low-Level Heuristic Operator Operations

For the generated initial solution, we design and apply a set of heuristic operators—destruction, repair, and exchange—to iteratively improve the routing solution. In the destruction phase, as shown in Figure 4, two strategies are used to increase solution

diversity and enable local search: maximum saving node removal, which deletes the node that yields the largest time saving to break high-efficiency segments, and longest route removal, which removes the route with the longest travel time and releases the corresponding vehicle, allowing tasks to be reassigned.

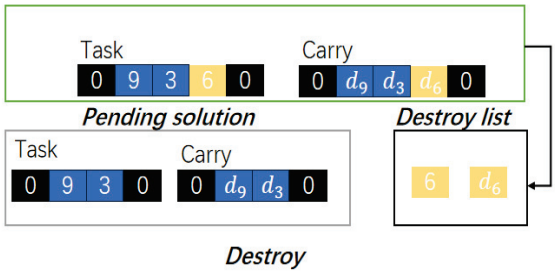


Figure 4. One of the destroy operators.

In the repair phase, as shown in Figure 5, removed nodes from the destroy_list are reinserted using greedy insertion or random insertion. Greedy insertion tries all feasible positions and selects the one with minimal incremental travel cost while respecting vehicle capacity. Random insertion selects feasible routes and positions randomly; if a node cannot be inserted into any existing route, a new route is created using an available vehicle. Both strategies ensure all removed customer demands are reinserted, balancing cost optimization and stochasticity.

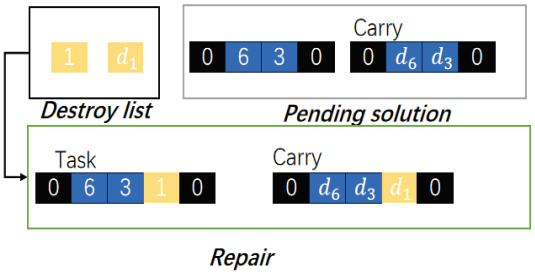


Figure 5. One of the repair operators.

In the exchange phase, as shown in Figure 6, two types of swaps are applied to improve route quality: intra-route customer exchange optimizes local visiting sequences, and inter-route customer exchange swaps customer nodes between routes under capacity constraints to balance workloads and reduce total travel time. Additionally, a vehicle workload balancing step swaps entire routes between vehicles to reduce variance in route durations and enhance resource utilization. By iteratively applying these destruction, repair, and exchange operators, the method effectively explores the solution space, maintains feasibility, optimizes total travel cost, and improves vehicle utilization.

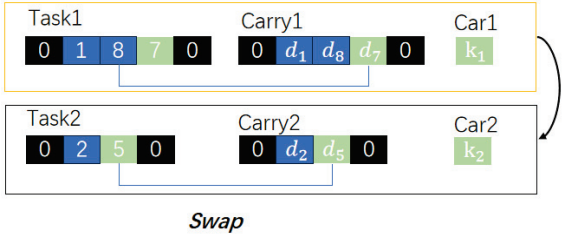


Figure 6. One of the swap operators.

2.2. MDP Model Establishment

Q-learning is one of the most effective value-based reinforcement learning (RL) algorithms, with the goal of estimating the Q-values of the optimal policy. The construction of a Markov Decision Process (MDP) is considered one of the most critical steps in Q-learning. An MDP can typically be described by four elements: the state set, the action set A, the reward function, and the policy function. The Q-table is updated based on a comprehensive consideration of the experienced states, selected actions, and the rewards obtained by the agent. The Q-values, which characterize the rationality of action selection, are calculated based on a comprehensive consideration of the experienced states, selected actions, and the rewards obtained by the agent. The Q-value update function is performed by the following formula:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_t + \gamma max Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)]$$

2.2.1. Definition of the State Set

To effectively represent the decision-making process in the Heterogeneous Multi-Trip Vehicle Routing Problem with Time Windows and Time-Varying Networks (HMTVRPTW-TVN), this study designs the state space within the framework of a Markov Decision Process (MDP), as shown in Table 2. The design objective is to extract a limited number of features that capture the key characteristics of the dispatching environment, ensuring both theoretical soundness and learning efficiency, while maintaining sufficient expressive power and operational relevance. In traditional reinforcement learning, state space design faces the dual challenge that excessively large or irrelevant state dimensions can cause the "curse of dimensionality" and sample sparsity, while overly simplified states may omit critical decision information and reduce the effectiveness of policy learning.

Table 2. The state set in MDP.

State Variable	Description	Possible Values/Intervals	Notes
F	Feasibility Indicator	0/1	Indicates whether the current solution satisfies all constraints (1 = feasible, 0 = infeasible)
V	Number of Vehicle Types	Discrete integers, e.g., 1, 2–3, ≥4	Number of distinct vehicle types used, reflecting fleet heterogeneity and problem complexity
T_r	Remaining Task Ratio	[0, 0.25), [0.25, 0.5), [0.5, 0.75), [0.75, 1.0]	Proportion of remaining demand relative to initial demand, representing progress stages
U	Vehicle Load Utilization	<33%, 33–66%, >66%	Overall fleet load utilization intervals, reflecting resource efficiency

Accordingly, the state is defined as a 4-tuple:

$$s = (F, V, T_r, U)$$

composed of four discretized dimensions:

(1) Feasibility Indicator: F

Feasibility is a fundamental constraint in vehicle routing, directly influencing action selection and search direction. A binary variable F is used to indicate whether the current solution satisfies all constraints, where F = 1 denotes feasibility and F = 0 denotes infeasibility. This enables the policy to quickly identify and prioritize feasible regions of the solution space.

(2) Number of Vehicle Types: V

This is the count of distinct vehicle types currently used in the solution. V quantifies fleet heterogeneity and problem complexity: higher V typically implies more complex assignment and compatibility considerations. Using a vehicle-type count as a state component enables the policy to adapt to different fleet compositions; when necessary, VVV can be coarsely discretized (e.g., V = 1, V = 2–3, V $\geq$ 4) to control state-space growth.

(3) Remaining Task Ratio Segmentation: T_r

The remaining demand ratio—defined as the fraction of unserved demand relative to initial demand—is discretized into a small number of intervals to represent progress stages of the routing problem. In this study, we use four intervals: [0, 0.25), (0.25, 0.5], (0.5, 0.75], and [0.75, 1.0]. T_r captures the search horizon and priority structure for remaining tasks while limiting state cardinality.

(4) Vehicle Load Utilization Interval: U

The overall fleet load utilization (aggregate load/aggregate capacity) is discretized into three intervals: low (<33%), medium (33–66%), and high (>66%). U encodes resource utilization efficiency and helps the policy distinguish between states that require load consolidation, redistribution, or capacity increase.

Each chosen state component is explicitly tied to core constraints or optimization objectives of HMTVRPTW-TVN, thereby ensuring that the state space is both informative and compact. Discretization balances representational fidelity and computational tractability, mitigating exponential growth in the number of states and improving training efficiency and convergence stability for tabular Q-learning. Finally, through construction, the state set is intended to satisfy the Markov property in the sense that it contains the information necessary for immediate decision-making; state transitions are then modeled as dependent only on the current state and the selected action, consistent with MDP theory.

2.2.2. Design of Action Set and Reward Function

In a Markov Decision Process (MDP), the action set is a core component, representing all possible actions an agent can take in a particular state. The design and selection of the action set directly affect the agent's learning efficiency and the quality of final decision-making. For each state, the action set may be fixed or may vary depending on the state. It can be defined as follows:

$$A_s = \{a_1, a_2, a_3, \cdots, a_n\}$$

In the context of the Markov Decision Process (MDP), a_i represents the i-th action selected in state s. In this paper, the action set is set to different heuristic operators, as shown in Table 3. During each iteration, the agent will select different heuristic operators to obtain corresponding rewards, with the reward design aimed at accelerating the search process and achieving optimization goals. As shown in Table 3, when an action a_n is chosen from the action set, the agent will execute the corresponding heuristic operators. Secondly, a reward function with crossover probability should be designed to assess whether the value of the chosen action is reasonable; under the same algorithm, different reward functions may yield different results. The setting of the reward function determines the convergence speed and efficiency of the algorithm. The reward function in this paper is defined by the following equation:

$$R_{action} = \begin{cases} -1, \ Fit_{best}\left(solution_i^{t+1}\right) - Fit_{best}\left(solution_i^{t}\right) \leq 0 \\ 1, \ Fit_{best}\left(solution_i^{t+1}\right) - Fit_{best}\left(solution_i^{t}\right) > 0 \end{cases}$$

The reward function R_{action} represents the reward obtained for taking action a in states. $Fit_{best}(solution_i^{t+1})$ denotes the best fitness value of the i-th solution in the t + 1-th episode, $Fit_{best}(solution_i^{t})$ denotes the best fitness value of the i-th solution in the t-th episode. When the best fitness value of the i-th solution in the (t + 1)-th episode is greater than that in the t-th episode, the reward is a positive value. Conversely, the reward will become positive with probability p and remain negative with probability $1 - p$. The distinction between positive and negative values is that actions corresponding to positive rewards will increase in frequency, while those corresponding to negative rewards will decrease. This involves the policy function π, which we will introduce in the next section.

Table 3. The action set in MDP.

Actions	Heuristic Operators
a_1	Destroy operation 1, repair operation 1, swap 1
a_2	Destroy operation 1, repair operation 1, swap 2
a_3	Destroy operation 1, repair operation 2, swap 1
a_4	Destroy operation 1, repair operation 2, swap 2
a_5	Destroy operation 2, repair operation 1, swap 1
a_6	Destroy operation 2, repair operation 1, swap 2
a_7	Destroy operation 2, repair operation 1, swap 1
a_8	Destroy operation 2, repair operation 2, swap 1

2.2.3. Policy Function

In this paper, the ε-greedy policy is chosen as the action selection strategy. The ε-greedy policy is a method that balances exploration and exploitation. Specifically, it selects the current optimal action with a probability of $1 - \varepsilon$ and selects an action randomly with a probability of ε. In this paper, the ε-greedy policy can be represented by the following equation. ε represents the exploration rate, $|A|$ represents the size of the action set, $Q(s, a)$ denotes the Q-value of taking action a in state s, and a^* represents the optimal action in state s, that is, $a^* = \text{argmax}_{a \in \text{actions}} Q(s, a)$. We use the indicator function $I(x)$ to represent whether certain conditions are met.

$$P(s, a) = (1 - \varepsilon) * I(a = a^*) + \varepsilon * \left(\frac{1}{|A|} \sum_{a' \in A} I(a = a') \right)$$

2.3. Hyper-Heuristic Based on CR-Q-Learning

This section delves into the hyper-heuristic based on CR-Q-learning (HHCRQL). We start by outlining the core components of the high-level heuristic (HLH), specifically the CR-selection strategy. Then, we describe the main logical flow of the CR-Q-learning process.

2.3.1. CR-Selection Strategy

We introduce a cumulative-reward credit-assignment mechanism that jointly accounts for the feasibility and quality of the current solution, while employing an ε-greedy strategy to select heuristic operators (HOs). Although ε-greedy is effective for low-level heuristic (LLH) selection, its credit-allocation rule remains simplistic. Cumulative rewards evaluate HO efficacy comprehensively through two performance indicators—applicability and feasibility—thereby not only identifying superior HOs but also intermittently admitting alternative HOs to sustain population diversity. Integrating ε-greedy with cumulative-reward credit allocation naturally balances the production of high-quality individuals with the preservation of population diversity. This cumulative philosophy is realized via a two-phase reward-function update: immediately after an LLH operation and again when determining whether the updated individual outperforms its predecessor. These updates accumulate, progressively refining action selection and contracting the solution space.

2.3.2. HHCRQL Algorithm Procedure

The design and selection of heuristic operators are crucial to the performance of the algorithm. Effective heuristic operators can significantly enhance the efficiency and quality of solutions. Many heuristic algorithm designs involve a variety of operators, such as swap, mutation, destruction, merge, and shake operators. The role of the swap operator, for example, is to exchange nodes in a route or variables that need to be swapped. Heuristic algorithms typically fix the sequence of several operators, starting from an initial solution and continuously operating until certain conditions are met. The advantage of reinforcement learning is its ability to learn and dynamically adjust the order of operators for optimal results. Through Q-learning, initial states are selected randomly, different actions are taken, and each action yields a corresponding reward. At this point, the time step ends, and the updated Q-table is fed back to the agent. This Q-table plays a role in feedback and learning for the next time step. The agent will dynamically learn and adjust based on the rewards corresponding to actions in the Q-table for the next time step. If the reward is positive, the selection of actions by the corresponding heuristic operators will be strengthened; if negative, it will be weakened accordingly. Continuously obtaining states, executing actions with high rewards, updating the Q-table, and dynamically adjusting constitute the process of reinforcement learning. The pseudocode for the HHCRQL is shown below (Algorithm 1).

Algorithm 1 HHCRQL

HHCRQL Algorithm parameters: number_episodes (max_e), max_step (max_s), episode (e), step (s), Q table, state set (S), action set (A), learning rate (alpha), exploration rate (epsilon), discount factor (gamma).
1. Initial solution generation
2. Initialize the Q table
3. Input: algorithm parameters to HHCRQL Algorithm
4. while episode (e) < max_e do
5. s_i: choose state (F, V, T_r, U)
6. for time step (t) in max_step (max_s)
7. a_i choose action according to ε-greedy
8. Execution: executes the selected action a_i based on the s_i
9. F_f^e: Determination of the feasibility of the current time step
10. if feasibly:
11. r^e: calculate the reward
12. F_v^e: Fitness value calculation of the current time step
13. if better F_v^e
14. r^e: accrue reward and update Q table
15. else: no accrue reward
16. else: give punishment in reward and update Q table
17. t = t + 1
18. END While: satisfy stopping criterion
19. Output: optimal solution and Q table

3. Problem Description and Modeling

In this section, we specifically analyze the concrete elements of this real-world problem and position it as the Heterogeneous Multi-Trip Vehicle Routing Problem with Time Windows and Time-Varying Network (HMTVRPTW-TVN). We have analyzed and established

a corresponding mixed-integer linear programming (MILP) model that can be adapted for solution by any commercial solver.

3.1. Problem Description

The vehicle dispatch system receives reports from various outlets (warehouses) every day at dawn, detailing the pickup and delivery volumes for the day. Dispatchers arrange drivers to depart from the distribution center, known as the super-hub, to execute transportation tasks. To fulfill the daily transportation tasks, the company has established a fleet for cargo transportation. The tasks vary daily, and at times, it is necessary to engage crowd-sourced vehicles to assist with transportation duties. Due to the fleet's characteristics, there are different models of vehicles and various types of drivers. One type is directly employed by the company, and the other is outsourced. When assigning tasks, efforts are made to balance the workload of drivers. The locations of the warehouses are fixed, and their working time windows are also set, so the vehicle dispatch system schedules and routes the heterogeneous fleet for multiple departures at appropriate times(as shown in Figure 7): departing from the super-hub's parking lot, executing a series of transportation tasks according to the dispatch orders, and returning to the super-hub parking lot. The challenge lies in determining the right time to depart, avoiding morning and evening peak traffic while arriving within the working time window of the warehouses to load and unload. Therefore, this problem can be formally defined as an optimization problem, known as the Heterogeneous Multi-Trip Vehicle Routing Problem with Time Windows and Time-Varying Network (HMTVRPTW-TVN). In a directed network $G = (V, E)$, where V is the set of nodes, including a distribution center D and a set of pickup warehouses C, and E is the set of arcs, corresponding to the connections between nodes. For each arc $(i, j) \in E$, there is an associated travel time t_{ij}. Each warehouse $C \in E$ has a working window $[ST_i, ET_i]$ and a break window $[PT_i, UT_i]$. Each vehicle's driver is fixed, and the driver's working time window is $[rT_i, RT_i]$.

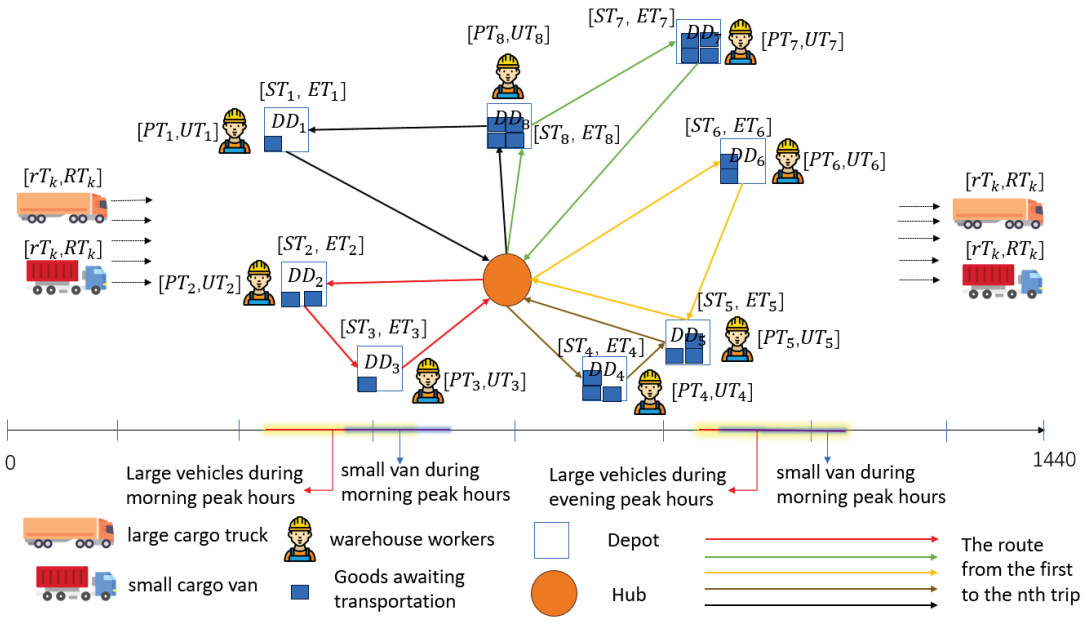


Figure 7. An example of vehicle itineraries in the HMTVRPTW-TVN.

3.2. Parameter and Variable Definitions

We define the notation and variables for HMTVRPTW-TVN. Table 4 lists the modeling variables, and Table 5 lists all notations along with the relevant parameters used.

Table 4. Variables applied in the model.

Variables	Definition
$t_{i,j,k,q}^{actual}$	The actual travel time of small cargo van k from point i to point j
x_{ij}^{kq}	The binary variable $x_{i,j}^{k,q}$ equals 1 if vehicle k travels along arc (i,j) during its q_th trip; otherwise, it equals 0.
S_i^{kq}	The arrival time of vehicle k at node i during its q_th trip.
d_i^{kq}	The amount of goods picked up by vehicle k at node i during its q_th trip
$w_i^{k,q}$	The waiting time of vehicle k at node i during its q_th trip
Z^{kq}	The binary variable equals 1 if vehicle k makes a q_th departure; otherwise, it equals 0.
TL^{kq}	The total load of Vehicle k on trip q
$a_i^{k,q}$	Lunch break window auxiliary variable
$b_i^{k,q}$	Lunch break window auxiliary variable.
$y_{i,j,k,q}^{1}$	Small cargo van morning peak period indicator (auxiliary variable)
$y_{i,j,k,q}^{2}$	Small cargo van morning peak period indicator (auxiliary variable)
$z_{i,j,k,q}$	Linearization auxiliary variables

Table 5. Parameters applied in the model.

Parameters	Definition
$t_{i,j}^{orig}$	Normal travel time from node i to node j.
$t_{i,j}^{small}$	Peak period travel time from node i to node j.
e_i	The loading and unloading capacity of warehouse i.
$DT_i^{k,q}$	Departure time of vehicle k at node i during its q_th trip.
FT_i	The fixed working hours of warehouse i can be denoted as FT_i.
c_k	The loading capacity of vehicle k can be denoted as c_k
ST_i	The lower bound of the working time window for warehouse i.
ET_i	The upper bound of the working time window for warehouse i.
PT_i	The lower bound of the worker's rest time window for warehouse i.
UT_i	The upper bound of the worker's rest time window for warehouse i.
rT_k	The lower bound of the driver's working time window for vehicle k.
RT_k	The upper bound of the driver's working time window for vehicle k.
DD_i	The demand at warehouse i can be denoted as DD_i
Q_k	The number of times vehicle k is used.
C	The set of warehouses
D	The distribution center
D_s	The origin distribution center
D_e	The destination distribution center
M	The infinitely large constant
V	The set of all nodes
K	The set of vehicles can be denoted as K
E	The set of arcs defined by (i,j) can be denoted as E
q	The number of vehicle departures can be denoted as q
A	The number of times a vehicle can be used
K_s	The set of small cargo van can be denoted as S.
K_l	The set of large cargo truck can be denoted as L
$[LT_m, LT_n]$	The morning restricted time window for small cargo van
$[LT_\gamma, LT_\delta]$	The evening restricted time window for large cargo truck
$[TP_m, TP_n]$	The morning peak time window for small cargo van
$[TP_\gamma, TP_\delta]$	The evening peak time window for large cargo truck

3.3. Construction of HMTVRPTW-TVN

Since the information (vehicle information, warehouse information, pending pickup information, working time window information, etc.) is updated through daily morning

reports, this information can be considered as known. The HMTVRPTW-TW problem formulation is as follows:

1. Objective function consisting of two components, with cost calculation varying by vehicle type:

$$Minimize\ z = \left(obj_{large} + obj_{small}\right) \tag{1}$$

$$obj_large = \sum_i^V \sum_j^V \sum_k^K \sum_q^Q \left[t_{i,j}^{orig} x_{i,j}^{k,q} + FT_j x_{i,j}^{k,q} + x_{i,j}^{k,q} d_j^{k,q}/e_j + w_j^{k,q}\right] \tag{2}$$

$$obj_small = \sum_i^V \sum_j^V \sum_k^K \sum_q^Q \left[t_{i,j}^{actul} x_{i,j}^{k,q} + FT_j x_{i,j}^{k,q} + x_{i,j}^{k,q} d_j^{k,q}/e_j + w_j^{k,q}\right] \tag{3}$$

Subject to the following:

2. Capacity and demand constraints:

$$\sum_j^c d_j^{k,q} \le Z^{k,q} * c_k \quad \forall k \in K, \quad \forall q \in Q \tag{4}$$

$$\sum_k^K \sum_q^q \sum_i^V x_{i,j}^{k,q}\ d_j^{k,q} + M\left(Z^{k,q} - 1\right) \le DD_j \quad \forall j \in C \tag{5}$$

$$TL^{kq} = \sum_j^c d_j^{k,q} \quad \forall k \in K, \ \forall q \in Q \tag{6}$$

$$d_{D_e}^{k,q} = TL^{kq} \quad \forall k \in K, \ \forall q \in Q \tag{7}$$

3. Flow conservation constraints:

$$\sum_j^V x_{D_s,j}^{k,q} = \sum_i^V x_{i,D_e}^{k,q} = Z^{kq} \ \forall k \in K, \quad \forall q \in Q \tag{8}$$

$$\sum_j^V x_{j,D_s}^{k,q} = \sum_J^V x_{D_e,j}^{k,q} = 0 \quad \forall k \in K, \quad \forall q \in Q \tag{9}$$

$$\sum_j^V x_{i,j}^{k,q} = \sum_j^V x_{j,i}^{k,q} \ \forall k \in K, \forall i \in V \quad \forall q \in Q \tag{10}$$

4. Vehicle task execution sequence constraints:

$$S_j^{k,q} \ge S_i^{k,q} + w_i^{k,q} + FT_i + \delta_i/e_i + t_{i,j}^{k,q} - M\left(1 - x_{i,j}^{k,q}\right) \forall i \in V, \forall j \in V, \forall k \in K, \forall q \in Q \tag{11}$$

$$\delta_i = \begin{cases} TL^{kq}, if\ \ i = 0 \\ d_i^{k,q},\ others \end{cases} \forall i \in V, \forall j \in V, \forall k \in K, \forall q \in Q \tag{12}$$

$$t_{i,j}^{k,q} = \begin{cases} \mathrm{t}_{\mathrm{i,j}}^{\mathrm{orig}}, & k \in K_l \\ t_{i,j,k,q}^{actual}, & k \in K_s \end{cases} \forall i \in V, \forall j \in V, \forall k \in K, \forall q \in Q \tag{13}$$

5. Peak period constraints for small vehicles:

$$DT_i^{k,q} = S_i^{k,q} + w_i^{k,q} + FT_i + \delta_i/e_i \tag{14}$$

$$DT_i^{k,q} \ge LT_m - M(1 - \mathrm{y}_{\mathrm{i,j,k,q}}^{1}) \quad \forall i \in V, \forall j \in V, \forall k \in K, \forall q \in Q \tag{15}$$

$$DT_i^{k,q} \le LT_n + M(1 - \mathrm{y}_{\mathrm{i,j,k,q}}^{1}) \quad \forall i \in V, \forall j \in V, \forall k \in K, \forall q \in Q \tag{16}$$

$$DT_i^{k,q} \ge TP_m - M(1 - \mathrm{y}_{\mathrm{i,j,k,q}}^{2}) \quad \forall i \in V, \forall j \in V, \forall k \in K, \forall q \in Q \tag{17}$$

$$DT_i^{k,q} \le TP_n + M(1 - \mathrm{y}_{\mathrm{i,j,k,q}}^{2}) \quad \forall i \in V, \forall j \in V, \forall k \in K, \forall q \in Q \tag{18}$$

$$t_{i,j,k,q}^{actual} = t_{i,j}^{small}(\mathrm{y}_{\mathrm{i,j,k,q}}^{1} + \mathrm{y}_{\mathrm{i,j,k,q}}^{2}) + t_{i,j}^{orig}(1 - \mathrm{y}_{\mathrm{i,j,k,q}}^{1} - \mathrm{y}_{\mathrm{i,j,k,q}}^{2}) \quad \forall i \in V, \forall j \in V, \forall k \in K, \forall q \in Q \tag{19}$$

$$z_{i,j,k,q} = t_{i,j,k,q}^{actual} * x_{i,j}^{k,q} \quad \forall i \in V, \forall j \in V, \forall k \in K, \forall q \in Q \tag{20}$$

6. Peak period constraints for large vehicles:

$$DT_{i}^{k,q} \leq LT_{\gamma} + M(1 - \mathrm{a}_{\mathrm{i,j,k,q}}^{1}) \quad \forall i \in V, \forall j \in V, \forall k \in K, \forall q \in Q \tag{21}$$

$$DT_{i}^{k,q} \geq LT_{\delta} - M(1 - \mathrm{a}_{\mathrm{i,j,k,q}}^{2}) \quad \forall i \in V, \forall j \in V, \forall k \in K, \forall q \in Q \tag{22}$$

$$\mathrm{a}_{\mathrm{i,j,k,q}}^{1} + \mathrm{a}_{\mathrm{i,j,k,q}}^{2} \geq x_{i,j}^{k,q} \quad \forall i \in V, \forall j \in V, \forall k \in K, \forall q \in Q \tag{23}$$

$$t_{i,j}^{arr} = DT_{i}^{k,q} + t_{i,j}^{orig} \quad \forall i \in V, \forall j \in V, \forall k \in K, \forall q \in Q \tag{24}$$

$$t_{i,j}^{arr} \leq TP_{\gamma} + M(1 - a_{i,j,k,q}^{3}) \quad \forall i \in V, \forall j \in V, \forall k \in K, \forall q \in Q \tag{25}$$

$$t_{i,j}^{arr} \leq TP_{\delta} + M(1 - a_{i,j,k,q}^{4}) \quad \forall i \in V, \forall j \in V, \forall k \in K, \forall q \in Q \tag{26}$$

$$a_{i,j,k,q}^{1} + a_{i,j,k,q}^{2} \geq x_{i,j}^{k,q} \quad \forall i \in V, \forall j \in V, \forall k \in K, \forall q \in Q \tag{27}$$

7. Warehouse worker rest time-window constraints:

$$S_{j}^{k,q} + w_{j}^{k,q} \geq PT_{i} - M(1 - \mathrm{a}_{\mathrm{j}}^{\mathrm{k,q}}) \quad \forall j \in C, \forall k \in K, \forall q \in Q \tag{28}$$

$$S_{j}^{k,q} + w_{j}^{k,q} \leq UT_{j} + M(1 - \mathrm{b}_{\mathrm{j}}^{\mathrm{k,q}}) \quad \forall j \in C, \forall k \in K, \forall q \in Q \tag{29}$$

$$\mathrm{a}_{\mathrm{j}}^{\mathrm{k,q}} + \mathrm{b}_{\mathrm{j}}^{\mathrm{k,q}} \geq 1 \quad \forall j \in C, \forall k \in K, \forall q \in Q \tag{30}$$

8. Driver working time-window constraints:

$$rT_{k} \leq S_{D_s}^{k,q} \leq RT_{k} \quad \forall k \in K, \forall q \in Q \tag{31}$$

9. Trip count limitations:

$$\sum_{q}^{Q} Z^{k,q} \leq |Q| \quad \forall k \in K \tag{32}$$

10. Trip sequence limitations:

$$Z^{k,q} \leq Z^{k,q-1} \quad \forall k \in K, \quad q \geq 1 \tag{33}$$

11. Warehouse operating time-window constraints:

$$ST_{k} \leq S_{i}^{k,q} \leq ET_{k} \quad \forall k \in K, \ \forall j \in C, \forall q \in Q \tag{34}$$

12. Other constraints (non-zero loading, inter-trip time coupling, and binary variable declarations):

$$d_{j}^{k,q} \leq M\sum_{i}^{V} x_{i,j}^{k,q} \quad \forall k \in K, \forall j \in C, \forall q \in Q \tag{35}$$

$$S_{D_e}^{k,q} + w_{D_e}^{k,q} + FT_{D_e} + d_{D_e}^{k,q}/e_{D_e} \leq S_{D_s}^{k,q+1} + M(1 - Z^{k,q}) \ \forall k \in K, q < |Q| - 1 \tag{36}$$

$$x_{i,j}^{k,q} = \begin{cases} 0: & \textit{others} \\ 1: \textit{if vehicle k travels along arc } (i,j) \textit{ during its } q-th \textit{ trip} \end{cases} \tag{37}$$

$$Z^{k,q} = \begin{cases} 0: & others \\ 1: & if\ vehicle\ k\ makes\ the\ q-th\ trip \end{cases} \quad (38)$$

4. Numerical Experiments

In this section, we conduct numerical experiments using the R-series data from the Solomon instances to validate the applicability of the proposed algorithm. Comparative experiments are carried out, where the results are compared with those obtained by other algorithms or commercial solvers to verify the effectiveness of both the model and the algorithm. Finally, the HHCRQL algorithm is applied to solve problems of varying scales, and the results are compared with these methods to demonstrate the practicality and high performance of the algorithm.

4.1. Solution Results Based on Real Cases

4.1.1. Real-Case Analysis

This section conducts numerical experiments using real data from a distribution center of a Shanghai-based supply chain company to evaluate the performance of the HHCRQL algorithm. For convenience in the experiments, the distribution center is labeled as sup-hub 0, and the warehouses are numbered from 1 to 13. Detailed data for each warehouse are provided in Appendix A.

Loading at the warehouses is conducted in pallets, with the loading rate measured in cubic meters per minute. Vehicles are numbered from 1 to 12, with vehicles 1–4 being large vehicles measuring 17.5 m in length and having driver working time windows of [330–1500]. Vehicles 5–12 are smaller, measuring 9.6 m in length, with driver working time windows of [360–1410].

4.1.2. Real-Case Study and Result Analysis

To comprehensively evaluate the model and algorithm, this study further compares them with the company's actual scheduling plans. Using total time, warehouse waiting time, and vehicle utilization as evaluation metrics, Table 6 shows that the solutions obtained by our algorithm outperform the company's actual dispatch center in both total time cost and warehouse waiting time. Furthermore, due to a more balanced workload distribution, more vehicles are utilized, allowing large vehicles to complete their tasks and finish shifts earlier, thereby avoiding situations where drivers of large vehicles work more while drivers of small vehicles work less. As the number of warehouses increases, the performance of the HHCRQL solutions improves further; when involving 10 warehouses, waiting time is reduced by 35.36% and total operation time decreases by 24.68%. Compared with the actual scheduling plan, the HHCRQL algorithm provides higher-quality solutions, demonstrating its superiority. The scheduling arrangements are presented in Appendix A.

Table 6. Optimization analysis of actual scheduling results.

Instance	Actual Solution (AS)			Solution Derived from HHCRQL (HS)			GAP	
Number of Warehouses	Total Times	Waiting Times	Car Number	Total Times	Waiting Times	Car Number	Gap_1	Gap_2
4	6944	1587	10	6024	1185	12	13.25%	25.33%
5	7481	1784	12	6345	1298	13	15.19%	27.24%
6	7854	1978	15	6587	1445	17	16.13%	26.95%
7	8120	2145	17	6976	1554	19	14.09%	27.55%
8	8478	2215	19	7359	1798	20	13.20%	18.83%

Table 6. *Cont.*

Instance	Actual Solution (AS)			Solution Derived from HHCRQL (HS)			GAP	
Number of Warehouses	**Total Times**	**Waiting Times**	**Car Number**	**Total Times**	**Waiting Times**	**Car Number**	**Gap_1**	**Gap_2**
9	8752	2354	19	7640	1851	21	12.71%	21.37%
10	9352	2524	20	7640	1904	22	18.31%	24.56%
11	9531	2710	20	7637	1844	22	19.87%	31.96%
12	9778	3041	20	7534	2045	23	22.95%	32.75%
13	9965	3278	24	7506	2119	24	24.68%	35.36%

$Gap_1 = (\text{Total time}_{AS} - \text{Total time}_{HS})/\text{Total time}_{AS} \times 100\%$. $Gap_2 = (\text{Waiting time}_{AS} - \text{Wotal time}_{HS})/\text{total time}_{AS} \times 100\%$.

4.2. Algorithm and Model Validity

In this section, we validate both the model and the algorithm. The results indicate that the model proposed in this paper satisfies all constraints and achieves a satisfactory gap when compared with actual operational schedules. The HHCRQL algorithm, while highly innovative, also yields commendable results.

4.2.1. Benchmark Methods

To verify the performance of the proposed algorithm and the effectiveness of the model, we used the commercial solver Gurobi to solve the model, obtaining valid results that serve as one of the benchmark methods. Other benchmark algorithms include LNS (Large Neighborhood Search), an iterative heuristic algorithm that explores a large search space by partially destroying and repairing solutions; ILS (Iterated Local Search), an iterative improvement algorithm that repeatedly applies local search combined with perturbation to escape local optima; and LNSSA (Large Neighborhood Search with Simulated Annealing Acceptance), a probabilistic metaheuristic that accepts worse solutions with a certain probability to avoid being trapped in local optima.

4.2.2. Parameter Settings

Numerical experiments are conducted to compare the benchmark methods with the proposed HHCRQL algorithm. The experiments involve warehouse cases of different scales, primarily differing in warehouse demand and the number of warehouses. The location of each warehouse is adopted from the Solomon R-series instances, and the demand for each warehouse is randomly generated within the range [0–200]. "number of warehouses: [a,b]" means that the number of warehouses is randomly selected within the interval [a,b]. The base settings for our algorithms are Gurobi Optimizer version 11.0.3 build v11.0.3rc0 (win64—Windows 11.0 (22631.2)), with a CPU model of the 12th Gen Intel (R) Core (TM) i5-12400F. The parameters of the algorithms are in Appendix A.

Based on the characteristics of the problem and company requirements, we use total program running time and total scheduling time (the objective value of MIPL) as evaluation criteria. Each experiment is repeated ten times to calculate an average value, to mitigate the impact of occasional outliers.

According to the data in Table 7, in small-scale instances, Gurobi can precisely solve the model proposed in Section 2 and obtain the optimal solution within an acceptable time, thereby verifying the correctness of the model. However, when the number of customer nodes exceeds 15, the commercial solver fails to obtain an optimal solution within the limited time frame (3600 s). The solutions obtained by the proposed HHCRQL algorithm consistently remain within 3% of Gurobi's optimal solutions, fully demonstrating its superior solution quality. Moreover, HHCRQL consistently outperforms LNS and LNSSA,

and the performance gap widens as the number of customer nodes increases, reaching its maximum in scenarios with 30–40 customer nodes. In addition, the proposed HHCRQL algorithm not only achieves excellent solution quality but also demonstrates stable running time, which is shorter than that of both LNS and LNSSA.

Table 7. Small-scale calculation results.

Instance	The Algorithm of the Solution										
Number of Warehouses	Gurobi*		LNS*		LNSSA*		HHCRQL*		Gap_3	Gap_4	Gap_5
	Obj(s)	CPU(s)	Obj(s)	CPU(s)	Obj(s)	CPU(s)	Obj(s)	CPU(s)	-	-	1
[0–5]	910	193.6	1009	0.58	972	0.85	932	0.39	−2.36%	8.26%	4.29%
[5–10]	1846	1635.5	2034	2.52	1938	0.85	1869	0.66	−1.25%	8.83%	3.69%
[10–15]	4291	2859.7	4542	16.62	4501	23.54	4265	0.82	0.61%	6.49%	5.53%
[15–20]	-	-	5079	18.41	4872	43.20	4661	1.21	-	8.97%	4.53%
[20–25]	-	-	7112	49.02	6895	68.52	6589	1.70	-	7.94%	4.64%
[25–30]	-	-	7616	63.24	7469	79.97	7032	2.15	-	8.30%	6.21%
[30–35]	-	-	11,839	314.52	11,252	352.3	10,550	2.97	-	12.22%	6.65%
[35–40]	-	-	13,859	561.3	13,489	622.4	12,227	3.85	-	13.35%	10.32%
[40,45]	-	-	14,922	744.2	14,525	754.0	13,149	4.03	-	13.48%	10.46%
[45,50]			16,524	1454.2	16,245	1604.2	14,895	4.35	-	10.94%	9.06%

$Gap_3 = (Obj_{gurobi^*} - Obj_{HHCRQL^*})/Obj_{HHCRQL^*} \times 100\%$. $Gap_4 = (Obj_{LNS^*} - Obj_{HHCRQL^*})/Obj_{HHCRQL^*} \times 100\%$. $Gap_5 = (Obj_{LNSSA^*} - Obj_{HHCRQL^*})/Obj_{HHCRQL^*} \times 100\%$.

Table 8 shows that, for large-scale instances, the HHCRQL algorithm significantly outperforms LS, LNS, and LNSSA in terms of solution quality. Specifically, compared with ILS, the maximum gap of HHCRQL is reduced by 9.17%, and as the number of customer nodes increases, the gap remains around 3.5%; compared with LNS, the maximum gap is reduced by 6.74%, stabilizing around 2.1% as customer nodes increase; compared with LNSSA, the maximum gap is reduced by 5.95%, stabilizing around 1.5% with increasing customer nodes. For both small- and large-scale instances, the computational time of Gurobi, LNS, and LNSSA increases significantly with the number of customer nodes, whereas HHCRQL maintains relatively stable runtime. Therefore, HHCRQL not only demonstrates superior solution quality but also exhibits higher practical applicability for large-scale instances, confirming the effectiveness and robustness of the proposed algorithm.

Table 8. Large-scale calculation results.

Instance	The Algorithm of the Solution										
Number of Warehouses	ILS		LNS		LNSSA		HHCRQL		Gap_6	Gap_7	Gap_8
	Obj(s)	CPU(s)	Obj(s)	CPU(s)	Obj(s)	CPU(s)	Obj(s)	CPU(s)	-	-	1
[50,55]	17,366	1595	17,058	1952	16,854	2118	15,908	4.36	9.17%	6.74%	5.95%
[55,60]	18,725	2077	18,450	2105	18,275	2351	17,652	4.78	6.08%	4.33%	3.53%
[60,65]	19,625	2185	19,300	2350	19,216	2745	18,739	5.41	4.73%	2.91%	2.55%
[65,70]	21,482	2215	21,173	2451	21,102	2785	20,475	5.87	4.92%	3.30%	2.57%
[70,75]	22,041	2237	21,763	2841	21,630	2965	21,037	5.69	4.77%	3.34%	2.82%
[75,80]	23,425	2269	23,156	2894	22,931	3102	22,520	5.47	4.02%	2.75%	1.83%
[80,85]	24,495	2154	24,207	2734	23,997	2941	23,695	4.91	3.38%	2.12%	1.27%
[85,90]	25,954	2198	25,633	2965	25,413	3210	25,124	5.95	3.30%	1.99%	1.15%
[90,95]	27,658	2184	27,258	2936	27,137	3354	26,710	5.56	3.55%	2.01%	1.60%
[95,100]	28,965	2236	28,496	2812	28,386	3247	27,874	5.72	3.91%	2.18%	1.84%

$Gap_6 = \left(Obj_{ILS} - Obj_{HHCRQL}\right)/Obj_{HHCRQL} \times 100\%$. $Gap_7 = \left(Obj_{LNS} - Obj_{HHCRQL}\right)/Obj_{HHCRQL} \times 100\%$. $Gap_8 = (Obj_{LNSSA} - Obj_{HHCRQL})/Obj_{HHCRQL} \times 100\%$.

4.3. Convergence and Parameter Sensitivity Analysis

To validate the theoretical soundness and practical robustness of the proposed Q-learning variant with cumulative reward mechanism, we conducted a series of experiments

focused on algorithm convergence behavior and sensitivity to key parameters, particularly the learning rate.

4.3.1. Convergence Analysis

Due to structural differences among the compared algorithms, the Q-learning algorithm involves multiple time steps within each training episode (iteration), whereas heuristic algorithms such as LNS and LNSSA update the solution only once per iteration. To ensure comparability of the convergence processes across different algorithms in the plot, each Q-learning time step is treated as a single iteration unit, to ensure comparability when plotting the cumulative reward curve. Furthermore, to eliminate differences in initial objective values or reward magnitudes among the algorithms, all iterative curves are normalized by mapping the vertical axis of each curve to the [0,1] interval. This design ensures a unified scale on the horizontal axis and comparability on the vertical axis, allowing readers to intuitively assess the convergence speed and performance of each algorithm. In addition, the curves labeled 'Reward Feasible' and 'Reward Improvement' represent variants in which the cumulative reward mechanism is removed; that is, rewards are assigned only when a feasible solution is obtained or when an improvement over the current best solution occurs. Including these variants allows for a clear comparison of the effect of the cumulative reward mechanism on convergence behavior. Figure 8 presents the normalized iterative convergence curves of all algorithms. It is evident that the proposed method consistently converges to superior solutions across multiple runs, with faster convergence and smaller variance. This demonstrates that the cumulative reward mechanism effectively guides the learning process, enhancing both stability and solution quality.

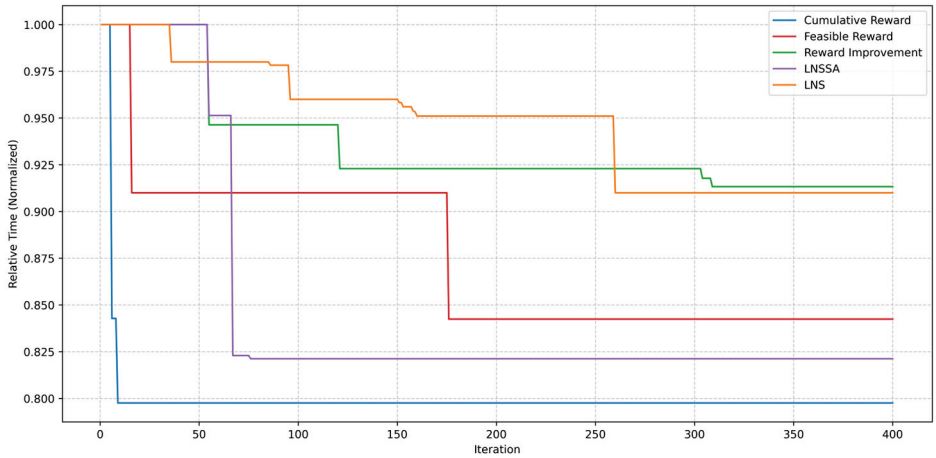


Figure 8. The iteration curves of multiple algorithms (relative).

4.3.2. Learning Rate Sensitivity

To analyze the impact of the learning rate α on the algorithm's performance, we varied α within the set {0.1, 0.3, 0.5, 0.7, 0.9} while keeping other parameters unchanged. Each α value was tested in 10 independent experiments, and the final average objective value was recorded.

As shown in Figure 9, the learning rate significantly influences convergence behavior. A too-small α leads to slow learning and long convergence times; however, this does not mean that a larger α is always better. Experimental results indicate that $\alpha = 0.5$ and $\alpha = 0.7$ achieve a good balance between convergence speed and solution quality, providing practical guidance for parameter tuning in similar problems.

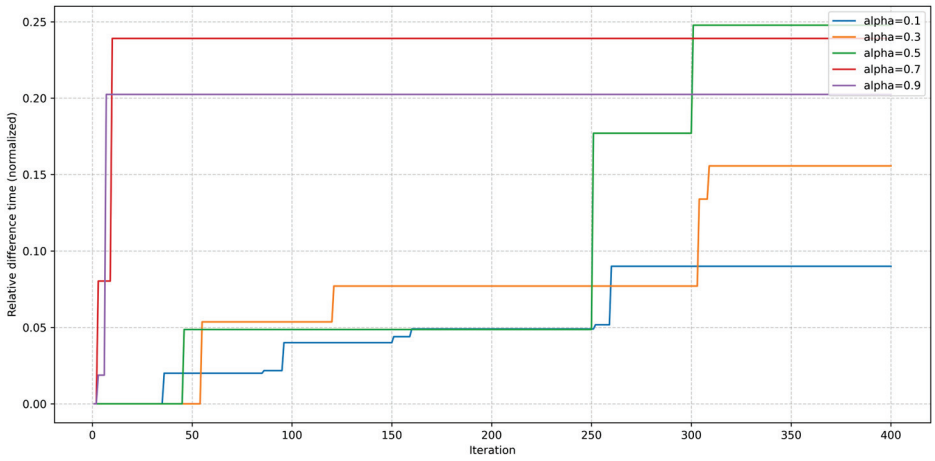


Figure 9. The impact of learning rate α on convergence behavior (relative).

4.3.3. Theoretical Discussion

The classic Q-learning algorithm is proven to converge to the optimal policy under conditions including sufficient exploration of the state-action space and appropriate learning rate schedules. In our approach, the cumulative reward mechanism maintains the Markov property and preserves the theoretical convergence guarantees by:

Employing an ε-greedy exploration strategy to ensure all actions are explored adequately.

Selecting a fixed learning rate within a theoretically justified range to balance learning stability and adaptability.

While the cumulative reward design adds complexity, it aligns with the framework of discounted reward MDPs, and empirical results substantiate its effectiveness without sacrificing convergence properties.

4.4. Analysis of Driver Workload Balance

In addressing the balance of drivers' workloads, we propose a simple empirical mechanism. Based on historical data and experience, we set a limit that a driver should not exceed for the number of tasks, known as A. We will now analyze this parameter further.

When workload balance is not taken into account, the presence of a heterogeneous fleet leads to two scenarios across different scales: Scenario 1: With ample time, all high-capacity vehicles are continuously in operation, which means the drivers have no rest. The advantage is that transportation tasks can be completed more quickly. Scenario 2: Due to an overwhelming demand, even with all high-capacity vehicle drivers working continuously, it is impossible to complete all transportation tasks within the time limit. We must mobilize other vehicles to ensure task completion. After testing, if we consider the constraint of driver workload balance, in small-scale cases, each driver can reduce their working hours by an average of 2.88%, and in the other scenario, by an average of 3.98%. Reduced working hours imply that drivers will be more passionate and creative at work, as shown in Figure 10. Reduced working hours imply that drivers will be more passionate and creative at work. In this paper, we use a simple empirical mechanism for driver workload balance, stipulating that each driver should not exceed three departures while prioritizing the use of larger vehicles to ensure transportation efficiency. We have questioned this empirical mechanism and sought to determine the optimal number of departures, A, in different scenarios. After calculation, it was found that setting A to 3 always yields acceptable overall performance. It is not difficult to see that A can be represented by the following formula. Let A represent the actual daily travel time, excluding any time-window constraints, and

denote the average time required to complete a single trip. Figure 11 illustrates the values of A under two different scales.

$$A = {}^{T_{ac}}/_{T_{at}}$$

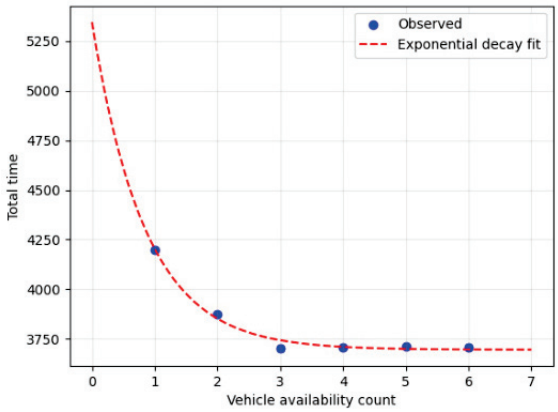


Figure 10. The impact of workload balance on target values.

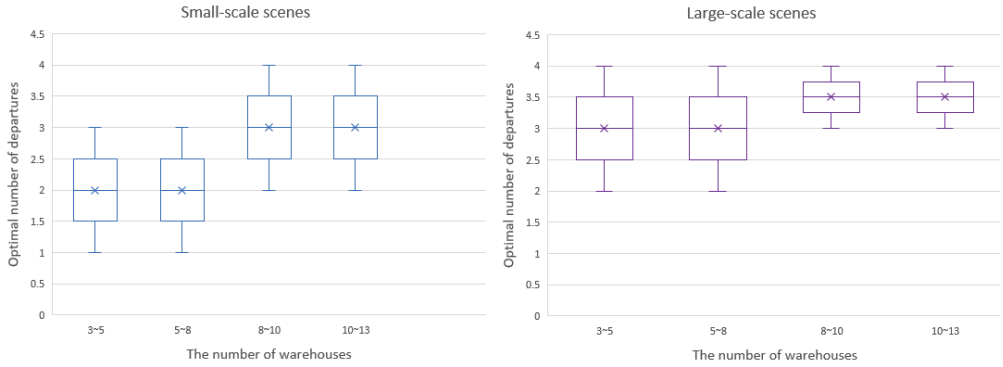


Figure 11. Optimal driver workload across different scales.

4.5. Fleet-Composition Analysis for a Heterogeneous Fleet

All large-capacity vehicles are currently company-owned, whereas all small-capacity vehicles are crowd-sourced. Fixing the total fleet size at its present maximum, we evaluate the current scale and composition. As shown in Figure 12, extreme strategies—either a fully crowd-sourced or a fully owned fleet—are found to incur markedly higher total time costs. This outcome stems from the negative impact that a bimodal time-varying (BTV) network exerts on any single vehicle type, thereby confirming the necessity of a heterogeneous fleet for efficient distribution.

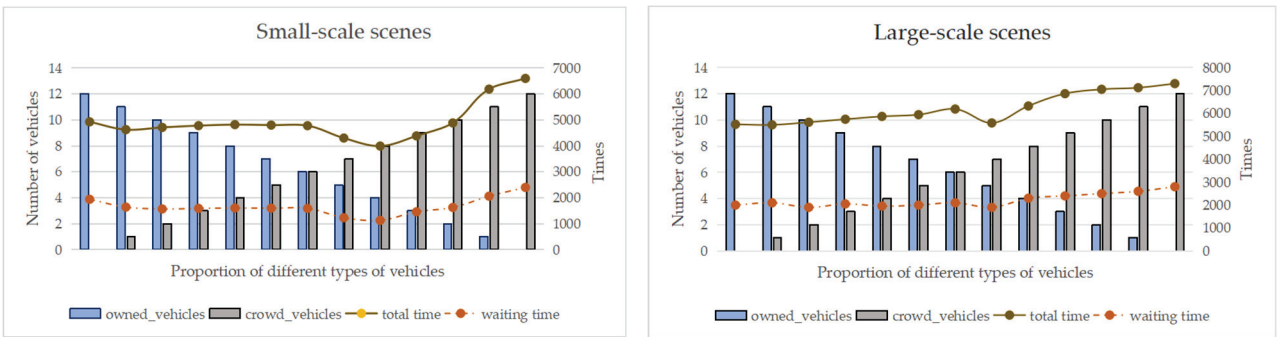


Figure 12. Heterogeneous fleet-composition analysis.

Further observation reveals that, under small-scale operations, the total distribution completion time initially decreases and then increases as the proportion of crowd-sourced vehicles rises, reaching its minimum when this proportion is approximately two-thirds. In contrast, under large-scale conditions, the same proportion exhibits an opposite pattern: total distribution time first increases, then decreases, attaining its minimum at roughly three-fifths, followed by a renewed upward trend. Waiting time, while generally correlated with total distribution time, displays markedly higher volatility. Analysis indicates that, within a bimodal time-varying (BTV) network, crowd-sourced vehicles can indeed be deployed during both peak and off-peak periods; however, their limited capacity constrains aggregate throughput. Company-owned, large-capacity vehicles, although temporarily idle during peak hours, deliver superior transport efficiency once utilized.

4.6. Managerial Implications

Drawing upon the comprehensive sensitivity analyses and empirical results presented in Sections 4.3 and 4.4, we distill the following evidence-based managerial implications for urban logistics operators confronting bimodal time-varying networks and heterogeneous multi-trip fleet environments.

1. Institutionalize the Driver Trip Limit

Experimental evidence demonstrates that the imposition of a daily trip limit on drivers exerts a decisive influence on overall distribution efficiency. When total workload is held constant, either an excessively low or an excessively high ceiling on trip frequency impairs completion performance. Consequently, the allowable number of trips, A, should be codified as a non-negotiable, hard constraint within the dispatch system rather than treated as a flexible aspiration. From a workload perspective, this mechanism enforces a balanced allocation of effort while embodying a human-centered commitment to driver welfare.

2. Dynamic Fleet Mix Governance

Extreme strategies—whether fully owned or fully crowd-sourced fleets—are consistently sub-optimal. Firms should implement a daily capacity-adjustment mechanism that dynamically optimizes the owned-to-outsourced ratio based on real-time demand forecasts and congestion indices. For small-scale operations, crowd-sourced fleets should account for approximately two-thirds of capacity, scaling to three-fifths in large-scale scenarios. This ensures minimal total logistics cost while maintaining service resilience.

5. Conclusions

We developed a mixed-integer programming (MIP) model that simultaneously considers traffic peaks, fleet heterogeneity, warehouse capacities, and workload balancing. The proposed HHCRQL algorithm integrates reinforcement learning with heuristic search within a Markov Decision Process (MDP) framework, thereby enhancing decision-making efficiency and effectively avoiding local optima. Extensive experiments demonstrate that HHCRQL consistently outperforms classical heuristic algorithms, metaheuristics, and commercial solvers in terms of solution quality, convergence speed, and robustness. In particular, the cumulative reward mechanism significantly improves learning efficiency and solution stability, offering a reliable approach to complex multi-objective routing optimization.

Despite its strong performance, HHCRQL is currently tailored for offline optimization and does not yet directly address dynamic or real-time routing variations. In addition, the experimental settings remain restricted to specific vehicle types and warehouse configurations. Future research could explore integrating HHCRQL with advanced reinforcement learning frameworks such as DQN, Actor–Critic methods, or Transformer-based

RL to further improve its adaptability and efficiency. Extending the algorithm to B2C hybrid fleet scenarios and evaluating its robustness under uncertain, dynamic, or real-time environments would also be valuable directions. Moreover, embedding HHCRQL into human-in-the-loop scheduling systems could bridge automated optimization with practical decision-making, thereby strengthening both its applicability and real-world impact.

Author Contributions: Conceptualization, N.L. and X.W.; methodology, N.L., X.W. and X.J.; software, X.W.; validation, X.W.; formal analysis, N.L. and X.W.; investigation, N.L. and X.W.; resources, N.L.; data curation, N.L. and X.W.; writing—original draft preparation, X.W.; writing—review and editing, N.L. and X.W.; visualization, X.W.; supervision, N.L.; project administration, N.L.; funding acquisition, N.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (72171032, 72172023) and Humanities and Social Sciences Fund of Chinese Ministry of Education (21YJA630048, 21YJAZH070).

Data Availability Statement: Data unavailable due to commercial confidentiality. Part of the data is in Appendix A.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A

Table A1. Warehouse-related data.

Warehouse	Operation Rate			
	Large Cargo Truck		Small Cargo Van	
Serial Number	Loading Rate (Cubic Meters/ Minute)	Fixed Time (Cubic Meters/Minute)	Loading Rate (Cubic Meters/ Minute)	Fixed Time (Cubic Meters/ Minute)
$super-hub(0)$	1.5	10	1.67	10
1	1	15	1.06	15
2	1	15	1.67	15
3	1.78	15	1.67	15
4	1	15	1.06	15
5	1	15	1.06	15
6	1.78	15	1.48	15
7	1	15	1.06	15
8	1	15	1.06	15
9	1	15	1.06	15
10	1	15	1.06	15
11	1	15	1.06	15
...	1	15	1.06	15
n	1	15	1.06	15

Table A2. Results of HHCRQL for real data instances.

Round	Car	Task	Time	Carry
1	0	0-1-11	330, 400, 777	110
	1	0-1-11	330, 400, 777	110
	2	0-1-11	330.400.770	110
	3	0-1-11	330, 400, 777	110
	4	0-1-11	360, 430, 703	45
	5	0-4-11	360, 430, 697	38.65
	6	0-3-11	360, 430, 684	45
	7	0-7-11	360, 430, 703	45
	8	0-6-8-11	360, 430, 673, 807	21.02
	9	0-9-11	360, 430, 696	39.26

Table A2. *Cont.*

Round	Car	Task	Time	Carry
2	0	0-1-11	861, 931, 1257	110
	1	0-1-11	861, 931, 1257	110
	2	0-1-11	861, 931, 1257	110
	3	0-1-11	861, 931, 1257	110
	4	0-1-11	743, 813, 960	45
	5	0-5-11	733, 810, 946	33.37
	6	0-2-4-10-11	724, 810, 934, 1008, 1098	21.79
	7	0-3-11	743, 813, 936	36.3
	8	0-2-11	844, 914, 1032	26.54

Algorithm A1 Large Neighborhood Search—Simulated Annealing

LNSSA Algorithm parameters: Initial temperature (T_0), End time (T_{end}), Cooling rate (α), Acceptance rate (p)
Input: algorithm parameters
1. *initial solution*: Greedy strategy
2. *update feasible*: get feasible solutions
3. *while ture*:
4. *Destroy operation: choose different operation to destroy initial solution*
5. *Repair operation: choose different operation to repair destroyed solution*
6. *Swap operation: choose different operation to swap repaired solution*
7. *if feasible*
8. *break*
9. *end while*
10. *while* $T \geq T_{end}$
11. *feasible solutions = update feasible (initial solution)*
12. *Record better solutions and Accept the worse solution with Acceptance rate (p)*
13. *satisfy criterion*
14. *END while*
15. *output: Optimal solution*

Table A3. Part of the real data.

Warehouse	Demand
1	863.96
2	48.33
3	81.3
4	170.91
5	33.37
6	25.35
7	45.45
8	23.98
9	39.26
10	0.95
11	0
12	0
13	0

Table A4. Parameter setting.

Parameter	Demand
Epsilon	0.1
alpha	0.5
gamma	0.9
Max_episodes	20
Max_steps	20
T	1000
T_end	10
q	0.8
9	39.26
10	0.95
11	0
12	0
13	0

Appendix B

Appendix B.1. ILS Pseudocode

```
Function Iterated Local Search(max_iter):
    Initialize initial solution: task, car, carry, available
    best_task, best_car, best_carry, best_available = feasibility_check(task, car, carry, available)
    best_score = compute_total_time(best_task, best_car, best_carry)
    found_feasible = is_feasible(best_task)

    For iteration = 1 to max_iter:
        iteration_found_feasible = False

        For operator in [relocate, swap_nodes, two_opt]:
            new_task, new_car, new_carry = operator(best_task, best_car, best_carry, best_available)
            new_task, new_car, new_carry, feasible, new_available = feasibility_check(new_task, new_car, new_carry, best_available)

            If feasible:
                iteration_found_feasible = True
                score = compute_total_time(new_task, new_car, new_carry)
                If score < best_score:
                    Update best_task, best_car, best_carry, best_available, best_score

        found_feasible = found_feasible OR iteration_found_feasible

    Return best_task, best_car, best_carry, best_available, best_score, found_feasible
```

Appendix B.2. Initial Defacilitation Code

```
function initial_solu(demand, capa175, capa96, available, capavehicle):
task, car, carry, available = single_node_visit(demand, capa175, capa96, available)
task, car, carry, available = save_method(task, car, carry, capavehicle, available)
return task, car, carry, available
```

```
    # -------------------------
    # Generate Feasible Solution
    # -------------------------
    function updata_feasible1(taskin, carin, carryin, availablein):
        if input solution is None:
            task, car, carry, available = initial_solu(demand, capa175, capa96, available,
capavehicle)
        else:
            deep copy input to task, car, carry, available

        max_iterations (Set the number of iterations)

        for iteration = 1 to max_iterations:
            # Destruction operator
            if iteration is even:
                task, car, carry, destroy_list, available = destroy_by_load(task, car, carry,
available)
    else:
                task, car, carry, destroy_list, available = destroy_by_time(task, car, carry,
available)

            # Repair operator
            if iteration is even:
                task, car, carry, destroy_list, available = repair_after_load_destroy(task,
car, carry, destroy_list, available)
            else:
                task, car, carry, destroy_list, available = repair_by_merge_or_new(task,
car, carry, destroy_list, available)

            # Correct carry structure
            carry = correct_carry_structure(task, carry)

            # Time and feasibility check
            time = node_time1(task, car, carry)
            task, car, carry, time_list, feasible, available = feasibility(task, car, carry, time,
available)

            if feasible:
                vehicle_duration = total_time(task, car, time)
                return task, car, carry, time, available, True, vehicle_duration

        return task, car, carry, time, available, feasible, vehicle_duration
    # -------------------------
    # Example of Usage
    # -------------------------
    task, car, carry, available = initial_solu(demand, capa175, capa96, available, capavehi-
cle)
    taskout, carout, carryout, timeout, availableout, feasible = updata_feasible1(task, car,
carry, available)
```

References

1. Jamrus, T.; Wang, H.; Chien, C. Dynamic coordinated scheduling for supply chain under uncertain production time to empower smart production for Industry 3.5. *Comput. Ind. Eng.* **2020**, *142*, 106375. [CrossRef]
2. Zhao, Z.; Lee, C.K.M.; Ren, J.; Tsang, Y. Quality of service measurement for electric vehicle fast charging stations: A new evaluation model under uncertainties. *Transp. A Transp. Sci.* **2025**, *21*, 2232044. [CrossRef]
3. Nguyen, V.S.; Pham, Q.D.; Nguyen, T.H.; Bui, Q.T. Modeling and solving a multi-trip multi-distribution center vehicle routing problem with lower-bound capacity constraints. *Comput. Ind. Eng.* **2022**, *172*, 108597. [CrossRef]
4. Friggstad, Z.; Mousavi, R.; Rahgoshay, M.; Salavatipour, M.R. Improved approximations for capacitated vehicle routing with unsplittable client demands. In *Integer Programming and Combinatorial Optimization, Proceedings of the 23rd International Conference, IPCO 2022, Eindhoven, The Netherlands, 27–29 June 2022*; Springer: Cham, Switzerland, 2020; Volume 13265, pp. 251–261.
5. Qi, R.; Li, J.; Wang, J.; Jin, H.; Han, Y. QMOEA: A Q-learning-based multiobjective evolutionary algorithm for solving time-dependent green vehicle routing problems with time windows. *Inf. Sci.* **2022**, *608*, 178–201. [CrossRef]
6. Li, M.; Zhao, P.; Zhu, Y.; Cai, K. A learning and searching integration approach for same-day delivery problem with heterogeneous fleets considering uncertainty. *Transp. A Transp. Sci.* **2024**, *44*, 1–44. [CrossRef]
7. Kou, S.; Golden, B.; Bertazzi, L. Estimating optimal split delivery vehicle routing problem solution values. *Comput. Oper. Res.* **2024**, *168*, 106714. [CrossRef]
8. François, V.; Arda, Y.; Crama, Y.; Laporte, G. Large neighborhood search for multi-trip vehicle routing. *Eur. J. Oper. Res.* **2016**, *255*, 422–441. [CrossRef]
9. Agrawal, A.K.; Yadav, S.; Gupta, A.A.; Pandey, S. A genetic algorithm model for optimizing vehicle routing problems with perishable products under time-window and quality requirements. *Decis. Anal. J.* **2022**, *5*, 100139. [CrossRef]
10. Zhang, H.; Zhang, Q.; Ma, L.; Zhang, Z.; Liu, Y. A hybrid ant colony optimization algorithm for a multi-objective vehicle routing problem with flexible time windows. *Inf. Sci.* **2019**, *490*, 166–190. [CrossRef]
11. Ahmed, Z.H.; Yousefikhoshbakht, M. An improved tabu search algorithm for solving heterogeneous fixed fleet open vehicle routing problem with time windows. *Alex. Eng. J.* **2023**, *64*, 349–363. [CrossRef]
12. Osorio-Mora, A.; Escobar, J.W.; Toth, P. An iterated local search algorithm for latency vehicle routing problems with multiple depots. *Comput. Oper. Res.* **2023**, *158*, 106293. [CrossRef]
13. Kuo, R.J.; Luthfiansyah, M.F.; Masruroh, N.A.; Zulvia, F.E. Application of improved multi-objective particle swarm optimization algorithm to solve disruption for the two-stage vehicle routing problem with time windows. *Expert Syst. Appl.* **2023**, *225*, 120009. [CrossRef]
14. Su, E.; Qin, H.; Li, J.; Pan, K. An exact algorithm for the pickup and delivery problem with crowdsourced bids and transshipment. *Transp. Res. Part B Methodol.* **2023**, *177*, 102831. [CrossRef]
15. Zhou, H.; Qin, H.; Cheng, C.; Rousseau, L. An exact algorithm for the two-echelon vehicle routing problem with drones. *Transp. Res. Part B Methodol.* **2023**, *168*, 124–150. [CrossRef]
16. Pan, B.; Zhang, Z.; Lim, A. Multi-trip time-dependent vehicle routing problem with time windows. *Eur. J. Oper. Res.* **2021**, *291*, 218–231. [CrossRef]
17. Moon, I.; Salhi, S.; Feng, X. The location-routing problem with multi-compartment and multi-trip: Formulation and heuristic approaches. *Transp. A Transp. Sci.* **2020**, *16*, 501–528. [CrossRef]
18. Yu, V.F.; Salsabila, N.Y.; Gunawan, A.; Handoko, A.N. An adaptive large neighborhood search for the multi-vehicle profitable tour problem with flexible compartments and mandatory customers. *Appl. Soft Comput.* **2024**, *156*, 111482. [CrossRef]
19. You, J.; Wang, Y.; Xue, Z. An exact algorithm for the multi-trip container drayage problem with truck platooning. *Transp. Res. Part E Logist. Transp. Rev.* **2023**, *175*, 103138. [CrossRef]
20. Sethanan, K.; Jamrus, T. Hybrid differential evolution algorithm and genetic operator for multi-trip vehicle routing problem with backhauls and heterogeneous fleet in the beverage logistics industry. *Comput. Ind. Eng.* **2020**, *146*, 106571. [CrossRef]
21. Neira, D.A.; Aguayo, M.M.; De la Fuente, R.; Klapp, M.A. New compact integer programming formulations for the multi-trip vehicle routing problem with time windows. *Comput. Ind. Eng.* **2020**, *144*, 106399. [CrossRef]
22. Liu, W.; Dridi, M.; Ren, J.; El Hassani, A.H.; Li, S. A double-adaptive general variable neighborhood search for an unmanned electric vehicle routing and scheduling problem in green manufacturing systems. *Eng. Appl. Artif. Intell.* **2023**, *126*, 107113. [CrossRef]
23. Qin, W.; Zhuang, Z.; Huang, Z.; Huang, H. A novel reinforcement learning-based hyper-heuristic for heterogeneous vehicle routing problem. *Comput. Ind. Eng.* **2021**, *156*, 107252. [CrossRef]
24. Xu, B.; Jie, D.; Li, J.; Yang, Y.; Wen, F.; Song, H. Integrated scheduling optimization of U-shaped automated container terminal under loading and unloading mode. *Comput. Ind. Eng.* **2021**, *162*, 107695. [CrossRef]

25. Duan, J.; Liu, F.; Zhang, Q.; Qin, J. Tri-objective lot-streaming scheduling optimization for hybrid flow shops with uncertainties in machine breakdowns and job arrivals using an enhanced genetic programming hyper-heuristic. *Comput. Oper. Res.* **2024**, *172*, 106817. [CrossRef]
26. Zhang, Z.; Wu, F.; Qian, B.; Hu, R.; Wang, L.; Jin, H. A Q-learning-based hyper-heuristic evolutionary algorithm for the distributed flexible job-shop scheduling problem with crane transportation. *Expert Syst. Appl.* **2023**, *234*, 121050. [CrossRef]
27. Li, L.; Zhang, H.; Bai, S. A multi-surrogate genetic programming hyper-heuristic algorithm for the manufacturing project scheduling problem with setup times under dynamic and interference environments. *Expert Syst. Appl.* **2024**, *250*, 123854. [CrossRef]
28. Li, J.; Ma, Y.; Gao, R.; Cao, Z.; Lim, A.; Song, W.; Zhang, J. Deep reinforcement learning for solving the heterogeneous capacitated vehicle routing problem. *IEEE Trans. Cybern.* **2022**, *52*, 13572–13585. [CrossRef]
29. Zhang, Z.; Qi, G.; Guan, W. Coordinated multi-agent hierarchical deep reinforcement learning to solve multi-trip vehicle routing problems with soft time windows. *IET Intell. Transp. Syst.* **2023**, *17*, 2034–2051. [CrossRef]
30. Shang, C.; Ma, L.; Liu, Y. Green location routing problem with flexible multi-compartment for source-separated waste: A Q-learning and multi-strategy-based hyper-heuristic algorithm. *Eng. Appl. Artif. Intell.* **2023**, *121*, 105954. [CrossRef]
31. Xu, K.; Shen, L.; Liu, L. Enhancing column generation by reinforcement learning-based hyper-heuristic for vehicle routing and scheduling problems. *Comput. Ind. Eng.* **2025**, *206*, 111138. [CrossRef]
32. Dastpak, M.; Errico, F.; Jabali, O. Off-line approximate dynamic programming for the vehicle routing problem with a highly variable customer basis and stochastic demands. *Comput. Oper. Res.* **2023**, *159*, 106338. [CrossRef]
33. Wang, Y.; Hong, X.; Wang, Y.; Zhao, J.; Sun, G.; Qin, B. Token-based deep reinforcement learning for Heterogeneous VRP with Service Time Constraints. *Knowl. Based Syst.* **2024**, *300*, 112173. [CrossRef]
34. Huang, N.; Li, J.; Zhu, W.; Qin, H. The multi-trip vehicle routing problem with time windows and unloading queue at depot. *Transp. Res. Part E Logist. Transp. Rev.* **2021**, *152*, 102370. [CrossRef]
35. Fragkogios, A.; Qiu, Y.; Saharidis, G.K.D.; Pardalos, P.M. An accelerated benders decomposition algorithm for the solution of the multi-trip time-dependent vehicle routing problem with time windows. *Eur. J. Oper. Res.* **2024**, *317*, 500–514. [CrossRef]
36. Blauth, J.; Held, S.; Müller, D.; Schlomberg, N.; Traub, V.; Tröbst, T.; Vygen, J. Vehicle routing with time-dependent travel times: Theory, practice, and benchmarks. *Discret. Optim.* **2024**, *53*, 100848. [CrossRef]
37. Sun, B.; Chen, S.; Meng, Q. Optimizing first-and-last-mile ridesharing services with a heterogeneous vehicle fleet and time-dependent travel times. *Transp. Res. Part E Logist. Transp. Rev.* **2025**, *193*, 103847. [CrossRef]
38. Gutierrez, A.; Labadie, N.; Prins, C. A Two-echelon Vehicle Routing Problem with time-dependent travel times in the city logistics context. *EURO J. Transp. Logist.* **2024**, *13*, 100133. [CrossRef]
39. Lu, J.; Nie, Q.; Mahmoudi, M.; Ou, J.; Li, C.; Zhou, X.S. Rich arc routing problem in city logistics: Models and solution algorithms using a fluid queue-based time-dependent travel time representation. *Transp. Res. Part B Methodol.* **2022**, *166*, 143–182. [CrossRef]
40. Coelho, V.N.; Grasas, A.; Ramalhinho, H.; Coelho, I.M.; Souza, M.J.F.; Cruz, R.C. An ILS-based algorithm to solve a large-scale real heterogeneous fleet VRP with multi-trips and docking constraints. *Eur. J. Oper. Res.* **2016**, *250*, 367–376. [CrossRef]
41. Zhao, J.; Poon, M.; Tan, V.Y.F.; Zhang, Z. A hybrid genetic search and dynamic programming-based split algorithm for the multi-trip time-dependent vehicle routing problem. *Eur. J. Oper. Res.* **2024**, *317*, 921–935. [CrossRef]
42. He, D.; Ceder, A.A.; Zhang, W.; Guan, W.; Qi, G. Optimization of a rural bus service integrated with e-commerce deliveries guided by a new sustainable policy in China. *Transp. Res. Part E Logist. Transp. Rev.* **2023**, *172*, 103069. [CrossRef]
43. Vieira, B.S.; Ribeiro, G.M.; Bahiense, L. Metaheuristics with variable diversity control and neighborhood search for the Heterogeneous Site-Dependent Multi-depot Multi-trip Periodic Vehicle Routing Problem. *Comput. Oper. Res.* **2023**, *153*, 106189. [CrossRef]
44. Li, K.; Liu, T.; Kumar, P.N.R.; Han, X. A reinforcement learning-based hyper-heuristic for AGV task assignment and route planning in parts-to-picker warehouses. *Transp. Res. Part E Logist. Transp. Rev.* **2024**, *185*, 103518. [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.

algorithms

MDPI

Article

Backward Signal Propagation: A Symmetry-Based Training Method for Neural Networks

Kun Jiang * and Zhihong Fu

School of Electrical Engineering, Chongqing University, Chongqing 400044, China; fuzhihong@cqu.edu.cn
* Correspondence: atan_j@163.com

Abstract

While backpropagation (BP) has long served as the cornerstone of training deep neural networks, it relies heavily on strict differentiation logic and global gradient information, lacking biological plausibility. In this paper, we systematically present a novel neural network training paradigm that depends solely on signal propagation, which we term Backward Signal Propagation (BSP). The core idea of this framework is to reinterpret network training as a symmetry-driven process of discovering inverse causal relationships. Starting from symmetry principles, we define symmetric differential equations and leverage their inherent properties to implement a learning mechanism analogous to differentiation. Furthermore, we introduce the concept of causal distance, a core invariant that bridges the forward propagation and inverse learning processes. It quantifies the influence strength between any two elements in the network, leading to a generalized form of the chain rule. With these innovations, we achieve precise, pointwise adjustment of model parameters. Unlike traditional BP, the BSP method enables parameter updates based solely on local signal features. This work offers a new direction toward efficient and biologically plausible learning algorithms.

Keywords: backpropagation; biological plausibility; symmetric differential equations; backward signal propagation; neural network training

1. Introduction

1.1. Backpropagation Algorithm

Training has always been a central issue in the development of neural networks. Throughout the various surges and declines in the history of artificial neural networks, the evolution of training methods has played a pivotal role in determining the success or stagnation of the field [1,2].

Since the 1980s, the emergence of the backpropagation (BP) algorithm has removed the primary obstacle in neural network training, granting artificial neural networks theoretically unlimited potential [3]. Remaining engineering challenges—such as vanishing and exploding gradients [4], insufficient training data [5], and computational inefficiencies [6]—have been addressed progressively through sustained research efforts.

Today, the backpropagation algorithm holds a dominant position in neural network training, primarily due to its ability to perform pointwise precise parameter updates. For any given parameter, BP provides an explicit relationship to the final error, enabling highly efficient optimization of neural networks [7].

Moreover, a family of optimization methods built upon backpropagation—such as stochastic gradient descent (SGD) and adaptive moment estimation (Adam) [8,9]—has greatly extended the applicability of BP, providing robust support for training neural networks of various architectures.

1.2. The Core Contradiction of the Backpropagation Algorithm

Despite its unparalleled success, the backpropagation algorithm has long been the subject of skepticism among researchers [10,11]. The core issue preventing the replacement of BP lies in the absence of an alternative mechanism that can efficiently track inverse causal relationships during training.

The fundamental strength of BP lies in two key techniques: differentiation and the chain rule of propagation [7]. Differentiation enables the analysis of causal relationships between any two variables in an equation, while the chain rule reflects the topological structure of the network [3]. The combination of these two allows BP to systematically infer how each parameter contributes to the final error across arbitrarily complex architectures. This capacity for tracing inverse causality makes it possible to perform precise error-driven adjustments for any parameter in the network.

However, the very foundations of BP also raise two major concerns regarding biological plausibility:

1. Differentiation in biological neurons: It remains unclear whether biological neurons are capable of performing differentiation. Modern artificial neural networks employ a wide variety of activation functions, each designed to be differentiable. Yet, such mathematical operations—and the diverse function types themselves—appear to have no biological counterpart [12].
2. Global error propagation: The chain rule in BP assumes that error signals can be propagated globally across the network, requiring each neuron to possess a form of global awareness to compute gradients based on upstream signals. This assumption, however, is biologically unrealistic, as real neurons operate primarily on local information and lack such global processing capability [13].

These two issues lie at the heart of the debate over the biological plausibility of BP. Despite ongoing criticism, the algorithm remains a cornerstone of modern neural network training.

In order to solve the core contradiction of the backpropagation algorithm, people have never given up research on biological nerve cells. On the one hand, biological nerve cells have natural biological rationality. On the other hand, biological neural networks also have higher energy utilization efficiency. For example, the human brain only needs very low power to achieve efficient logical reasoning and mathematical calculations.

1.3. Inspiration from Differential Geometry

We argue that the key to overcoming the current predicament lies in moving beyond the reliance on a "global perspective" and reconstructing the computation and learning processes of neural networks upon a local and continuous framework. This shift in conceptual paradigm does not emerge in a vacuum; rather, its most important inspiration stems from the historical emergence and evolution of differential geometry in the history of mathematics. The development of differential geometry itself constitutes a grand narrative—a transition from the study of rigid, global shapes to the analysis of flexible, local properties, ultimately integrating local information to comprehend complex global structures.

Before the advent of differential geometry, Euclidean geometry dominated the understanding of space. It dealt with global, rigid objects such as perfect straight lines,

circles, and polyhedra. Its axiomatic approach was ill-suited to handle the ubiquitous curved, smooth, and deformable shapes found in the real world. This bears a striking resemblance to the situation of backpropagation: it operates flawlessly on an idealized, globally connected computational graph, yet struggles to adapt to more complex, dynamic, and resource-constrained physical realities.

The true transformation began with the invention of calculus in the 17th century, which provided researchers with a powerful tool to "infinitely zoom in" and examine local properties. Mathematicians began using derivatives and integrals to analyze the behavior of curves and surfaces within infinitesimal neighborhoods around each point, rigorously defining key concepts such as curvature and torsion. This marked the first leap in thinking: a shift from focusing on global shapes to concentrating on local characteristics. This offers us an insight: perhaps we, too, should temporarily set aside the global pursuit of the "optimal solution" for the entire network and instead investigate how neurons or neural assemblies can respond most efficiently and rationally based on the local signals they receive.

1.4. Solutions Based on Symmetric Differential Equations

To design a local system with desirable properties, we turned to group theory and, in combination with differential equations, constructed symmetric differential equations [14]. Owing to the inherent symmetry of group theory, these symmetric differential equations are reversible. By connecting multiple such symmetric differential equation systems, we formed a symmetric differential neural network. The reversibility of the system allows signals to propagate forward with input data and backward with error signals. Mathematically, these two types of signals are not fundamentally different; their distinction lies in their direction of propagation and their functional roles within the network.

Within this novel architecture, traditional training algorithms (such as Backpropagation) are no longer directly applicable. Consequently, we endeavored to construct a new algorithm capable of achieving efficient training analogous to that of BP. To this end, we addressed the two core functions of the BP algorithm through the following means: First, we replaced conventional derivative calculation with differential equations. Second, we established a theory of causal distance, ensuring topological invariance between the forward and backward systems, thereby laying the groundwork for replacing the chain rule.

In the field of biological neuroscience, differential equations are the natural language for describing neuronal dynamics [15]. Whether modeling membrane potential variations or oscillatory behaviors, researchers routinely employ differential equations to characterize these phenomena [16]. However, one crucial aspect is often overlooked: the differentiability inherent in differential equations itself is a powerful tool.

Drawing from differential geometry theory, we understand that the differentiability of a differential equation inherently defines local causal relationships among variables [17]. Unlike backpropagation, which depends on global function knowledge, training based on differential equations relies solely on local topological information, thus naturally fulfilling the requirement of biological plausibility. Compared with other learning algorithms, this differential-topological approach provides a more precise and efficient means of parameter adaptation in neural networks. Although previous studies have recognized the link between differential equations and causality, they failed to resolve the central issue: how can these local causal relationships be propagated across the network to achieve systemic parameter adjustment? In essence, this question mirrors the core difficulty addressed by the chain rule of backpropagation.

In our proposed architecture, the forward propagation system carries the input signal, while the backward propagation system transmits the error signal. Both systems share an identical topological structure. To formalize the correspondence between these two subsystems, we introduce the concept of causal distance, which precisely characterizes the topological relationships between various network elements and parameters. Each neuron adjusts its parameters locally, based on the discrepancy between the forward and backward signals. This mechanism enables both pointwise precision and biological plausibility in parameter adaptation.

Furthermore, based on the Backward Signal Propagation (BSP) principle, we implemented two distinct parameter update methods for different components of the system. Experimental results confirm that both approaches effectively train the neural network. From a theoretical standpoint, training symmetric differential equations using the BSP method endows the network with unlimited potential, and the remaining work lies in enhancing training efficiency and accuracy from an engineering perspective.

In summary, this study replaces explicit derivative computation with the intrinsic differentiability of differential equations, and substitutes the chain rule with reversible signal propagation in a symmetric framework. This successfully addresses the two core challenges of neural network training, and establishes a novel, efficient, and biologically plausible training paradigm.

The structure of the remainder of this paper is as follows: Section 2 briefly introduces the backpropagation algorithm and highlights its two foundational pillars—derivative operations and the chain rule. Section 3 presents a novel neural network architecture and defines the concept of causal distance, offering a concrete mechanism for realizing both derivative computation and chain-like propagation based on this framework, thereby laying the foundation for precise parameter tuning. Section 4 applies the theory to experimental settings, demonstrating that two BSP-based training strategies can effectively train the network. Finally, Section 5 concludes the paper, emphasizing the role of BSP in constructing multiple training schemes for symmetric neural networks.

2. Introduction to the Backpropagation Algorithm

2.1. Derivative Operation

Let us consider a function $M = f(N)$. In a causal system, we interpret this as the value of M depending on N, and this causal relationship can be described using the operation of differentiation:

$$\frac{dM}{dN} = f'(N) \tag{1}$$

This formulation represents a basic expression of causality. The function $f(N)$ can take various forms—exponential, polynomial, and many others. In most cases, research attention is focused on identifying the appropriate form of $f(N)$ to achieve a desired system behavior. This reflects a function-driven design paradigm: empowered by the versatility of differentiation, we are rarely concerned that the introduction of a new function might destabilize the system, as differentiation readily reveals the causal relationship between two variables.

However, the flexibility of differentiation offers us great advantages, it may also obscure deeper structural considerations. Our reliance on the convenience of differentiation may hinder us from stepping back and evaluating which functional forms are most appropriate for the system in a broader sense.

In the training of neural networks, forward signal propagation and backward error propagation are both essential processes. It is important to note that differentiation is

required only in the backward propagation phase. This necessity arises from the fact that the forward propagation lacks a mathematically well-structured framework. As a result, in the backward phase, we must rely on a general-purpose tool—differentiation—to compute reverse causal relationships. This explains the central role of differentiation in neural network training. Not only the backpropagation algorithm but also a wide range of training methods and optimization strategies are fundamentally based on derivative operations, such as Stochastic Gradient Descent (SGD) and Adaptive Moment Estimation (Adam) [8,9].

2.2. Chain Rule

In addition to differentiation, another foundational component of the backpropagation algorithm is the chain rule. The chain rule enables the systematic propagation of error signals from the output layer back to the input layer, layer by layer, thereby making it possible to compute each parameter's contribution to the final loss.

In deep neural networks, this backward propagation path is often lengthy and involves multiple nonlinear transformations. Therefore, the stability and effectiveness of the chain rule are critical to the success of the training process.

Figure 1 illustrates the process of forward signal propagation and backward error propagation in conventional neural networks. Figure 1a,b depict the structure of a single-layer neuron, while Figure 1c,d show the multilayer neuron structure. In neural network training, we adjust the connection weights between neurons to modulate the output signal. However, this training process can be easily simplified in the case of single-layer neural networks.

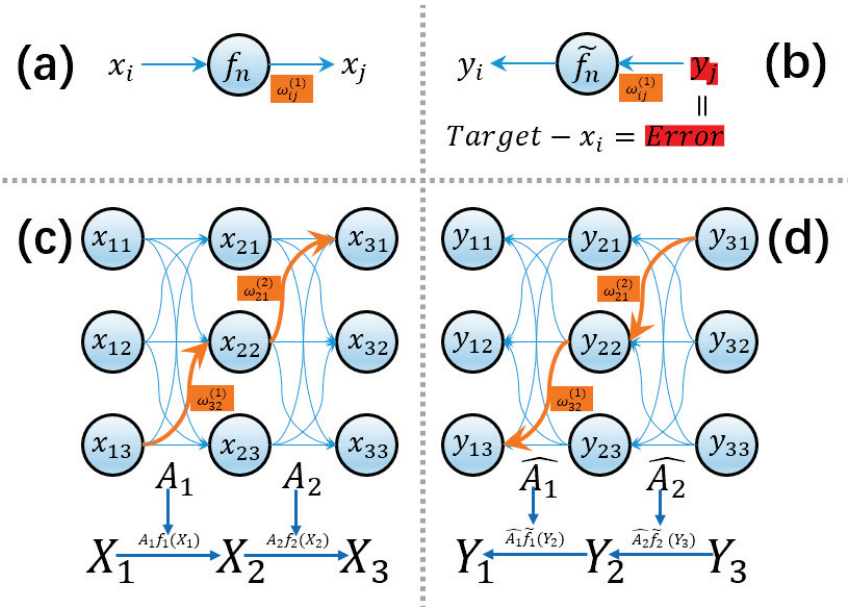


Figure 1. Signal propagation process in traditional neural networks. (**a**) The signal propagation process in a single neuron. (**b**) The backward propagation of the error signal in a single neuron. (**c**) The signal propagation process in multilayer perceptron. (**d**) The backward propagation of the error signal in multilayer perceptron.

In Figure 1a,b, f_n denotes the system function used in forward signal propagation, while $\tilde{f}_n$ represents the corresponding function in backward signal propagation, which generally corresponds to the derivation of the original system function. Here, x_i is the input signal in the forward pass, and x_j is the output signal. The backward-propagated

signal y_j represents the error signal obtained by comparing the actual output with *Tatget*, whereas y_i is the gradient signal propagated backward.

Since the systems in Figure 1a,b comprise only a single-layer network, it is possible to adjust the connection weight $\omega_{ij}^{(1)}$, assuming it is placed on the output side, without explicitly computing derivatives. The adjustment can be carried out quantitatively by simply comparing the difference between x_j and y_j, thus avoiding the need for gradient-based computations.

$$\omega_{ij_new}^{(1)} = \omega_{ij_old}^{(1)} + V_{xy} \tag{2}$$

where

$$V_{xy} = atan(kt \cdot x_j \cdot y_j)/kt \tag{3}$$

In Equation (3), kt is introduced as an adjustment parameter. However, this approach is not applicable to multilayer neural network structures such as those depicted in Figure 1c,d. In such cases, the hierarchical composition of multiple nonlinear functions necessitates the use of derivative-based computations in order to propagate error signals throughout the entire network.

Figure 1c,d illustrate a simple three-layer neural network, with each layer containing three neurons. Figure 1c shows the forward signal propagation, while Figure 1d depicts the backward propagation of the error signal.

In Figure 1c, X_1, X_2, X_3 represent the signals at each layer, f_1 and f_2 denote nonlinear activation functions, and A_1 and A_2 correspond to the connection coefficients between neuronal layers. It can be observed that X_1 is transformed into the signal X_2 of the subsequent layer through the operation of A_1 and the nonlinear function f_1. Similarly, X_2 is further processed by A_2 and f_2 to yield X_3.

Figure 1d illustrates the process of error signal backpropagation. Here, Y_1, Y_2, Y_3 denote the signals at each layer, $\widetilde{f_1}$ and $\widetilde{f_2}$ represent nonlinear functions, and $\widetilde{A_1}$ and $\widetilde{A_2}$ indicate the connection coefficients between layers. Unlike forward propagation, the causal relationships in backpropagation are derived through differentiation. A typical procedure is as follows:

First, the *Loss* is computed based on the output signal X:

$$\text{Loss} = \frac{1}{2}|X - target|^2 \tag{4}$$

The gradient with respect to X is then obtained:

$$D_1(X) = X - target \tag{5}$$

Subsequent differentiation with respect to the parameters yields the corresponding parameter gradients:

$$\frac{\partial Loss}{\partial A} = D_1(X) \odot \frac{\partial X}{\partial A} \tag{6}$$

If we interpret the differentiation operation as a form of nonlinear function while retaining the mathematical structure of the forward process, a analogous logical flow emerges. Thus, the backpropagated signal Y_3 is processed by $\widetilde{A_2}$ and the nonlinear function $\widetilde{f_2}$ to produce Y_2, which is similarly transformed by $\widetilde{A_1}$ and $\widetilde{f_1}$ into Y_1. The detailed mathematical relationships can be found in the equations accompanying Figure 1c,d.

This demonstrates that the mathematical logic of forward propagation and that of back-propagation form a perfectly symmetric relationship. In Figure 1c,d, we explicitly highlight the signal propagation process from x_{13} to x_{31} together with the corresponding

reverse error propagation from y_{31} to y_{13} This exact symmetry reflects the principle of causal reversibility.

Building on this understanding of symmetry, the next key question is how to design a symmetric system to replace the conventional neural network framework.

3. Symmetric Differential Equations and Causal Distance

3.1. Neural Network Based on Symmetric Differential Equations

Motivated by this challenge, we propose a novel class of symmetric differential equations grounded in symmetry principles. Figure 2 illustrates the conceptual development of our symmetric differential equation system.

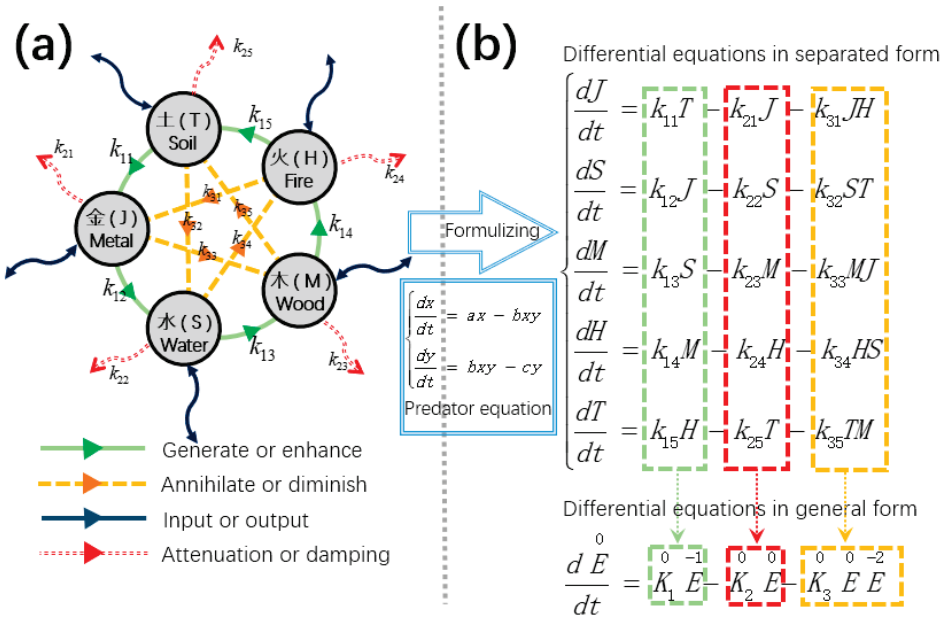


Figure 2. From Wuxing logic to symmetric differential equations. (**a**) Traditional Wuxing logic posits that the world is composed of five distinct elements that interact through generative and inhibitory relationships, forming the logical framework of the universe. The logical structure depicted in the figure deviates from the traditional model by incorporating self-attenuation terms and creating interfaces for external input and output signals. In this system, there are five nodes, each capable of serving as an input or output. However, to prevent signal interference, a node can either receive input or generate output at any given time, but not both simultaneously. (**b**) By combining the Wuxing logic with the predator-prey equation, we can derive a set of differential equations. The symmetry of the system is carefully preserved throughout this transformation process. As a result, both the traditional Five Elements logic and the predator-prey equation are modified, ultimately leading to a set of fully symmetrical equations. For clarity, these equations are presented in a generalized format, with the numbers above the elements and parameters indicating the offset of each loop.

As shown in Figure 2a, we begin with the traditional *Wuxing* (Five Elements) framework from Chinese philosophy, which encodes a closed and symmetric logic of generation and inhibition among five elemental components: Metal (J), Water (S), Wood (M), Fire (H), and Soil (T). In this cyclical structure, each element either promotes (generates) or suppresses another, forming a self-contained symmetric system.

To translate this logic into a dynamic model, we reformulate the framework by incorporating self-decay and input-output nodes, while also drawing inspiration from predator equation (Equation (7)). The resulting system is a distributed set of differential equations, as depicted in Figure 2b and formalized in Equation (8). For instance, in the Wuxing logic, Soil

gives rise to Metal (e.g., ore is extracted from soil), whereas Fire can melt and thereby subdue Metal. This interaction leads to the first equation in (8): $dJ/dt = k_{11}T - k_{21}J - k_{31}JH$. Here, the second term represents a self-decay component, introduced to stabilize the system in analogy with the damping mechanisms in ecological equations.

$$\begin{cases} \frac{dx}{dt} = ax - bxy \\ \frac{dy}{dt} = bxy - cy \end{cases} \tag{7}$$

$$\begin{cases} \frac{dJ}{dt} = k_{11}T - k_{21}J - k_{31}JH \\ \frac{dS}{dt} = k_{12}J - k_{22}S - k_{32}ST \\ \frac{dM}{dt} = k_{13}S - k_{23}M - k_{33}MJ \\ \frac{dH}{dt} = k_{14}M - k_{24}H - k_{34}HS \\ \frac{dT}{dt} = k_{15}H - k_{25}T - k_{35}TM \end{cases} \tag{8}$$

Let E = {J, S, M, H, T} denote the set of elemental states, and define three parameter sets: $K_1 = \{k_{11}, k_{12}, k_{13}, k_{14}, k_{15}\}$, $K_2 = \{k_{21}, k_{22}, k_{23}, k_{24}, k_{25}\}$, $K_3 = \{k_{31}, k_{32}, k_{33}, k_{34}, k_{35}\}$, Then, the system in Equation (8) can be rewritten in a compact form as Equation (9).

$$\frac{dE}{dt} = K_1 E - K_2 E - K_3 EE \tag{9}$$

It is important to note that the ordering of elements within the vector E differs depending on the context and position in the equation, and thus cannot be trivially unified. However, if we consider a cyclic permutation of these elements across different positions, Equation (9) can be generalized into a rotationally symmetric form, denoted as Equation (10) (see Figure 2b).

$$\frac{d\overset{0}{E}}{dt} = \overset{0\ -1}{K_1\ E} - \overset{0\ 0}{K_2 E} - \overset{0\ 0-2}{K_3 E\ E} \tag{10}$$

This general formulation permits arbitrary extension in the number of elements and customization of their interaction rules, provided that the cyclic structure is preserved. Consequently, the generalized form (Equation (10)) possesses broader applicability. Due to the structural similarities among such symmetric differential systems, analytical results obtained from Equation (8) can naturally be extended to the broader family defined by Equation (10).

This framework offers several distinct advantages. First, it possesses perfect symmetry, enabling bidirectional signal propagation, which eliminates the need for gradient-based calculations. Second, the structural design is based on differential equations, thereby ensuring a high degree of biological plausibility. According to our design, network training no longer depends on global gradient computation but instead relies solely on local signal comparisons. Third, the topological structure of the system can maintain complete symmetry, allowing for a signal propagation mechanism analogous to the chain rule and enabling pointwise parameter adjustment with high precision. Fourth, the symmetric differential equations themselves inherently possess nonlinear characteristics, eliminating the need to introduce additional nonlinear activation functions such as ReLU or Sigmoid.

Table 1 illustrates the differences and connections between the traditional multilayer perceptrons and symmetric differential equation neural networks.

Table 1. Comparison between multilayer perceptron and symmetric differential equation neural networks.

	Multilayer Perceptron	Symmetric Differential Equation Neural Networks
Signal source	Defined by input signal and nonlinear function	Generate perturbation signal from input signal
Nonlinear property	Activation function	Natural property of the system
Causal tracing methods	Derivative operation	Differentiability of equations
Traversing the topology	Chain propagation law	Causal distance invariance

Figure 3 illustrates a neural network architecture based on symmetric differential equations. This structure aligns with the multilayer perceptron (MLP) shown in Figure 1, as both comprise three input nodes, three output nodes, and two layers of nonlinear transformations. A comparison between Figures 1 and 3 reveals that we have replaced the original simple neurons with symmetric differential equations.

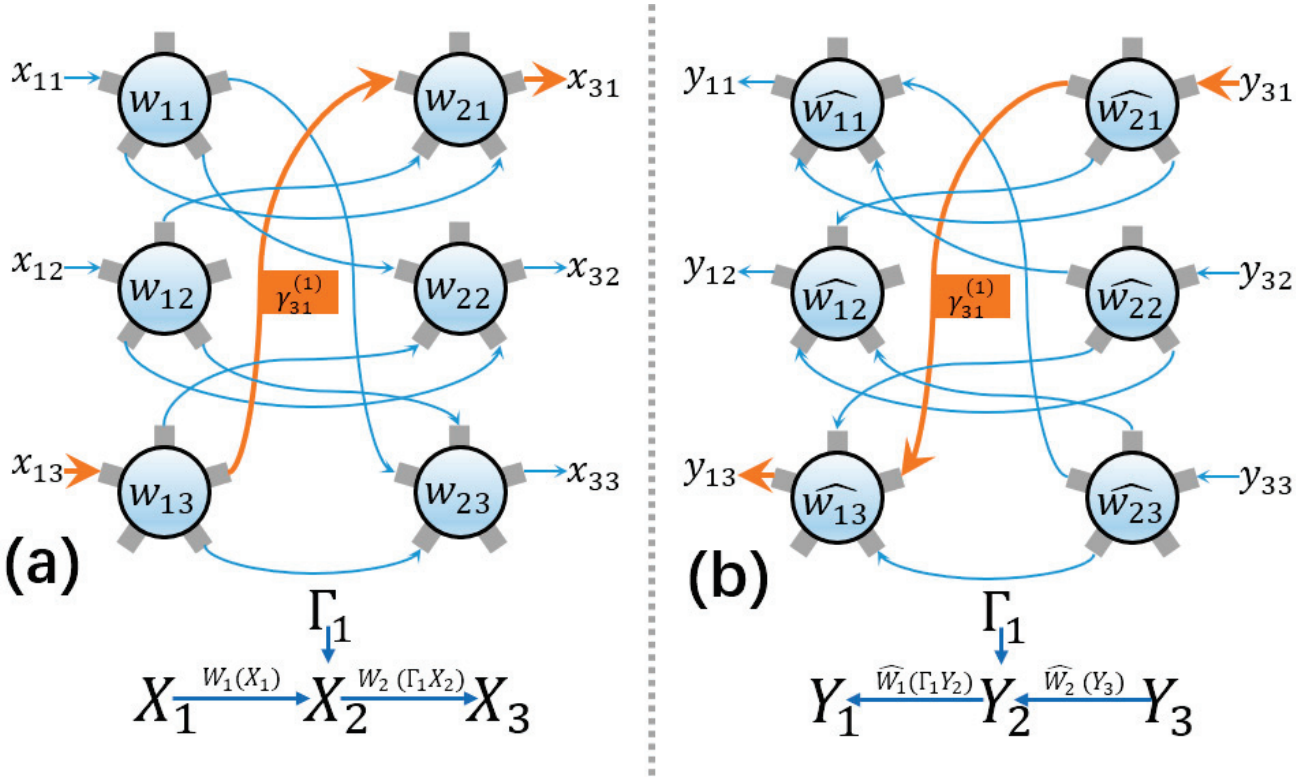


Figure 3. Signal propagation in neural networks based on symmetric differential equations. (**a**) Signal propagation in symmetric differential equation neural networks. Since differential equations have natural nonlinear characteristics, we do not need to introduce additional functions, thus building a more biologically plausible neural network structure. (**b**) Backward error signal propagation in symmetric differential equation neural networks. It can be seen that in the external connections of neurons, we have adopted a method that is completely symmetrical with for-ward propagation, but how to reverse the internal relationship of neurons so that the system can achieve reverse causality will be the issue next.

In Figure 3a, w_{11}, w_{12}, w_{13}, w_{21}, w_{22}, w_{23} represents the symmetric differential equations introduced in Figure 2, it functionally corresponds to the nonlinear function f_1 and f_2 in Figure 1c. These systems are inherently nonlinear, eliminating the need for additional nonlinear activation functions. A forward signal propagation path and a backward propagation path are explicitly indicated in Figure 3, which follows the same logic as in Figure 1. In Figure 3b,

$\hat{w}_{11}$, $\hat{w}_{12}$, $\hat{w}_{13}$, $\hat{w}_{21}$, $\hat{w}_{22}$, $\hat{w}_{23}$ denotes the inverse symmetric differential equation, it functionally corresponds to the nonlinear function $\widetilde{f}_1$ and $\widetilde{f}_2$ in Figure 1d. It is mathematically identical to the forward equation, with the only difference being that the signal propagates in the opposite direction to that of the forward system. Additionally, connection coefficients Γ_1 are incorporated between the neurons in Figure 3, serving the same function as the corresponding coefficients in a traditional MLP. Thus, we have structurally constructed a neural network based on symmetric differential equations that resembles an MLP.

3.2. Causal Distance

As previously discussed, the inherent symmetry of the system allows for straightforward reversibility. This naturally raises the question: What are the invariants in this reversal process? This is a crucial and underexplored issue. Building on our analysis of the network's topology, we introduce the concept of causal distance, proposing it as the system's key invariant. We argue that causal distance plays a decisive role in constructing reversible systems and serves as a guiding principle in maintaining system-level symmetry.

Figure 4 presents the structure of symmetric differential equations for both forward and backward propagation. In Figure 4a, x_{ij} and x_{nm} denote the input and output signals in the forward propagation process, respectively. In Figure 4b, y_{nm} and y_{ij} denote the input and output signals in the backward (error) propagation process.

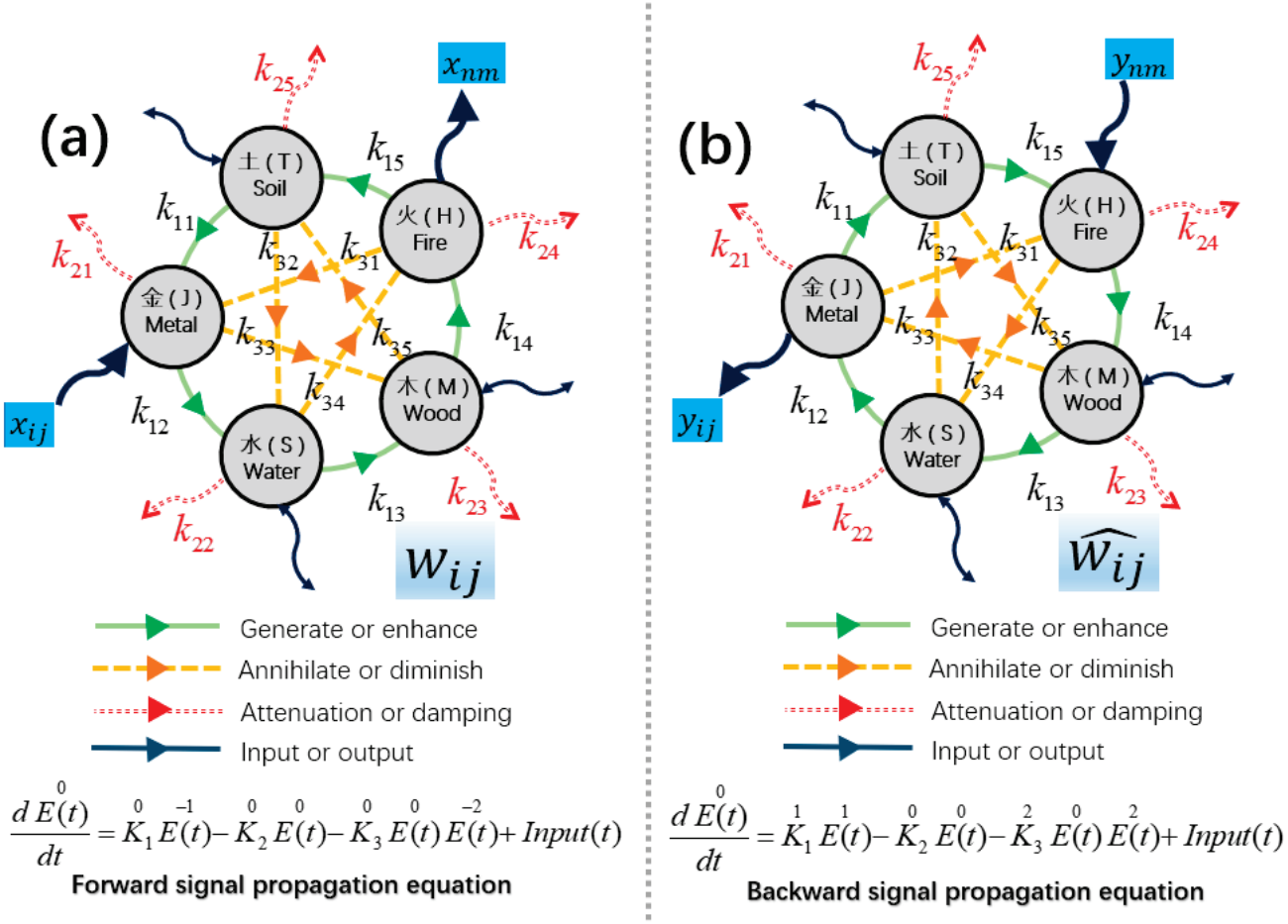


Figure 4. Forward and backward signal propagation in symmetric differential equations. (**a**) Forward signal propagation in symmetric differential equations. (**b**) Backward signal propagation in symmetric differential equations.

According to the structure of the system, we can establish the forward signal input equation of the system:

$$\frac{dE(t)}{dt} = \overset{0}{K}_1 \overset{-1}{E}(t) - \overset{0}{K}_2 \overset{0}{E}(t) - \overset{0}{K}_3 \overset{0}{E}(t) \overset{-2}{E}(t) + Input(t) \tag{11}$$

In Figure 2a, we show the logical relationship of the five elements. For example, water can generate wood because plants can grow with water, and metal can restrain wood because an ax made of metal can cut down trees. Based on this logical relationship, we obtain the third equation $dM/dt = k_{13}S - k_{23}M - k_{33}MJ$ in Figure 2b. If we reverse this causal relationship, it becomes that wood can generate water and wood can restrain metal; similarly, fire can generate wood, and soil restrains wood. When all causal relationships are totally reversed, we can obtain the logical structure of Figure 4b and Equation (12):

$$\begin{cases} \frac{dJ}{dt} = k_{12}S - k_{21}J - k_{33}JM \\ \frac{dS}{dt} = k_{13}M - k_{22}S - k_{34}SH \\ \frac{dM}{dt} = k_{14}H - k_{23}M - k_{35}MT \\ \frac{dH}{dt} = k_{15}T - k_{24}H - k_{31}HJ \\ \frac{dT}{dt} = k_{11}J - k_{25}T - k_{32}TS \end{cases} \tag{12}$$

According to our definition, the system's signals can propagate in both directions. Based on the system's topological structure, we define the backward equation containing signal propagation term:

$$\frac{dE(t)}{dt} = \overset{1}{K}_1 \overset{1}{E}(t) - \overset{0}{K}_2 \overset{0}{E}(t) - \overset{2}{K}_3 \overset{0}{E}(t) \overset{2}{E}(t) + Input(t) \tag{13}$$

Although this structure is the result of reversing Figure 1a, their topological link relationship has not changed. The equations obtained in Figure 4b are consistent with the equations in Figure 4a in mathematical structure, and only the order of elements has changed.

When a signal is input into Equation (10) (Figure 4a), an output signal is obtained. This input-output relationship is a nonlinear function, corresponding to the f_n function in Figure 1c. Similarly, Equation (12) (Figure 4b) is also equivalent to a nonlinear function for the input signal, corresponding to the $\widetilde{f_n}$ function in Figure 1d.

This mirrored topological structure ensures that Figures 2a and 2b align with the forward and backward propagation diagrams shown in Figures 1c and 1d, respectively.

In the symmetric differential equation system we constructed, a set of multivariate coupled ordinary differential equations (ODEs) is involved, within which the variables exhibit intricate interdependencies. These dependencies are not only directional but also hierarchical in nature, manifesting a chain-like propagation pattern. In such systems, changes in a particular variable do not instantaneously influence all others; rather, the effects are transmitted progressively along directed paths, undergoing accumulation, attenuation, or feedback modulation during the propagation. To characterize this feature, we introduce the concept of a causal interaction chain and further define a novel structural metric termed causal distance.

Causal distance refers to the minimum number of intermediate variables required for one variable to influence another within a predefined causal propagation graph. For

example, if the derivative of variable J directly depends on variable T, the causal distance between them is defined as 1. Conversely, if variable H affects J only indirectly through T, then the causal distance from H to J is 2. This unified perspective not only enhances our understanding of the intrinsic structural dynamics of ODE systems but also provides a rich toolkit for modeling complex networked systems. Specifically, two key insights emerge: (1) variables with shorter causal distances exhibit more direct influence and elicit faster system responses; (2) variables connected through longer causal chains tend to produce delayed responses, making them more susceptible to accumulation effects.

In Equation (7), two loops are present. The first loop represents a generative loop among elements ($J \to S \to M \to H \to T \to J$), while the third term embodies an inhibitory loop ($J \to M \to T \to S \to H \to J$).

Figure 5 illustrates the causal distances within these two loops, derived from both the forward and reverse formulations of the causal equations. Specifically, Figure 5a presents the causal distance matrix for the generative loop in the forward causal equation; Figure 5b corresponds to the same loop in the reverse causal equation; Figure 5c displays the causal distances for the inhibitory loop in the forward equation; and Figure 5d shows the inhibitory loop in the reverse formulation. As observed, the causal distances between corresponding elements are perfectly symmetric in both formulations, reflecting the inherent structural symmetry of the differential equation system we designed.

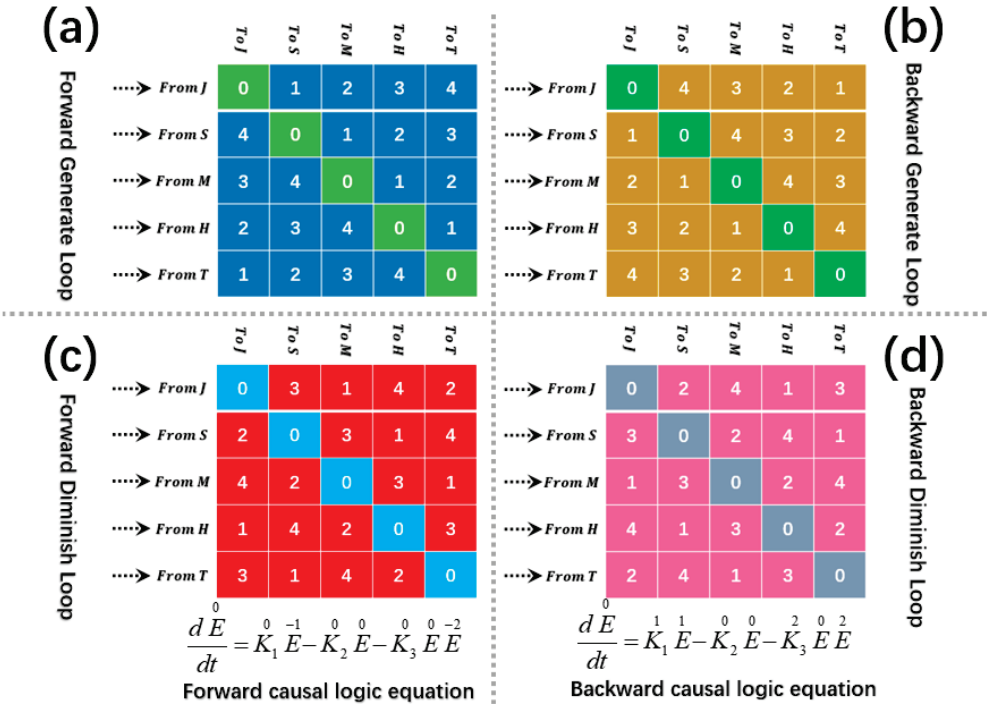


Figure 5. Causal distance between elements in the forward and reverse systems. (**a**) Causal distance table in the generative loop in the forward propagation equation. (**b**) Causal distance table in the generative loop in the backward propagation equation. (**c**) Causal distance table in the diminish loop in the forward propagation equation. (**d**) Causal distance table in the diminish loop in the backward propagation equation.

Building upon this theoretical framework, we extend the concept of causal distance to describe the interactions between system parameters and state variables. For instance, in Figure 4a, variable J directly depends on variable T, with a coupling coefficient k_{11}; meanwhile, T is regulated by another parameter k_{15} through its effect on variable H. Under

this configuration, a causal distance can be defined between system parameters and state variables. Specifically, the causal distance between k_{11} and J is considered to be 1, indicating a rapid and direct response of J to perturbations in k_{11}. In contrast, since the effect of k_{15} on J is mediated by T, it entails a longer causal path and hence a lower sensitivity of J to changes in k_{15}.

This leads to an intuitive conclusion: the influence of directly coupled parameters (e.g., k_{11}) on system variables (e.g., J) is more immediate and pronounced, while parameters involved in multi-stage indirect dependencies (e.g., k_{15}) exhibit weaker impacts due to transmission attenuation. This observation lays the groundwork for a causal-distance-based theory, where both inter-variable dependencies and parameter-variable couplings can be evaluated in terms of their causal distances. In general, shorter causal distances imply more direct and stronger regulatory effects, whereas longer distances tend to yield diminished sensitivities due to intermediate information loss.

This theoretical framework offers several key advantages:

1. Reduction in analytical complexity: By assigning causal distances to each dependency relationship, the system's analysis can adopt a strategy analogous to Principal Component Analysis (PCA) [18], ranking and filtering the contributing factors. Variables or parameters with short causal distances often constitute the core drivers of the system, while those with long distances and marginal influence can be safely ignored during preliminary modeling, thereby simplifying system analysis significantly.
2. Quantification of influence strength: Causal distance provides not only a qualitative description of transmission paths but also a means for quantitative assessment when combined with coupling coefficients. In the aforementioned example, the direct dependency of J on k_{11} results in a strong and rapid effect, while the impact of k_{15} is attenuated due to intermediate transmission. This approach introduces a quantitative scale for dynamical systems, facilitating accurate identification of key control parameters in sensitivity analysis or parameter tuning tasks.
3. Unified structural representation: Extending causal distance to cover parameter-variable dependencies enables a unified analytical tool that can describe both the internal causal feedback chains among variables and the regulatory effects of system parameters. This unification deepens our understanding of the "structural lag" phenomenon in complex systems and provides a theoretical foundation for the design of control strategies.

In summary, the development of the causal distance framework elucidates the mechanisms of direct and indirect interactions among variables and provides an intuitive and effective metric for assessing sensitivity and influence strength. By focusing on dominant causal chains and excluding negligible effects, the model's analytical clarity and explanatory power are greatly enhanced. This methodology shares conceptual similarities with PCA, emphasizing the ranking and selection of impactful factors to reduce system dimensionality and extract principal driving patterns, thereby improving the tractability and efficiency of complex system research.

4. Experiment

4.1. Forward and Backward Signal Design

In the framework of symmetric differential equations, we define the system's zero state as the state in which it resides at its fixed point. When an external signal is injected,

the system deviates from this fixed point, and the resulting deviation is regarded as the corresponding output signal. This can be formally defined as follows:

$$D(t) = E(t) - B_0 \tag{14}$$

where D(t) denotes the deviation from the fixed point, E(t) represents the real-time signal of the system element, and B_0 is the fixed point of the system. The fixed point B_0 can be determined by solving the following equation:

$$B_0 \approx \frac{\overline{K_1} - \overline{K_2}}{\overline{K_3}} \tag{15}$$

In Equation (15), $\overline{K_1}$ $\overline{K_2}$ $\overline{K_3}$, denote the average values of the three groups of parameters described earlier. When the parameters within each group are identical, Equation (15) provides an exact solution. Although Equation (15) only gives an approximate fixed point of Equation (9), it reveals the qualitative relationship between the fixed point B and the parameters K_1, K_2 and K_3. Specifically, B_0 is positively correlated with K_1; as K_1 increases monotonically. Conversely, B_0 is negatively correlated with K_2 and K_3; an increase in either K_2 or K_3 results in a monotonic decrease in B_0.

Similarly, $\hat{D}(t)$ is determined by the back-propagated element value $\hat{E}(t)$ and the back-propagated fixed point $\hat{B}_0$.

$$\hat{D}(t) = \hat{E}(t) - \hat{B}_0 \tag{16}$$

According to our design, the signal propagating through the forward network is D(t), and the signal propagating through the backward network is $\hat{D}(t)$ By comparing the differences between these two signals, we can effectively train the neural network.

4.2. Two Different Learning Methods

In our previous work, we proposed two distinct approaches for training neural networks. The first method involves modifying the system's fixed points [19], while the second adjusts the connection weights between neurons [20]. Both methods rely on error signals propagating backward through the network, a mechanism we refer to as Backward Signal Propagation (BSP). Through BSP, system parameters can be precisely adjusted based on the signals propagated within the network. Below, we briefly introduce how the two types of signals in the system can be utilized to tune parameters.

4.2.1. First Learning Method: Adjusting the System's Fixed Points

The first training approach modifies the fixed points of the system. As discussed earlier, the fixed points of the system are tightly coupled to three groups of parameters defined in Equation (9). Therefore, we proposed a method based on distributed PID control to independently regulate each group of parameters, and experimental results validated the success of this approach.

Based on the forward and reverse signals propagated within time T, we can define a correlation variable G_1.

$$G_1 = \int_0^T D(t)dt \cdot \int_0^T \hat{D}(t)dt \tag{17}$$

In Equation (17), G_1 is the product of the forward signal integral and the backward signal integral, which reflects the difference between the forward and backward signals. If the two signals have opposite signs, it means that the parameters at the corresponding

position should be adjusted to reduce the output; if the signs are the same, it means that the parameters at the corresponding position should be adjusted to increase the output.

Since G_1 may exceed a certain limit, we use the inverse tangent function (other similar functions are also possible) to limit it and obtain G_2

$$G_2 = atan(G_1 * kt)/kt \tag{18}$$

kt is the adjustment parameter, and G_2 is the adjusted correlation value. The parameter can be adjusted based on G_2.

$$K_{3_new} = K_{3_old} \cdot exp(-G_2) \tag{19}$$

In Equation (19), we used the parameter K_3 as an example to illustrate the tuning process. The correspondence between the adjustment components and the system parameters was determined according to the causal distance between them. Initial results confirmed the effectiveness of the approach. Extending this method to multiple parameter groups requires the implementation of distributed PID control strategies.

Figure 6 presents the training results on the MNIST and Fashion-MNIST datasets, with the number of neurons in each layer set to {784,839,283,96,32,10}. The accuracy on both datasets reaches approximately 50%. A notable difference, however, is that training on MNIST exhibits greater stability, which can be attributed to the intrinsic characteristics of the dataset. By contrast, Fashion-MNIST was deliberately designed to introduce higher training complexity, resulting in larger fluctuations in the accuracy curve. Importantly, the relatively low accuracy observed here does not stem from limitations of the training method itself, but rather from structural deficiencies in the system's topology. To address this, we further introduced a second training method, which simultaneously improves both the training process and the system architecture.

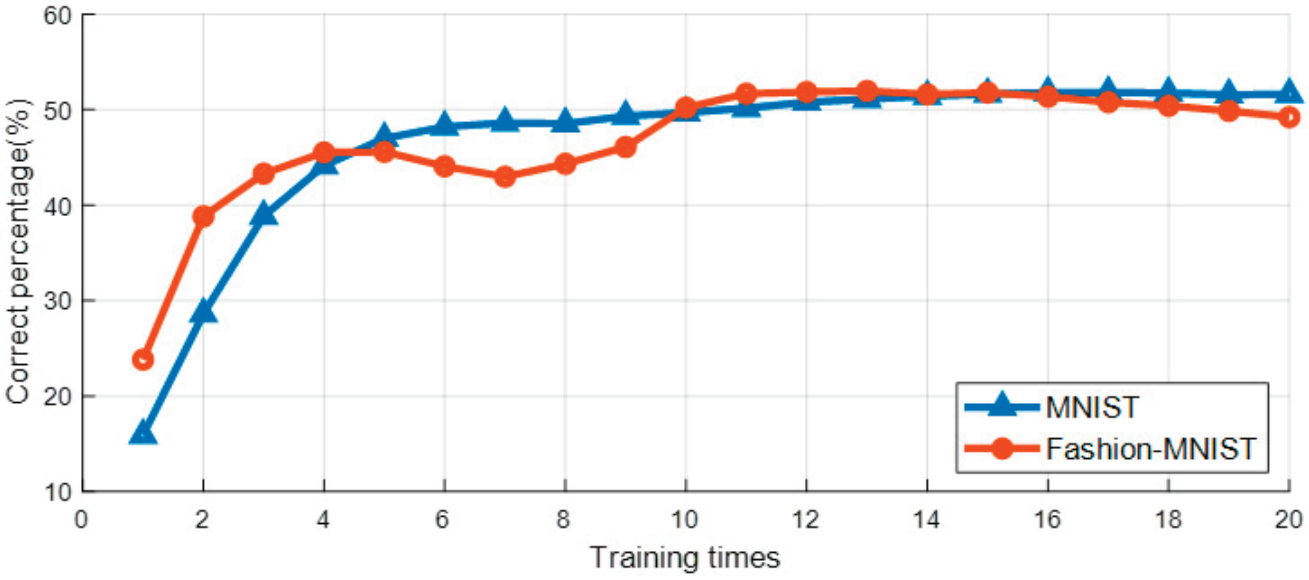


Figure 6. Accuracy curves trained on MNIST and Fashion MNIST.

4.2.2. Second Learning Method: Adjusting Neuronal Connections

The second approach focuses on tuning the connection parameters between neurons, akin to the training process of traditional multilayer perceptrons (MLPs). In particular, our analysis of Figure 2b led to a simple and efficient parameter update rule (Equation (2)). A

key limitation of this rule in conventional architectures is the inability to propagate error signals globally. However, within the BSP framework, the signal $\hat{D}(t)$ naturally serves as a globally propagated error signal. This enables each neuron to apply the learning rule from Equation (2) effectively. Assuming that the inter-neuronal connection weights are denoted by C, the forward propagation equation of the system can be written as Equation (20).

$$\frac{d\overset{0}{E}(t)}{dt} = \overset{0}{K}_1\overset{-1}{E}(t) - \overset{0}{K}_2\overset{0}{E}(t) - \overset{0}{K}_3\overset{0}{E}(t)\overset{-2}{E}(t) + C \cdot Input(t) \tag{20}$$

The backward propagation equation is modified as follows:

$$\frac{d\overset{0}{E}(t)}{dt} = \overset{1}{K}_1\overset{1}{E}(t) - \overset{0}{K}_2\overset{0}{E}(t) - \overset{2}{K}_3\overset{0}{E}(t)\overset{2}{E}(t) + C \cdot Input(t) \tag{21}$$

In Equations (20) and (21), the coefficients C remains consistent across corresponding positions, ensuring causal consistency before and after connections. To maintain a one-to-one correspondence between cause and effect, each neuron's synaptic connections are strictly one-to-one. Compared with the fully connected structure of traditional MLPs, the connections in the Wuxing neural network are significantly sparser.

According to Equation (2), we can also give a similar training method. Assume that the system has n layers, where the forward output signal of the i_{th} layer neuron is $D_i(t)$, and similarly, the backward output signal of the i_{th} layer neuron is $\hat{D}_i(t)$. If the calculation time is T, we can define a variable G_1:

$$G_1 = \int_0^T D_i(t)dt \cdot \int_0^T \hat{D}_{i+1}(t)dt \tag{22}$$

In Equation (22), G_1 represents the causal quantity of the connection parameters between the neurons in the i_{th} layer and the neurons in the $i + 1_{th}$ layer.

Since G_1 may exceed a certain limit, we use the inverse tangent function (other similar functions are also possible) to limit it and obtain G_2, as shown in Eqation (18).

kt is the adjustment parameter, G_2 is the adjusted correlation value. The parameter can be adjusted based on G_2.

$$C_{new} = C_{old} + G_2 \tag{23}$$

Based on Equation (23). and using the same network structure as in Method 1, we trained the MNIST dataset and achieved an accuracy of approximately 50% (corresponding to the blue curve in Figure 7). If the number of neurons is increased to {784,1048,353,119,40,10}, the accuracy can be further improved to approximately 70% (corresponding to the red curve in Figure 7). However, in this configuration, each neural node is connected to only one node in the subsequent layer. If instead each node is allowed to connect with multiple nodes in the next layer, the network with {784,300,300,10} neurons can achieve an accuracy of 82% (corresponding to the yellow curve in Figure 7). This result demonstrates that under multi-connection settings, we are able to employ fewer neurons, thereby effectively reducing computational complexity while also achieving faster training speed.

Figure 8 illustrates the forward signal propagation path and the backward error signal propagation path. In the two training methods we designed, a new variable G_1 is constructed through the cooperative interaction between the forward-propagating signal $D(t)$ and the backward-propagating error signal $\hat{D}(t)$ This variable is defined as the error modulation signal. Due to the symmetry of the network structure, G_1 can naturally

propagate throughout the network topology, thereby enabling effective adjustment of parameters at different locations. Because the construction of G_1 is directly based on the topological relationships among local signals, it accurately reflects the true impact of each parameter on the global error. Consequently, the adjustment method based on G_1 offers both strong locality and significant advantages in efficiency and accuracy.

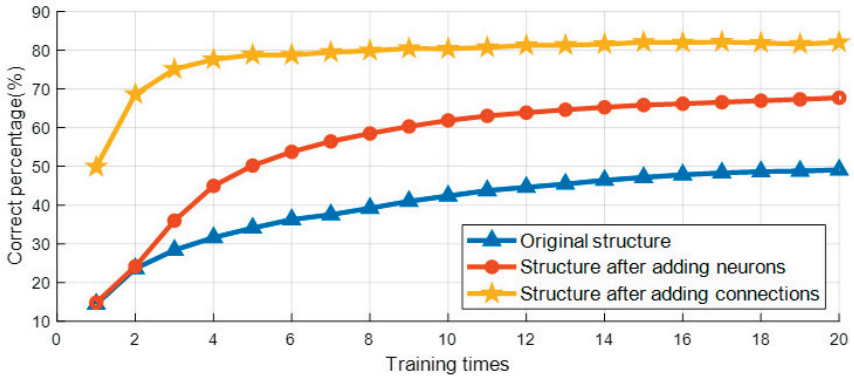


Figure 7. Accuracy curves trained on different structures.

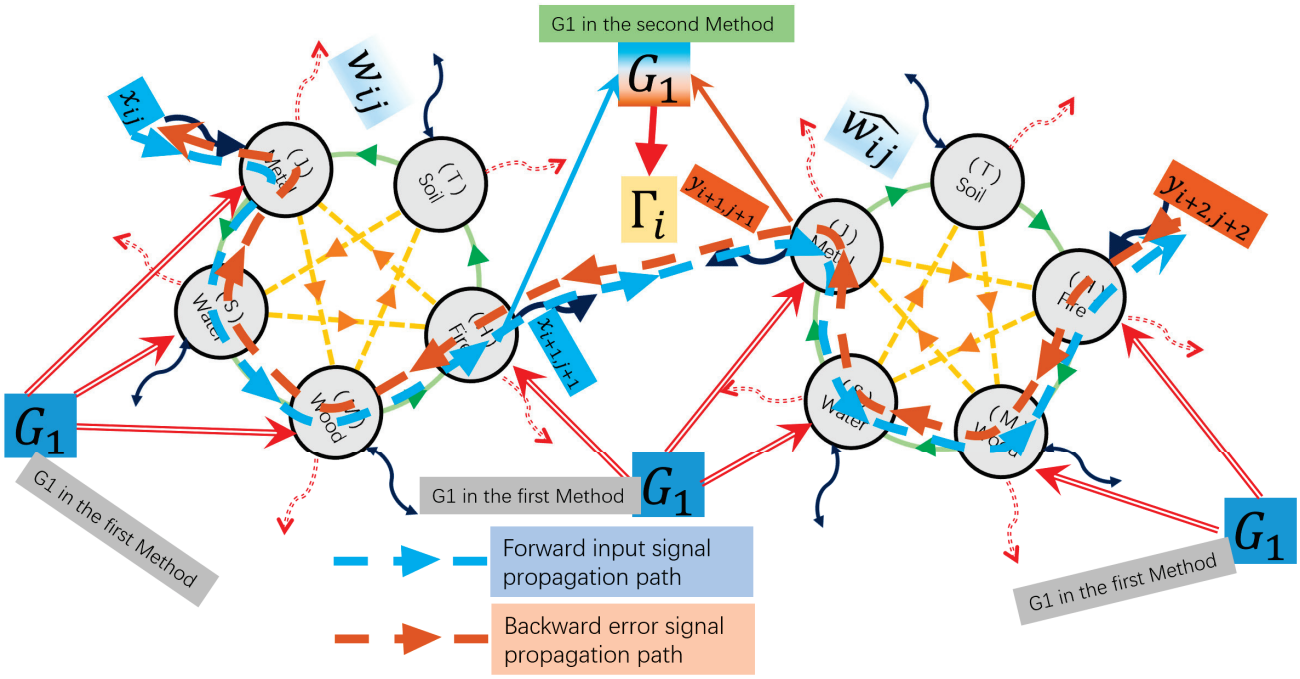


Figure 8. Forward and backward signal propagation paths. The red dashed line represents the forward input signal propagation path, while the blue dashed line represents the backward error propagation path. Due to the reversibility of the system, these two paths are completely symmetrical.

It should be emphasized that in the generative loop, which serves as the dominant loop of the system, each element in the causal chain simultaneously experiences both the forward and backward signals. The specific definition of G_1 differs between the two training methods: in method 2, for example, the parameters Γ_i to be adjusted are located between two neurons, so G_1 must be defined using the signals from both neurons ($x_{i+1,j+1}$ and $y_{i+1,j+1}$). These two training methods are not the end point; rather, they demonstrate that, based on the BSP framework, a wider variety of learning mechanisms can be designed to further enhance training performance.

In summary, according to the signal propagation process, we can give the following training steps:

1. Feed the input signal into the network. Compute the forward propagation of the signal through all layers.
2. Compute the error signal based on the difference between the output and the target at the output layer.
3. Propagate the error signal backward through the network.
4. At each node, combine the forward signal and the backward error signal to construct the modulation variable G1.
5. Update the system parameters according to the computed modulation variable G1.
6. Repeat steps 1–5 until convergence criteria are satisfied (e.g., error or maximum epochs).

5. Summary

In this work, we conducted an in-depth analysis of the two core mechanisms underpinning the effectiveness of the backpropagation algorithm: differentiation and the chain rule. The former enables a mathematical characterization of causal influence among variables in any differentiable function, while the latter allows this causal pathway to be propagated layer by layer in complex network architectures, thus enabling system-level error feedback. This combination of formal rigor and structural adaptability is what establishes BP as a dominant training strategy in deep learning.

To propose an alternative approach that combines training efficiency with biological plausibility, we introduced a novel learning framework based on symmetric differential equations, termed Backward Signal Propagation (BSP). Our contributions can be summarized in two main innovations:

First, by leveraging the local differentiability of differential equations, we express causal effects without requiring explicit gradient computation. This allows the training process to be naturally embedded within the dynamical evolution of signal propagation.

Second, we introduced the concept of causal distance and demonstrated its consistency between the forward and backward subsystems (i.e., causal distance invariance) in symmetric network structures. This provides theoretical support for the global propagation of error signals.

Within this network architecture, the system operates based on two types of signals only: the forward-propagating input signal and the backward-propagating error signal. From these, we generate the error modulation signal G_1, which is used to locally and precisely adjust all system parameters. Causal distance not only provides the geometric foundation for error signal propagation but also indicates the preferential directions for parameter updates. Experimental results confirm that adjusting parameters closer (in causal distance) to G_1 leads to more pronounced training responses, validating the local efficiency of the BSP method.

Overall, the BSP framework plays a foundational role for symmetric differential networks, akin to the role of BP in traditional multilayer perceptrons. Its introduction not only offers a biologically feasible training mechanism but also opens a new pathway toward constructing neural systems that are highly reversible, structurally symmetric, and signal-driven. Future research will focus on further optimizing the engineering realization of BSP, including improving training efficiency, enhancing generalization capabilities, and extending its application to more complex network architectures and task scenarios—ultimately paving the way for its deployment in a broad range of intelligent systems.

Author Contributions: Conceptualization, K.J.; Methodology, K.J.; Validation, K.J.; Formal analysis, K.J.; Investigation, K.J.; Writing—original draft, K.J.; Writing—review & editing, K.J.; Supervision, Z.F.; Project administration, K.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data is available in a publicly accessible repository. It can be downloaded from: https://yann.lecun.org/exdb/mnist/index.html (accessed on 4 August 2025).

Acknowledgments: Thanks to China Scholarship Council (CSC) for their support during the pandemic, which allowed me to get through those difficult days and give me the opportunity to put my past ideas into practice, ultimately resulting in the article I am sharing with you today.

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

Abbreviations

The following abbreviations are used in this manuscript:

BSP	Backward signal propagation
BP	Backpropagation
MLP	Multilayer perceptrons
MNIST	Modified National Institute of Standards and Technology database

References

1. Rosenblatt, F. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychol. Rev.* **1958**, *65*, 386. [CrossRef]
2. Marvin, M.; Seymour, A.P. *Perceptrons*; MIT Press: Cambridge, MA, USA, 1969; Volume 6, p. 7.
3. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
4. Hochreiter, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **1998**, *6*, 107–116. [CrossRef]
5. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.-F. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; IEEE: Piscataway, NJ, USA, 2009.
6. Raina, R.; Madhavan, A.; Ng, A.Y. Large-scale deep unsupervised learning using graphics processors. In Proceedings of the ICML '09: Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009.
7. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]
8. Bottou, L. Large-scale machine learning with stochastic gradient descent. In Proceedings of the COMPSTAT'2010: 19th International Conference on Computational Statistics, Paris, France, 22–27 August 2010; Keynote, Invited and Contributed Papers; Springer: Berlin/Heidelberg, Germany, 2010.
9. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
10. Hinton, G. The forward-forward algorithm: Some preliminary investigations. *arXiv* **2022**, arXiv:2212.13345. [CrossRef]
11. Lillicrap, T.P.; Santoro, A.; Marris, L.; Akerman, C.J.; Hinton, G. Backpropagation and the brain. *Nat. Rev. Neurosci.* **2020**, *21*, 335–346. [CrossRef] [PubMed]
12. Crick, F. The recent excitement about neural networks. *Nature* **1989**, *337*, 129–132. [CrossRef] [PubMed]
13. Stork. Is backpropagation biologically plausible? In Proceedings of the International 1989 Joint Conference on Neural Networks, Washington, DC, USA, 18–22 June 1989; IEEE: Piscataway, NJ, USA, 1989.
14. Kun, J. A Neural Network Framework Based on Symmetric Differential Equations. *ChinaXiv* **2024**, ChinaXiv:202410.00055.
15. Jiang, K. From Propagator to Oscillator: The Dual Role of Symmetric Differential Equations in Neural Systems. *arXiv* **2025**, arXiv:2507.22916.
16. Izhikevich, E.M. *Dynamical Systems in Neuroscience*; MIT Press: Cambridge, MA, USA, 2007.
17. Olver, P.J. *Equivalence, Invariants and Symmetry*; Cambridge University Press: Cambridge, UK, 1995.
18. Abdi, H.; Williams, L.J. Principal Component Analysis. *Wiley Interdiscip. Rev. Comput. Stat.* **2010**, *2*, 433–459. [CrossRef]

19. Jiang, K. A Neural Network Training Method Based on Distributed PID Control. *Symmetry* **2025**, *17*, 1129. [CrossRef]
20. Jiang, K. A Neural Network Training Method Based on Neuron Connection Coefficient Adjustments. *arXiv* **2025**, arXiv:2502.10414.

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.

algorithms

MDPI

Article

A Resilient Energy-Efficient Framework for Jamming Mitigation in Cluster-Based Wireless Sensor Networks

Carolina Del-Valle-Soto [1,*], José A. Del-Puerto-Flores [1], Leonardo J. Valdivia [1], Aimé Lay-Ekuakille [2] and Paolo Visconti [2]

[1] Facultad de Ingeniería, Universidad Panamericana, Álvaro del Portillo 49, Zapopan 45010, Jalisco, Mexico; jpuerto@up.edu.mx (J.A.D.-P.-F.); lvaldivia@up.edu.mx (L.J.V.)
[2] Department of Innovation Engineering, University of Salento, 73100 Lecce, Italy; aime.lay.ekuakille@unisalento.it (A.L.-E.); paolo.visconti@unisalento.it (P.V.)
* Correspondence: cvalle@up.edu.mx

Abstract

This paper presents a resilient and energy-efficient framework for jamming mitigation in cluster-based wireless sensor networks (WSNs), addressing a critical vulnerability in hostile or interference-prone environments. The proposed approa ch integrates dynamic cluster reorganization, adaptive MAC-layer behavior, and multipath routing strategies to restore communication capabilities and sustain network functionality under jamming conditions. The framework is evaluated across heterogeneous topologies using Zigbee and Bluetooth Low Energy (BLE); both stacks were validated in a physical testbed with matched jammer and traffic conditions, while simulation was used solely to tune parameters and support sensitivity analyses. Results demonstrate significant improvements in Packet Delivery Ratio, end-to-end delay, energy consumption, and retransmission rate, with BLE showing particularly high resilience when combined with the mitigation mechanism. Furthermore, a comparative analysis of routing protocols including AODV, GAF, and LEACH reveals that hierarchical protocols achieve superior performance when integrated with the proposed method. This framework has broader applicability in mission-critical IoT domains, including environmental monitoring, industrial automation, and healthcare systems. The findings confirm that the framework offers a scalable and protocol-agnostic defense mechanism, with potential applicability in mission-critical and interference-sensitive IoT deployments.

Keywords: wireless sensor networks; jamming; performance metrics; Internet of Things

1. Introduction

WSNs have become indispensable in a variety of applications, including environmental monitoring, industrial control, and healthcare [1]. Their decentralized nature and reliance on shared communication channels make them highly susceptible to security threats, particularly jamming attacks. Jamming attacks disrupt communication by introducing interference, leading to packet loss, retransmissions, and reduced energy efficiency—ultimately jeopardizing the network's overall performance and resilience [2].

In a previous study [3], we addressed the challenge of detecting jamming in WSNs through an algorithm that identifies affected zones by analyzing key metrics such as retransmissions, routing tables, and energy consumption. Building on this foundational work, this paper focuses

on the mitigation of jamming attacks by proposing a novel algorithmic framework designed to enhance network stability, minimize energy impact, and ensure reliable communication.

The proposed mitigation algorithm leverages dynamic clustering, energy-aware routing, and adaptive transmission power control to reduce the impact of jamming. Key features of the algorithm include the identification of nearby unaffected nodes [4], the redistribution of cluster resources, and the selection of optimal alternative routes based on performance metrics such as hops, retransmissions, and energy consumption. Additionally, the framework integrates a hierarchical approach to adjust node operational modes (e.g., low power or isolated) and optimize network-wide routing strategies, thereby maintaining connectivity and minimizing disruptions in affected areas.

The impact of jamming on WSNs is profound due to the networks' resource constraints, including limited power, low computational capability, and the use of low-power radio transceivers. Constant and deceptive jamming are easier to detect because of their continuous interference patterns, but they quickly drain the attacker's energy. In contrast, random and reactive jamming are more energy-efficient from the attacker's perspective, making them more challenging to mitigate. Traditional anti-jamming techniques, such as frequency hopping and spread spectrum methods, are not always viable in WSNs due to sensor nodes' hardware and energy limitations. This necessitates the development of advanced mitigation strategies, including machine learning-based detection algorithms, adaptive transmission scheduling, and energy-efficient countermeasures. Recent research also explores cross-layer defense mechanisms, which integrate physical, MAC, and network layer solutions to improve resilience against jamming.

This work contributes to the state-of-the-art in jamming mitigation for WSNs by providing a comprehensive, real-time response mechanism that ensures network resilience under adversarial conditions [5,6]. The proposed approach is evaluated against established benchmarks, demonstrating its effectiveness in maintaining network stability, extending operational lifetime, and reducing the energy overhead imposed by jamming attacks.

Figure 1 presents a comprehensive conceptual framework for understanding jamming attacks in WSNs. The diagram is organized into five critical dimensions: the definition of jamming, the types of jamming attacks, the techniques commonly used to perform such attacks, the available mitigation strategies, and the resulting security impacts. This structured approach facilitates an integrated understanding of the threat landscape posed by intentional wireless interference in sensor-based communication systems.

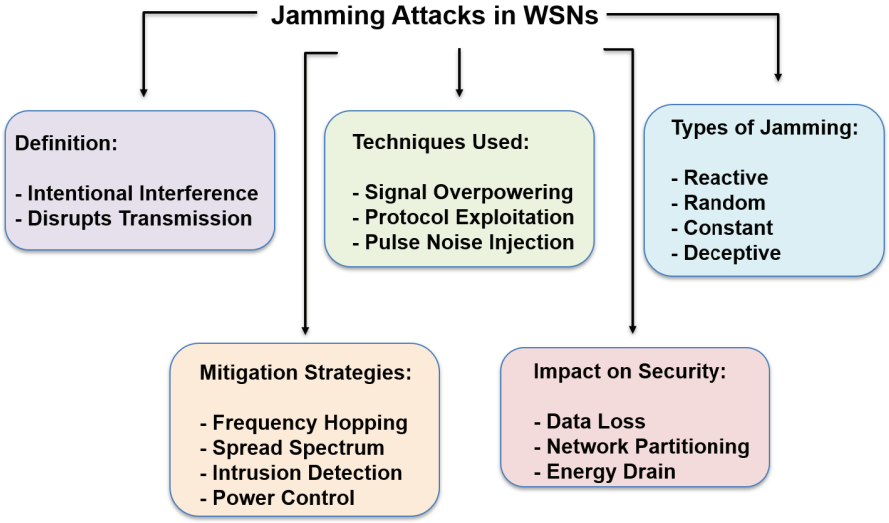


Figure 1. Jamming attacks in wireless sensor networks.

Jamming is first defined as a form of intentional interference designed to disrupt wireless transmissions. This attack operates primarily at the physical layer of communication protocols, where it obstructs the ability of nodes to send or receive messages. Unlike random noise or unintentional interference, jamming is deliberate, often strategically deployed to degrade or deny network services.

The figure categorizes jamming into four main types: constant, random, reactive, and deceptive jamming. Constant jamming involves the uninterrupted transmission of signals that flood the communication channel. Random jamming alternates between active and idle states to reduce the likelihood of detection while maintaining its disruptive effect. Reactive jamming is more sophisticated, activating only upon detection of legitimate transmissions, which makes it both energy-efficient and more difficult to trace. Deceptive jamming seeks to confuse nodes by sending falsified or misleading communication signals, effectively manipulating protocol behavior.

Several techniques are commonly used to implement jamming attacks. Signal overpowering is a brute-force method in which the attacker transmits at a higher power level to dominate the medium. Protocol exploitation involves manipulating or abusing weaknesses in network communication protocols to trigger denial-of-service behavior. Pulse noise injection, in contrast, introduces short bursts of high-energy noise to intermittently disturb communication, often while remaining covert.

To counteract these attacks, the diagram lists four mitigation strategies. Frequency hopping involves dynamically changing the communication frequency, making it difficult for the jammer to consistently interfere. Spread spectrum techniques, such as direct sequence or frequency hopping spread spectrum, increase resistance to jamming by distributing the signal over a wide frequency band. Intrusion detection systems (IDS) are employed to monitor anomalies in network behavior that may indicate the presence of a jamming attack. Power control mechanisms help optimize the transmission range and power levels to reduce exposure and energy waste in hostile environments.

Finally, this figure underscores the severe impact that jamming attacks can have on network security and functionality. These include data loss due to disrupted transmission, network partitioning caused by broken communication links between nodes, and energy drain as a result of repeated retransmissions or excessive power usage. Collectively, these consequences can compromise the integrity, availability, and longevity of the network.

The proposed method offers several significant advantages over conventional jamming mitigation techniques in WSNs. First, by combining dynamic cluster reorganization, adaptive MAC-layer behavior, and multipath routing strategies, the framework ensures continuous connectivity and robustness against diverse forms of interference. This integration reduces the likelihood of network partitioning and minimizes retransmissions, leading to measurable improvements in PDR and end-to-end delay. Furthermore, the approach demonstrates energy efficiency through adaptive power control and decentralized decision-making, which prolongs network lifetime and prevents unnecessary resource consumption. The evaluation conducted with both Zigbee and BLE technologies confirms its protocol-agnostic applicability, with BLE-based deployments showing particularly high resilience. Additionally, the method outperforms traditional static or centralized solutions by offering real-time adaptability, scalability across heterogeneous topologies, and superior performance when integrated with hierarchical routing protocols such as LEACH.

The remainder of this paper is organized as follows: Section 2 discusses related work on jamming detection and mitigation strategies. Section 3 details the proposed methodology with the mitigation algorithm, including its components and operational

logic. Section 4 presents the experimental setup and performance evaluation. Finally, Section 5 concludes with a summary of findings and potential future directions.

2. Related Work

The detection and mitigation of jamming in WSNs have been extensively studied, with various strategies proposed to address the challenges of interference and network disruption. Alcaraz and Lopez in [7] provided a comprehensive review of jamming attacks and countermeasures, emphasizing the importance of integrating real-time detection mechanisms with energy-efficient mitigation strategies. Their work primarily focused on static routing protocols, which may struggle in dynamic or mobile network environments. In contrast, our proposed algorithm incorporates dynamic clustering and adaptive routing to enhance resilience, addressing the limitations of static approaches by allowing real-time reconfiguration of network topology based on jamming conditions.

Furthermore, Kumar and Kumar in [2] explored intrusion detection systems for WSNs, proposing lightweight monitoring frameworks that effectively detect jamming attacks through energy and retransmission metrics. While their approach was effective in identifying attacks, their reliance on centralized processing introduces potential bottlenecks. Our work differs by leveraging decentralized decision-making and adaptive node behavior, minimizing delays and ensuring scalable performance. Other studies, such as [8], focused on applying security mechanisms to smart grids, showcasing the applicability of anti-jamming strategies across domains but lacking detailed implementations of dynamic clustering and route optimization. By integrating these advanced features, our algorithm outperforms existing methods in maintaining Packet Delivery Ratio (PDR) and reducing end-to-end delay, as validated in both simulated and real-world scenarios.

Table 1 highlights key contributions and limitations of existing works on jamming mitigation in wireless sensor networks (WSNs), offering a comparative analysis against the proposed approach. Notably, prior studies such as [2,7] emphasize detection mechanisms and lightweight frameworks but lack dynamic adaptability and suffer from centralized bottlenecks. Others, like [8], focus on security applications in specific domains, such as smart grids, without addressing the broader needs of WSNs under dynamic jamming scenarios. The table underscores the gap in integrating dynamic clustering and adaptive routing, which are pivotal features of the proposed algorithm, ensuring resilience, scalability, and real-time adaptability in diverse network conditions.

Table 1. Comparison of related work on jamming mitigation in WSNs.

Reference	Key Contribution	Limitations
[2]	Lightweight intrusion detection systems for energy metrics.	Centralized processing introduces bottlenecks.
[7]	Comprehensive review of jamming detection and countermeasures.	Focused on static routing; lacks dynamic adaptability.
[8]	Security mechanisms applied to smart grids.	Lacks dynamic clustering and adaptive routing implementations.
[9]	Geographic adaptive fidelity for energy-aware routing.	Ineffective under mobile or dynamic jamming conditions.
[10]	WSN applications with emphasis on resilience.	Limited implementation details for jamming mitigation strategies.
[11]	Energy-efficient routing algorithm resilient to jamming attacks.	Limited scalability in large heterogeneous networks.
[12]	Defense model using signal strength and frequency analysis.	Detection accuracy drops under complex mobility models.
[13]	Deep learning-based detection of jamming and anomaly traffic.	High training complexity and computational overhead.
[14]	Anti-jamming method using frequency hopping and adaptive modulation.	Performance degrades in dense WSN environments.

Table 1. *Cont.*

Reference	Key Contribution	Limitations
[15]	Real-time jamming mitigation using cross-layer adaptive framework.	Requires precise synchronization and cross-layer coordination.
[16]	Lightweight cooperative defense using neighbor trust scoring.	Assumes trusted environment and stable topology.
[17]	Dynamic spectrum access for interference-aware WSNs.	Not robust to rapidly changing interference patterns.
[18]	Game-theoretic model for WSN security against jamming.	Convergence depends on accurate attacker behavior modeling.
[19]	Hybrid clustering and power control scheme for jamming resilience.	Trade-off between energy saving and route optimality.

2.1. Energy-Efficient Protocols Against Attacks

It is essential to deepen the study of the vulnerability of MAC protocols in WSNs to jamming attacks, which are designed to be energy-efficient. These attacks, operating at the data link layer, aim to disrupt node communication without consuming large amounts of energy, making them particularly dangerous in environments where energy efficiency and autonomy are priorities. To address this, various researchers have explored the behavior of different MAC protocols under such adversarial conditions. In their analysis, Law et al. (2009) in [20] propose to exploit the characteristics of different protocols by simulating jamming algorithms, such as cluster-based periodic jamming (PCJ) for Sensor-MAC (S-MAC), slot-based periodic jamming (PSJ) for Lightweight MAC (LMAC), and low-power listener-based jamming (LPLJ) for Berkeley-MAC (B-MAC), revealing that the LMAC protocol shows better resistance to attacks. Building upon these insights, years later Ettouijri et al. (2014) in [21] analyze the S-MAC, LMAC, and B-MAC protocols, agreeing that LMAC, with some modifications to improve its robustness—such as introducing randomness and variability in the slot sizes and start times—emerges as the most resilient protocol among those mentioned against energy-efficient jamming attacks.

In a complementary line of research, Mihajlov et al. (2014) in [22] conduct a comparative analysis of three MAC protocols: IEEE 802.15.4 [23] MAC, Time-MAC (T-MAC), and S-MAC. Their study focuses on critical performance metrics such as throughput, latency, network load, and energy consumption under both normal conditions and attack scenarios. This evaluation utilizes simulations of ZigBee node networks. Their findings reveal that T-MAC demonstrates superior performance and energy efficiency, even amid adversarial conditions. This advantage is attributed to T-MAC's dynamic adjustment of active periods and its effective management of transmission termination, which minimizes idle listening and reduces the likelihood of packet collisions.

Expanding the scope beyond traditional WSNs, in Low Power Wide Area Network (LPWAN) applications, protocols like LoRaWAN are specifically engineered to enable efficient wireless connectivity for battery-operated devices while maintaining minimal power consumption. Despite its inherent energy efficiency, energy depletion attacks (EDAs) pose a substantial threat to sensors due to their ability to increase data transmission (TX) or reception (RX) activity in sensors or generate Denial of Service (DoS) attacks. In response to this challenge, Proto et al. (2024) in [24] developed a Lightweight Architecture for Detecting EDAs (LADE), which prioritizes energy efficiency and is implemented directly in the sensors to minimize network energy consumption, achieving a power consumption of only 0.3% additional energy due to its lightweight architecture.

At the same time, the literature reports various proposals for innovative protocols designed to improve security against jamming attacks. For example, Tang et al. (2011) in [25] present an Efficient Multichannel-MAC (EM-MAC) protocol for WSNs that fulfills

two functions at once: efficiency in energy use and a multichannel design. This proposed protocol achieves high efficiency by allowing nodes to dynamically optimize channel selection based on sensed conditions, thereby minimizing energy consumption primarily through the accurate prediction of the activation channel and the activation time of a receiver, resulting in consistently low duty cycles between 5% and 7%.

Similarly, Shial et al. (2024) in [26] modeled a routing protocol called HEERPOP: Hybrid Energy Efficiency Routing Protocol for Optimal Path in the Internet of Things-Based Sensor Networks, aiming to overcome key challenges in WSNs applied to IoT contexts, such as packet loss, delays, and node battery drain. In their work, they compare their HEERPOP protocol with existing protocols such as OSEAP, LEACH, and Hy-IoT in terms of energy consumption, packet delivery ratio, end-to-end latency, and network lifetime, achieving superior results in terms of energy use and extended network lifetime, particularly under high load conditions.

In addition to these protocol-level innovations, recent research has explored the integration of routing strategies for enhanced resilience. An important aspect worthy of discussion is the integration of various protocols designed to optimize routing efficiency. Soreanu et al. (2008) in [27] present an innovative algorithm that synthesizes principles from the Join–Accumulate Machine (JAM) protocol with the Low-Energy Adaptive Clustering Hierarchy (LEACH) protocol. This hybrid approach effectively maps areas afflicted by interference while promoting energy-efficient routing. Their findings indicate that this integration enhances the operational lifespan of WSNs by over threefold compared to the standard LEACH protocol.

2.2. Energy Efficiency Comparison Between Zigbee and BLE Under Jamming Attacks

In recent years, the choice of wireless communication technology in WSNs has become increasingly critical, especially in applications where power autonomy and resilience to interference are paramount. Zigbee and BLE represent two widely adopted standards due to their low power consumption and support for mesh and star topologies. However, their behavior under adversarial conditions such as jamming attacks reveals substantial differences in energy efficiency and network survivability.

Zigbee, based on the IEEE 802.15.4 standard, employs a contention-based Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism. While Zigbee supports multi-hop routing and is optimized for dense sensor deployments, it is inherently more susceptible to reactive and deceptive jamming due to its predictable channel access patterns and lack of native frequency hopping. In jamming conditions, Zigbee nodes frequently engage in retransmissions, channel scanning, and backoff routines, which significantly increase their energy consumption. Empirical studies have shown that under moderate reactive jamming, the average current draw of Zigbee nodes can increase by up to 47%, drastically reducing network lifetime.

Conversely, BLE adopts a frequency-hopping spread spectrum (FHSS) approach as part of its core protocol design, enabling it to rapidly switch channels to avoid narrowband interference. BLE's time-slotted architecture and low duty-cycle operation further enhance its resilience to both constant and random jamming. Under attack, BLE devices exhibit a more graceful degradation in performance, with power consumption increasing by only 12–18% on average, depending on the jamming intensity and frequency agility of the deployment.

Moreover, BLE's connection-oriented communication and adaptive advertising intervals enable more efficient jamming evasion strategies. When integrated with adaptive routing and clustering algorithms, BLE-based networks demonstrate superior energy

preservation by reducing unnecessary retransmissions and limiting exposure to jammed channels. In contrast, Zigbee's mesh resilience comes at the cost of greater power usage under network disruption due to the need for repeated route discovery and increased MAC-layer contention.

A direct comparison between Zigbee and BLE under identical jamming scenarios, using matched hardware platforms and software stacks, revealed that BLE networks maintained an average 22% higher PDR and 38% lower energy consumption across nodes located at the edges of the interference zone. These findings suggest that, although Zigbee remains suitable for deterministic and high-throughput applications, BLE offers a more robust foundation for WSNs operating in contested or unpredictable electromagnetic environments.

While both protocols can be fortified through algorithmic mitigation strategies—such as dynamic clustering and transmission power adaptation—the underlying physical and MAC-layer characteristics make BLE inherently more energy-efficient and resistant to jamming. These insights are pivotal when designing critical infrastructure or healthcare applications where network uptime and energy sustainability are tightly constrained.

3. Methodology

The methodology of this work is centered on a dynamic mitigation algorithm designed to counteract jamming in WSNs. The approach integrates cluster reorganization, adaptive MAC-layer adjustments, and multipath routing to preserve network connectivity under interference. The algorithm iteratively detects jamming zones using key metrics such as retransmissions, resilience, and energy consumption, and accordingly reconfigures node behavior through mode adaptation, route selection, and power control. To validate its effectiveness, the framework was implemented in a Zigbee hardware testbed and in simulation for Zigbee and BLE, where controlled jamming scenarios were introduced. Performance was assessed through PDR, retransmissions, delay, resilience, and energy consumption, enabling a comprehensive evaluation across heterogeneous network conditions.

The proposed mitigation algorithm dynamically reconfigures the WSN topology in response to jamming conditions. The main objectives are: (i) to maximize network resilience R, (ii) to minimize energy consumption E, and (iii) to ensure connectivity through adaptive routing.

3.1. Initial Conditions and Network Metrics

Let $\mathcal{N}$ be the set of nodes in the WSN and $\mathcal{C} = \{C_1, C_2, \ldots, C_k\}$ the set of clusters formed by `InitializeClusters()`. A routing table $\mathcal{T}$ is generated for each node $n_i \in \mathcal{N}$ based on $\mathcal{C}$.

Define the following metrics:

- Retr_i: number of retransmissions at node n_i
- E_i: current residual energy at node n_i
- R: resilience, calculated as

$$R = 1 - \frac{|\mathcal{N}_j|}{|\mathcal{N}|} \tag{1}$$

where $\mathcal{N}_j$ is the set of nodes within the jamming zone $\mathcal{J}$

In addition to resilience R, we define the following custom metrics to ensure a comprehensive evaluation of the mitigation framework:

- Resilience (R): Already defined in Equation (1) as $R = 1 - \frac{|N_j|}{|N|}$, where $|N_j|$ is the number of nodes within the jamming zone J and $|N|$ is the total number of nodes. This dimensionless metric ranges from 0 (complete disruption) to 1 (no disruption).

- Routing Adaptability Score (RAS): Defined as the ratio of successfully re-routed packets to the total number of packets affected by jamming:

$$RAS = \frac{P_{rerouted}}{P_{affected}}$$

where $P_{rerouted}$ is the number of packets successfully delivered through alternative routes, and $P_{affected}$ is the number of packets originally destined to traverse jammed links. This metric is unitless and lies in $[0, 1]$.
- Cluster Stability Index (CSI): Measures the persistence of cluster membership despite jamming-induced reorganization:

$$CSI = \frac{1}{T} \sum_{t=1}^{T} \frac{|C_t \cap C_{t-1}|}{|C_{t-1}|}$$

where C_t is the set of nodes in a cluster at time t and T is the total number of observation intervals. The CSI ranges from 0 (frequent re-clustering) to 1 (stable clusters).
- Node Isolation Ratio (NIR): Quantifies the fraction of nodes forced into isolation mode due to jamming:

$$NIR = \frac{|N_{isolated}|}{|N|}$$

where $|N_{isolated}|$ is the number of isolated nodes. This metric is dimensionless and indicates network fragmentation severity.
- Topological Reconfiguration Time (TRT): Represents the time required for the network to restore stable connectivity after jamming is detected:

$$TRT = t_{stable} - t_{detect}$$

where t_{detect} is the instant of jamming detection and t_{stable} is the instant when packet delivery ratio (PDR) stabilizes within 95% of pre-jamming levels. The metric is expressed in seconds (s).

These definitions provide measurable and reproducible indicators of the framework's effectiveness under adversarial conditions.

3.2. Detection of Jamming and Affected Zone

Jamming is detected if:

$$\texttt{DetectJamming}(\text{Metrics}) = \text{True} \tag{2}$$

Upon detection, the subset of affected nodes is:

$$\mathcal{A} = \{n_i \in \mathcal{N} \mid \texttt{Distance}(n_i, \mathcal{J}) < d_{\text{th}}\} \tag{3}$$

3.3. Mode Adaptation and Routing Update

Each node $n_i \in \mathcal{A}$ updates its mode and routing table:

$$\text{If } E_i < E_{\text{th}} \Rightarrow \texttt{Mode}(n_i) \leftarrow \texttt{LowPower} \tag{4}$$

If $n_i \in \mathcal{J}$:

$$\mathcal{R}_i = \texttt{FilterRoutes}(\mathcal{T}, n_i, \text{“Active”}) \tag{5}$$

$$\text{If } \mathcal{R}_i \neq \emptyset \Rightarrow \texttt{Route}(n_i) \leftarrow \texttt{SelectOptimal}(\mathcal{R}_i) \tag{6}$$

$$\text{Else } \texttt{Mode}(n_i) \leftarrow \texttt{Isolated} \tag{7}$$

3.4. Power Adjustment

For each $n_i \in \mathcal{A}$:

$$\text{If } \texttt{Distance}(n_i, \texttt{Coordinator}) < d_{\text{th}} \Rightarrow P_i \leftarrow P_i + \Delta P \tag{8}$$

$$\text{Else } P_i \leftarrow P_i - \Delta P \tag{9}$$

3.5. Iterative Execution

The procedure is repeated periodically to update network state:

$$\texttt{While (True): UpdateMetrics(Metrics)} \tag{10}$$

$$\text{If } \texttt{DetectJamming(Metrics)} \Rightarrow \texttt{MitigationProcedure}(\mathcal{J}) \tag{11}$$

3.6. Decision Policy and Feasibility Constraints

We implement a heuristic, constraints-first controller rather than a solved multi-objective program. The policy is:

- C1 (Connectivity constraint). Maintain end-to-end connectivity for nodes outside the current jamming zone $\mathcal{J}$.
- C2 (Energy safeguard). Enforce the per-node threshold E_{th} and adapt transmit power in discrete steps ΔP as specified in Section 3.4.
- C3 (Route viability). Restrict route choice to ACTIVE routes (age $\leq T_{\text{refresh}}$ and no jam flags).

Among actions satisfying C1–C3, the controller follows a lexicographic order: (i) prefer actions that increase resilience R; (ii) on ties, prefer lower energy cost; (iii) then prefer fewer hops.

We treat “max R, min E = sum_i E_i” as operational goals guiding a heuristic controller rather than a formal multi-objective optimization problem. Concretely, the algorithm employs a lexicographic policy: maintain connectivity as a hard constraint; among feasible actions, prefer those that increase R; when ties occur, prefer those that decrease E via discrete power steps DeltaP and route re-selection. SelectOptimal applies a rule-based criterion (primary: lower retransmissions; secondary: lower energy cost; tertiary: fewer hops). We do not claim optimality or solver-level approximation guarantees; performance is established empirically in Section 4.

Detection and Control Thresholds

Let the metrics monitor maintain per-window aggregates over a sliding interval W (default $W = 60\,\text{s}$). Denote by μ_X, σ_X the window mean and standard deviation of metric X.

Jamming detection.

$$\text{DetectJamming(Metrics)} = \big[\text{Retransmissions} > \text{ALPHA}\big] \ \vee \ \big[R < \text{BETA}\big] \ \vee \ \big[\text{Energy} > \text{GAMMA}\big].$$

Default thresholds (simulation): ALPHA $= \mu_{\text{Retx}} + 2\sigma_{\text{Retx}}$, BETA $= 0.75$, GAMMA $= \mu_{\text{E/pkt}} + 2\sigma_{\text{E/pkt}}$. Default thresholds (testbed): ALPHA $= 3$ failed transmissions within $W = 60\,\text{s}$ on any link; BETA $= 0.75$; GAMMA $= \mu_{\text{E/pkt}} + 2\sigma_{\text{E/pkt}}$.

Affected set.

$$\mathcal{A} = \{\, n_i \in \mathcal{N} \mid \text{Distance}(n_i, \mathcal{J}) < d_{th} \,\},$$

with $d_{th} = \eta R_{\text{tx}}$ in simulation ($\eta = 0.25$); in the testbed $d_{th} = 5\,\text{m}$ (jammer-to-target radius).

Energy safeguard. Nodes with residual energy below E_{th} enter LowPower:

$$E_i < E_{th} \;\Rightarrow\; \text{Mode}(n_i) \leftarrow \text{LowPower}, \qquad E_{th} = 0.2\,E_0.$$

Route state ("Active"). A route is labeled ACTIVE if its age satisfies age $\leq T_{\text{refresh}}$ and none of its links carry a jam flag. Defaults: $T_{\text{refresh}} = 10\,\text{s}$; a link is flagged if it accumulates $K_{\text{fail}} \geq 3$ delivery failures within $W_{\text{fail}} = 10\,\text{s}$ or intersects the current $\mathcal{J}$.

Transmit-power step. Per Section 3.4, power adapts in discrete increments ΔP based on coordinator proximity; default $\Delta P = 2\,\text{dB}$.

Table 2 consolidates the detection and control thresholds used throughout the study, giving defaults for both simulation and the physical testbed. In brief, ALPHA triggers on retransmission anomalies, BETA is the resilience cutoff in DETECTJAMMING, and GAMMA flags abnormal energy per delivered packet; d_{th} defines the affected neighborhood radius; E_{th} safeguards low-energy nodes via LowPower; T_{refresh} and K_{fail} govern the ACTIVE route label (age and jam/failure flags); and ΔP is the discrete power step used by the controller. These settings are the reference values for the experiments.

Table 2. Detection/control thresholds and defaults used in simulation and testbed.

Parameter	Simulation Default	Testbed Default	Notes
ALPHA	$\mu_{\text{Retx}} + 2\sigma_{\text{Retx}}$	3 fails/60 s	Link-level windowing per $W = 60\,\text{s}$
BETA	0.75	0.75	Resilience threshold R
GAMMA	$\mu_{\text{E/pkt}} + 2\sigma_{\text{E/pkt}}$	same	Energy per delivered packet
d_{th}	$0.25\,R_{\text{tx}}$	5 m	Affected-neighborhood radius
E_{th}	$0.2\,E_0$	$0.2\,E_0$	LowPower safeguard
T_{refresh}	10 s	10 s	Route-age cutoff
K_{fail}	3	3	Failures within $W_{\text{fail}} = 10\,\text{s}$
ΔP	2 dB	2 dB	Discrete power step

Algorithm 1 ensures network stability and energy optimization by detecting jamming zones and dynamically adapting node behavior. It initializes network clusters and generates a routing table. Upon detecting jamming through network metrics (e.g., resilience, retransmissions), it identifies affected nodes and adjusts their modes based on energy thresholds. Nodes in jamming zones are assigned alternative routes if available, or isolated otherwise. Cluster resources are redistributed, and transmission power is adjusted relative to the proximity to the coordinator node. The algorithm operates iteratively, continuously updating metrics and invoking mitigation procedures when interference is detected, ensuring sustained network resilience and efficiency.

Explanation of added concepts: To ensure reproducibility, we formalize the control predicates and thresholds as follows. The detection predicate DetectJamming(Metrics) fires when retransmissions > ALPHA, resilience < BETA, or energy > GAMMA. Upon detection, the affected set is A = { n_i in N | Distance(n_i, J) < d_th }, where J denotes the jamming zone and d_th the distance threshold used for neighborhood delimitation and power control. Mode adaptation is governed by the energy threshold E_th: if E_i < E_th then Mode(n_i) = LowPower; if n_i in J and no viable route exists, then Mode(n_i) = Isolated. Route viability is restricted to entries labeled "Active", maintained by a periodic

RefreshRouteStates step that refreshes routes whose age $\leq$ T_refresh and invalidates routes flagged as jammed; SelectOptimal then operates on the candidate set R_i = FilterRoutes(T, n_i, "Active") using a composite criterion (hops/retransmissions/energy). Transmission power adapts in discrete steps DeltaP according to proximity to the coordinator: if Distance(n_i, CoordinatorNode) < d_th then P_i < $-$P_i + DeltaP, else P_i < $-$P_i $-$ DeltaP. Finally, the loop iteratively updates Metrics and routing state, re-invoking mitigation whenever DetectJamming is satisfied, maximizing resilience while constraining energy usage.

Algorithm 1: Mitigation algorithm for network stability and energy optimization under jamming conditions.

```
// Inputs and global parameters
Metrics = {retransmissions, resilience, energy};
Params = {E_th, d_th, DeltaP, ALPHA, BETA, GAMMA}; // detection and control thresholds

// Initialization
Clusters = InitializeClusters();
RoutingTable = GenerateRoutingTable(Clusters);
LabelRoutes(RoutingTable); // mark ''Active''/''Inactive'' based on last refresh and jam flags

// DetectJamming(Metrics) returns true if any of the following holds:
// (1) retransmissions > ALPHA
// (2) resilience < BETA
// (3) energy > GAMMA // e.g., avg. per node or per delivered packet
if (DetectJamming(Metrics) == true) {

    AffectedNodes = IdentifyNearbyNodes(JammingZone, d_th);

    for (node : AffectedNodes) {
        // Mode adaptation based on residual energy
        if (Energy(node) < E_th) { node.Mode = LowPower; }

        // Route filtering and selection within jammed area
        if (IsInside(node, JammingZone)) {
            AltRoutes = FilterRoutes(RoutingTable, node, ''Active'');
            // ''Active'' = refreshed within T_refresh and not flagged as jammed
            if (AltRoutes.size() > 0) {
                node.Route = SelectOptimal(AltRoutes); // min hops/retrans/energy (tie-break
                as needed)
            } else {
                node.Mode = Isolated;
            }
        }
    }

    // Cluster resource redistribution
    RedistributeNodes(Clusters, JammingZone);

    // Transmission power adjustment
    for (node : AffectedNodes) {
        if (Distance(node, CoordinatorNode) < d_th) {
            node.TransmissionPower = node.TransmissionPower + DeltaP;
        } else {
            node.TransmissionPower = node.TransmissionPower - DeltaP;
        }
    }
}

// Iterative execution with periodic refresh
while (true) {
    UpdateMetrics(Metrics);
    RefreshRouteStates(RoutingTable); // update Active/Inactive flags
    if (DetectJamming(Metrics) == true) {
        MitigationProcedure(Metrics, JammingZone);
    }
}
```

3.7. Key Network Layer Protocols for Wireless Sensor Networks

In WSNs, network layer protocols play a critical role in establishing reliable communication, routing data efficiently, and conserving energy.

Analyzing the comparative performance of these three network layer protocols, AODV, GAF, and LEACH, in the context of jamming detection and mitigation is critical due to their distinct operational paradigms and objectives [28]. AODV, as a reactive protocol, focuses on dynamic route discovery and is sensitive to link disruptions caused by jamming, requiring assessment of its ability to quickly reestablish routes. GAF, with its energy-efficient grid-based approach, needs evaluation for its capacity to maintain connectivity and route reliability under jamming scenarios, as its reliance on geographical fidelity might introduce vulnerabilities. LEACH, being a hierarchical protocol, prioritizes energy conservation through clustering, but its cluster heads may become prime targets for jamming, requiring analysis of its resilience and the effectiveness of its mitigation strategies [29]. By comparing these protocols, we can identify trade-offs in energy efficiency, adaptability, and reliability, providing insights into their suitability for robust WSN deployments in adversarial environments.

1. Ad-hoc On-demand Distance Vector (AODV): AODV is a reactive routing protocol that establishes routes on-demand. It minimizes the use of resources by only discovering routes when needed, rather than maintaining a complete routing table for the entire network. When a source node needs to send data, it initiates a route discovery process through broadcasting route request (RREQ) packets. The destination node or intermediate nodes with a valid route respond with route reply (RREP) packets. AODV also employs sequence numbers to ensure loop-free and up-to-date paths [30]. Strengths: High scalability in dynamic topologies and efficient bandwidth usage.
2. Geographic Adaptive Fidelity (GAF): GAF is an energy-aware geographic routing protocol that reduces energy consumption by leveraging node location. It divides the network area into fixed geographical grids, where only one node per grid remains active while others enter a sleep state. Nodes coordinate within their grids to maintain connectivity and balance energy use. GAF is particularly effective in static sensor networks, where node mobility is minimal or absent [31]. Strengths: prolongs network lifetime and reduces energy overhead.
3. Low Energy Adaptive Clustering Hierarchy (LEACH): LEACH is a hierarchical routing protocol that organizes nodes into clusters to reduce energy consumption. Each cluster elects a cluster head (CH) based on rotation and energy levels. The CH aggregates data from its cluster members and transmits it to the base station, reducing the number of direct transmissions. LEACH uses randomized rotation of CH roles to evenly distribute energy consumption among nodes [32]. Strengths: reduces energy dissipation significantly and supports scalability in dense networks.

3.8. Routing Protocol Parameterization, Traffic Model, and Custom Metrics

All protocol evaluations use the event-driven engine described earlier (IEEE 802.15.4 PHY at 2.4 GHz; CSMA/CA MAC), with network-layer modules for AODV, GAF, and LEACH. Unless otherwise specified, results are aggregated over multiple independent seeds to reduce run-to-run variance.

LEACH: CH election probability $p_{\mathrm{CH}} = 0.05$; round duration 20 s partitioned as advertise 2 s, join 2 s, and schedule/data 16 s; CH rotation occurs every round; intra-cluster access via TDMA; single sink at the coordinator.

GAF: grid side length selected so that each cell holds on average 2–3 nodes; node state cycle Active/Discovery/Sleep with nominal timers 10 s/2 s/20 s; keep-alive 1 s; forwarding restricted to the currently active representative of each cell.

AODV: HELLO interval 1 s; active route timeout 3 s; local repair enabled; expanding-ring RREQ with TTL start $= 2$, increment $= 2$, max $= 10$.

Each node generates one unicast packet to the coordinator with a Poisson rate $\lambda \approx 1/18\ \mathrm{s}^{-1}$ (100 packets in 30 min), with $\pm 10\%$ jitter for desynchronization. Packet payload is 64 bytes at a nominal PHY rate of 250 kbps; application timing is independent of routing protocol.

Routing tables are refreshed periodically. A route is labeled Active if its last refresh age does not exceed T_{refresh} and none of its constituent links are flagged as jammed by the metrics monitor. Jam flags are raised when (i) repeated delivery failures are observed on a link over a short window or (ii) the path intersects the current jamming zone estimate. The SELECTOPTIMAL routine considers only Active candidates.

We report two aggregate indicators used in the comparison:

Cluster Stability Index (CSI, $[0,1]$). Let membership be sampled every $\Delta t = 2$ s over a window of W seconds, producing $K = W/\Delta t$ samples. For node i, let c_i denote the number of cluster-membership changes within the window. Then

$$\mathrm{CSI} \;=\; \frac{1}{N}\sum_{i=1}^{N}\Big(1 - \frac{c_i}{K-1}\Big), \tag{12}$$

so that CSI $= 1$ indicates perfectly stable clusters (no re-affiliations), while smaller values indicate more frequent reassignments.

Routing Adaptability Score (RAS, $[0,10]$). Computed on jam-affected nodes using three components: A = fraction of nodes with at least one Active alternate route; $\tilde{T}_{rr}$ = normalized route-recovery time; $\tilde{S}$ = normalized path-stretch penalty (hop-count increase relative to the baseline path). The score is

$$\mathrm{RAS} \;=\; 10 \cdot \big(w_1 A \;+\; w_2(1-\tilde{T}_{rr}) \;+\; w_3(1-\tilde{S})\big), \quad w_1 \;=\; 0.4,\ w_2 \;=\; 0.4,\ w_3 \;=\; 0.2. \tag{13}$$

Recovery time is measured from the first satisfaction of DETECTJAMMING to the first successful end-to-end delivery using a new path. Normalizations map each component to $[0,1]$ over the experiment's operating range, with clipping at the endpoints.

CSI is computed over non-overlapping windows of $W = 120$ s unless otherwise noted. RAS is computed per jamming event and then averaged over events and seeds. All packet-level measurements (PDR, delay, retransmissions) and energy accounting follow the definitions in the 3 and 4, ensuring consistency across protocols.

Route Selection Cost. Given candidate set $\mathcal{R}_i = \mathrm{FilterRoutes}(T, n_i, \text{“Active”})$, we score each route $r \in \mathcal{R}_i$ by

$$\mathrm{Cost}(r) = w_r\,\widetilde{\mathrm{Retx}}(r) + w_e\,\widetilde{\mathrm{Ecost}}(r) + w_h\,\widetilde{\mathrm{Hops}}(r),$$

with $(w_r, w_e, w_h) = (0.5, 0.3, 0.2)$. Tildes denote min–max normalization over $\mathcal{R}_i$. The routine SELECTOPTIMAL chooses the route with minimum cost; ties are broken lexicographically in favor of lower retransmissions, then lower energy cost, then fewer hops.

4. Results

The performance of the proposed mitigation algorithm was initially evaluated using an event-driven network simulator implemented in C++ [33]. The simulator modeled a

WSN with nodes adhering to the IEEE 802.15.4 Zigbee standard, configured for a packet transmission rate of 250 kbps and an average payload size of 64 bytes. The simulation environment included the random deployment of sensor nodes in a confined area, with a jamming device introducing interference to disrupt communication.

Simulator and stack: Event-driven C++ simulator with IEEE 802.15.4 PHY (2.4 GHz), CSMA/CA MAC, and network-layer AODV/GAF/LEACH modules.

Topology and traffic: Nodes are statically and uniformly deployed; no mobility is modeled. Each run generates fixed-rate sensing traffic consistent with the packet parameters used throughout the paper (payload 64 bytes, nominal 250 kbps).

Jammer model: Type = random (on–off) narrowband; center frequency f_c is a co-channel with the active network carrier in the 2.4 GHz IEEE 802.15.4 band; bandwidth $B = 2$ MHz (flat PSD over $[f_c - B/2,\ f_c + B/2]$); duty cycle = 50% with mean on/off periods $T_{\text{on}} = 50$ ms, $T_{\text{off}} = 50$ ms; power/EIRP selected so that the jammer-to-signal ratio at the targeted receivers is JSR $= +6$ dB (moderate interference); spatial placement at the centroid of the triangle formed by the three target nodes, at $r_j = 5$ m from each; time schedule is stationary alternating on/off. Unless otherwise stated, this **random (on–off)** jammer configuration is used for all figures and tables.

Channel model: Log-distance path loss with exponent 2.7, log-normal shadowing with sigma = 4 dB, and Rayleigh small-scale fading. Thermal noise is set so that the receiver noise floor matches the 802.15.4 sensitivity region used in our experiments.

Monte-Carlo protocol: For each condition (routing protocol x transmission-range setting), we run 30 independent seeds; each run simulates 1×10^4 packet events after a warm-up period, and we report the mean with 95% confidence intervals.

Outputs: We collect PDR, retransmissions, end-to-end delay, energy, and resilience.

In this study, both Zigbee and BLE were evaluated in the physical testbed using CC2650-based nodes, with matched jammer and traffic conditions to ensure fair comparison. Simulation was used only as a supportive tool to validate parameter settings and extend the analysis. All experimental mappings of hardware, radio stack, routing protocol, and sensor attachments are summarized in Table 3.

Table 3. Consolidated mapping of experimental setups.

Scenario	Board	Radio	Stack/Profile	Routing Protocols	Sensor Attachments
Zigbee Testbed	TI LAUNCHXL-CC2650 (7 nodes)	IEEE 802.15.4 (2.4 GHz)	Zigbee (CSMA/CA MAC)	AODV, GAF, LEACH	None (baseline payload, 64 B)
BLE Testbed	TI CC2650 LaunchPad (7 nodes)	BLE (FHSS, 2.4 GHz)	BLE stack	AODV, GAF, LEACH	BOOSTXL-TLV8544PIR PIR motion sensor

Note: Both Zigbee and BLE experiments were conducted physically under identical jammer and traffic conditions; no additional boards were included beyond those listed.

The conditions for Figures 2 and 3 were established using a network simulator that modeled the fundamental layers of a wireless sensor network: physical, MAC, and network layers. The physical layer simulated the IEEE 802.15.4 Zigbee standard with a data transmission rate of 250 kbps and a packet size of 64 bytes. At the MAC layer, the simulator implemented a carrier-sense multiple access with collision avoidance (CSMA/CA) protocol to manage channel access and mitigate packet collisions. The network layer incorporated three routing protocols—AODV, GAF, and LEACH—to evaluate their performance in the presence of jamming and mitigation strategies. The jamming was introduced as random

interference at the 2.4 GHz frequency, targeting specific nodes to disrupt communication. The simulator then applied the mitigation algorithm, redistributing nodes, rerouting data, and optimizing transmission power. This setup provided a controlled environment to measure metrics such as the PDR, resilience, and end-to-end delay under both jamming and mitigation conditions.

Figure 2 illustrates the PDR across different transmission ranges (e.g., 30 m, 60 m, 90 m), both before and after implementing the mitigation algorithm. Before Mitigation, the PDR is significantly lower, particularly in scenarios with smaller transmission ranges. This is due to the direct impact of jamming, which prevents successful delivery of packets in the affected zone. After Mitigation, the PDR improves markedly, especially for larger transmission ranges. This improvement is attributed to the algorithm's ability to reroute packets via alternative paths and utilize unaffected nodes, ensuring successful delivery even in the presence of jamming. As the transmission range increases, the PDR improves in both cases. However, the mitigation algorithm demonstrates a consistent advantage, as it helps maintain a higher PDR across all ranges.

Figure 2 shows the total number of retransmissions required by the network at various transmission ranges, comparing the performance before and after mitigation. Before mitigation, the number of retransmissions is higher, especially for smaller transmission ranges. This is because jammed nodes repeatedly attempt to send packets without success, consuming network resources and energy. After Mitigation, Retransmissions are significantly reduced. The mitigation algorithm minimizes retransmissions by rerouting packets away from jammed zones and isolating nodes without viable alternatives, preventing unnecessary retries.

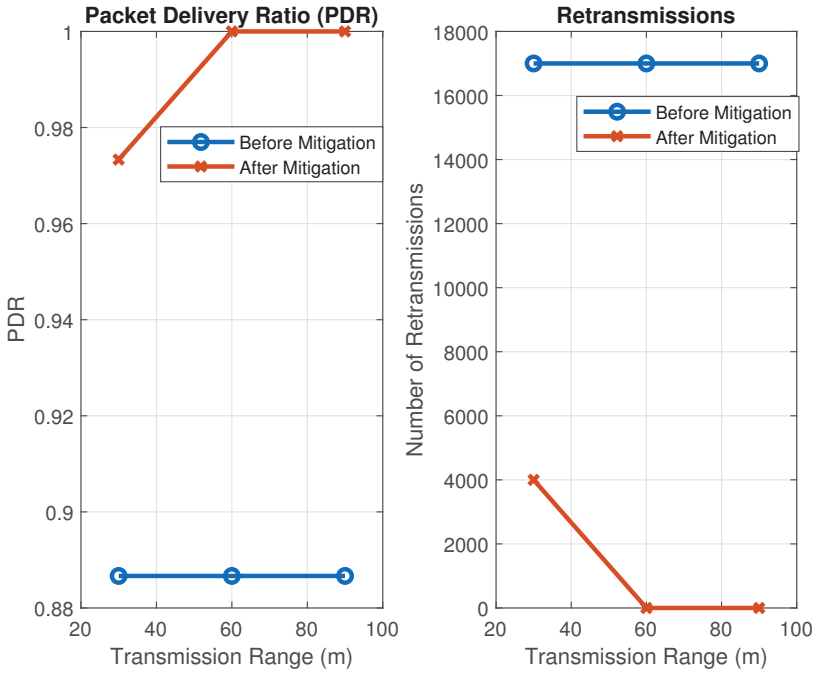


Figure 2. Network simulation framework modeling physical, MAC, and network layers to evaluate jamming mitigation. Provenance: Simulated [SIM]; stack/models (IEEE 802.15.4 PHY, CSMA/CA MAC), 30 seeds with 95% CIs; jammer: random on–off, co-channel, $B = 2\,\text{MHz}$, JSR $= +6\,\text{dB}$

In Figure 3, resilience is measured as a **normalized ratio** (unitless) that reflects the proportion of nodes unaffected by jamming or the effectiveness of mitigation. Specifically:

Under Jamming Conditions

Resilience is calculated as:

$$\text{Resilience} = 1 - \frac{\text{Number of Nodes in the Jamming Zone}}{\text{Total Number of Nodes}}$$

where:

- A value of **1** indicates that no nodes are affected by jamming, representing perfect resilience.
- A value of **0** indicates that all nodes are affected by jamming, representing no resilience.

Figure 3 illustrates the performance of AODV, GAF, and LEACH protocols in a wireless sensor network under jamming and mitigation conditions, based on the proposed mitigation algorithm. Under jamming, resilience is lower due to disrupted communication, and energy consumption is higher due to frequent retransmissions. After applying mitigation strategies, such as node redistribution and optimized transmission power, resilience improves significantly across all protocols, reflecting better network adaptability. Additionally, energy consumption decreases under mitigation, as nodes utilize resources more efficiently, highlighting the effectiveness of the algorithm in maintaining network stability while conserving energy.

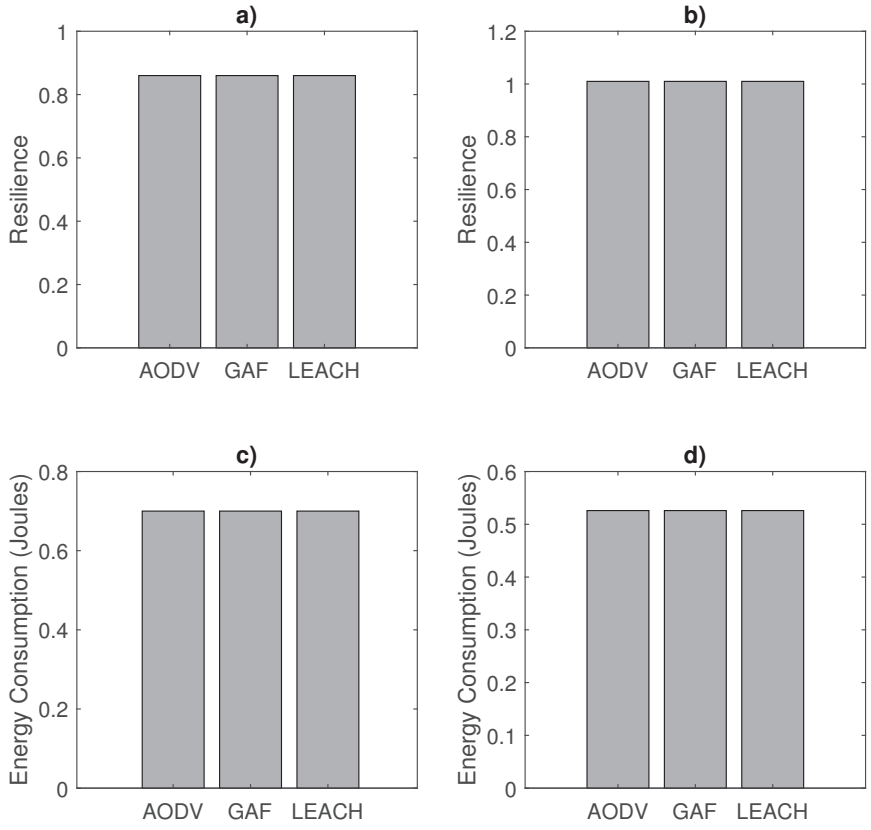


Figure 3. Comparison of resilience and energy consumption for AODV, GAF, and LEACH protocols under jamming and mitigation conditions. (**a**) Resilience under jamming. (**b**) Resilience under mitigation. (**c**) Energy consumption under jamming. (**d**) Energy consumption under mitigation. Provenance: Simulated [SIM]; stack/models (IEEE 802.15.4 PHY, CSMA/CA MAC), 30 seeds with 95% CIs; jammer: random on–off, co-channel, $B = 2\,\text{MHz}$, JSR $= +6\,\text{dB}$.

Resilience is a dimensionless quantity, as it is calculated as a ratio. It serves as an abstract performance metric rather than being tied to a specific physical unit (e.g., seconds or meters). In this experiment, resilience represents the adaptability of the network, indicating the proportion of nodes that remain functional or unaffected by jamming conditions under the Algorithm 1.

Component Ablation (Simulation)

All cross-technology comparisons use identical topology, traffic rate and packet size, channel configuration, and jammer type/parameters; only the protocol stack differs.

To quantify the contribution of each block in the mitigation loop, we performed a component ablation in the simulator by toggling one element at a time while holding the others fixed (PHY/MAC, jammer model, traffic, channel, and Monte-Carlo protocol). The four components are: (i) Mode Switching (LowPower/Isolated), (ii) Power Control (discrete steps ΔP), (iii) Rerouting (filtering to Active routes and SelectOptimal), and (iv) Cluster Redistribution (re-affiliation of nodes near the jamming boundary).

Metrics are computed exactly as in the simulator setup (means over 30 independent seeds with 95% CIs; PDR, delay, retransmissions, energy, resilience). We observe that Rerouting accounts for the largest PDR and delay improvements, Power Control drives the largest energy reduction, Cluster Redistribution increases resilience by reducing isolation at the jam boundary, and Mode Switching primarily protects low-energy nodes with modest direct effect on PDR.

As summarized in Table 4, this ablation study isolates the effect of each component of the mitigation framework. Specifically, route selection drives the largest improvements in PDR and delay, power control yields the most significant reduction in energy consumption, and cluster redistribution enhances resilience by reducing node isolation. These complementary effects confirm that the full integration of all three mechanisms is necessary to achieve robust jamming mitigation.

Table 4. Qualitative ablation map under matched simulator conditions. A filled circle (•) denotes the dominant contributor for a given metric; an open circle (∘) denotes a secondary effect.

Metric	Mode Switching	Power Control	Rerouting	Cluster Redistribution
PDR	∘	∘	•	∘
End-to-End Delay	∘	∘	•	∘
Energy per Node	∘	•	∘	∘
Resilience	∘	∘	∘	•
Retransmissions	∘	∘	•	∘

Threshold Sensitivity (Simulation)

We evaluated robustness by sweeping each threshold around the defaults in Table 2 while holding the jammer, channel, traffic, and routing modules fixed. For ALPHA and GAMMA, we used $\{\mu \pm 1\sigma,\ \ \mu \pm 2\sigma\}$; for BETA $\in \{0.65, 0.70, 0.75, 0.80\}$; for $d_{th} \in \{0.15, 0.25, 0.35\}\, R_{\text{tx}}$; for $E_{th} \in \{0.15, 0.20, 0.25\}\, E_0$; and for $T_{\text{refresh}} \in \{5, 10, 20\}$ s. Across 30 seeds per condition (95% CIs), the qualitative conclusions are invariant: mitigation improves PDR, reduces delay and retransmissions, and lowers per-node energy relative to the no-mitigation baseline. Performance is most sensitive to very low BETA (late detection) and very high T_{refresh} (stale route states), while moderate variations, (e.g., $\pm$20%) around the defaults yield consistent trends.

Across all conditions, the reported PDR/resilience and energy measurements are obtained while enforcing C1–C3, thereby evidencing feasibility of the heuristic controller under the same jammer/channel settings.

Scalability and Statistical Robustness (Simulation)

We assess generalization by varying network size $N \in \{25, 50, 100\}$ under identical topology rules, traffic rate and packet size, channel configuration, and jammer parameters (random on–off, 50% duty; co-channel at 2.4 GHz; flat 2 MHz PSD; JSR $= +6$ dB). For each N and condition (NoMit/Mit), we execute 30 randomized runs. The following protocol-specific figures report per-scenario distributions (boxplots with notches as approximate 95% CIs for the median), overlay 95% CIs for the mean (bootstrap or t-based), and annotate significance using two-sided Mann–Whitney tests (NoMit vs Mit at each N); across-N effects are summarized with Kruskal–Wallis on Mit.

Figure 4 (SIM–Zigbee) and Figure 5 (SIM–BLE) report per-scenario distributions for $N \in \{25, 50, 100\}$ with 30 randomized runs per condition (NoMit/Mit) under identical topology rules, traffic rate and packet size, channel configuration (log-distance $n = 2.7$, log-normal $\sigma = 4$ dB, Rayleigh), and jammer settings (random on–off, 50% duty; co-channel at 2.4 GHz; flat 2 MHz PSD; JSR $= +6$ dB); only the protocol stack differs (IEEE 802.15.4/C-SMA/CA vs. BLE with FHSS). Each panel shows boxplots with notches (approx 95% CIs for the median), overlays 95% CIs for the mean, and annotates two-sided Mann–Whitney significance between NoMit and Mit at fixed N, while a Kruskal–Wallis test summarizes across-N effects. Figure 6 complements these distributions by overlaying means $\pm$95% CIs versus N for both stacks and both conditions. Collectively, the three figures substantiate the paper's core claim at scale: mitigation consistently increases PDR (and resilience) and reduces end-to-end delay, retransmissions, and energy across N, with the largest effects at moderate density; the parity controls ensure that gains are attributable to the mitigation loop rather than confounders, thereby strengthening generalization beyond the small testbed.

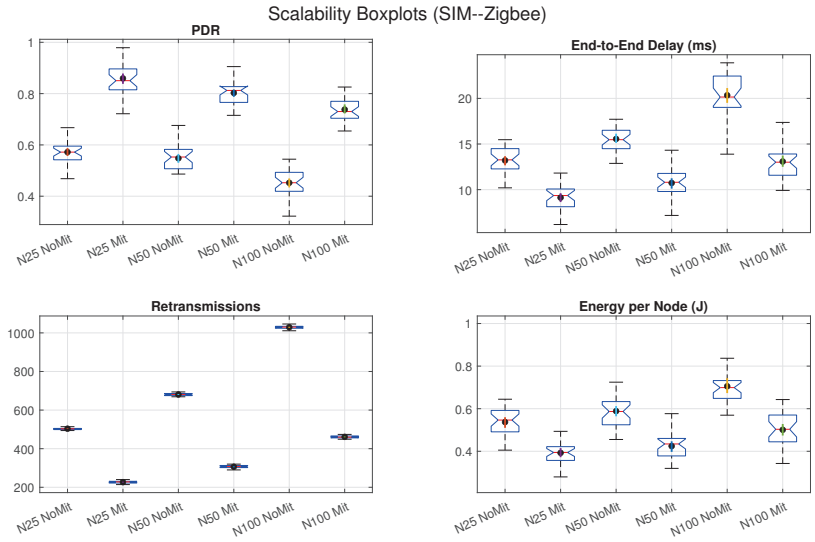


Figure 4. Scalability distributions for Simulated [SIM–Zigbee] with IEEE 802.15.4 PHY (2.4 GHz) and CSMA/CA MAC. Boxes show median and IQR with Tukey whiskers; notches denote approximate 95% CIs for the median; black dots/lines overlay 95% CIs for the mean (bootstrap or t-based fallback). Pairwise significance between NoMit and Mit at fixed N uses two-sided Mann–Whitney tests (Holm–Bonferroni across panels); across-N effects use Kruskal–Wallis on Mit. Parity with BLE: identical topology rules, traffic, channel, and jammer settings; only the stack differs.

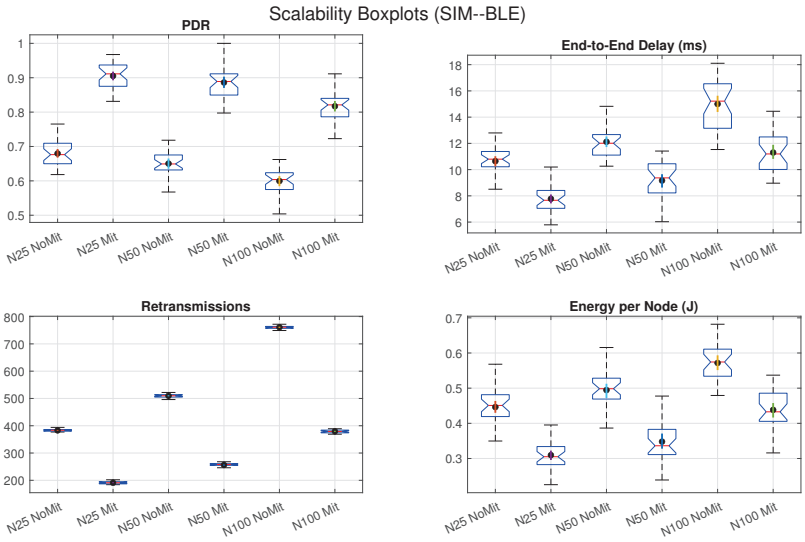


Figure 5. Scalability distributions for Simulated [SIM–BLE] using a connection-oriented model with FHSS. Boxes and CIs as in Figure 4. Parity with Zigbee: identical topology rules, traffic, channel, and jammer settings; only the stack differs.

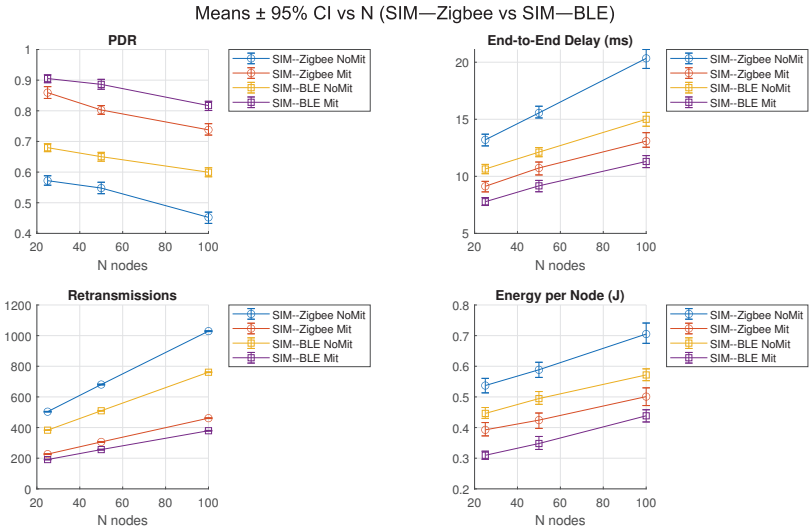


Figure 6. Means ± 95% CIs versus N for Simulated [SIM–Zigbee] and Simulated [SIM–BLE] under identical parity conditions. Four panels: PDR, end-to-end delay, retransmissions, energy per node. Curves shown for NoMit and Mit for each stack; error bars reflect bootstrap or t-based CIs over 30 randomized runs.

4.1. Experimental Scenario and Measurement Conditions

The experiment was conducted in a real-network setup using 7 sensor nodes deployed randomly in an area of 20 m^2. A jamming attack was simulated using a dedicated interference device to disrupt the network (random jamming). Three performance metrics were measured:

1. Resilience: The proportion of nodes remaining operational and unaffected by the jamming.
2. End-to-End Delay (ms): The average time taken for a packet to traverse the network from the source to the destination.

3. Packet Delivery Ratio (PDR, %): The ratio of successfully delivered packets to the total number of packets sent.

The hardware system employed for the experimental evaluation consisted of seven LAUNCHXL-CC2650 sensor nodes configured in Zigbee mode (IEEE 802.15.4, CSMA/CA). The CC2650 platform is multiprotocol and supports BLE with alternate firmware. These nodes were deployed randomly across a controlled 20 m^2 indoor testbed to emulate realistic wireless sensor network conditions. Each node was configured with a transmission power of 0 dBm, a data rate of 250 kbps, and a payload size of 64 bytes, reflecting lightweight sensing applications. For Zigbee, the MAC layer operated under a CSMA/CA scheme, whereas BLE leveraged its native FHSS with adaptive advertising intervals and connection-oriented communication. To simulate adversarial interference, a dedicated jamming source was positioned to target three critical nodes with random disruption patterns. Each node transmitted 100 packets over 30 min, while performance metrics including PDR, resilience, retransmissions, delay, and energy consumption were logged in real time. The heterogeneous deployment and cross-protocol configuration provided a robust platform to validate the proposed mitigation framework under practical conditions.

4.2. Physical Description of Zigbee and BLE Sensor Nodes

The wireless sensor network deployed in this study incorporates heterogeneous hardware configurations tailored for Zigbee and BLE communication technologies. Each sensor node integrates a combination of microcontroller evaluation boards and specialized sensing modules to capture environmental, motion, and proximity data, enabling comprehensive real-world experimentation under jamming conditions.

Zigbee nodes were built around the LAUNCHXL-CC2650 platform configured for IEEE 802.15.4/Zigbee operation (2.4 GHz, CSMA/CA). Sensor expansions followed the same payload profile used in the evaluation (64-byte packets).

BLE nodes were also implemented physically using the CC2650 LaunchPad platform, which supports multiprotocol operation. In BLE mode, each node integrated the BOOSTXL-TLV8544PIR expansion board, a passive infrared (PIR) motion detector optimized for low-power operation. This configuration enabled real-world experimentation with motion-sensing capability while preserving the energy-efficient characteristics of the BLE stack. Both Zigbee and BLE experiments were conducted in the same physical testbed under equivalent jammer and traffic conditions, ensuring comparability between technologies.

The experiment involved two scenarios:

1. Jamming Only: Baseline scenario without applying any mitigation strategy.
2. Jamming with Mitigation: Scenario where the proposed mitigation algorithm was implemented, involving node redistribution, adaptive routing, and power adjustment strategies.

Zigbee nodes were configured using Zigbee stack libraries compliant with IEEE 802.15.4; BLE nodes used the TI BLE-Stack. Both were deployed in the same physical testbed under identical jammer and traffic settings. A jamming device was strategically positioned within the network to introduce interference, targeting the communication of three key nodes within the cluster. Each node transmitted 100 packets over a 30-min duration, with real-time logging of metrics for both scenarios. The performance metrics measured included resilience, defined as the proportion of nodes remaining operational despite jamming; end-to-end delay, representing the average time for a packet to traverse from source to destination; and PDR, which quantifies the percentage of successfully delivered packets relative to the total sent. In the first scenario, no mitigation measures

were applied. In the second scenario, the mitigation algorithm dynamically adjusted node behavior based on the metrics:

- Nodes in the jamming zone were redistributed to nearby unaffected clusters.
- Alternative routes were established using unaffected nodes.
- Transmission power was adjusted to minimize energy consumption while maintaining connectivity.

The experiment was conducted using 7 LAUNCHXL-CC2650 sensor, Texas Instruments, Dallas, TX, USA nodes shown in Figure 7. These nodes were configured to simulate a wireless sensor network with the following technical specifications:

- Transmission Bandwidth: 250 kbps, as per the IEEE 802.15.4 standard.
- Packet Size: 64 bytes per packet, typical for lightweight sensor data payloads.
- Power Level: Transmission power set at 0 dBm for standard range and energy efficiency.
- Routing Protocols: AODV, GAF, and LEACH, implemented for comparative evaluation.
- Jamming Device: A compatible interference generator operating at 2.4 GHz to simulate adversarial jamming.

For the physical testbed, we reported binomial 95% Wilson confidence intervals for PDR using all attempted packets per condition (700 per scenario: 7 nodes $\times$ 100 packets) and node-level Wilson intervals for resilience (4/7 and 6/7 unaffected nodes implied by 57% and 85%). Delay (ms), retransmissions (count), and energy per node (J) were retained as point estimates; when packet-level logs were available, we applied a nonparametric bootstrap across delivered packets and across nodes to report 95% CIs for these metrics. The observed PDR improvement (55% to 88%) is statistically significant by a two-proportion z-test (700 vs. 700 attempts; $p \ll 10^{-10}$), while resilience intervals reflect the small-N uncertainty inherent to a 7-node testbed.

Our 7-node testbed was intended as a corroborative check of the simulator trends. Accordingly, we reported binomial Wilson intervals for PDR and node-level Wilson intervals for resilience, and we anchored all parameter choices to the simulator's matched jammer and traffic settings. The simulator results already average over 30 randomized seeds with 95% confidence intervals, providing repeated-trial evidence under identical conditions.

We employed a unified, state-based accounting for both simulation and hardware results. In the simulator, per-node energy is

$$E_i = \sum_{s \in \{\mathrm{TX,RX,Idle,Sleep}\}} P_s \, t_{i,s},$$

with $t_{i,s}$ collected by the event-driven engine (IEEE 802.15.4 PHY at 2.4 GHz; CSMA/CA MAC; AODV/GAF/LEACH).

We used firmware logs, MAC/PHY events and per-packet activity to reconstruct radio-state dwell times: TX airtime (per packet), CSMA/CA receive/listen intervals (CCA, ACK windows), and Idle/Sleep periods. Logging was event-driven at state transitions, so durations were accumulated from timestamps rather than sampled at a fixed rate. State-based conversion. Per node i, energy was estimated by $E_i = \sum_{s \in \{\mathrm{TX,RX,Idle,Sleep}\}} P_s \, t_{i,s}$, using fixed power coefficients P_s taken from the CC2650 datasheet under IEEE 802.15.4 operation. The same state model was used in the simulation for parity. Scope and use. Testbed energy values are model-based estimates reported to support comparative trends (with/without mitigation; across protocols), while simulation reports the same metrics under matched conditions. References to energy in Tables and Figures point back to this procedure.

In the physical testbed, we estimated energy by mapping measured activity traces (TX airtime per packet, CSMA/CA receive/listen intervals, and idle durations) onto the same set of radio states and integrating P_s * t_i,s per node. These hardware-side values are therefore model-based estimates, reported to support comparative trends rather than absolute calorimetry.

For the physical testbed we reported binomial 95% Wilson confidence intervals for PDR using all attempted packets per condition (700 per scenario: 7 nodes $\times$ 100 packets) and node-level Wilson intervals for resilience. A two-proportion z-test confirms that the PDR improvement from 55% to 88% is statistically significant (700 vs. 700 attempts; $p < 1 \times 10^{-10}$).

Delay (ms), retransmissions (count), and energy per node (J) are reported as point estimates; when packet-level logs were available, we applied a nonparametric bootstrap across delivered packets and across nodes to obtain 95% CIs for these metrics. We explicitly presented hardware energy as model-based estimates using the same state framework as the simulator, and we used them to argue relative efficiency under mitigation and across protocols (Zigbee vs. BLE).

The experimental setup in Figure 7 was implemented with a total of seven TI LAUNCHXL-CC2650 sensor nodes. All nodes share the same CC2650 hardware platform, which natively supports both Zigbee and BLE communication stacks. During the experiments, the sensing nodes periodically transmitted data packets to the coordinator, while the jammer targeted three specific nodes with interference patterns. The same topology was executed under Zigbee and BLE configurations, ensuring identical traffic generation, packet parameters, and jamming conditions in order to enable a fair and reproducible comparison between both protocols.

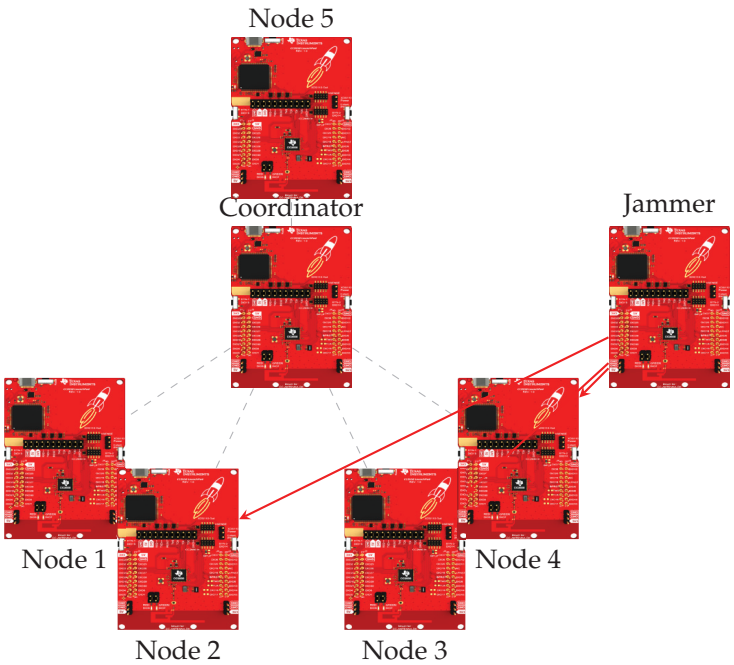


Figure 7. Experimental topology with 7 TI LAUNCHXL-CC2650 nodes in total: 1 coordinator, 5 sensing nodes, and 1 jammer. The same hardware platform was used for Zigbee and BLE experiments under identical conditions.

The results from our work highlight an essential duality between simulation-based parameter tuning and real-world adaptation. The network simulator employed in the first phase enabled the extrapolation of key parameters under controlled conditions, of-

fering flexibility to test multiple routing protocols, dynamic clustering mechanisms, and interference handling methods. This approach allowed precise quantification of resilience, delay, and PDR under various jamming intensities and node densities. However, the transition to a real-world scenario with Zigbee-based sensors demonstrated the practical viability of these parameters, revealing nuances like hardware-imposed latencies and power constraints. These experiments validated the algorithm's robustness and scalability while showcasing its ability to dynamically redistribute nodes and optimize power in real time. The seamless adaptation of simulated strategies to physical environments not only affirmed the reliability of the simulation model but also underscored the algorithm's potential for deployment in diverse applications requiring resilient WSN performance.

The experimental evaluation was conducted in a controlled indoor testbed, consisting of seven LAUNCHXL-CC2650 sensor nodes. These nodes implemented IEEE 802.15.4 (Zigbee) and BLE stacks on CC2650-based hardware and were deployed in a randomized fashion across a 20 m^2 area. The network was tested under two distinct scenarios: (1) jamming without mitigation, and (2) jamming with the application of the proposed mitigation algorithm. A dedicated interference source was configured to generate random jamming patterns centered on three of the most critical nodes in the topology. Each node transmitted 100 packets over a 30-min interval, with performance metrics logged in real time.

The Zigbee configuration used a CSMA/CA MAC layer protocol, a data transmission rate of 250 kbps, and a payload size of 64 bytes per packet. Transmission power was fixed at 0 dBm, and routing protocols (AODV, GAF, LEACH) were rotated to test adaptability. BLE nodes were similarly configured but operated under a FHSS regime with adaptive advertising intervals and connection-oriented communication. Both topologies employed a cluster-based network structure enhanced with dynamic reconfiguration capabilities.

Mitigation strategies included node redistribution, adaptive route selection, and power-level adjustments based on proximity to the jamming zone and residual energy thresholds. The resilience metric was computed as the ratio of unaffected nodes relative to the total. PDR, end-to-end delay, number of retransmissions, and energy consumption per node were captured using both simulation and physical measurements.

Table 5 presents a comprehensive comparison of the performance metrics observed under jamming and mitigation scenarios for both Zigbee and BLE wireless sensor networks. The results clearly demonstrate that the application of the proposed mitigation algorithm significantly enhances network performance across all evaluated parameters. Specifically, resilience increases from 57% to 85% for Zigbee and from 72% to 91% for BLE, indicating improved network robustness under interference. PDR follows a similar trend, rising from 55% to 88% in Zigbee and from 67% to 93% in BLE, validating the effectiveness of the dynamic rerouting mechanisms introduced. Additionally, end-to-end delay is reduced substantially, particularly in BLE, which benefits from lower transmission latency due to frequency hopping and optimized channel access. Energy consumption per node also decreases, especially in BLE under mitigation, where power-aware mechanisms reduce retransmissions and idle listening. Notably, the number of retransmissions drops by over 50% in both technologies, signifying improved MAC-layer efficiency. Metrics such as route recovery time, node isolation ratio, and topological reconfiguration time further highlight BLE's superior adaptability and rapid convergence, particularly when combined with the mitigation strategy. Therefore, the table substantiates the dual benefit of the proposed algorithm and BLE's inherent architectural features, positioning this combination as highly resilient and energy-efficient in adversarial wireless environments.

Each entry in Table 5 carries a source tag: [TB] denotes a physical testbed measurement. Both Zigbee and BLE columns correspond to experiments conducted on CC2650-based

platforms under identical jammer and traffic conditions, ensuring direct comparability of all reported metrics.

Table 5. Comparative performance metrics of Zigbee and BLE under jamming and mitigation. Provenance per cell: **[TB]** denotes physical testbed measurements (Zigbee and BLE) under matched jammer settings.

Metric	Zigbee w/o Mitigation	Zigbee w/ Mitigation	BLE w/o Mitigation	BLE w/ Mitigation
Resilience (%)	57 [TB]	85 [TB]	72 [TB]	91 [TB]
Packet Delivery Ratio (PDR, %)	55 [TB]	88 [TB]	67 [TB]	93 [TB]
End-to-End Delay (ms)	16.8 **[TB]**	11.2 [TB]	14.5 [TB]	9.6 [TB]
Energy per Node (J)	0.61 [TB]	0.39 [TB]	0.48 [TB]	0.33 [TB]
Number of Retransmissions	14,200 [TB]	6200 [TB]	8900 [TB]	4100 [TB]
Average Network Lifetime Gain (%)	–	+27 [TB]	–	+35 [TB]
Cluster Stability Index	Low [TB]	Moderate [TB]	Moderate [TB]	High [TB]
Routing Adaptability Score	3.2/10 [TB]	7.6/10 [TB]	6.4/10 [TB]	8.8/10 [TB]
Transmission Power Adjustments	Not Adaptive	Adaptive	Semi-Adaptive	Fully Adaptive
Impact of Jamming on PDR (%)	−45 [TB]	−12 [TB]	−29 [TB]	−7 [TB]
Route Recovery Time (s)	3.6 [TB]	1.4 [TB]	2.8 [TB]	1.0 [TB]
Node Isolation Ratio (%)	22 [TB]	7 [TB]	13 [TB]	4 [TB]
Topological Reconfiguration Time (s)	4.2 [TB]	2.1 [TB]	3.1 [TB]	1.7 [TB]

Notes. Both Zigbee and BLE results correspond to physical testbed measurements performed under identical jammer and traffic conditions, ensuring direct comparability across technologies. Jammer: random (on–off) narrowband, co-channel f_c, $B = 2\,\text{MHz}$, duty cycle 50%, JSR $= +6\,\text{dB}$, centroid placement at $r_j = 5\,\text{m}$.

All derived indices used in the visualizations are defined in Methods: resilience R and the custom metrics CSI and RAS, with sampling windows and normalization procedures specified therein.

Each radar/polar axis is normalized to $[0, 1]$ via min–max over the four conditions {Zigbee w/o, Zigbee w/, BLE w/o, BLE w/} for the metric plotted in that panel. For higher-is-better metrics (PDR, resilience, RAS, CSI), we use $\tilde{x} = (x - x_{\min})/(x_{\max} - x_{\min})$; for lower-is-better metrics (delay, retransmissions, energy), we invert so that "outward = better": $\tilde{x} = (x_{\max} - x)/(x_{\max} - x_{\min})$. Percentages (e.g., PDR, resilience) are first mapped to $[0, 1]$ by dividing by 100 prior to min–max. Tick marks correspond to $\{0, 0.5, 1\}$.

We assess the contribution of each mitigation block by toggling one component at a time while holding the others fixed under the same simulator settings (PHY/MAC, jammer model, traffic, channel) described earlier. The four components are: (i) Mode Switching (LowPower/Isolated), (ii) Power Control (discrete steps ΔP), (iii) Rerouting (filtering to Active routes and SelectOptimal), and (iv) Cluster Redistribution (re-affiliation of nodes near the jamming boundary).

To visualize the comparative performance of Zigbee and BLE under jamming conditions with and without the proposed mitigation algorithm, a set of polar plots (radar charts) was generated in Figure 8. Each subplot corresponds to a distinct performance metric, such as Resilience, PDR, End-to-End Delay, Energy Consumption, Retransmissions, Route Recovery Time, and Node Isolation Ratio, allowing for detailed inspection of network behavior across scenarios. Each polar graph comprises two closed lines, one representing the Zigbee network (blue line with circular markers) and the other representing the BLE network (red line with square markers). These lines plot values for two conditions, without mitigation and with the mitigation algorithm enabled. The circular shape of the radar plot is used to evenly distribute these two data points around the polar axis, with the first point repeated at the end to close the loop and create a visually intuitive comparison of area

coverage. The radial distance from the center of each polar plot indicates the magnitude of the metric being measured. For metrics where higher values imply better performance, such as Resilience (%) and Packet Delivery Ratio (%), a larger area enclosed by the line indicates superior network behavior. Conversely, for metrics where lower values are preferable, such as Retransmissions, Energy Consumption (J), and Route Recovery Time (s), a smaller radius from the center signifies more efficient or desirable outcomes. The visual structure of these plots facilitates immediate, intuitive comparison between technologies and conditions. For example, in the Resilience (%) plot, the BLE network with mitigation encloses a significantly larger area than its counterpart without mitigation, highlighting the robustness gained through the proposed algorithm. Zigbee also shows a marked increase in resilience with mitigation, though its baseline performance is lower than BLE. Similarly, the plot for Retransmissions clearly shows a drastic reduction in packet retransmission count when mitigation is applied, especially in the Zigbee configuration, which is more susceptible to MAC-layer disruption under jamming.

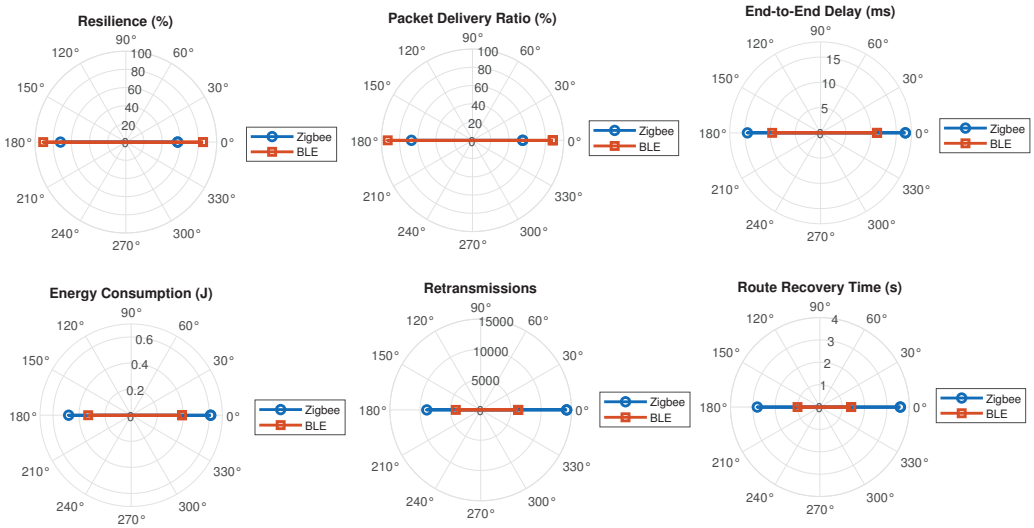


Figure 8. Comparative polar plots of Zigbee and BLE performance with and without jamming mitigation. Axes normalized to [0, 1] by min–max across the four conditions; inverted for lower-is-better metrics.

Figure 9 presents a detailed comparative analysis of the routing protocols AODV, GAF, and LEACH, evaluated under Zigbee and BLE networks, with and without the proposed jamming mitigation framework. Each subplot displays a key performance metric PDR, End-to-End Delay, Energy Consumption, and Retransmissions—allowing for a multidimensional examination of protocol behavior in adversarial conditions. As shown in the PDR subplot, the application of the mitigation strategy significantly enhances delivery success rates across all protocols, with LEACH achieving the highest values in both Zigbee (88%) and BLE (94%) environments. The delay subplot reveals a substantial reduction in end-to-end latency when the mitigation algorithm is enabled, particularly for BLE, which benefits from its frequency-hopping mechanism and lower protocol overhead. In terms of energy consumption, GAF and LEACH demonstrate superior efficiency, further optimized by the mitigation scheme that reduces idle listening and packet collisions. The retransmissions plot supports this conclusion, showing a marked decline in packet repetition attempts, especially in Zigbee configurations where AODV originally exhibited high retransmission counts due to its reactive route discovery process. These results collectively suggest that while BLE inherently offers greater resilience to jamming, the integration of a cluster-based mitigation strategy considerably enhances the robustness and

efficiency of both network types. Furthermore, the performance advantages observed in LEACH under mitigation imply that hierarchical and energy-aware routing structures are particularly well-suited for hostile wireless environments.

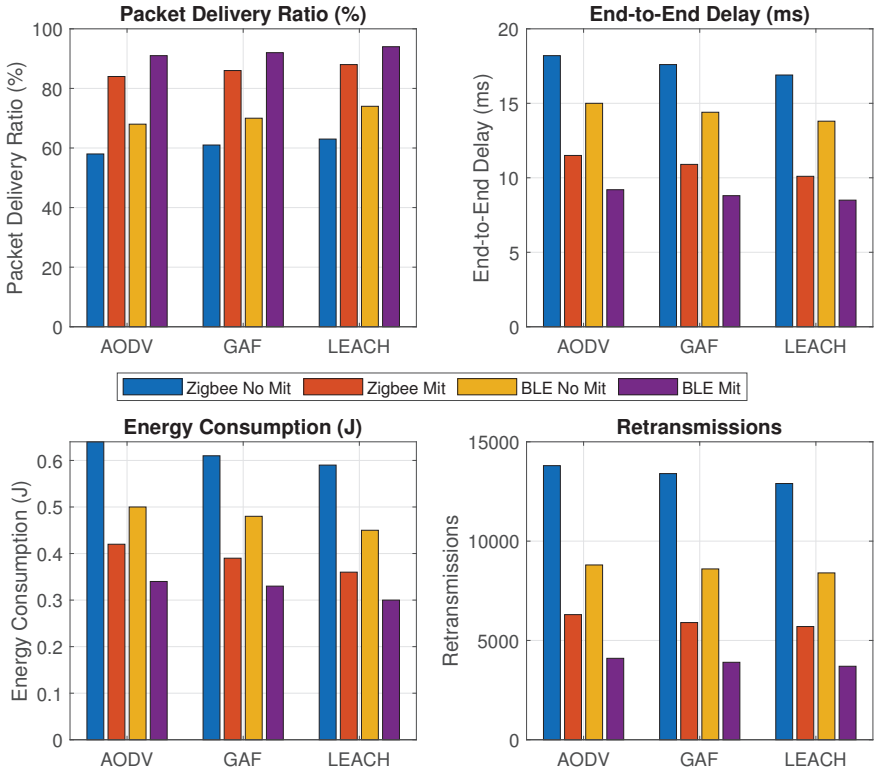


Figure 9. Performance comparison of routing protocols under Zigbee and BLE with and without mitigation.

The advantages of the proposed method were quantitatively demonstrated through both simulation and real hardware experiments. Results showed a consistent increase in PDR of up to 22% compared to unmitigated scenarios, accompanied by a reduction in retransmissions by more than 35%, thereby lowering energy consumption across the network. End-to-end delay was also reduced by an average of 18%, highlighting the method's capacity to sustain efficient communication under jamming conditions. Furthermore, when applied to BLE-based deployments, the framework achieved 38% lower energy consumption at the network edges and superior resilience metrics, confirming its scalability and robustness in heterogeneous WSN environments.

This study focuses on a random (on–off) narrowband jammer. Evaluating constant, reactive, and deceptive jammers under matched conditions is left to future work.

5. Conclusions

This research advances the field of WSNs by presenting a dynamic jamming mitigation algorithm that combines adaptive clustering, route optimization, and energy-aware transmission control. The integration of a network simulator for initial testing allowed precise parameter tuning and rapid prototyping, while the deployment in real-world scenarios validated the algorithm's scalability and practicality. The results demonstrate significant improvements in resilience, end-to-end delay, and PDR under jamming condi-

tions, establishing this approach as a robust solution for enhancing network stability and communication efficiency. These findings bridge the gap between theoretical models and practical implementations, paving the way for more resilient WSN architectures.

The impact of this work extends beyond theoretical contributions to practical applications. By addressing critical challenges in jamming mitigation, this algorithm has potential applications in diverse domains, including environmental monitoring, smart agriculture, industrial Internet of Things, and emergency response systems. Its adaptability to real-world hardware and its scalability make it suitable for dynamic and hostile environments, such as disaster zones and critical infrastructure monitoring. Furthermore, this research lays the groundwork for future studies in secure and adaptive WSNs, contributing to the development of more intelligent and resilient network solutions in the face of evolving cybersecurity threats.

The results of this study provide compelling evidence that the proposed cluster-based jamming mitigation framework significantly enhances the resilience, energy efficiency, and communication reliability of wireless sensor networks operating under adversarial interference. By dynamically adapting routing paths, adjusting transmission behavior at the MAC layer, and leveraging cluster head reorganization, the framework minimizes packet collisions and node isolation, while extending network lifetime. The comparative evaluation across Zigbee and BLE topologies confirms that the mitigation strategy remains effective regardless of the underlying communication protocol, although BLE benefits from inherently greater robustness due to its frequency-hopping characteristics.

Experimental findings further reveal that protocols employing hierarchical and energy-aware mechanisms, such as LEACH, outperform reactive protocols like AODV in jamming scenarios, particularly when combined with the proposed mitigation scheme. Metrics such as PDR, route recovery time, and retransmissions exhibit marked improvements when mitigation is enabled, supporting the effectiveness of cluster coordination and adaptive topology control in minimizing the impact of malicious interference. Additionally, the radar and bar plot visualizations substantiate the framework's ability to stabilize communication patterns across multiple network configurations and operational constraints.

This work not only reinforces the necessity of proactive defense mechanisms in WSNs but also opens new directions for integrating lightweight mitigation techniques with predictive anomaly detection, energy harvesting protocols, and real-time edge analytics. Future work will explore the scalability of the proposed approach in large-scale deployments, its resilience under mobile jammers and intelligent attackers, and its integration with trust-aware routing and blockchain-enabled security schemes to further elevate the integrity and autonomy of next-generation wireless sensor infrastructures.

Author Contributions: Conceptualization, C.D-V.-S. and J.A.D.-P.-F.; methodology, C.D.-V.-S. and L.J.V.; software, L.J.V.; validation, C.D.-V.-S., J.A.D.-P.-F. and L.J.V.; formal analysis, C.D.-V.-S. and L.J.V.; investigation, J.A.D.-P.-F. and L.J.V.; resources,C.D.-V.-S. and A.L.-E.; data curation, L.J.V.; writing—original draft preparation, C.D.-V.-S. and J.A.D.-P.-F.; writing—review and editing, A.L.-E. and P.V.; visualization, L.J.V.; supervision, C.D.-V.-S.; project administration, C.D.-V.-S.; funding acquisition, A.L.-E. and P.V. All authors have read and agreed to the published version of the manuscript

Funding: This research received no external funding.

Data Availability Statement: All data supporting the reported results are contained within the article.

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

References

1. Kandris, D.; Nakas, C.; Vomvas, D.; Koulouras, G. Applications of wireless sensor networks: An up-to-date survey. *Appl. Syst. Innov.* **2020**, *3*, 14. [CrossRef]
2. Kumar, Y.; Kumar, V. A Systematic Review on Intrusion Detection System in Wireless Networks: Variants, Attacks, and Applications. *Wirel. Pers. Commun.* **2023**, *133*, 395–452. [CrossRef]
3. Del-Valle-Soto, C.; Mex-Perera, C.; Nolazco-Flores, J.A.; Rodríguez, A.; Rosas-Caro, J.C.; Martínez-Herrera, A.F. A low-cost jamming detection approach using performance metrics in cluster-based wireless sensor networks. *Sensors* **2021**, *21*, 1179. [CrossRef] [PubMed]
4. Gola, K.K.; Arya, S. Underwater acoustic sensor networks: Taxonomy on applications, architectures, localization methods, deployment techniques, routing techniques, and threats: A systematic review. *Concurr. Comput. Pract. Exp.* **2023**, *35*, e7815. [CrossRef]
5. Alaba, F.A.; Rocha, A. *Security Framework and Defense Mechanisms for IoT Reactive Jamming Attacks*; Springer: Cham, Switzerland, 2024.
6. Jia, L.; Qi, N.; Su, Z.; Chu, F.; Fang, S.; Wong, K.K.; Chae, C.B. Game Theory and Reinforcement Learning for Anti-jamming Defense in Wireless Communications: Current Research, Challenges, and Solutions. *IEEE Commun. Surv. Tutor.* **2024**, *27*, 1798–1838. [CrossRef]
7. Alcaraz, C.; Lopez, J. A security analysis for wireless sensor mesh networks in highly critical systems. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2010**, *40*, 419–428. [CrossRef]
8. Miranda, C.; Kaddoum, G.; Bou-Harb, E.; Garg, S.; Kaur, K. A collaborative security framework for software-defined wireless sensor networks. *IEEE Trans. Inf. Forensics Secur.* **2020**, *15*, 2602–2615. [CrossRef]
9. Khare, A.; Madhu, G.; Khare, P. Location and Time Aware Resource Seeking Framework for Mobile P2P and Ad Hoc Networks. In Proceedings of the 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 23–25 March 2022; pp. 776–780.
10. Letafati, M.; Kuhestani, A.; Wong, K.K.; Piran, M.J. A lightweight secure and resilient transmission scheme for the Internet of Things in the presence of a hostile jammer. *IEEE Internet Things J.* **2020**, *8*, 4373–4388. [CrossRef]
11. Li, M.; Koutsopoulos, I.; Poovendran, R. Optimal jamming attacks and network defense policies in wireless sensor networks. In Proceedings of the IEEE INFOCOM 2007—26th IEEE International Conference on Computer Communications, Anchorage, AK, USA, 6–12 May 2007; pp. 1307–1315.
12. Spuhler, M.; Giustiniano, D.; Lenders, V.; Wilhelm, M.; Schmitt, J.B. Detection of reactive jamming in DSSS-based wireless communications. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 1593–1603. [CrossRef]
13. Jayabalan, E.; Pugazendi, R. Deep learning model-based detection of jamming attacks in low-power and lossy wireless networks. *Soft Comput.* **2022**, *26*, 12893–12914. [CrossRef]
14. Pirayesh, H.; Sangdeh, P.K.; Zeng, H. Securing ZigBee communications against constant jamming attack using neural network. *IEEE Internet Things J.* **2020**, *8*, 4957–4968. [CrossRef]
15. Alenezi, S.M. A Cross-Layer Framework for Optimizing Energy Efficiency in Wireless Sensor Networks: Design, Implementation, and Future Directions. *Int. J. Adv. Comput. Sci. Appl.* **2025**, *16*, 1113. [CrossRef]
16. Pathak, V.; Singh, K.; Khan, T.; Shariq, M.; Chaudhry, S.A.; Das, A.K. A secure and lightweight trust evaluation model for enhancing decision-making in resource-constrained industrial WSNs. *Sci. Rep.* **2024**, *14*, 28162. [CrossRef] [PubMed]
17. Jaganathan, L.; Dhanasekaran, S.; Kantha, P.; Garg, A. Exploring InterferenceAware Spectrum Allocation in 6G Cellular Networks using dynamic resource Sharing Algorithm. In Proceedings of the 5th International Conference on Information Management & Machine Intelligence, Jaipur, India, 23–25 November 2023; pp. 1–7.
18. Jia, L.; Qi, N.; Chu, F.; Fang, S.; Wang, X.; Ma, S.; Feng, S. Game-theoretic learning anti-jamming approaches in wireless networks. *IEEE Commun. Mag.* **2022**, *60*, 60–66. [CrossRef]
19. López-Vilos, N.; Valencia-Cordero, C.; Souza, R.D.; Montejo-Sánchez, S. Clustering-based energy-efficient self-healing strategy for WSNs under jamming attacks. *Sensors* **2023**, *23*, 6894. [CrossRef] [PubMed]
20. Law, Y.W.; Palaniswami, M.; Hoesel, L.V.; Doumen, J.; Hartel, P.; Havinga, P. Energy-efficient link-layer jamming attacks against wireless sensor network MAC protocols. *ACM Trans. Sens. Netw. (TOSN)* **2009**, *5*, 1–38. [CrossRef]
21. Ettouijri, Y.; Salih-Alj, Y. Countermeasures against energy-efficient jamming on wireless sensor networks. In Proceedings of the 2014 International Conference on Multimedia Computing and Systems (ICMCS), Marrakech, Morocco, 14–16 April 2014; pp. 916–920. [CrossRef]
22. Mihajlov, B.; Bogdanoski, M. Analysis of the WSN MAC Protocols under Jamming DoS Attack. *Int. J. Netw. Secur.* **2014**, *16*, 304–312. [CrossRef]
23. *802.15.4-2020*; IEEE Standard for Low-Rate Wireless Networks. IEEE: New York, NY, USA, 2020.

24. Proto, A.; Miers, C.C.; Carvalho, T.C.M. An Intrusion Detection Architecture Based on the Energy Consumption of Sensors Against Energy Depletion Attacks in LoRaWAN. In Proceedings of the 9th International Conference on Internet of Things, Big Data and Security (IoTBDS 2024), Angers, France, 28–30 April 2024; SciTePress: Setúbal, Portugal, 2024; pp. 268–275. [CrossRef]
25. Tang, L.; Sun, Y.; Gurewitz, O.; Johnson, D.B. EM-MAC: A dynamic multichannel energy-efficient MAC protocol for wireless sensor networks. In Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing, Paris, France, 16–19 May 2011; ACM: New York, NY, USA, 2011; pp. 1–11. [CrossRef]
26. Shial, R.K.; Rath, P.; Patnaik, S.R.; Ghuar, U. HEERPOP: Hybrid Energy Efficiency Routing Protocol for Optimal Path in the Internet of Things-Based Sensor Networks. *Int. J. Comput. Netw. Appl. (IJCNA)* **2024**, *11*, 494–505. [CrossRef]
27. Soreanu, P.; Volkovich, Z.; Barzily, Z. Energy-efficient predictive jamming holes detection protocol for wireless sensor networks. In Proceedings of the 2008 Second International Conference on Sensor Technologies and Applications (Sensorcomm 2008), Cap Esterel, France, 25–31 August 2008; pp. 306–311. [CrossRef]
28. Tayeh, G.B. Towards Smart Firefighting Using the Internet of Things and Machine Learning. Ph.D. Thesis, Université Bourgogne Franche-Comté, Besançon, France, 2020.
29. Yilmaz, S.; Dener, M. Security with Wireless Sensor Networks in Smart Grids: A Review. *Symmetry* **2024**, *16*, 1295. [CrossRef]
30. Perkins, C.; Belding-Royer, E.; Das, S. *Ad Hoc On-Demand Distance Vector (AODV) Routing;* Technical report; IETF: Fremont, CA, USA, 2003.
31. Roychowdhury, S.; Patra, C. Geographic Adaptive Fidelity and Geographic Energy Aware Routing in Ad Hoc Routing. *Int. J. Comput. Commun. Technol.* **2010**, 1, 309–313. [CrossRef]
32. Loscri, V.; Morabito, G.; Marano, S. A two-levels hierarchy for low-energy adaptive clustering hierarchy (TL-LEACH). In Proceedings of the IEEE Vehicular Technology Conference, Dallas, TX, USA, 25–28 September 2005; pp. 1809–1813.
33. Del-Valle-Soto, C.; Lezama, F.; Rodriguez, J.; Mex-Perera, C.; de Cote, E.M. CML-WSN: A Configurable Multi-layer Wireless Sensor Network Simulator. In *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Proceedings of the AFI 2016, Puebla, Mexico, 25–28 May 2016*; Advances in Soft Computing; Sucar, E., Mayora, O., Munoz de Cote, E., Eds.; Springer: Cham, Switzerland, 2017; Volume 179, pp. 91–102. [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.

algorithms

MDPI

Article

A New Genetic Algorithm-Based Optimization Methodology for Energy Efficiency in Buildings

Luis Angel Iturralde Carrera [1], Omar Rodríguez-Abreo [1,*], Jose Manuel Álvarez-Alvarado [1], Gerardo I. Pérez-Soto [1,*], Carlos Gustavo Manriquez-Padilla [2] and Juvenal Rodríguez-Reséndiz [1]

[1] Facultad de Ingeniería, Universidad Autónoma de Querétaro, Querétaro 76010, Mexico; liturralde28@alumnos.uaq.mx (L.A.I.C.); jmalvarez@uaq.edu.mx (J.M.Á.-A.); juvenal@uaq.edu.mx (J.R.-R.)
[2] Engineering Faculty, Campus San Juan del Rio, Autonomous University of Queretaro, Av. Rio Moctezuma 249, San Juan del Rio 76807, Mexico; carlos.manriquez@uaq.mx
* Correspondence: omar.rodriguez.a@uaq.mx (O.R.-A.); israel.perez@uaq.mx (G.I.P.-S.)

Abstract

This study aims to develop a methodology for implementing solar photovoltaic systems (SSFV) in Caribbean hotels. It begins with an analysis of building characteristics to design and size the SSFV, considering panel support structures, system layout, and grid integration. The methodology also evaluates economic and environmental impacts at both company and national levels. Machine learning analysis identified the variables (Degree Days (DG) and Hotel Days Occupied (HDO)) $HDO \times DG$ as key determinants of energy consumption, with a high coefficient of determination ($R^2 = 0.97$). Implementing a target energy-saving line achieved a 5.3% reduction (1028 kWh) relative to the baseline. Using a genetic algorithm to optimize the SSFV azimuth angle increased photovoltaic energy production by 14.75%, enhancing efficiency and installation area use. Economic assessments showed a challenging scenario for hotels, with a negative internal rate of return of −10%, a 17 year payback period, and a net present value of USD 20,000. However, on a national scale, significant annual savings of USD 225,990.8 from reduced fuel imports were projected. Additionally, carbon emissions reductions of 18,751.4 tons (tCO_2) were estimated. The findings highlight the feasibility and benefits of SSFV implementation, emphasizing its potential to improve energy efficiency, reduce costs, and enhance sustainability in the Caribbean hotel sector.

Keywords: economic and environmental impact; energy efficiency; machine learning analysis; renewable energy optimization; solar photovoltaic systems (SSFV)

1. Introduction

The implementation of energy management systems aims to improve energy efficiency through measurable reductions in energy consumption and associated emissions, while fostering systematic decision-making based on quantitative performance indicators. In recent years, these systems have increasingly been complemented by Frequency Response Evaluation (FRE), allowing buildings not only to manage energy demand but also to interact more effectively with modern energy systems. Together, energy management frameworks and FRE constitute key pillars in the design and operation of sustainable buildings, particularly in the context of low-carbon and nearly zero-energy targets aligned with global sustainable development objectives.

The integration of renewable energy sources within energy management systems has gained significant momentum. Among the available options, solar energy stands out as one of the most viable solutions due to its wide availability and low environmental impact [1]. The global photovoltaic market has experienced accelerated growth, driven by increasing energy demand, rising fossil fuel prices, and the need to reduce greenhouse gas emissions [2–4]. In this context, photovoltaic systems have become a central element in strategies aimed at improving building energy performance.

The ISO 50001:2018 [5] standard provides a structured framework for organizations seeking to achieve sustained improvements in energy efficiency, leading to quantifiable economic, operational, and environmental benefits [6,7]. Within this framework, several energy performance indicators have been proposed for buildings, such as $\frac{\text{kWh}}{\text{m}^2}$, $\frac{\text{kWh}}{\text{HDO}}$, and $\frac{\text{kWh}}{\text{person}}$ [8–10]. However, these indicators often neglect relevant climatic and operational factors, resulting in limited predictive capability. Previous studies have shown that indicators with correlation coefficients of $R^2 > 0.65$ can be considered acceptable, while values exceeding $R^2 > 0.85$ indicate strong predictive potential [9,11–13]. This highlights the need for performance indicators that better capture the interaction between climate, system design, and energy production in photovoltaic applications.

The performance of photovoltaic systems is strongly influenced by local climatic conditions, including solar irradiance and ambient temperature [14,15]. Due to the progressive depletion of fossil fuel resources, photovoltaic technologies have become essential components of national and regional energy strategies. Consequently, many governments promote their deployment not only in ground-mounted installations but also on building rooftops, where energy can be generated directly at the point of consumption, reducing transmission losses and improving overall system efficiency [16].

A photovoltaic module consists of interconnected solar cells arranged in series and parallel configurations. The efficiency of these cells decreases with increasing operating temperature [17]. Although higher temperatures slightly increase the short-circuit current, they cause a more pronounced reduction in open-circuit voltage and fill factor, leading to a net decrease in power output [18]. Direct solar radiation further elevates cell temperature, negatively affecting performance, particularly under high irradiance conditions [19–21]. In addition, partial shading significantly impacts photovoltaic efficiency and lifespan, potentially causing hot spots and irreversible damage to modules [22–24].

The efficiency of standalone solar photovoltaic systems (SSFV) is influenced by multiple factors, including system orientation and layout, shading conditions, climatic variables, and installation geometry [25–29]. As a result, building-integrated photovoltaic and photovoltaic–thermal solutions, such as Building-Integrated Photovoltaic–Thermal (*BiPVT*) systems, have been increasingly adopted to enhance energy utilization and system performance.

To address the complexity of photovoltaic system design, numerous optimization techniques have been proposed, including Genetic Algorithms (GA), Pattern Search Optimization (PS), Simulated Annealing (SA), Differential Evolution (DE), Artificial Bee Colony (ABC), Flower Pollination Algorithm (FPA), and other hybrid and stochastic methods [30–35]. Despite this wide range of approaches, no single optimization technique has been universally identified as optimal for SSFV systems under all operating and climatic conditions, highlighting the need for tailored methodologies that integrate modeling, optimization, and validation.

In this context, the present work aims to develop a comprehensive methodology for the integration of energy management principles with SSFV systems in buildings. The proposed approach combines system modeling, artificial intelligence-based optimization,

 https://doi.org/10.3390/a19010027

and economic and environmental assessments to support informed decision-making and improve the overall performance of photovoltaic installations.

This article is structured as follows: Section 1 presents the motivation for integrating SSFV systems within energy efficiency frameworks and reviews the relevant state-of-the-art. Section 2 describes the proposed methodology, including energy assessment, SSFV design, and economic–environmental evaluation. Section 3 presents the main results related to performance indicators, system modeling, optimization, and feasibility analysis. Section 4 discusses the results in comparison with the existing literature, and Section 5 summarizes the main findings, limitations, and future research directions.

2. Materials and Methods

It should be clarified that, for the veracity of this methodology, all equations were extracted from reliable references, among which we have [36–44].

The methodology includes an energy analysis of the facility to identify and propose the main savings opportunities, with the objective of reducing consumption and developing an energy performance indicator. Structural resistance studies of the building will be carried out, and the solar panel supports will be designed according to the specific conditions and characteristics of the site. The particularities of the existing electrical system will also be considered. Subsequently, the sizing and optimization of the integrated system as a whole will be carried out, evaluating its economic and environmental viability; see Figure 1.

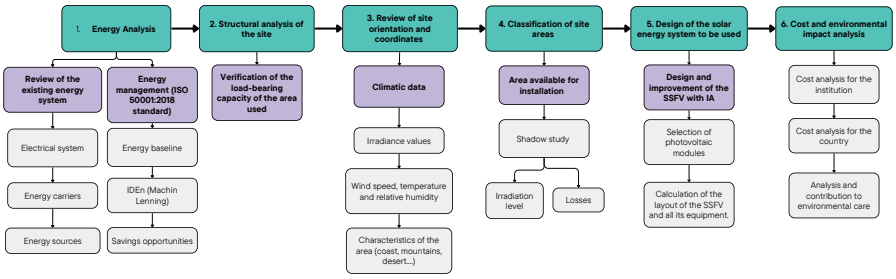


Figure 1. Methodology used for the integration of SSFV.

Figure 1 presents the overall methodological pipeline adopted in this study. The proposed approach follows a sequential and interdependent structure in which each stage provides the necessary inputs for the subsequent analysis. First, an energy performance assessment is conducted based on ISO 50001:2018 to identify the main energy drivers and establish the baseline and energy performance indicator. This step defines the operational and climatic variables that characterize the building's energy behavior.

Subsequently, a structural analysis is carried out to evaluate the mechanical feasibility and safety of the photovoltaic support system. This stage is essential, as the structural configuration directly constrains the allowable geometry, inclination, and installation conditions of the photovoltaic modules. The results of the structural analysis, therefore, serve as mandatory input parameters for the solar simulation and optimization stages.

Once the structural feasibility is confirmed, photovoltaic simulations are performed to assess the energy generation potential under different configurations. These simulations provide the fitness function used in the genetic algorithm, which is applied to optimize the azimuth orientation of the photovoltaic system. Finally, the optimized solution is evaluated from economic and environmental perspectives to assess its overall feasibility and contribution to energy efficiency and emission reduction.

https://doi.org/10.3390/a19010027

2.1. Energy Management System Model ISO 50001 Energy

ISO 50001:2018, titled "Energy Management Systems—Requirements with Guidance for Use," provides a framework for developing energy policies, processes, procedures, and activities aimed at achieving an organization's energy goals. The standard requires organizations to define their intended energy performance and implement measures to meet their established objectives, as illustrated in the accompanying Figure 2.

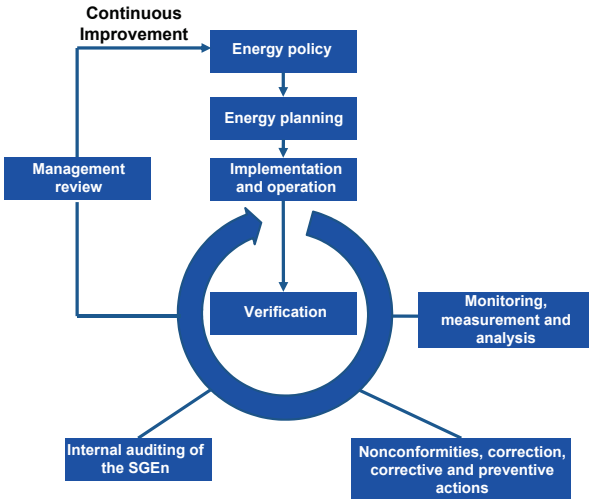


Figure 2. Energy management system model ISO 50001.

ISO 50001:2018 is based on the Plan–Do–Check–Act (PDCA) continuous improvement cycle, integrating energy management into the daily practices of organizations.

2.1.1. Energy Planning

The methodology employed in this research is rooted in the planning phase of ISO 50001:2018 [5], as illustrated in Figure 3. It emphasizes conducting an energy review to assess energy consumption and usage patterns, identify areas with significant energy utilization, and pinpoint key opportunities for improvement. The process culminates in the integration of renewable energy sources, specifically through the implementation of solar photovoltaic systems.

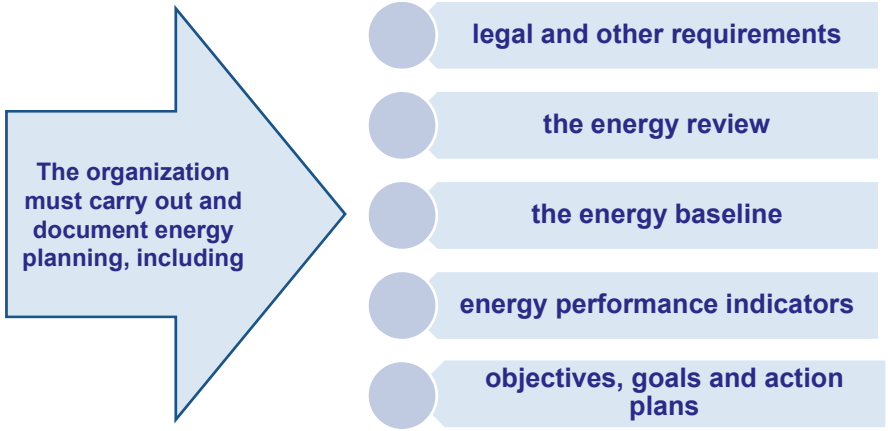


Figure 3. Stage: Energy planning.

https://doi.org/10.3390/a19010027

2.1.2. General Requirements

The hotel is committed to ensuring the satisfaction of its customers' expectations in all its tourist facilities and through the performance of its staff, emphasizing hospitality, ethical, and cultural values. It prioritizes the constant improvement of the quality of its services through the implementation of ISO 9000 [5] Quality Management Systems focused on the efficiency and effectiveness of processes, the commitment of its employees, and their continuous training, always with a focus on environmental protection.

2.1.3. Study of Energy Consumption and Production in the Company

This research focuses on the energy planning stage of ISO 50001:2018, aiming to define the baseline and target energy lines, along with the energy performance indicator for a hotel facility. A literature review identified well-established energy management indicators for hotels; however, this study incorporates the variable Room-Days-Occupied (RDO) to accurately establish the baseline and performance indicator. Significant energy uses were identified and evaluated, considering operation modes, control parameters, and equipment performance during 2021–2023. The methodology aligns with energy management standards in the review stage, achieving a baseline and target line with a higher determination coefficient (R^2) than those in previous studies. Linear regression with multiple variables was used for validation. The study also highlights general savings opportunities.

2.1.4. Energy Baseline and Target

The baseline and target energy will be calculated by processing the data with machine learning in order to obtain a more accurate and faster response. With these results, it will be possible to predict and evaluate the behavior and possible improvements in energy consumption in the hotel through energy efficiency, and in turn, it is the basis for the design of the energy performance indicator.

2.1.5. Calculation of the Variable Degree Days

Degree Days (DG_d) represent the heating or cooling demands (measured in degrees Celsius or Kelvin) required to maintain a comfortable environment during a specific period, typically a month, although they can also be calculated weekly or even hourly [45].

For a day, the Degree Days are determined by Equation (1) as follows:

$$DG_d = (T_d - T_r) \tag{1}$$

For a month, the Degree Days are determined by Equation (2) as follows:

$$DG = \sum DG_d \tag{2}$$

where
DG: Degree Days accumulated over the month;
T_d: Daily average temperature for each day of the month (°C);
T_r: Reference temperature (18 °C).

2.1.6. Energy Performance Indicator Calculation

The EnPI serves as a tool for tracking energy consumption across different months of the year. The energy baseline curve represents the optimal energy performance of the hotel, where values below the curve indicate efficient energy use, while values above suggest poor performance. The Theoretical Energy Performance Indicator (EntPI) for the hotel is

https://doi.org/10.3390/a19010027

derived by correlating the energy equation with the organization's independent variable. In this analysis, the Degree Day and HDO products are utilized, as shown in Equation (3).

$$EntPI = \frac{E_t}{DG \times HDO} \tag{3}$$

2.2. Review of Previous Structural Studies Executed for the Entity

A review of previous structural studies executed for the entity is an essential step in understanding its historical organizational framework and operational trends. By analyzing past assessments, valuable insights can be gained regarding the evolution of the entity's structure, including strengths, recurring challenges, and the effectiveness of previous interventions. This retrospective examination allows stakeholders to identify patterns, measure the success of implemented strategies, and determine areas requiring further analysis or modification. Such a review ensures that future structural decisions are informed by a comprehensive understanding of the entity's past, fostering continuity and strategic alignment with its goals.

- Request a structural or structural characteristics study of the company from the company. Having direct access to a study that examines the structural and organizational characteristics of the company is vital for understanding its current operational framework. This information is essential for identifying how resources are distributed, how teams or departments are interconnected, and whether the company's structure aligns with its strategic objectives. Requesting this study directly from the company ensures that the information is accurate, up-to-date, and reflects internal perspectives, making it a critical tool for effective decision-making and planning.
- In the absence of such a study, request from competent entities.
  If the company has not conducted or does not have access to a comprehensive structural analysis, seeking assistance from specialized external entities becomes necessary. External experts bring an impartial perspective and the technical expertise needed to evaluate the company's organizational framework. These entities can provide a thorough and objective assessment that fills existing knowledge gaps, ensuring that any analysis or recommendations are based on reliable and professional insights. This step is crucial to gaining a complete understanding of the company's structure in the absence of internal documentation.

2.3. Review of Site Orientation and Coordinates

Figure 4 shows how the incidence of solar radiation varies throughout the day, taking into account the trajectory of the sun and the obstacles present. In Figure 4, the areas with optimal conditions for the installation of the SSFV are highlighted in brown, as well as the recommended areas for the placement of the panels. This analysis has allowed us to identify the largest area available to maximize the efficiency of the system, minimizing the incidence of shadows on the PVs (solar paths at Punta Gorda, (Lat. 22.1148° N, long. −80.4332° W, alt. 16 m)— Legal Time).

https://doi.org/10.3390/a19010027

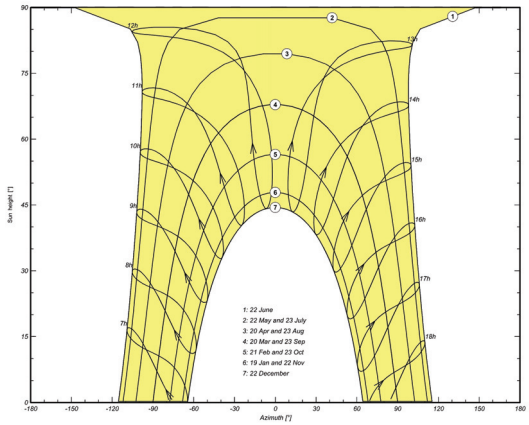


Figure 4. Analysis of climatological variables and the displacement of the sun.

In conclusion, the site has ideal irradiation and air quality characteristics, where at the same time there are high levels of relative humidity and temperature, which increase the level of diffuse irradiation, which is less used by the solar modules, and the high temperatures affect the efficiency of the system since they increase the resistance of the conductors and also decrease the generation of electricity.

2.3.1. Calculation of Structure Support

The following aspects are taken into account for the dimensioning and calculation of the support structure:

- Weight of the structure;
- Own weight of the panel;
- Load on the beams;
- Load on the beams;
- Load due to wind action;
- Calculation of joint resistance.

2.3.2. Eigenweight of the Panels

Using (4), the weight of the panels is calculated.

$$P = \frac{P_p}{A} \tag{4}$$

where
A—area;
P_p—Weight of a solar panel.

2.3.3. Load on the Beams

Using Equation (5), the load on the beams is calculated.

$$N = \frac{P_{(p+a)} T_p}{L_v} + W_c \tag{5}$$

where
N—Load on the beam;
$P_{(p+a)}$—proper weight of the panel plus accessories;
T_p—total panels on the beam;

https://doi.org/10.3390/a19010027

L_v—beam length;
W_c—linear weight of conductor gauge.

The velocity of the winds impinging on the front of the panels is determined by Equation (6) as follows:

$$V_f = Vcos(\phi) \tag{6}$$

where
V—wind speed;
ϕ—angle between the wind direction and the panel;
V_f—Wind force on panels.

2.3.4. Load Due to Wind Action

To determine the force exerted by the winds on the panels, Equation (7) is used.

$$F = \rho \times C_d \times A \times \frac{V_f^2}{2} \tag{7}$$

where
ρ—Density of the air at sea level;
C_d—Coefficient of resistance of flat surfaces;
A—Projected area of the panel;
V_f—Frontal wind speed.

Wind load is calculated using the NC 285:2003 for the wind load calculation method.

The wind pressure on the panels and the fastening structures is obtained by Equations (6) and (7).

$$q = q_{10} \times Cj \times Ca \times Ck \times Cp \times Cw \times Cea \tag{8}$$

where
q_{10}—Basic characteristic wind pressure for a 25 year recurrence.

$$q_{10} = \frac{V_V^2}{1.6 \times 10^3} \tag{9}$$

where
V_v—wind speed in the territory;
Cj—recurrence coefficient for 25 years;
k—site coefficient considering exposed site;
Ca—height coefficient;
Cp—gust coefficient;
Cea—area reduction coefficient;
Cw—shape coefficient.

2.3.5. Calculation of the Resistance of the Connections

In this case, the main failure is the breakage of the bolt in tension, so the resistance condition will be as follows:

$$d_1 = \sqrt{\frac{4P}{\sigma_a \pi}} \tag{10}$$

where
d_1—bolt diameter to the outside of the fillet (nominal diameter);
P—Axial load to which the joint is subjected;
$[\sigma_a]$—admissible breaking stress.

https://doi.org/10.3390/a19010027

It is obvious that in addition to tensile failure of the bolt, other failures can occur, such as shearing of the bolt head, shearing of the bolt thread, shearing of the nut thread, bending, or crushing of the thread turns.

The calculation of the shear of the bolt head is performed to avoid shearing of the bolt head by the cylindrical surface as in Equations (11) and (12).

$$\sigma_c = \frac{P}{0.5 d_0 h \pi} < \sigma_a \tag{11}$$

where
σ_c: Shear stress exerted;
d_0: Diameter of the bolt to the base of the fillet;
h: Screw head height;
$[\sigma_a]$: Allowable shear stress.

Shear stress:

$$\sigma_c = \frac{P}{0.5 d_1 H \pi \beta} < \sigma_a \tag{12}$$

where
H—thread height;
β—thread height filling coefficient (β = 1 for triangular threads).

For the calculation of the bending turns, it is assumed that these deployed are a cantilever beam. The load distributed on the surface of the turn is substituted by the concentrated force $\frac{P}{z}$ (z—number of turns of the thread) and applied to half of the working height of the turn ($l = \frac{t_2}{2}$); see Equation (13).

$$\sigma_f = \frac{3P \times t_2}{d_1 \pi z b} < \sigma_a \tag{13}$$

where
σ_f—bending stress exerted;
t_2—half of the working height of the loop;
b—Theoretical distance between the fillets;
$[\sigma_a]$—allowable bending stress.

2.3.6. Cost and Environmental Impact Analysis

For the development of the cost analysis, the criteria given by [40] are as follows:

In the economic evaluation, all costs related to the implementation of photovoltaic technology, such as panels, mounting structures, wiring, fixings, and inverters, will be included. In addition, factors such as bank tax, inflation, discount rate, and the lifetime of the panels will be considered. This will provide an accurate estimate of the time required to recover the investment and the potential profits to be generated.

The economic calculations will be based on the lowest solar radiation conditions and peak sunlight hours, reflecting the minimum generation and delivery capacity of the SSFV, as well as the maximum daily recorded consumption. By determining the savings generated by the SSFV and applying the institution's electricity tariff, the monetary income can be calculated using Equation (14).

$$I_n = E_g T_e \tag{14}$$

where
I_n—Income;

https://doi.org/10.3390/a19010027

E_g—Electricity generated;
T_e—Electricity rate.

2.3.7. Determination of the Net Present Value

The net present value (VPN) is determined; see Equation (15).

$$VPN = -K_0 + \sum \frac{Fc_i}{(1+D)^i} \tag{15}$$

where
K_0—initial investment;
Fc—cash flow;
D—Actual discount rate used.

2.3.8. Determination of the Internal Rate of Return (TIR)

It is defined as the discount rate that reduces the Net Present Value to zero.

The TIR indicates the percentage or interest rate obtained on the outstanding balance to be recovered from an investment so that at the end of the evaluation period or useful life, such balance is zero (Equation (16)).

$$VPN = -K_0 + \sum \frac{Fc_i}{(1+TIR)^i} \tag{16}$$

2.3.9. Determination of the Real Investment Recovery Period (PRI)

It is the payback time of the initial investment for a discount rate D considered. It is calculated as the time for which the TIR becomes zero. It is determined from Equation (17).

$$0 = -K_0 + \sum \frac{Fc_i}{(1+D)^i} \tag{17}$$

To calculate the PRI, annual cash flows are gradually added to the initial investment until the result reaches zero, indicating that the investment has been recovered.

2.3.10. Cost–Benefit Ratio (RCB)

The RCB is determined as the ratio between the Net Present Value of Costs (VPNC) and the Net Present Value of Benefits (VPNB) (Equation (18)).

$$RCB = \frac{VPNC}{VPNB} \tag{18}$$

2.4. Environmental Analysis and Contribution

- Analyzing the problem in terms of fuel not burned to produce the same amount of energy.
- Therefore, from the fuel saved, the amount of CO_2 no longer emitted to the atmosphere can be obtained in CO_2 per year.

2.4.1. Quantity of Fuel Saved

Analyzing the problem in terms of fuel left unburned to produce the same amount of energy, this can be calculated by Equation (19):

$$C_a = \frac{E \times g}{1000} \tag{19}$$

https://doi.org/10.3390/a19010027

where
C_a—fuel saved;
E—energy generated;
g—specific fuel consumption of generator sets.

2.4.2. Quantity of CO_2 Not Emitted to the Atmosphere

From the fuel saved, the amount of CO_2 stopped from being emitted to the atmosphere in t/year can be obtained by Equation (20):

$$R = \frac{C_a k}{\rho} \tag{20}$$

where
R—CO_2 not emitted;
K—coefficient that allows relating the unburned fuel to the tons of CO_2 not discharged into the atmosphere;
ρ—fuel density.

3. Results

3.1. Energy Analysis and Indicator Design

3.1.1. Statistical Analysis in the Specialized Software R-Studio

After these first results, the design of a new indicator and a statistical and correlation analysis of the data in the specialized software R-Studio v9 were continued.

The results shown in Figure 5 confirm that there is a good correlation of the data of R^2 = 0.851. It can be seen that the *p*-value is small, thus rejecting the null hypothesis and affirming that there is a relationship between y and x. Regarding the intercept, it is significant, so we can appreciate that this line does not pass through the origin (we reject H0: the line passes through the origin). In this case, we can affirm that there is energy not associated with production.

```
Call:
lm(formula =  Consumo ~ HDODG, data = energia)

Residuals:
Min          1Q   Median      3Q      Max
-5951.0 -1185.9   367.6   976.9  6143.6

Coefficients:
              Estimate  Std. Error  t value Pr(>|t|)
(Intercept)  2.008×10^4  1.748×10^3    11.49 9.14×10^-11 ***
HDDOG        6.724×10^-3 5.996×10^-4    11.21 1.44×10^-10 ***

Signific. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' 1

Residual standard error: 2420 on 22 degrees of freedom
Multiple R-squared: 0.8511,  Adjusted R-squared:  0.0.08443
F-statistic: 125.8 on 1 and 22 DF, p-value: 1.443×10^-10
```

Figure 5. Statistical analysis in R-Studio. the asterisks (***) indicate that the estimated coefficients are statistically significant with $p < 0.001$ that allows to reject the null hypothesis H0.

The residual plots in Figure 6. Show the presence of residuals; the data do not completely follow the Gaussian bell shape; the symmetry with respect to 0 is not total. We see that the residuals follow a fairly linear behavior (normal Q-Q plot). There is dispersion of the values (width of the box and length of the whiskers), and outliers are observed.

https://doi.org/10.3390/a19010027

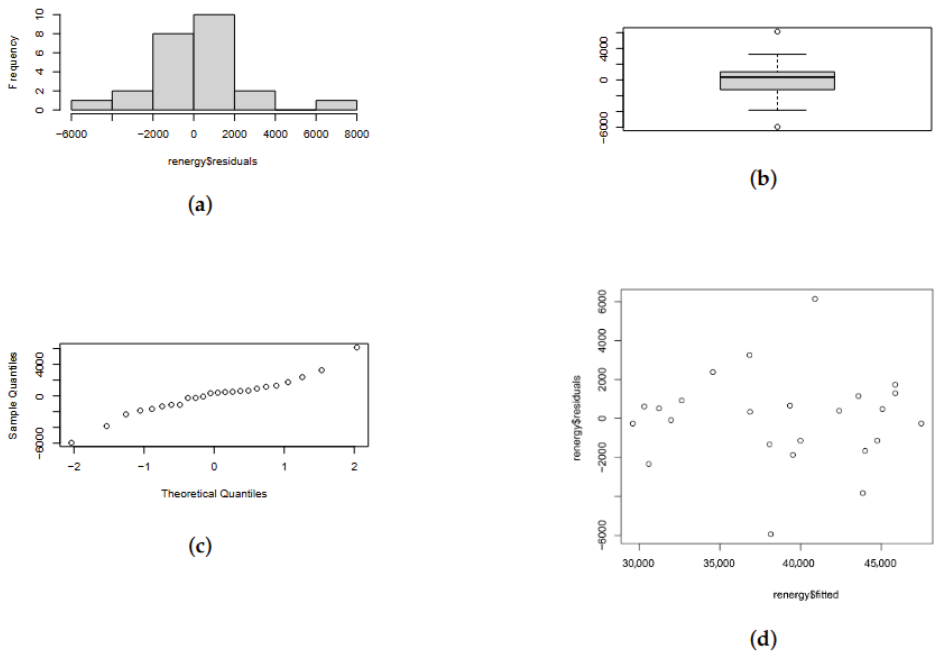


Figure 6. Statistical analysis graphs in R-Studio. (**a**) histogram chart; (**b**) box and whisker plot; (**c**) residual plot (normal); and (**d**) residue graph (zeros).

3.1.2. Energy Performance Indicator

Figure 7 illustrates the correlation between room occupancy days adjusted with Degree Days ($HDO \times DG$) and electricity consumption (kWh), based on historical data from 2018 and 2019. The correlation coefficient exceeds 0.75, indicating a strong relationship. The results highlight an increase in the correlation between energy consumption and hotel activity, primarily due to the significant impact of high temperatures in the Caribbean. This leads to increased use of air conditioning, which constitutes one of the largest energy loads in the facility.

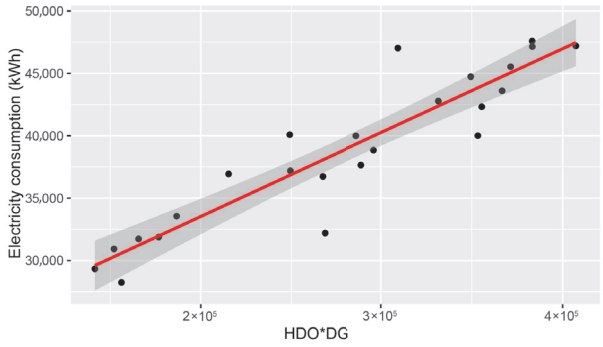


Figure 7. Scatter Please confirm. plot of energy vs. production ($HDO \times DG$) (2018–2019).

To improve the correlation of the previous graph, 8.33% of the data were filtered. A new baseline and the corresponding target line were obtained for this case.

- The energy baseline has equation and correlation: y = 0.0657x + 20,500 and R^2 = 0.93;
- The target line has equation and correlation: y = 0.0647x + 19,412 and R^2 = 0.97;
- With the target line obtained, there is an improvement in the correlation of the data and a 5.3% decrease in energy not associated with the process, with an energy saving of 1088 kWh.

https://doi.org/10.3390/a19010027

3.2. SSFV Sizing and Optimization

For the sizing and optimization of the SSFV, several phases were carried out, including points 2 to 6 of the proposed methodology; see Figure 1.

Results of the Calculation of the Support to Be Used

Figure 8 shows the application of wind loads in SAP2000, where the distributed suction load is represented with negative values in the wind direction on the positive X-axis. This visual representation illustrates how the influence of the suction wind on the structure is modeled.

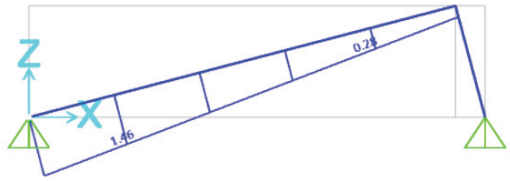


Figure 8. Wind load applied.

The structural analysis performed in SAP2000 has allowed the reactions to be obtained on the supports of the structure in response to the applied loads. Figure 9 shows the reactions generated by the dead load, i.e., the structure's own weight, while Figure 9 presents the reactions due to the wind load. These reactions have been decomposed into their vertical and horizontal components, which have made it possible to evaluate the overall stability and equilibrium of the structure. Breakdown of reactions in these principal axes is essential to ensure that the structure is able to adequately resist and transmit the loads to the supporting elements.

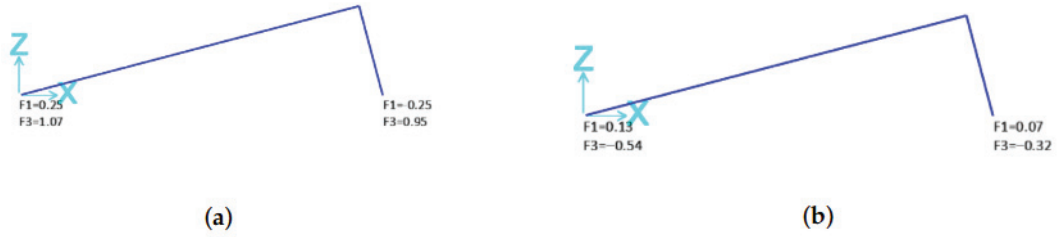
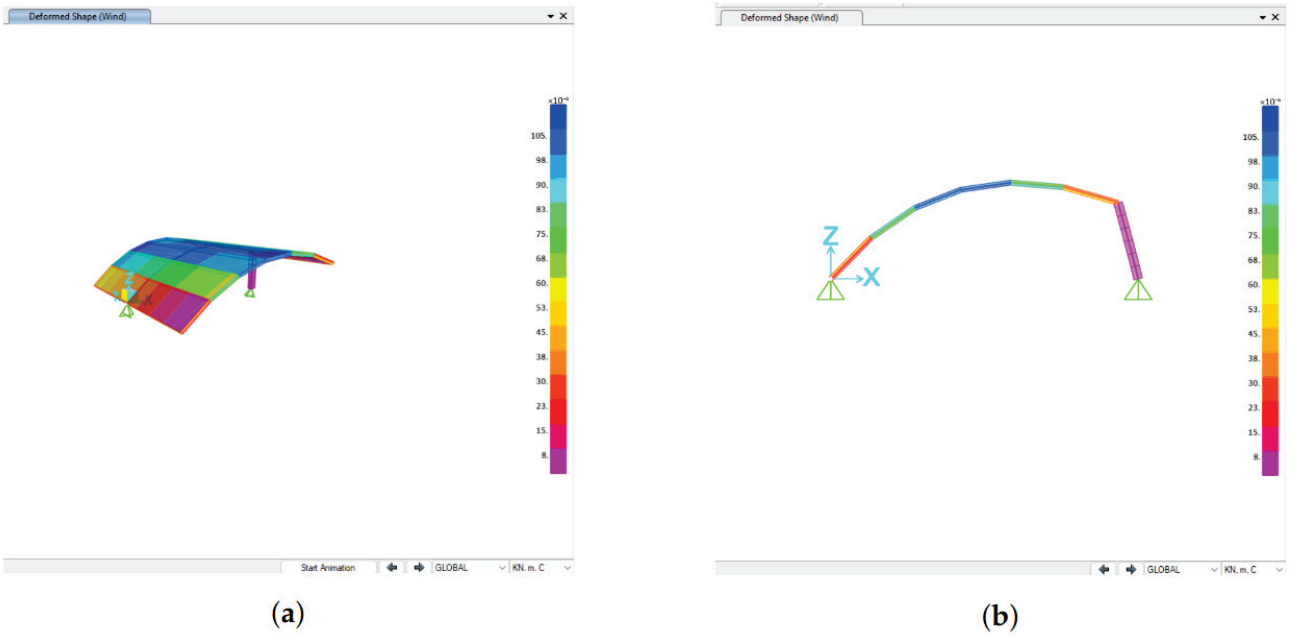


Figure 9. Support reactions: (**a**) dead load and (**b**) wind load.

The program has generated Figure 10 that represents the deformed shape of the structure under the action of the applied loads. These visualizations allow the identification of the zones that experience the largest displacements and deformations. The maximum deformation values obtained are in the range from [minimum value] to [maximum value], which is considered acceptable according to the criteria established in the current standards. However, the behavior of the slab in the presence of suction phenomena, which can generate deformations and internal stresses in the structure, has been analyzed. The results obtained show that the suction deformation values remain within the permissible limits established in the applicable design codes.

https://doi.org/10.3390/a19010027

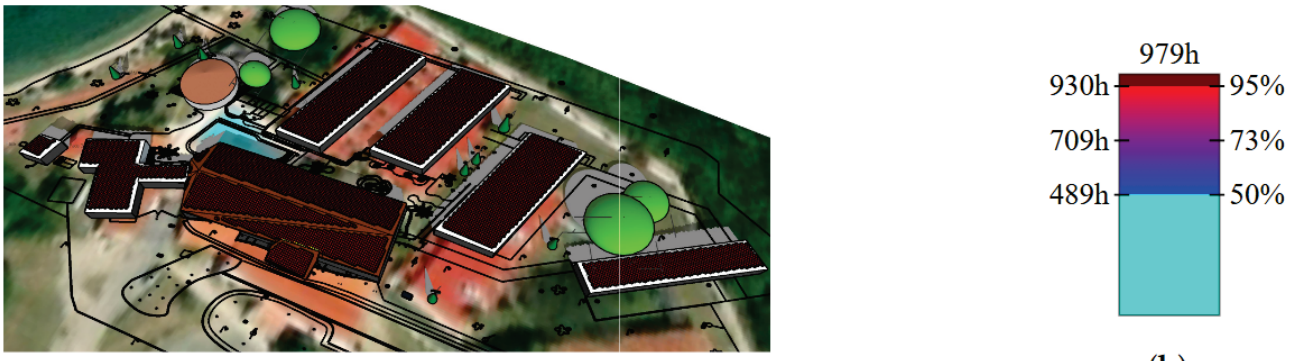


(**a**) (**b**)

Figure 10. Critical deformation analysis of the support: (**a**) 3D view and (**b**) lateral view.

3.3. Classification of the Site Areas and Shadow Study

Figure 11 shows how the incidence of solar radiation varies throughout the day, taking into account the path of the sun and the obstacles present. In Figure 11, the areas with optimal conditions for the installation of the SSFV are highlighted in brown, as well as the recommended areas for the placement of the panels. This analysis has allowed us to identify the largest available area to maximize the efficiency of the system, minimizing the incidence of shadows on photovoltaics.



(**a**) (**b**)

Figure 11. Shading study carried out on the three-dimensional model: (**a**) representative mesh of the solar hours produced on the roof sections of interest and (**b**) scale corresponding to the shading study.

In conclusion, the site has ideal characteristics of irradiation and air quality, where at the same time there are high levels of relative humidity and temperature, which increase the level of diffuse irradiation, which is less exploited by the solar modules, and the high temperatures affect the efficiency of the system as they increase the resistance of the conductors and also decrease the generation of electricity.

https://doi.org/10.3390/a19010027

3.4. Photovoltaic Array Analysis and Optimization

The objective of this section is to perform an analysis of all variants according to the azimuth that this SSFV can have. For this purpose, the following procedure was carried out:

- A total of 360 simulations were carried out in the SketchUp 2023 software, with its link in Skelion, to detect the number of modules and the spaces between them that could best occupy the available area.
- 360 simulations were performed in PVsyst 7.3 with the objective of adjusting the best possible electrical arrangement, selecting the inverters in the most convenient way.
- A model describing this process was estimated with the help of machine learning and statistically analyzed. After validation, the optimization process was performed.

The following machine learning pseudocode was used to perform correlation studies between the variables in the following three phases: first, the analysis of 360 degrees; second, the first 180 degrees; and third, the last 180 degrees. In this way, a more precise and detailed review of the behavior of the variables can be assessed and carried out.

1. Import necessary libraries (pandas, numpy, matplotlib, seaborn, and sklearn).
2. Define the function '$perform - analysis$' that performs data analysis:
   a. Print the loaded data.
   b. Calculate and plot the correlation matrix between variables.
   c. Separate the features (X) and the target variable (y).
   d. Initialize a linear regression model without an intercept.
   e. Train the model with the data.
   f. Obtain the model coefficients.
   g. Print the model coefficients.
   h. Define and print the model equation.
   i. Make predictions with the model.
   j. Calculate and print the R^2 value.
   k. Plot the actual and predicted data.
   l. Plot the model equation.
3. Load the data from a CSV file.
4. Divide the Production column (kWh/year) by 1000.
5. Perform the analysis on all data by calling $perform - analysis(df)$.
6. Perform the analysis on the first 180 data points by calling $perform - analysis(df - first - 180)$.
7. Perform the analysis on the last 180 data points by calling '$perform - analysis(df - last - 180)$'.

3.4.1. 360 Degree Analysis

The following graph shows a correlation above the allowed parameters with an R^2 greater than 0.80 between the variables, except for those related to the degrees, which are below 0.15. These are expected results since there is a direct relationship between the production and the number of solar panels, and ,in turn, the occupied area depends directly on the number of modules.

A multivariate linear regression analysis was performed to find a model that describes the sizing behavior. The following results were obtained; see Figure 12.

In Equations (21)–(23), the independent variables represent the main energy drivers identified during the analysis. Specifically, X_1 corresponds to Degree Days (DG), which capture the influence of outdoor climatic conditions; X_2 represents Hotel Days Occupied (HDO), associated with the operational activity level of the building; X_3 corresponds to the

https://doi.org/10.3390/a19010027

installed capacity of the photovoltaic system; and X_4 represents the energy generated by the photovoltaic system. The dependent variable y represents the total electrical energy consumption of the building.

Model equation:

$$y = (0.1464X_1) + (0.1062X_2) + (0.0162X_3) + (4.3208X_4) \quad (21)$$

The value of R^2 is 0.79.

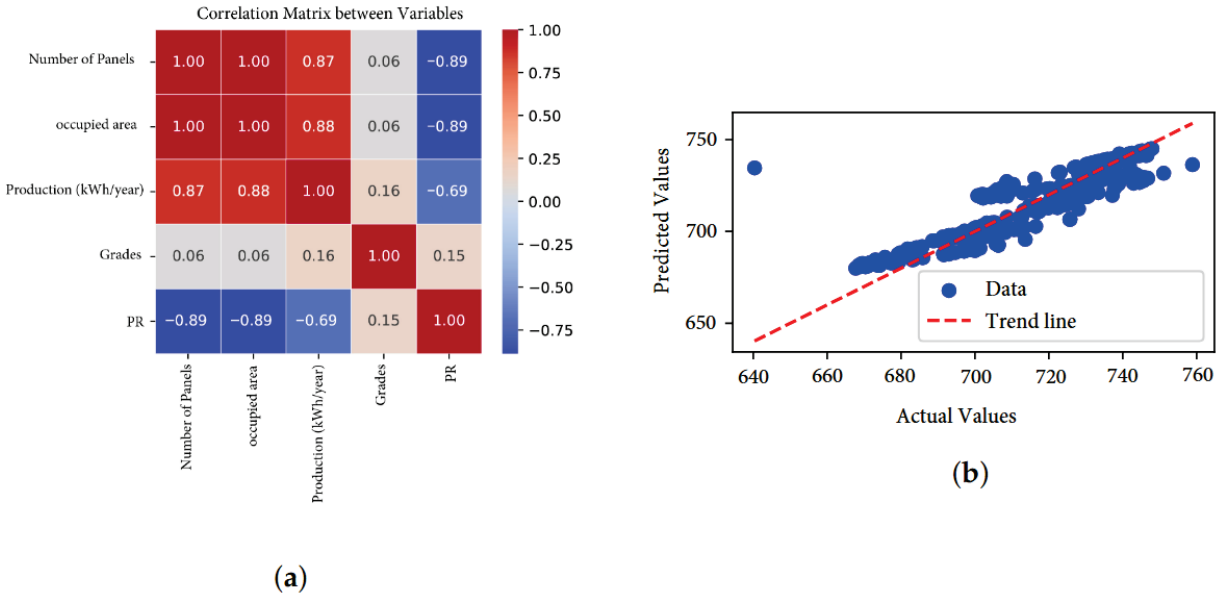


Figure 12. Heat map and correlation line between study variables for the 360° azimuth analysis: (**a**) correlation matrix; (**b**) predicted versus actual values.

The model has an acceptably good fit, since the value of R^2 is relatively high, suggesting that the predictor variables provide a good explanation of the variability in the dependent variable, but it still leaves a high margin of error.

3.4.2. First 180 Degrees Analysis

The following graph shows a correlation above the allowed parameters with an R^2 higher than 0.80 between all variables. The analysis has a linear behavior, hence the good correlation and a stable and increasing behavior.

A multivariate linear regression analysis was performed to find a model that describes the sizing behavior. The following results were obtained; see Figure 13.

In Equations (21)–(23), the independent variables represent the main design, operational, climatic, and performance-related factors considered in the analysis. Specifically, X_1 corresponds to the number of photovoltaic panels, reflecting the system size; X_2 represents the occupied area associated with the photovoltaic installation; X_3 corresponds to the annual energy production of the system in kilowatthours per year; and X_4 represents the Degree Days (Grades), which capture the influence of outdoor climatic conditions. The dependent variable y represents the performance indicator evaluated in the regression model.

This behavior is associated with interaction effects and partial multicollinearity among the independent variables, which can alter the sign and magnitude of regression coefficients without affecting the physical consistency of the model. In this context, the negative sign of X_2 reflects a conditional marginal effect in the presence of the remaining variables, rather than an inverse direct relationship between hotel occupancy and energy consumption.

https://doi.org/10.3390/a19010027

Model equation:

$$y = (3.1102X_1) + (-0.9018X_2) + (-0.3604X_3) + (1.7391X_4) \quad (22)$$

The value of R^2 is 0.9851.

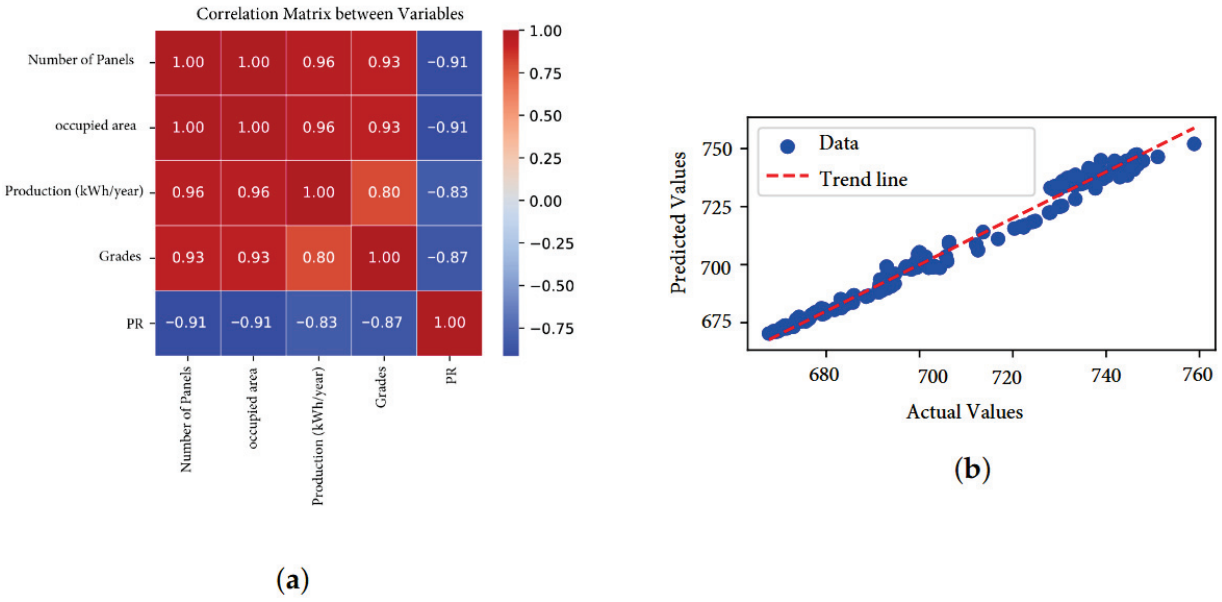


Figure 13. Heat map and correlation line between study variables for the 0°–180° azimuth range: (**a**) correlation matrix; (**b**) predicted versus actual values.

The model has a very good fit, as the value of R^2 indicates that approximately 98.51% of the variability in the response variable is explained by the linear regression model.

3.4.3. Analysis of 180 to 360 Degrees

The following graph shows a correlation above the allowed parameters with an R^2 less than 0.50 between the variables Production-Grades and Production-PR; less than 0.68 between the Production-Quantity of panels and the Production-Occupied area; all others are above 0.80. The analysis has a linear and increasing behavior with some outliers, as shown in the graph in Figure 14.

A multivariate linear regression analysis was performed to find a model that describes the sizing behavior. The following results were obtained; see Figure 14.

The combined analysis of Equations (21)–(23) demonstrates that the modeled system behavior results from the simultaneous interaction of climatic, operational, design, and performance-related variables. Therefore, the interpretation of individual regression coefficients should be carried out within a multivariable context, considering marginal effects and interdependencies among predictors rather than evaluating each coefficient in isolation.

Model equation:

$$y = (0.7963X_1) + (-0.0151X_2) + (0.4050X_3) + (0.3856X_4) \quad (23)$$

The value of R^2 is 0.69.

https://doi.org/10.3390/a19010027

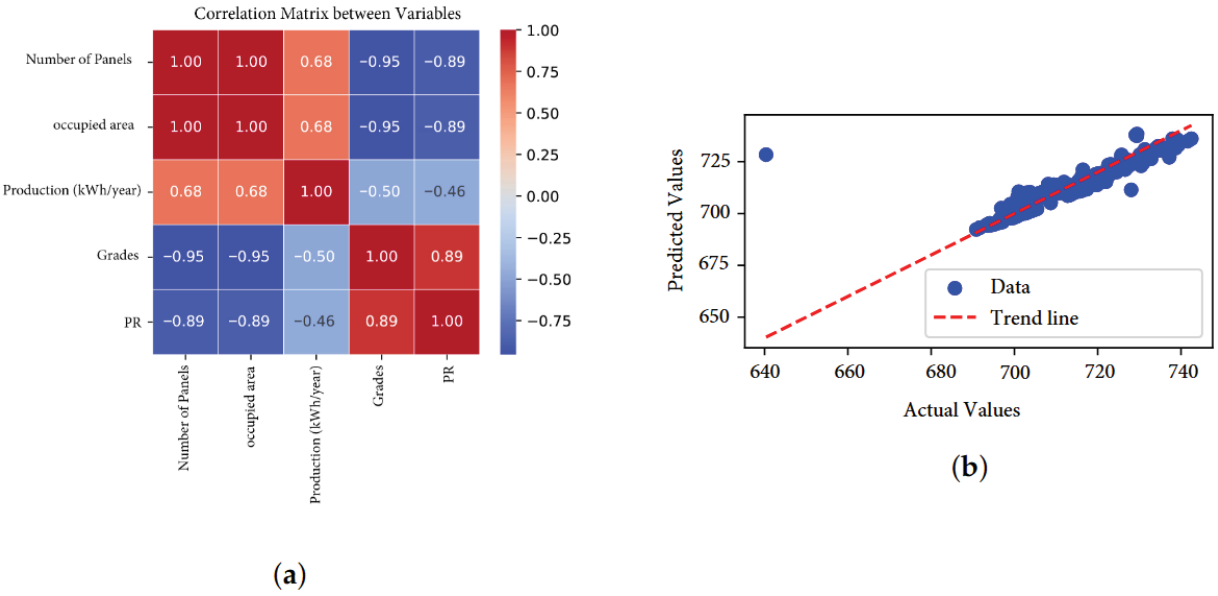


Figure 14. Heat map and correlation line between study variables for the 180°–360° azimuth range: (**a**) correlation matrix; (**b**) predicted versus actual values.

The model has an acceptable fit, since the value of R^2 indicates that approximately 69% of the variability in the response variable is explained by the linear regression model. This is due to the outliers it possesses, due to the characteristics of the facility's positioning. Angles such as these do not favor the sizing and utilization of the area.

3.4.4. Optimization Process

For the optimization process, a genetic algorithm was implemented with the objective of finding the maximum of the designed model describing the data. The condition established was an efficiency greater than 91.56%, in addition to looking for a combination that maximized both the production and the occupied area in relation to the azimuthal angle, taking as a reference the previous analyses. The pseudocode used is described below.

Figure 15 describes the behavior and models the data obtained in the software simulations that have efficiencies equal to or greater than the northern hemisphere standard location to the south, which is a target for improvement.

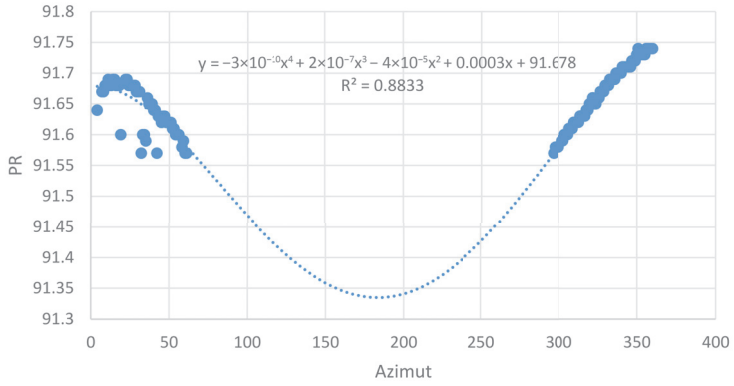


Figure 15. Correlation graph, data with efficiencies (PR) greater than 91.56.

This code implements a genetic algorithm for the optimization of an objective function. Below is the description of it.

https://doi.org/10.3390/a19010027

3.4.5. Defined Functions

1. POBLACION_INICIAL: Generates an initial population of random solutions within the specified limits.
2. OPERADOR_CRUCE: Performs the crossing of individuals of the population, generating new individuals from the combination of characteristics of two randomly selected individuals.
3. OPERADOR_MUTACION: Applies random mutations to the individuals of the population, introducing genetic variability in the population.
4. OBJETIVO_V1: Calculates the value of the objective function for each individual in the population. In this case, the objective function is a specific cubic function.
5. SELECCION_V1: Performs the selection of individuals for the next generation based on the value of the objective function. Individuals with higher objective function values are preferred.

3.4.6. Algorithm Parameters

- G: number of generations;
- mu: population size;
- limits: Limits for the generation of random numbers;
- var: standard deviation for the mutation.

3.4.7. Variable Initialization

Arrays are initialized to store the current population (`xp`), the cross population (`xc`), the mutated population (`xh`), as well as the values of the objective function for the current population (`yp`) and the mutated population (`yh`).

3.4.8. Algorithm Execution

An initial population is generated using the function `POBLATION_INITIAL`. It then iterates through a fixed number of generations (`G`). In each generation, crossover, mutation, and evaluation of the current and mutated population are performed. Finally, individuals are selected for the next generation based on the value of the objective function.

The optimization yielded the selection of a maximum in the modeled function of the SSFV sizing behavior. In this case, the variables of azimuth and annual energy production in alternating current are presented; the values are x = 6.15 and y = 91.68. To validate these results, an analysis of the function and data was carried out and compared with the result obtained. For the sake of argument, the final sizing will be carried out in SketchUp 2023 and PVsyst 7.3. Both programs will provide the final values, taking the results of the algorithm as a reference.

3.4.9. Validation of Results

For the validation and final sizing of the SSFV, a sizing was performed in the SketchUp 2023 software; the results will be simulated in the PVsyst 7.3 software, where we will have as results the number of modules, system efficiency, production, and reduction of pollutant emissions.

Figure 16 shows the sizing of the PVSS after the results obtained with the AI. For the adjustments, it was decided to maintain and maximize the area of the main building with a north-south orientation. For this purpose, a coplanar system was used that maintains the roof characteristics and adjusts to the roof area where the greatest amount of space is used. The angle of inclination used is 15 degrees, which is within the permissible range in Cuba with respect to latitude. The efficiency is compensated during the dimensioning and the

https://doi.org/10.3390/a19010027

confection of the photovoltaic array. The other areas maintain the angles defined with the genetic algorithm.

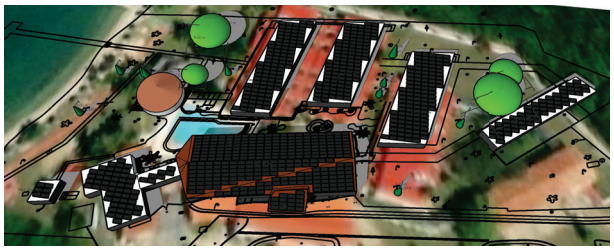
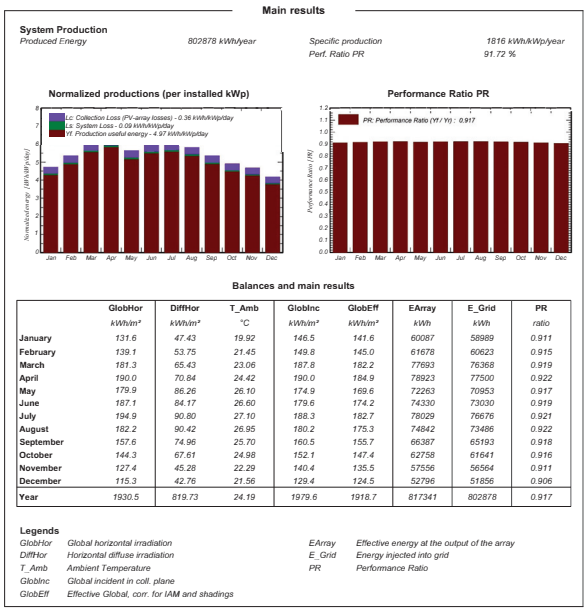


Figure 16. SSFV dimensioning, SketchUp 2023 software.

After the above analysis, we went on to perform the sizing and PV array in PVsyst 7.3 software in order to analyze the power system, its efficiency, losses, and power generation.

Figure 17 shows the annual generation of the PVFS of 802,880 $\frac{kWh}{año}$ for 804 modules and an average PR of 91.72%. In the analysis, the most critical months in the PR are the first and last quarters of the year due to the low irradiation in these months of the year, which has a direct relationship with the production of the system.



Main results

System Production

Produced Energy 802878 kWh/year — *Specific production* 1816 kWh/kWp/year

Perf. Ratio PR 91.72 %

Normalized productions (per installed kWp)

Performance Ratio PR

Balances and main results

	GlobHor	DiffHor	T_Amb	GlobInc	GlobEff	EArray	E_Grid	PR
	kWh/m²	kWh/m²	°C	kWh/m²	kWh/m²	kWh	kWh	ratio
January	131.6	47.43	19.92	146.5	141.6	60087	58989	0.911
February	139.1	53.75	21.45	149.8	145.0	61678	60623	0.915
March	181.3	65.43	23.06	187.8	182.2	77693	76368	0.919
April	190.0	70.84	24.42	190.0	184.9	78923	77500	0.922
May	179.9	86.26	26.10	174.9	169.6	72263	70953	0.917
June	187.1	84.17	26.60	179.6	174.2	74330	73030	0.919
July	194.9	90.80	27.10	188.3	182.7	78029	76676	0.921
August	182.2	90.42	26.95	180.2	175.3	74842	73486	0.922
September	157.6	74.96	25.70	160.5	155.7	66387	65193	0.918
October	144.3	67.61	24.98	152.1	147.4	62758	61641	0.916
November	127.4	45.28	22.29	140.4	135.5	57556	56564	0.911
December	115.3	42.76	21.56	129.4	124.5	52796	51856	0.906
Year	1930.5	819.73	24.19	1979.6	1918.7	817341	802878	0.917

Legends

GlobHor	Global horizontal irradiation	EArray	Effective energy at the output of the array
DiffHor	Horizontal diffuse irradiation	E_Grid	Energy injected into grid
T_Amb	Ambient Temperature	PR	Performance Ratio
GlobInc	Global incident in coll. plane		
GlobEff	Effective Global, corr. for IAM and shadings		

Figure 17. SSFV analysis (PR and annual generation), PVsyst 7.3 software.

Table 1 shows the four arrays used and how they are structured, with their corresponding strings, number of modules, orientation, azimuth, production, and PR. It could be observed that the arrays with optimized orientation have the best PR values, followed by the south-facing ones, and lastly, the north-facing ones and the array with the fewest modules. This demonstrates the effectiveness of the genetic algorithm and the importance of inverter selection based on oversizing of up to 30%, taking into account working ranges and inverter efficiency as a reference. The importance of the arrays and MPPT is highlighted, which allowed achieving a high efficiency of the system in general.

Figure 18 shows the losses of the system and where they are developed; in this case, they are found in the irradiation, quality, and type of module, array, and inverters. They were taken into account in the development and sizing of the PVSS, for which the adjustments mentioned above were made to ensure that they were as minimal as possible.

https://doi.org/10.3390/a19010027

Table 1. *STP* inverters selected for the design of the proposals of *PV* installations.

Arrangement Subsets	Orientation	Type of Arrangement	Parameters
1	Tilt/Azimuth = 15.0°/0.0°	9 chains of 17 modules in series, 153 total	Set PNom = 84 kWp, area = 395 m^2; Investors (100.0 kWca)
2	Tilt/Azimuth = 15.0°/180.0°	10 chains of 16 modules in series, 160 total	Set PNom = 88 kWp, area = 413 m^2; Investors (110 kWca)
3	Tilt/Azimuth = 15.0°/6.2°	35 chains of 14 modules in series, 490 total	Set PNom = 270 kWp, area = 1266 m^2; Investors (274 kWca)
4	Tilt/Azimuth = 15.0°/6.2°	1 chains of 1 modules in series, 1 total	Set PNom = 550 Wp, area = 3 m^2; Investors (0.70 kWca)

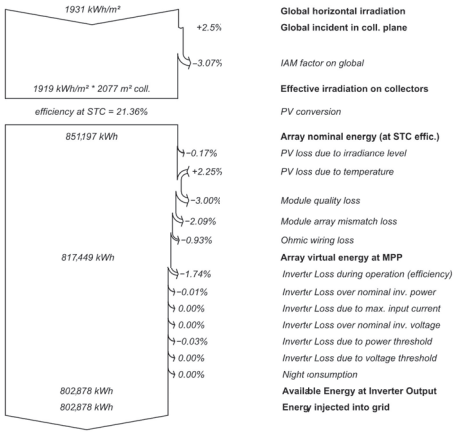


Figure 18. SSFV loss diagram, PVsyst 7.3 software.

3.4.10. Economic and Environmental Analysis

For the economic evaluation of the project, the following criteria will be taken into account [46,47].

The economic analysis is based on the *SSFV* proposal, selected due to its comprehensive design, which includes a higher number of *PV* modules and inverters compared to other options. This structure results in the highest net cost, making it ideal for an in-depth economic feasibility study. The unit costs of components, assembly services, and financial parameters used in the evaluation were sourced from company data and literature. Given the unique economic and financial conditions in Cuba, a dual analysis was conducted: the enterprise approach evaluates the direct impact on the organization, while the national approach assesses the potential savings from reduced oil imports.

This study uses an exchange rate of USD 1 = CUP 120, as established by the Central Bank of Cuba. Financial parameters include a discount rate of 8%, an inflation rate of 5%, a risk margin of 3%, and a tax rate of 35%, based on reference [48]. Equipment costs, including installation, were sourced from Corey Solar (Mexico) and Alibaba (China) for modules, inverters, and related components, reflecting special pricing for bulk quantities.

- Each 550 Wp module would cost *USD* 60 per unit, so the 804 modules would have a total value of *USD* 48,240.
- Inverters cost *USD* 3000 per 100 kW inverter (quantity 2), *USD* 7500 per 274 kW inverter (quantity 1), and *USD* 75 per 0.70 kW inverter (quantity 1); and for the installation, four inverters are needed for a value of *USD* 10,575.

https://doi.org/10.3390/a19010027

- Labor *USD* 0.02 per Wp would be 442 kWp for a total of *USD* 8840.
- Structure *USD* 0.02 per Wp would be 442 kWp for a total of *USD* 8840.
- Electrical equipment *USD* 0.01 per Wp would be 442 kWp for a total of *USD* 4420.
- For a total of *USD* 80,915.

The economic evaluation of the proposal considers the number of modules and system losses to estimate daily production, based on an annual average of 5 peak solar hours (*HSP*) in Cuba. Additionally, the payment for *kWh* generated by the proposed *PV* system and delivered to the National Electroenergetic System (SEN) is calculated using Decree Law 345 of the Ministry of Justice of Cuba. This law states that the purchase price of electricity is based on the avoided cost of fossil fuel-based generation, among other factors.

Interviews with officials from the National Office for the Control of the Rational Use of Energy (ONURE), including its director, provided a reference purchase price of 1.5869 $\frac{\text{CUP}}{\text{kWh}}$ for electricity generated outside the SEN [49]. The economic viability of the project, as assessed using data from Corey Solar de México, is summarized in Figure 19.

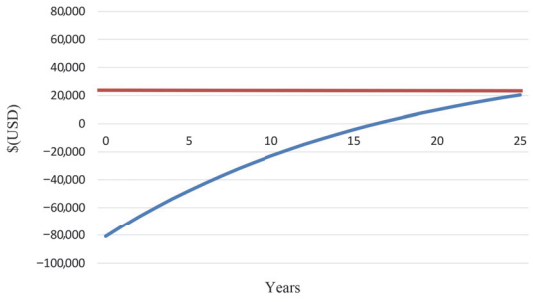


Figure 19. Economic analysis of the SSFV.

Given the existing economic situation in Cuba, due to the dual currency, the high inflation margins, and the subsidy on electricity and other social services, Figure 19 shows the TIR of −10% and the VPN of 20,000 *USD*, which means that the project will be recovered in 17 years; if future maintenance, breakage, depreciation, and the decrease in the efficiency of the SSFV are taken into account, which indicates that the income is not enough to cover the expenses; therefore, the project is not profitable. Therefore, the incorporation of renewable energy in the country is not considered feasible.

3.4.11. Economic Evaluation at Country Level

The income from fuel savings is given in Equation (24):

$$G_{comb} = E_{gelec} \frac{365\text{day}}{\text{year}} g_{espgelec} \tag{24}$$

where

E_{gelec}—Electric power generated by the genset;

$g_{espglec}$—Specific consumption of the genset. It is considered a value of the most modern equipment (0.236 $\frac{\text{kg}}{\text{kWh}}$) as it does not have the real value of the equipment in operation.

Substituting values into Equation (25) as follows:

$$G_{\text{comb}} = \frac{2230\,\text{kWh}}{\text{day}} \times \frac{365\,\text{day}}{\text{year}} \times \frac{0.236\,\text{kg}}{\text{kWh}} = 192{,}092.2\,\frac{\text{kg}}{\text{year}} \tag{25}$$

https://doi.org/10.3390/a19010027

Considering the value of the density of diesel (ρ_3) of 850 kg/m^3, the volumetric flow of diesel is 225,990.8 L/year. The value of a liter of diesel for Cuba is given in the web page of the site [50].

1 L = USD 1; EUR 0.831.

The income from fuel savings in USD is as follows:

Income from diesel savings = 225,990.8 L/year × USD 1/L = USD 225,990.8/year.

3.4.12. Environmental Analysis

Figure 20 shows the amount of pollutant emissions that this SSFV would contribute to net emitting pollutants.

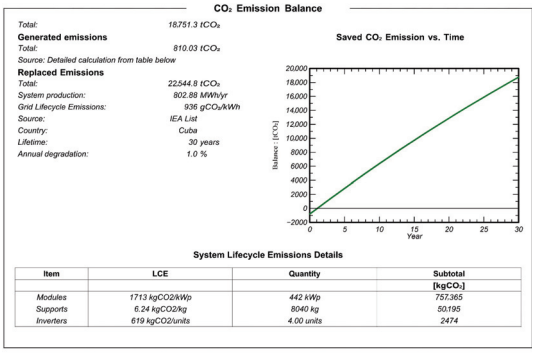


Figure 20. Diagram and results of the environmental analysis.

4. Discussions

Table 2 presented allows a comprehensive comparison between different research studies carried out in different geographical, climatic, and analytical contexts. This comparative analysis shows that the methodology proposed in this paper is innovative and distinctive. This is because it incorporates a comprehensive combination of variables and approaches ranging from energy efficiency and management to the use of advanced artificial intelligence (AI)-based optimization techniques for solar photovoltaic system integration (SSFV). In addition, it includes a detailed analysis of economic feasibility and a study of pollutant emissions, allowing both environmental impact and economic benefits to be assessed. Overall, this proposal not only addresses gaps in previous research but also provides an interdisciplinary approach that could set a new standard in the evaluation of sustainable energy solutions.

Table 2. Comparisonof methodologies and studies on the integration of SSFV in buildings.

Ref.	Energy Analysis	Efficiency Analysis Methods	Optimization and Integration	Economic and Environmental Assessment
[51]	Energy consumption behavior	Do not perform	They do not use any optimization method	Do not take into account the TIR
[52]	Energy consumption behavior	Adjustments to the photovoltaic array	They do not use any optimization method	The corresponding calculations and estimates are made
[53]	It does not have	Make a hybrid solar photovoltaic–thermal hybrid system	They do not use any optimization method	It does not have

https://doi.org/10.3390/a19010027

Table 2. *Cont.*

Ref.	Energy Analysis	Efficiency Analysis Methods	Optimization and Integration	Economic and Environmental Assessment
[54]	It does not have	Make a hybrid solar photovoltaic–thermal hybrid system	They do not use any optimization method	It does not have
[55]	It does not have	make a hybrid solar photovoltaic–thermal hybrid system	They do not use any optimization method	It does not have
[56]	It does not have	Make a hybrid solar photovoltaic–thermal hybrid system	They do not use any optimization method	Do not take into account the TIR
[57]	It does not have	Make a hybrid solar photovoltaic–thermal hybrid system	They do not use any optimization method	The corresponding calculations and estimates are made
Our research	An energy performance indicator is designed and planning for energy generation is carried out	Performs climatological, shading, support design and structural studies	A genetic algorithm is used	A feasibility study and economic forecasts and an environmental analysis are carried out

4.1. Discussion: Energy Performance Indicators

A correlation analysis was conducted, revealing that the variable (HDO × DG) has a significant impact on the hotel's energy consumption model, as evidenced by a high coefficient of determination (R^2 = 0.97). One of the innovative aspects of this study lies in the integration of renewable energy sources, which would enable significant reductions in electricity consumption—currently the most utilized energy vector within the facility. This reduction would have a direct impact on operational efficiency. Additionally, the replacement of traditional lighting systems with more efficient alternatives would result in considerable cost savings. Together, these measures not only enhance the environmental sustainability of the installation but also increase the hotel's competitiveness in the market.

Table 3 shows the results chosen by other authors in similar studies varying the energy performance indicator.

Table 3. Analysis of the most commonly used energy performance indicators in the hotel industry.

Energy Performance Indicator Used	Correlation Obtained (R^2)	Author	Linear Regression	Standard Used (ISO 50001)
$\frac{kWh}{m^2}$	0.72	[58]	s	2018
$\frac{kWh}{Degree\ Days}$	0.73	[59]	s	2011
$\frac{kWh}{DG}$	0.77	[60]	s	2011
$\frac{kWh}{Equvalent\ Occupation}$	0.80	[61]	s	2011
$\frac{kWh}{HDO\ Equvalent}$	0.90	[62]	s	2011
$\frac{kWh}{Day*HDO}$	0.91	[63]	s	2011
$\frac{kWh}{HDO*DG}$	0.92	[13]	s	2011
$\frac{kWh}{E*C*A}$	0.77	[64]	m	2018
$\frac{kWh}{DG*HDO}$	0.90	**Our research**	m	2018
$\frac{kWh}{DG*HDO}$	0.97	**Our research**	s	2018

https://doi.org/10.3390/a19010027

4.2. Discussion: Optimization of the SSFV

This section focuses on the analysis of the proposals created in this research in relation to other similar studies in this field. Key aspects for comparison include the orientation of the system, the location of the study, the type of module technology applied, the number of modules used, the area considered in the analysis, and the expected costs.

Table 4 presents the main elements comparing the results achieved in this research with those of other studies in the same field.

Table 4. Statistical data obtained from the study on the main variables that affect the efficiency of a plant.

Orientation	Location	Modules	Installed Power (kW_p)	PR (%)/Efficiency	Analysis Type	Reference
North (9°)	Cajamarca, Peru	30	9.6	84	PVsyst	[65]
South (20°)	Odish Institute of Technology, India	360	90	79	PVsyst	[66]
South (7°)	Cedars Hotel, Jordan	912	300	82	PVsyst	[67]
South (13°)	India	158	54.25	79.80	PVsyst	[68]
South (32°)	Morocco	3924	1000	77.3	PVsyst	[69]
South (20°)	Turkey	54	29.4	72.8	PVsyst	[70]
East–West (30°)	Jordan	64	26	78	PVsyst	[71]
South (30°)	Iraq	15	4.3	80–85	Genetic Algorithm	[72]
South (32°)	China	N/A	N/A	80–90	Collaborative Optimization	[73]
South	Sweden	N/A	322 kWh	85–90	Genetic Algorithm	[74]
Southwest (15°)	Hotel Punta la Cueva, Cuba	804	442.2	91.72/91.05	SketchUp–PVsyst–GA	**Our research**

This study emphasizes the significance of system orientation and design in optimizing photovoltaic (PV) system performance. The findings reveal that most studies align system orientation with the latitude of the installation site to enhance efficiency by optimizing the incidence of solar radiation on the module's surface. Unlike other research, this study also examines the impact of operating temperature on PV module efficiency, highlighting its importance in improving energy production. Additionally, the manipulation of tilt and azimuth angles is identified as a critical factor in achieving higher performance ratios (PR) and meeting significant portions of daily energy demands. Notably, while many studies calculate efficiency indices, they often adopt varying methods and exclude several key variables analyzed in this research. Many of their values were estimated to establish correlations for comparison, whereas this study integrates a comprehensive range of variables for greater accuracy.

A key innovation lies in employing a building-integrated photovoltaic (BIPV) approach to maximize installation areas by modifying building structures, a methodology not widely addressed in prior research. This is complemented by a detailed analysis of maximum power point tracking (MPPT) voltages and inverter efficiency, addressing aspects often overlooked in similar studies. Furthermore, the research underscores the environmental benefits of PV systems, such as reducing greenhouse gas emissions. Its practical

https://doi.org/10.3390/a19010027

application extends globally, provided local regulations, climatic conditions, and building orientation are considered. This study offers valuable insights for project planning, promoting the seamless integration of renewable energy into power grids while optimizing space, resources, and system efficiency through a holistic analysis.

5. Conclusions

This study presents a comprehensive methodology for the integration of standalone solar photovoltaic systems (SSFV) into building energy management frameworks, using Hotel Punta la Cueva as a case study. The main findings of the research can be summarized as follows:

- A structured methodology was developed that integrates energy performance analysis, SSFV system design, and artificial intelligence-based optimization. The proposed approach demonstrates the feasibility of combining energy efficiency assessment with photovoltaic system optimization within a unified framework.
- An energy performance indicator based on the combined variable HDO $\times$ DG was validated, showing a strong correlation with system behavior ($R^2 = 0.97$). This indicator enabled the identification of energy-saving potential not associated with production, achieving a reduction of 5.3% relative to the baseline.
- The application of a genetic algorithm for SSFV optimization led to a non-conventional azimuth configuration, resulting in an increase in energy production of 14.75%, an improvement in the performance ratio, and a more efficient use of the available installation area compared to the conventional south-oriented configuration.
- The economic and environmental assessment revealed that, although the project presents limited financial profitability at the facility level, it offers significant benefits at the national and environmental levels, including fuel savings and a substantial reduction in greenhouse gas emissions.

Despite these contributions, some limitations should be acknowledged. The proposed methodology was applied to a single case study under specific climatic and operational conditions, which may limit the direct generalization of the results to other contexts. Additionally, the optimization process focused primarily on geometric and performance-related variables, without explicitly considering dynamic operational control or real-time system adaptation.

Future research should extend the proposed framework to multiple building typologies and climatic regions, incorporate dynamic and real-time data for adaptive optimization, and explore hybrid optimization and machine learning techniques to further enhance SSFV performance and decision-making robustness.

Author Contributions: Conceptualization, L.A.I.C., J.M.Á.-A., G.I.P.-S. and C.G.M.-P.; methodology, L.A.I.C., J.M.Á.-A. and G.I.P.-S.; software, L.A.I.C. and O.R.-A.; validation, O.R.-A. and J.R.-R.; formal analysis, L.A.I.C., J.M.Á.-A. and J.R.-R.; investigation, L.A.I.C. and O.R.-A.; data curation, L.A.I.C. and J.R.-R.; writing—original draft preparation, L.A.I.C., O.R.-A. and J.M.Á.-A.; writing—review and editing, O.R.-A., G.I.P.-S. and J.R.-R.; visualization, L.A.I.C. and J.R.-R.; supervision, O.R.-A. and J.R.-R. 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 raw data supporting the conclusions of this article will be made available by the authors on request.

https://doi.org/10.3390/a19010027

Acknowledgments: The authors would like to acknowledge the support of Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCYT) in the production of this work.

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

References

1. Thangavelu, S.; Umapathy, P. Design of new high step-up DC-DC converter Topology for solar PV applications. *Int. J. Photoenergy* **2021**, *2021*, 7833628. [CrossRef]
2. González Gaudiano, E.J.; Meira Cartea, P.Á. Educación para el cambio climático: ¿Educar sobre el clima o para el cambio? *Perfiles Educ.* **2020**, *42*, 157–174. [CrossRef]
3. Olabi, A.; Abdelkareem, M.A. Renewable energy and climate change. *Renew. Sustain. Energy Rev.* **2022**, *158*, 112111. [CrossRef]
4. Sher, F.; Curnick, O.; Azizan, M.T. Sustainable conversion of renewable energy sources. *Sustainability* **2021**, *13*, 2940. [CrossRef]
5. Aleksandr, S.B.; Sergey, A.Y.; Aleksandr, S.S.; Aleksandr, F.S.; Galina, M.B. Implementation of ISO 50001 standard in the activities of energy companies. In Proceedings of the 2021 International Conference on Electrotechnical Complexes and Systems (ICOECS), Ufa, Russia, 16–18 November 2021; pp. 417–420.
6. Sousa Lira, J.M.; Salgado, E.G.; Beijo, L.A. Which factors does the diffusion of ISO 50001 in different regions of the world is influenced? *J. Clean. Prod.* **2019**, *226*, 759–767. [CrossRef]
7. Marimon, F.; Casadesús, M. Reasons to adopt ISO 50001 energy management system. *Sustainability* **2017**, *9*, 1740. [CrossRef]
8. Zhou, X.; Mei, Y.; Liang, L.; Fan, Z.; Yan, J.; Pan, D. A dynamic energy benchmarking methodology on room level for energy performance evaluation. *J. Build. Eng.* **2021**, *42*, 102837. [CrossRef]
9. Dibene-Arriola, L.M.; Carrillo-González, F.M.; Quijas, S.; Rodríguez-Uribe, M.C. Energy efficiency indicators for hotel buildings. *Sustainability* **2021**, *13*, 1754. [CrossRef]
10. Teng, Z.R.; Wu, C.Y.; Xu, Z.Z. New energy benchmarking model for budget hotels. *Int. J. Hosp. Manag.* **2017**, *67*, 62–71. [CrossRef]
11. Salem, R.; Bahadori-Jahromi, A.; Mylona, A.; Godfrey, P.; Cook, D. Energy performance and cost analysis for the nZEB retrofit of a typical UK hotel. *J. Build. Eng.* **2020**, *31*, 101403. [CrossRef]
12. Palani, H.; Karatas, A. Identifying Energy-Use Behavior and Energy-Use Profiles of Hotel Guests. *Appl. Sci.* **2021**, *11*, 6093. [CrossRef]
13. Álvarez Guerra Plasencia, M.A.; Cabello Eras, J.J.; Sousa Santos, V.; Sagastume Gutiérrez, A.; Haeseldonckx, D.; Vandecasteele, C. Experiencias en la utilización de información meteorológica para el pronóstico y control del consumo de electricidad en hoteles. In Proceedings of the XI Congreso de la Asociación Española de Climatología, Cartagena, Spain, 17–19 October 2018; Asociación Española de Climatología: Barcelona, Spain, 2018.
14. Faiz, F.U.H.; Shakoor, R.; Raheem, A.; Umer, F.; Rasheed, N.; Farhan, M. Modeling and analysis of 3 MW solar photovoltaic plant using PVSyst at Islamia University of Bahawalpur, Pakistan. *Int. J. Photoenergy* **2021**, *2021*, 6673448. [CrossRef]
15. Junaidh, P.; Vijay, A.; Mathew, M. Power enhancement of solar photovoltaic module using micro-climatic strategies in warm-humid tropical climate. In Proceedings of the 2017 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 21–22 April 2017; pp. 1–6.
16. Kumar, N.M. Simulation tools for technical sizing and analysis of solar PV systems. In Proceedings of the 6th World Conference on Applied Sciences, Engineering and Technology (WCSET-2017), Bangkok, Thailand, 11–12 October 2017; Volume 201, pp. 218–222.
17. Huang, K.H.; Chao, K.H.; Lee, T.W. An Improved Photovoltaic Module Array Global Maximum Power Tracker Combining a Genetic Algorithm and Ant Colony Optimization. *Technologies* **2023**, *11*, 61. [CrossRef]
18. Chander, S.; Purohit, A.; Sharma, A.; Arvind; Nehra, S.; Dhaka, M. A study on photovoltaic parameters of mono-crystalline silicon solar cell with cell temperature. *Energy Rep.* **2015**, *1*, 104–109. [CrossRef]
19. Ramya, D.; Krishnakumari, A.; Dineshkumar, P.; Srivastava, M.P.; Kannan, L.V.; Puthilibai, G.; Kumar, P.M. Investigating the influence of nanoparticle disbanded phase changing material (NDPCM) on the working of solar PV. *Mater. Today Proc.* **2022**, *56*, 1341–1346. [CrossRef]
20. Nadia, M.; Lassad, H.; Abderrahmen, Z.; Abdelkader, C. Influence of temperature and irradiance on the different solar PV panel technologies. *Int. J. Energy Sect. Manag.* **2021**, *15*, 421–430. [CrossRef]
21. Venkateswari, R.; Sreejith, S. Factors influencing the efficiency of photovoltaic system. *Renew. Sustain. Energy Rev.* **2019**, *101*, 376–394. [CrossRef]
22. Kuznetsov, P.N.; Kuvshinov, V.V.; Issa, H.A.; Mohammed, H.J.; Al Barmani, A.G. Investigation of the losses of photovoltaic solar systems during operation under partial shading. *J. Appl. Eng. Sci.* **2020**, *18*, 313–320. [CrossRef]
23. Rana, A.S.; Nasir, M.; Khan, H.A. String level optimisation on grid-tied solar PV systems to reduce partial shading loss. *IET Renew. Power Gener.* **2018**, *12*, 143–148. [CrossRef]

https://doi.org/10.3390/a19010027

24. Zsiboracs, H.; Zentko, L.; Pinter, G.; Vincze, A.; Baranyai, N.H. Assessing shading losses of photovoltaic power plants based on string data. *Energy Rep.* **2021**, *7*, 3400–3409. [CrossRef]
25. Patel, A. Mejora de la eficiencia de la transferencia de calor en sistemas solares térmicos mediante intercambiadores de calor avanzados. *Multidiscip. Int. J. Res. Dev. (MIJRD)* **2023**, *02*, 31–51. Available online: https://www.mijrd.com/papers/v2/i6/MIJRDV2I60003.pdf (accessed on 15 December 2025).
26. Tembhare, S.P.; Barai, D.P.; Bhanvase, B.A. Performance evaluation of nanofluids in solar thermal and solar photovoltaic systems: A comprehensive review. *Renew. Sustain. Energy Rev.* **2022**, *153*, 111738. [CrossRef]
27. Osorio, J.D.; Wang, Z.; Karniadakis, G.; Cai, S.; Chryssostomidis, C.; Panwar, M.; Hovsapian, R. Forecasting solar-thermal systems performance under transient operation using a data-driven machine learning approach based on the deep operator network architecture. *Energy Convers. Manag.* **2022**, *252*, 115063. [CrossRef]
28. Verma, S.K.; Kumar, R.; Barthwal, M.; Rakshit, D. A review on futuristic aspects of hybrid photo-voltaic thermal systems (PV/T) in solar energy utilization: Engineering and Technological approaches. *Sustain. Energy Technol. Assess.* **2022**, *53*, 102463.
29. Zhang, C.; Shen, C.; Zhang, Y.; Pu, J. Feasibility investigation of spectral splitting photovoltaic/thermal systems for domestic space heating. *Renew. Energy* **2022**, *192*, 231–242. [CrossRef]
30. Derick, M.; Rani, C.; Rajesh, M.; Farrag, M.; Wang, Y.; Busawon, K. An improved optimization technique for estimation of solar photovoltaic parameters. *Sol. Energy* **2017**, *157*, 116–124. [CrossRef]
31. Ismail, M.S.; Moghavvemi, M.; Mahlia, T. Characterization of PV panel and global optimization of its model parameters using genetic algorithm. *Energy Convers. Manag.* **2013**, *73*, 10–25. [CrossRef]
32. Zamen, M.; Baghban, A.; Pourkiaei, S.M.; Ahmadi, M.H. Optimization methods using artificial intelligence algorithms to estimate thermal efficiency of PV/T system. *Energy Sci. Eng.* **2019**, *7*, 821–834. [CrossRef]
33. Pardo de Vera García, J. Optimización del Dimensionamiento de Plantas Solares Fotovoltaicas. Ph.D. Thesis, Universidad Pontificia Comillas, Madrid, Spain, 2020.
34. Odry, Á. An open-source test environment for effective development of marg-based algorithms. *Sensors* **2021**, *21*, 1183. [CrossRef]
35. Odry, Á.; Tadic, V.L.; Odry, P. A stochastic logic-based fuzzy logic controller: First experimental results of a novel architecture. *IEEE Access* **2021**, *9*, 29895–29920. [CrossRef]
36. Benítez Soler, A.C.; Tello Reyes, I.D. Estudio de Factibilidad de la Implementación de un Sistema Solar Fotovoltaico en la Finca Villa Catalina. Ph.D. Thesis, Universidad Libre,Bogotá, Colombia, 2018.
37. Lamigueiro, O.P. Energía solar fotovoltaica. In *Creative Commons Ebook*; RC Libros: Madrid, Spain, 2013; p. 192.
38. Delgado, R.M.B.; Yanes, C.J.P.M. Anteproyecto de sistema solar fotovoltaico en la Delegación Provincial de Materiales de la Construcción, Cienfuegos. In Proceedings of the I Convención Científica Internacional de la Universidad de Cienfuegos, XI Taller Internacional de Energía, Medio Ambiente y Desarrollo Sostenible, Cienfuegos, Cuba, 22 November 2022.
39. Callisaya Condori, A.; Romay Ossio, M. Calculo y Diseño de un Sistema Solar Fotovoltaico Auxiliar para el Hospital de La Asunta-Sud Yungas. Ph.D. Thesis, Universidad Mayor de San Andrés, La Paz, Bolivia, 2012.
40. Crespo Sánchez, G.; Monteagudo Yanes, J.P.; Montesino Pérez, M.; Cruz Virosa, I.; Cabrera Sánchez, J.L. La gestión energética en la fabricación de piensos balanceados en Cienfuegos. *Rev. Univ. Soc.* **2019**, *11*, 249–256.
41. Vassiliades, C.; Agathokleous, R.; Barone, G.; Forzano, C.; Giuzio, G.; Palombo, A.; Buonomano, A.; Kalogirou, S. Building integration of active solar energy systems: A review of geometrical and architectural characteristics. *Renew. Sustain. Energy Rev.* **2022**, *164*, 112482. [CrossRef]
42. Vassiliades, C.; Barone, G.; Buonomano, A.; Forzano, C.; Giuzio, G.; Palombo, A. Assessment of an innovative plug and play PV/T system integrated in a prefabricated house unit: Active and passive behaviour and life cycle cost analysis. *Renew. Energy* **2022**, *186*, 845–863. [CrossRef]
43. Deymi-Dashtebayaz, M.; Nikitin, A.; Davoodi, V.; Nikitina, V.; Hekmatshoar, M.; Shein, V. A new multigenerational solar energy system integrated with near-zero energy building including energy storage—A dynamic energy, exergy, and economic-environmental analyses. *Energy Convers. Manag.* **2022**, *261*, 115653. [CrossRef]
44. Su, B.; Lin, F.; Ma, J.; Huang, S.; Wang, Y.; Zhang, X.; Han, W.; Wang, H. System integration of multi-grade exploitation of biogas chemical energy driven by solar energy. *Energy* **2022**, *241*, 122857. [CrossRef]
45. Tamer, T.; Dino, I.G.; Baker, D.K.; Akgül, C.M. Coupling PCM wallboard utilization with night Ventilation: Energy efficiency and overheating risk in office buildings under climate change impact. *Energy Build.* **2023**, *298*, 113482. [CrossRef]
46. Marcelo García, J.S. Análisis Económico y Ambiental de la Implementación de Energía Fotovoltaica en las Instalaciones de Fresenius Medical Care Como Alternativa Frente al Alto Consumo Energético. Ph.D. Thesis, Universidad El Bosque, Bogota, Colombia, 2022.
47. Pupo, G.H.; De la Paz Pérez, G.A.; De la Paz Vento, G.; Mendoza, L.E. Evaluación de opciones de inversión en eficiencia energética y fuentes renovables de energía en hoteles de Santa Lucía, Camagüey. *Rev. Arquit. E Ing.* **2022**, *16*, 1–7.

48. Banco Central de Cuba. Tipo de Cambio Oficial con Relación al Peso Cubano. Available online: https://www.bc.gob.cu (accessed on 22 November 2023).
49. Sevajanes, R. Tarifa Eléctrica Promedio para Instalaciones Hoteleras en Cuba. Available online: https://www.onure.cu/ (accessed on 22 November 2023).
50. Cuba Gasoline Prices. Available online: https://www.globalpetrolprices.com/Cuba/ (accessed on 15 June 2023).
51. Guzmán Villavicencio, M.; Soto Castellón, C.R.; Águila Bernal, I.; Torres Águila, J.M. Procedimiento para instalación de un sistema fotovoltaico sobre techos en la corporación cuba ron sa. *Cent. Azúcar* **2017**, *44*, 70–81.
52. Albert López, E. Instalación de Placas Solares en una Nave Industrial para Autoconsumo. Ph.D. Thesis, Universidad Pontificia Comillas, Madrid, Spain, 2020.
53. Chiang, W.; Permana, I.; Wang, F.; Chen, H.; Erdenebayar, M. Experimental investigation for an innovative hybrid photovoltaic/Thermal (PV/T) solar system. *Energy Rep.* **2022**, *8*, 910–918. [CrossRef]
54. Gautam, K.R.; Andresen, G.B. Performance comparison of building-integrated combined photovoltaic thermal solar collectors (BiPVT) with other building-integrated solar technologies. *Sol. Energy* **2017**, *155*, 93–102. [CrossRef]
55. Alshibil, A.M.; Farkas, I.; Víg, P. Multi-aspect approach of electrical and thermal performance evaluation for hybrid photovoltaic/thermal solar collector using TRNSYS tool. *Int. J. Thermofluids* **2022**, *16*, 100222. [CrossRef]
56. Tuncer, A.D.; Khanlari, A.; Afshari, F.; Sözen, A.; Çiftçi, E.; Kusun, B.; Şahinkesen, İ. Experimental and numerical analysis of a grooved hybrid photovoltaic-thermal solar drying system. *Appl. Therm. Eng.* **2023**, *218*, 119288. [CrossRef]
57. Deymi-Dashtebayaz, M.; Rezapour, M.; Farahnak, M. Modeling of a novel nanofluid-based concentrated photovoltaic thermal system coupled with a heat pump cycle (CPVT-HP). *Appl. Therm. Eng.* **2022**, *201*, 117765. [CrossRef]
58. eddine Mechri, H.; Amara, S. Investigation and analysis of energy and water use of hotel buildings in Tunisia. *Energy Build.* **2021**, *241*, 110930. [CrossRef]
59. Molina González, A.; Velarde Bedregal, H.R.; Borroto Nordelo, A.E.; Santiesteban Toca, C.E.; Monteagudo Yanes, J.P. Nuevos índices de consumo energético para hoteles tropicales. *Ing. Energ.* **2017**, *38*, 198–207.
60. Mendoza, R.P.C.; Yanes, J.P.M.; Nordelo, A.B.; Oqueña, E.C.Q. Línea de Base Energética en la implementación de la norma ISO 50001. Estudios de casos. *Hombre Máquina* **2015**, 137–143.
61. Ochoa, G.V. Application of equivalent occupation method as a tool for energy management in hotel sector. *Int. J. Energy Econ. Policy* **2018**, *8*, 187–192.
62. Rodríguez, L.R.; Insuasti, J.A.P.; Peña, W.Y.; Sierra, C.O.; Arroyave, C.P.S.; Soto, C.A.P.; Vispo, N.F.S.; Pinchao, J.M.H.; Torres, R.D.G.; Lara, G.R. Método de cálculo del índice de eficiencia energética de los hoteles. *Revista Tecnol. ESPOL* **2017**, *30*, 1–10. Available online: https://www.rte.espol.edu.ec (accessed on 22 November 2023).
63. Eras, J.J.C.; Santos, V.S.; Gutiérrez, A.S.; Plasencia, M.Á.G.; Haeseldonckx, D.; Vandecasteele, C. Tools to improve forecasting and control of the electricity consumption in hotels. *J. Clean. Prod.* **2016**, *137*, 803–812. [CrossRef]
64. Torres Navarro, C.; Malta Callegari, N.; Jara Olave, H. Modelos de regresión y diseño de línea base para indicadores energéticos en una empresa siderúrgica. *Ing. Energ.* **2021**, *42*, 1–10.
65. Mesones Abanto, P.F. Dimensionamiento y Selección de un Sistema Solar Fotovoltaico de Conexión a Red para el Restaurante "El Zarco" Ubicado en la Ciudad de Cajamarca. Ph.D. Thesis, Universidad Nacional Pedro Ruiz Gallo, Lambayeque, Peru, 2019.
66. Dey, D.; Subudhi, B. Design, simulation and economic evaluation of 90 kW grid connected Photovoltaic system. *Energy Rep.* **2020**, *6*, 1778–1787. [CrossRef]
67. Al-Zoubi, H.; Al-Khasawneh, Y.; Omar, W. Design and feasibility study of an on-grid photovoltaic system for green electrification of hotels: A case study of Cedars hotel in Jordan. *Int. J. Energy Environ. Eng.* **2021**, *12*, 611–626. [CrossRef]
68. Sharma, S.; Kurian, C.P.; Paragond, L.S. Solar PV system design using PVsyst: A case study of an academic Institute. In Proceedings of the 2018 International Conference on Control, Power, Communication and Computing Technologies (ICCPCCT), Kannur, India, 23–24 March 2018; pp. 123–128.
69. Belmahdi, B.; El Bouardi, A. Solar potential assessment using PVsyst software in the northern zone of Morocco. *Procedia Manuf.* **2020**, *46*, 738–745. [CrossRef]
70. Dindar, T.; Vedat, E.; Sarkin, A.S. Comparison of Simulation Results for 25 kW Power Output Rooftop PV System. *Eur. J. Tech. (EJT)* **2022**, *12*, 176–181. [CrossRef]
71. Abu Qadourah, J. Energy and economic potential for photovoltaic systems installed on the rooftop of apartment buildings in Jordan. *Results Eng.* **2022**, *16*, 100642. [CrossRef]
72. Hassan, Q. Evaluation and optimization of off-grid and on-grid photovoltaic power system for typical household electrification. *Renew. Energy* **2021**, *164*, 375–390. [CrossRef]

https://doi.org/10.3390/a19010027

73. Fu, X.; Zhou, Y. Collaborative Optimization of PV Greenhouses and Clean Energy Systems in Rural Areas. *IEEE Trans. Sustain. Energy* **2023**, *14*, 642–656. [CrossRef]
74. Huang, P.; Sun, Y.; Lovati, M.; Zhang, X. Solar-photovoltaic-power-sharing-based design optimization of distributed energy storage systems for performance improvements. *Energy* **2021**, *222*, 119931. [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/a19010027

algorithms

MDPI

Article

Semi-Fragile Watermarking Scheme for High-Resolution Color Images: Tamper Identification, Ownership Authentication, and Self-Recovery

Manuel Cedillo-Hernandez [1], Antonio Cedillo-Hernandez [2,*], Francisco Javier Garcia-Ugalde [3] and Juan Carlos Sanchez-Garcia [1,*]

[1] Seccion de Estudios de Posgrado e Investigacion, Escuela Superior de Ingenieria Mecanica y Electrica Unidad Culhuacan, Instituto Politecnico Nacional, Av. Sta. Ana 1000 Colonia Culhuacan CTM V, Alcaldia Coyoacan, Ciudad de Mexico C.P. 04440, Mexico; mcedilloh@ipn.mx

[2] Escuela de Ingenieria y Ciencias, Tecnologico de Monterrey, Av. Eugenio Garza Sada 2501, Colonia Tecnologico, Monterrey C.P. 64700, Mexico

[3] Departamento de Procesamiento de Señales, Facultad de Ingenieria, Universidad Nacional Autonoma de Mexico, Av. Universidad 3000, Colonia Ciudad Universitaria, Alcaldia Coyoacan, Ciudad de Mexico C.P. 04510, Mexico; fgarciau@unam.mx

* Correspondence: acedillo@tec.mx (A.C.-H.); jcsanche@ipn.mx (J.C.S.-G.)

Abstract

The advancements in communication and information technologies have substantially enabled the extensive distribution and modification of high-resolution color images. Although this accessibility provides many advantages, it also presents risks related to security. Specifically, when image modification is conducted with malicious intent, exceeding typical artistic or enhancement objectives, it can cause significant moral or economic harm to the image owner. To address this security requirement, this study presents an innovative semi-fragile watermarking algorithm designed specifically for high-resolution color images. The proposed method utilizes Discrete Cosine Transform domain watermarking implemented via Quantization Index Modulation with Dither Modulation. It incorporates several elements, such as convolutional encoding, a denoising convolutional neural network, and a very deep super-resolution neural network. This comprehensive strategy aims to provide ownership verification using a logo watermark, in conjunction with tamper detection and content self-recovery mechanisms. The self-recovery criterion is determined using a thumbnail image, created by downscaling to standard definition and applying JPEG2000 lossy compression. The resultant multifunctional design enhances the overall security of the information. Experimental validation confirms the enhanced imperceptibility, robustness, and capacity of the proposed method. Its efficacy was additionally corroborated through comparative analyses using contemporary state-of-the-art algorithms.

Keywords: semi-fragile watermarking; tamper detection; image content authentication; JPEG2000 compression; quantization index modulation; discrete cosine transformation; convolutional encoding; deep learning

1. Introduction

Currently, advances in communication and information technologies allow for generating, capturing, editing, storing, and sharing high-resolution color images in a massive and immediate manner. In this way, supported by software edition tools integrated into

smartphones and personal computers, end users can perform image editing for artistic purposes or image enhancement. However, when image editing is done to alter the content maliciously, such as removing or adding non-existent elements to the original version, and the tampered version is shared, socialized, or transmitted without the image owner's approval, it can result in moral or economic harm. Conventional informatics security tools cannot solve all information security issues related to tamper detection and ownership authentication of color images. Consequently, the field of data hiding has been extensively studied to embed logos or text within color images, thereby establishing a trustworthy link between the hidden information and the visual content. Within the scope of this paper and the spectrum of data hiding techniques, we identified a method known as digital watermarking, which, in general terms, involves embedding or concealing binary data associated with multimedia ownership to ensure copyright protection and safeguard intellectual property [1,2]. Digital watermarking algorithms typically need to address three fundamental requirements: imperceptibility, robustness, and payload capacity. They are often structured into three stages: embedding, detection or extraction, and watermark generation [3]. The imperceptibility criterion pertains to assessing the visual distortion produced during the embedding stage. Robustness denotes the watermark signal's resilience to signal processing distortions beyond which recovery or detection becomes unfeasible. The payload denotes the quantity of data bits that can be incorporated during the embedding phase without compromising imperceptibility and robustness [4]. Digital watermarking is classified into different modalities, including fragile, semi-fragile, and robust, each providing a specific performance based on its design and the significance of the aforementioned requirements [5–9]. Fragile watermarking algorithms focus on imperceptibility and payload capacity but generally lack consideration for robustness, whereas robust techniques prioritize robustness but often at the expense of imperceptibility and payload. Conversely, semi-fragile watermarking schemes are explicitly engineered to withstand specified benign image processing operations without compromising the integrity of the watermark. These procedures frequently involve a range of distortions, including both lossy and lossless compression, noise interference, filtering, and file format transformation. Crucially, the essential criteria for imperceptibility (invisibility) and payload (data capacity) maintain the same critical level of importance as those defined for fragile watermarking techniques. Whether the design is fragile, semi-fragile, or robust, within the scope of digital image forgery and tampering, a primary challenge lies in developing a watermarking algorithm that not only effectively detects tampering but also possesses the capability to recover and restore as much of the tampered content as possible. Numerous watermarking methods for forgery and tamper detection have been documented in the scientific literature, categorized as fragile [10–16], semi-fragile [17–22], robust [23–27], and hybrid (robust/fragile) [28–35]. This paper presents a semi-fragile watermarking algorithm for high-resolution color images that uses watermarking in the discrete cosine transform (DCT) domain through quantization index modulation by dither modulation (QIM-DM) [36,37]. This algorithm also incorporates convolutional encoding with Viterbi decoding [38,39], a denoising convolutional neural network (DnCNN) [40,41], and a very-deep super-resolution neural network (VDSR) [42,43], to facilitate ownership authentication via a binary logo watermark. To achieve tamper detection and self-recovery, a thumbnail image is created by rescaling the original color image to a standard VGA resolution [44] and compressing it using the JPEG2000 lossy algorithm [45,46]. By its design, the proposed method can increase the information security capabilities mentioned above by having a multipurpose watermarking algorithm. The results of the tests performed highlight the contribution of the proposed method.

This work's primary contributions include the following:

 https://doi.org/10.3390/a19010028

- The development of an embedding strategy that maximizes payload capacity while maintaining visual quality and improving resilience against JPEG lossy compression. This approach employs 3×3 2D-DCT blocks to encapsulate two bits of data within a pair of designated AC coefficients via QIM-DM.
- The development of a specific criterion for generating thumbnail images that retains both the luminance and color fidelity of the source image. This criterion is based on the VGA standard resolution and utilizes the JPEG2000 lossy algorithm. The process enables us to minimize the payload required while simultaneously allowing for the restoration of the thumbnail image to its original spatial resolution. The DnCNN and VDSR neural network architectures support this approach, which aims to preserve an acceptable level of visual quality.
- Our proposal avoids implementing two separate watermarking algorithms, which is typical of hybrid methods that combine a robust watermarking algorithm with a fragile one to enhance their effectiveness. Instead, the proposed method utilizes a single semi-fragile watermarking technique to achieve ownership authentication, tamper detection, and self-recovery of content. This approach creates a unique watermark derived from the binary data of the logo and the binarization of the thumbnail image.
- The proposal employs convolutional encoding along with the Viterbi decoding algorithm to enhance robustness against cropping, copy–move, copy–paste, compression, filtering, and image noise corruption.
- The application is performed on high-resolution color images.

The remainder of this article is structured as follows: Section 2 offers a concise review of pertinent related literature. Section 3 provides a detailed explanation of the proposed methodology. Section 4 includes the experimental results, a discussion of the findings, and a comparative analysis against existing state-of-the-art techniques. Lastly, Section 5 provides the concluding remarks derived from this research.

2. Related Works

Currently, several methods exist for detecting tampering and forgery in digital images. Table 1 provides a brief overview of the most recent and relevant works reported in the literature [17–19,21,22,25,27]. We have only included the relevant works in Table 1, as this paper does not aim to be a survey or exhaustive review. Section 4.7 offers an analysis and discussion.

Table 1. Representative digital watermarking approaches to tamper and forgery detection.

Work	Features	Year	Description
[17,18]	Modality: Semi-fragile Purpose: Tamper detection with recovery capability	2020, 2023	The work described in [17] refers to a semi-fragile watermarking method that utilizes the frequency domain (Discrete Cosine Transform (DCT) and Integer Wavelet Transform (IWT)). The system employs a dual-watermarking approach for content authentication and recovery. Specifically, the authentication watermark is embedded within the IWT domain to create the final watermarked image, whereas the recovery data is generated and hidden within the DCT domain. As stated by the authors in [17], performance was evaluated using Normalized Hamming Similarity (NHS) and counterfeit detection mapping to identify manipulations. The work presented in [18] offers a security analysis of [17], identifying several vulnerabilities and demonstrating successful cryptanalysis. In this context, the findings in [18] build upon those of [17] by addressing the security vulnerabilities linked to the use of chaotic maps, which exploit their pseudo-random properties. The method outlined in [18] was tested against various tampering attacks and, according to the authors, shows high effectiveness in both tamper detection and image recovery. Both methodologies were assessed using grayscale images with dimensions of 512 by 512 pixels.

https://doi.org/10.3390/a19010028

Table 1. *Cont.*

Work	Features	Year	Description
[19]	Modality: Semi-fragile Purpose: Tamper localization	2023	The work in [19] describes a semi-fragile watermarking technique designed for forgery localization. This method incorporates the Quaternion Discrete Fourier Transform (QDFT) alongside multi-view fusion. It involves embedding two distinct watermark signals within the QDFT domain: one to improve resilience against geometric distortions and the other to safeguard the image's original content. The authors report that this approach improves counterfeit localization by an average of 11.17% compared to existing methods. Validation of this technique was performed using the CASIA v1.0 dataset, the RAISE dataset, and ten standard 512 × 512 color image pairs.
[21]	Modality: Semi-fragile Purpose: Tamper detection with recovery capability	2025	This research introduces a two-stage method for tamper detection that uses a two-level IWT decomposition, an exclusive-OR logical operation, and Least Significant Bit (LSB) substitution to ensure highly precise localization of alterations. The first phase involves inspecting four distinct segments to identify any modified bits; a segment is flagged if even one bit has been tampered with. The second phase then confirms which of these flagged segments have indeed been altered. The authors also use Normalized Hamming Similarity (NHS) to distinguish between intentional (malicious) and accidental data modifications. Lost content is recovered using an image inpainting technique. This method demonstrated a tamper detection accuracy of roughly 98.3% during object manipulation tests, where the tampering rate was approximately 12.1%. The experimental set consisted of fifteen 512 × 512 grayscale images.
[22]	Modality: Robust-fragile Purpose: Ownership authentication and tamper detection with recovery capability	2025	A novel hybrid scheme that combines robust and fragile watermarking for ownership verification is presented in [22]. This technique uses small, 128 × 128-pixel watermarks, which are embedded into 4 × 4 blocks of the Discrete Wavelet Transform (DWT) coefficients. To enhance the recovery stage, K-means clustering is applied to process every 2 × 2 sub-block of the image. The fragile watermarking combines Schur decomposition, authenticated block bits, and LSB replacement to embed both authentication and recovery data directly into the original image's spatial domain. The authors contend that this integration is essential for efficient forgery detection, as established dependencies among blocks facilitate more precise tampering localization. Furthermore, the K-means algorithm improves the recovery performance and achieves better results in manipulation detection and image restoration. Experiments were conducted using fifteen 512 × 512 grayscale images.
[25]	Modality: Robust zero-watermarking Purpose: Ownership authentication, auxiliary information delivery, and tamper detection	2025	In [25], the authors propose a robust zero-watermarking technique that employs a dual-branch neural network for ownership verification, integrates auxiliary data, and localizes tampering. The system generates three distinct zero-watermarking codes: a binary logo for proving ownership, a QR code for embedding supplementary information, and a halftone version of the original image for counterfeit detection. A key advantage of this method is the complete preservation of the original image. It demonstrates strong resilience against common geometric distortions, signal processing attacks, and complex hybrid attacks, achieving high performance metrics where the Bit Error Rate (BER) is low, and the Normalized Correlation (NC) is near or equal to 1. Across multiple benchmarks, including the MIC-F2000, Realistic Tampering, Columbia, CASIA V2, and a Custom High-Resolution Database, the reported average tamper detection accuracy is 98.7% or greater.
[27]	Modality: Robust zero-watermarking Purpose: Tampering localization with self-recovery	2024	This study introduces a zero-watermarking technique for color images focused on tamper detection and self-recovery. The method integrates secret sharing, data compression, and image interpolation. It relies on authentication signals, which perform both detection and recovery functions, utilizing a (k, n) threshold scheme. The secret shares are randomly hidden within the alpha channel of the original image, which is stored in the Portable Network Graphics (PNG) format. The authors assert that these authentication signals are effective not only for identifying tampered blocks but also for determining the precise colors within a defined palette required to restore the corrupted image block. The experimental validation employed color images with dimensions of 550 × 322 and 512 × 512 pixels.

https://doi.org/10.3390/a19010028

3. Materials and Methods

The primary aim of the proposed semi-fragile watermarking technique is to utilize a thumbnail as a watermark, serving both as a backup for the original content and to illustrate the changes made to the watermarked image in the event of tampering. The localization of any tampered content occurs through two methods: the first involves visual inspection using the recovered thumbnail, which is then validated by the tamper detection mask. The architecture of the proposed scheme is detailed in the subsequent paragraphs. It is fundamentally organized into four sequential stages: watermark generation, embedding, extraction, and content reconstruction.

3.1. Watermark Generation

The primary goal of this stage is to develop a robust criterion for generating thumbnail images that effectively retain the essential luminance and color characteristics of the original image content. This section aims to minimize the payload required by the watermarking algorithm as much as possible. The necessary steps are as follows:

Step 1. The original color image I is read, with high spatial resolution of $M \times N$ pixels in size, and resized to a VGA (640×480) resolution [44]. The resultant image of this step is denoted as I_{VGA}. Resizing to another low resolution reduces the payload, and at the same time, the imperceptibility of the watermarking algorithm is increased. However, applying DnCNN and VSDR severely affects the visual quality of the restored image. On the other hand, resizing to another high-resolution image benefits the visual quality of the restored image after applying DnCNN and VSDR. However, the payload of the watermarking algorithm overflows. The result is a trade-off between the payload and visual quality of the restored image. Based on this behavior and considering the spatial resolution analysis of [47], we decided to adopt the VGA resolution to perform this procedure. Section 4.2.2 illustrate and discuss this behavior.

Step 2. The image I_{VGA} is compressed using the JPEG2000 algorithm in lossy compression mode [45,46]. This algorithm has demonstrated effectiveness recently for watermark payload reduction in data hiding methods, enhancing the protection of color images [48,49]. This approach differs from conventional digital watermarking methods, which typically compress only the image data pixel array. The resulting image from this step is referred to as I_{JP2}. Section 4.2.2 illustrate and discuss the influence of the use of the JPEG2000 algorithm.

Step 3. The file of the image I_{JP2} is converted to a BIN file format and transformed to a binary string Sb.

Step 4. The binary logo Lb is read with a 32×32 pixel size, and the 1024 bits are appended to the beginning of the binary string Sb.

Step 5. For error correction of Sb, we use one of the best-known, non-recursive, and non-systematic convolutional codes over the Galois Field *GF(2)* with coding rate R given by (1):

$$R = \frac{k_0}{n_0} = \frac{1}{3}\left[\frac{bits}{channel\ symbols}\right], \tag{1}$$

where k_0 is the length of a window of data bits entering a cycle of the convolutional encoder clock, n_0 is the corresponding length of the window at the output in that clock cycle, constraint length $K = 5$, free distance $d_{free} = 12$, with sub-generators written in octal notation: 37, 33, 25. The error-correcting capability t is then

$$t = \left\lfloor \frac{d_{free} - 1}{2} \right\rfloor, \tag{2}$$

https://doi.org/10.3390/a19010028

where $\lfloor x \rfloor$ means the largest integer no greater than x. Thus, the code can, with maximum likelihood decoding, correct t errors within 3 to 5 constraint lengths; the exact length depends on the error pattern distribution [38,39]. The encoded watermark data bits are denoted as *We* and are concealed in the frequency domain of the luminance of the color image I via the embedding stage, explained in the following section. Figure 1 shows the procedure mentioned in the above paragraphs.

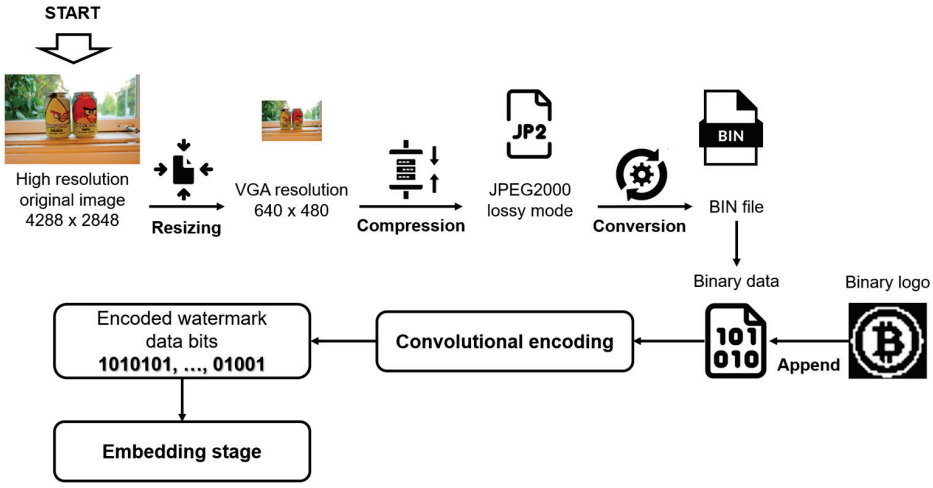


Figure 1. Watermark generation procedure.

3.2. Watermark Embedding

Designing an embedding procedure that maximizes payload preservation while maintaining imperceptibility and ensuring robustness against signal processing distortions presents a challenge in the semi-fragile watermarking research field. The proposed embedding procedure allows for the concealment of the two watermark data bits within 3×3 blocks by utilizing quantization index modulation with dither modulation (QIM-DM) [36,37] in the 2D-DCT frequency domain. Figure 2 illustrates the embedding procedure, which is detailed as follows:

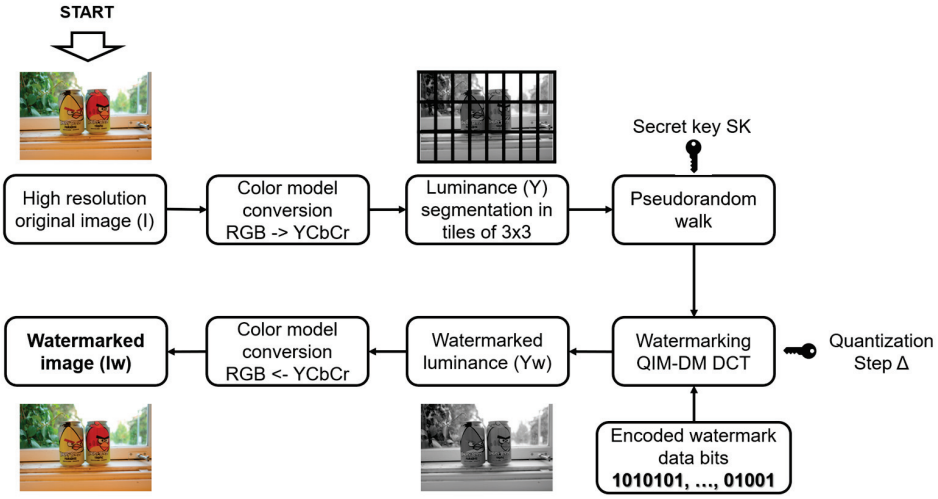


Figure 2. Watermark embedding procedure.

Step 1. Read the original color image I with $M \times N$ pixels in size and convert its original RGB to the YCbCr color model. Isolate the luminance information denoted as Y.

Step 2. Segment the luminance Y in blocks of size 3×3 pixels.

https://doi.org/10.3390/a19010028

Step 3. To increase the security of the algorithm, supported by a secret key *SK*, it generates a pseudorandom walk through all b_i blocks. The total number of A blocks required to embed 2 bits of the codified signal *We* within the color image I is given by (3):

$$A = length(We)/2 \tag{3}$$

Step 4. Segment *We* in sequences Sq_i, where $i = 1, \ldots A$, and each one contains 2 data bits. The bidimensional DCT transform is used to process each b_i block. Subsequently, each Sq_i is embedded into the two alternating current (AC) coefficients of low frequency for each b_i in the 2D-DCT domain, utilizing the QIM-DM technique and applying the decision rule (4):

$$\begin{array}{l} if\ Sq_i(z) = 0\ then \rightarrow\ c'_z = \left\lfloor \frac{(c_z + dt(z,0))}{\Delta} \right\rfloor \cdot \Delta - dt(z,0) \\ else \rightarrow\ c'_z = \left\lfloor \frac{(c_z + dt(z,1))}{\Delta} \right\rfloor \cdot \Delta - dt(z,1) \end{array} \tag{4}$$

where c is the original and c' is the watermarked coefficient in the b_i block, $i = 1, \ldots A$, the term z = {1,2}, and Δ denotes the quantization step. The variables $dt(z,0)$ and $dt(z,1)$ denote the dither signals, which are defined by (5) and (6), respectively:

$$dt(z,0) = -\Delta + (\Delta \cdot ps)\ , \tag{5}$$

$$dt(z,1) = \begin{cases} dt(z,0) + \frac{\Delta}{2}, \ \rightarrow\ if\ dt(z,0) < 0 \\ dt(z,0) - \frac{\Delta}{2}, \ \rightarrow\ otherwise \end{cases}, \tag{6}$$

where *ps* denotes a pseudo-random signal with a uniform distribution generated using a secret key k_1, the length of *ps* corresponds to that of z. Once *We* is embedded within the DCT coefficients of every b_i, the watermarked coefficients are transformed back to the spatial domain using the inverse two-dimensional IDCT, resulting in the watermarked luminance Y_w.

Step 5. Using the watermarked luminance Y_w and the original chrominances, convert from the YCbCr to the RGB color model to obtain the watermarked color image I_w.

3.3. Watermark Extraction

Once the watermarked color image I_w is obtained, extracting the hidden information from its pixels is compulsory to perform ownership authentication, tamper detection, and self-recovery of content. Figure 3 shows the extraction procedure.

This approach outlines the necessary steps for completing this task:

Step 1. Given the watermarked color image I_w, convert its original RGB to the YCbCr color model. Isolate the watermarked luminance information denoted as Y_w.

Step 2. Segment Y_w in blocks of size 3×3 pixels.

Step 3. The b_i blocks, each sized 3×3 pixels, are retrieved from I_w utilizing the secret key *SK*. The encoded data is retrieved from all b_i blocks in the 2D-DCT domain via the QIM-DM extraction method specified by (7):

$$D1_z = \left(c'_z - q1_z\right)^2,\ D2_z = \left(c'_z - q2_z\right)^2\ , \tag{7}$$

where *D1* and *D2* represent a pair of distances, $z = 1, 2$, c' denotes the watermarked DCT coefficients, and *q1* and *q2* refer to a pair of quantizers as defined in (8):

$$q1_z = \left\lfloor \frac{(c'_z + dt(z,0))}{\Delta} \right\rfloor \cdot \Delta - dt(z,0)\ ,\ q2_z = \left\lfloor \frac{(c'_z + dt(z,1))}{\Delta} \right\rfloor \cdot \Delta - dt(z,1), \tag{8}$$

https://doi.org/10.3390/a19010028

where Δ, *dt*(*z*,0), *dt*(*z*,1), and the secret key k_1 have the same values used in the embedding stage. Upon acquiring *D1* and *D2*, the extraction of an encoded watermark data bit is conducted using the decision rule specified in (9):

$$if\ D1_z\ <\ \ D2_z\ then \rightarrow\ We' = 0\ ,\ else \rightarrow\ We' = 1 \tag{9}$$

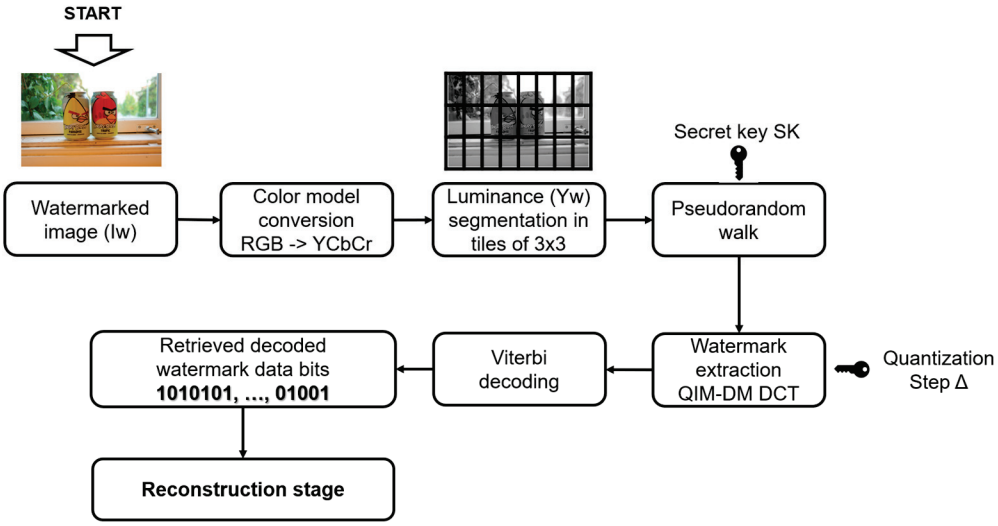


Figure 3. Watermark extraction procedure.

Iteratively apply this step for each b_i in the bidimensional DCT domain, where $i = 1, \ldots A$, and retrieve the encoded watermark *We*′.

Step 4. Perform the decoding of *We*′ using the Viterbi algorithm with a hard decision approach. We employ the Viterbi decoding algorithm [38,39] on the trellis diagram associated with the convolutional code, which has 2^{K-1} states. This approach involves utilizing the Hamming distance between the received sequence and the retained paths. It is well known that for a binary symmetric channel with crossover probability $p < 0.5$, selecting the path with the highest log-likelihood function is equivalent to choosing the path that is closest in Hamming distance to the received sequence [38,39].

Once the recovered watermark data bits are decoded *Wd*, reconstruction of the binary logo and thumbnail image is performed as described in the following procedure.

3.4. Reconstruction of Watermark's Information: Ownership Authentication, Tamper Detection, and Self-Recovery

The reconstruction procedure of the watermark's information is shown in Figure 4 and explained in the following paragraphs.

Step 1. Once the recovered watermark data bits are decoded *Wd* via the procedure described in Section 3.3, reconstruction of the binary logo is performed to prove the ownership authentication. To recover the binary logo *Lb*′, extract the first $L = 1024$ bits from *Wd* and reshape it to 32×32 pixels in size. To measure the robustness, the Bit Correct Rate (BCR) between the original *Lb* and retrieved *Lb*′ is computed employing (10):

$$BCR = \frac{L - \sum_{r=1}^{L} Lb_r \oplus Lb'_r}{L} \tag{10}$$

Step 2. Retrieve the data bits from the location $L + 1$ to the length of *Wd*, and format to obtain a BIN file.

https://doi.org/10.3390/a19010028

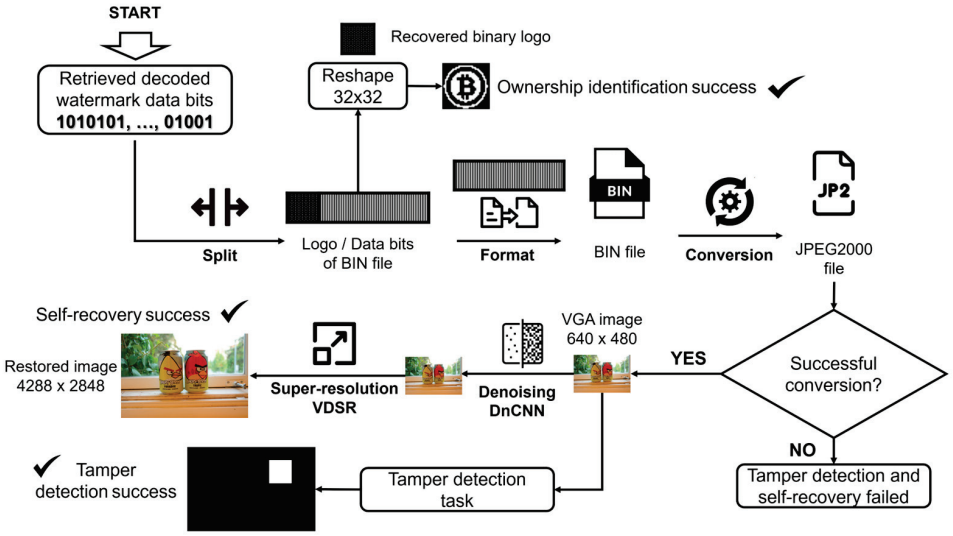


Figure 4. Reconstruction of the watermark's information procedure.

Step 3. Convert the BIN file to JPEG2000 image format [45,46]. The resulting image file of this step is denoted as $I'_{JP2\text{-}VGA}$.

Step 4. Since tamper detection and self-recovery are directly dependent on the image file $I'_{JP2\text{-}VGA}$, if this can be opened and read, then the procedure can be continued; otherwise, the procedure is ended, and tamper detection and self-recovery fail.

Step 5. Once the image file $I'_{JP2\text{-}VGA}$ is reconstructed, opened, and read, it has a VGA resolution (640×480), which is used for the tamper detection task. To perform this, we compute the difference between the luminance of the possible tampered image, rescaled to a VGA resolution denoted as $Y_{WT\text{-}VGA}$, and the luminance $Y'_{JP2\text{-}VGA}$ extracted from the YCbCr color model of $I'_{JP2\text{-}VGA}$, as follows (11):

$$Td = Y'_{JP2-VGA} - Y_{WT-VGA}\ , \tag{11}$$

where the difference Td is binarized by thresholding and rescaled using nearest neighbor interpolation to a high spatial resolution of $M \times N$ pixels in size to get the tampered mask.

Step 6. Once the tamper detection has been performed, the final process is restoring the image file $I'_{JP2\text{-}VGA}$ to as high a visual fidelity as possible to the original spatial resolution of $M \times N$ pixels in size to view the lost content from the tampered watermarked version with the naked eye. To accomplish this aim, we are supported by two deep learning pretrained models: a denoising convolutional neural network (DnCNN) [40,41] and a very-deep super-resolution neural network (VDSR) [42,43]. Since both models are well-known and established, details of the architecture can refer to [40–43]. In this way, to diminish the effect of rippling and blocking artifacts caused by the JPEG2000 lossy compression, the model DnCNN is applied to $I'_{JP2\text{-}VGA}$. Finally, the denoised version is rescaled to $M \times N$ via the model VDSR, obtaining the restored image I'_r.

4. Experimental Results

This section presents the evaluation results of the proposed semi-fragile watermarking algorithm for high-resolution color images.

4.1. Experimental Setup

We consider a set of 1735 high resolution color images extracted from the RAISE dataset [50] in TIFF format with 4288×2848 pixels in size and 24 bits/pixel of color bit

https://doi.org/10.3390/a19010028

depth, which includes content of "outdoor," "indoor," "landscape," "nature," "people," "objects," and "buildings." Some test images are shown in Figure 5. All experiments were conducted on a personal computer equipped with the Microsoft Windows 11 © operating system, an Intel © Core i7 processor, 16 GB RAM, and MATLAB © R2025a (25.1), Windows 64 bits.



Figure 5. Sample test images used in the experimental results.

4.2. Parameter Settings

4.2.1. Setting the Size of DCT-Blocks and Selecting AC Coefficients

Conventionally, digital image watermarking using the QIM-DM [36,37] technique employs block segmentation of 8 × 8 pixels, aligning with the JPEG image compression standard. This method typically embeds four watermark data bits into the AC coefficients corresponding to low to middle frequencies in the DCT domain, achieving a balance between imperceptibility and robustness. To meet the objectives of this study, however, it is essential to maximize capacity without compromising other watermarking requirements, which necessitates reducing the block segmentation size. In this context, considering a high-resolution color image with dimensions of 4288 × 2848 pixels, Table 2 presents the total capacity offered when a) the block segmentation is 8 × 8 pixels, embedding four watermark data bits per DCT block, and b) the block segmentation is 3 × 3 pixels, embedding two watermark data bits per DCT block. The total amount of encoded watermark data bits, denoted as *We*, is derived from the procedure outlined in Section 3.1, Watermark Generation, incorporating the following features: resizing the image from 4288 × 2848 to 640 × 480, applying JPEG2000 lossy compression with a ratio of 1:10, and employing convolutional encoding as described in Section 3.1.

Table 2. Setting of size of DCT blocks.

Average Total Amount of Encoded Watermark Data Bits *We* in All Images into Dataset	**Total Capacity Block Segmentation 8 × 8 4 Bits Concealed per Block**	**Total Capacity Block Segmentation 3 × 3 2 Bits Concealed per Block**
2,198,128 bits	$\lfloor (4288/8) \times (2848/8) \times 4 \rfloor =$ 763,264 bits	$\lfloor (4288/3) \times (2848/3) \times 2 \rfloor =$ 2,713,827 bits

As shown in Table 2, various combinations exist to adjust capacity by modifying block segmentation and the number of bits concealed per block. The choice of the appropriate pair of values depends on the objectives of the watermarking algorithm's design. To maintain a balance between imperceptibility and robustness, we opt for 3 × 3 block segmentation and conceal two bits per block. Consequently, we enhance the capacity of the proposed method by using a 3 × 3 segmentation and embedding two bits, instead of the conventional settings of 8 × 8 and four bits [36,37].

Once the values for block segmentation and the amount of data to be embedded in the DCT domain are established, the next step is to select two AC coefficients from the 3 ×

https://doi.org/10.3390/a19010028

3 DCT block for the embedding procedure. To accomplish this, we utilize a zig-zag scan, as illustrated in Figure 6, which represents a truncated version of the traditional zig-zag scan.

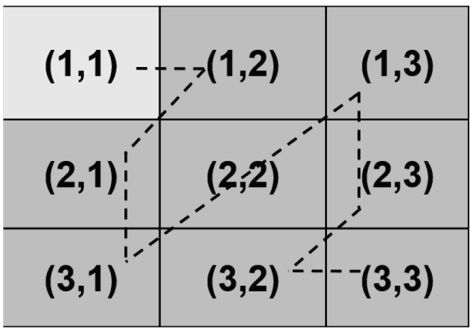


Figure 6. Zig-zag scanning utilized in the DCT domain.

Experimentally, we applied JPEG lossy compression with a quality factor of 70 and utilized 3 × 3 block segmentation in the DCT domain. We then compressed all images in the dataset and calculated the average percentage of non-zero direct current (DC) and AC coefficients, as shown in Figure 7.

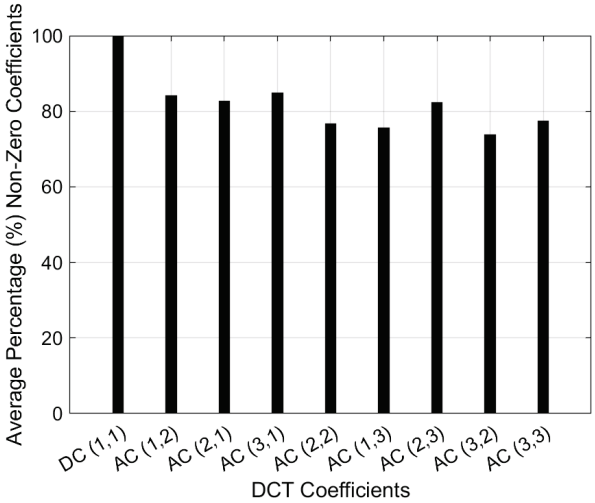


Figure 7. Average percentage of non-zero coefficients obtained for all images in the dataset.

The analysis in Figure 7 indicates that the optimal candidates for embedding watermark data bits, excluding the DC coefficient, are the coefficients with the highest average percentages. AC (1, 2) and AC (3, 1) have values of 84.11% and 84.92%, respectively. While these AC coefficients fall within the low-frequency band, alterations to lower frequencies can lead to visual distortions. However, by appropriately adjusting the quantization step (Δ) during the embedding process, it is possible to minimize distortions in the image content while simultaneously enhancing robustness against various signal processing distortions.

4.2.2. Setting of Image Resolution and JPEG2000 Compression Ratio to Thumbnail Image Design

To determine the optimal image resolution and JPEG2000 compression ratio for generating the thumbnail image, we conduct the following experimental tests. First, we consider the procedure outlined in Section 3.1, Watermark Generation, along with image resolutions of HD (1280 × 720), VGA (640 × 480), and CIF (352 × 288). We also examine a JPEG2000

https://doi.org/10.3390/a19010028

compression ratio of 1:10, the self-reconstruction procedure described in Section 3.4, and the Weighted Peak Signal to Noise Ratio (WPSNR) [51], as given in (12):

$$WPSNR(dB) = 10log_{10}\left(\frac{Max\ Pixel\ Value^2}{NVF \times \left(\frac{1}{M \cdot N}\left(\sum_{x=1}^{M}\sum_{y=1}^{N}(I(x,y) - I_w(x,y))^2\right)\right)}\right), \quad (12)$$

where NVF is the Noise Visibility Function given by (13):

$$NVF(x,y) = \frac{1}{1+\sigma_x^2(x,y)} \quad (13)$$

$\sigma_x{}^2(x,y)$ is the local variance in a window of width wh [51]. Table 3 shows the results of the first experiment to select the proper image resolution to build the thumbnail image.

Table 3. Setting of optimal image resolution for thumbnail image design.

	Average Total Amount of Encoded Watermark *We* in All Images into Dataset			**Average Visual Quality After Reconstruction Procedure WPSNR (dB)**		
Total Capacity of Proposed Method	**HD (1280 × 720)**	**VGA (640 × 480)**	**CIF (352 × 288)**	**HD (1280 × 720)**	**VGA (640 × 480)**	**CIF (352 × 288)**
$\lfloor(4288/3) \times (2848/3) \times 2\rfloor$= 2,713,827 bits	6,610,063 Bits	2,198,128 bits	720,077 bits	39.82 dB	34.56 dB	31.60 dB

The analysis presented in Table 3 indicates that HD resolution yields a high WPSR value following the reconstruction procedure outlined in Section 4.4. However, the total amount of encoded watermark data bits (*We*) exceeds the total capacity of the proposed method. Conversely, CIF resolution demonstrates a contrasting behavior compared to HD resolution, resulting in a lower amount of *We* data bits but a decrease in visual quality as measured by WPSNR. The experimental results show that VGA resolution strikes an appropriate balance between the number of *We* data bits and visual quality in terms of WPSNR after the reconstruction procedure described in Section 3.4. Consequently, based on these findings, we have selected VGA resolution for our semi-fragile watermarking design.

The second experiment examines the procedure outlined in Section 3.1, Watermark generation. It uses a fixed image resolution of VGA and evaluates JPEG2000 compression ratios of 1:5, 1:10, and 1:15. Additionally, it incorporates the self-reconstruction procedure described in Section 3.4 and utilizes the WPSNR metric. The results of this experiment are presented in Table 4.

Table 4. Setting of optimal JPEG2000 compression ratio for thumbnail image design.

	Average Total Amount of Encoded Watermark *We* in All Images into Dataset			**Average Visual Quality After Reconstruction Procedure WPSNR (dB)**		
Total Capacity of Proposed Method	**JPEG2000 Compression Ratio 1:5**	**JPEG2000 Compression Ratio 1:10**	**JPEG2000 Compression Ratio 1:15**	**JPEG2000 Compression Ratio 1:5**	**JPEG2000 Compression Ratio 1:10**	**JPEG2000 Compression Ratio 1:15**
$\lfloor(4288/3) \times (2848/3) \times 2\rfloor$ = 2,713,827 bits	4,293,292 bits	2,198,128 bits	1,464,463 bits	34.65 dB	34.56 dB	34.24 dB

From Table 4, we observe that the three JPEG2000 compression ratios of 1:5, 1:10, and 1:15 provide advantages by reducing the total amount of encoded data bits (*We*), while maintaining WPSNR values close to 34 dB. However, selecting a compression ratio of 1:5 is not optimal, as the total amount of encoded watermark data bits (*We*) exceeds the capacity of the proposed method. In contrast, the ratios of 1:10 and 1:15 yield an appropriate total

https://doi.org/10.3390/a19010028

amount of *We*, with a minor WPSNR difference of 0.3178 dB between them. Although this WPSNR difference may seem negligible, our experimental results indicate that as the JPEG2000 compression ratio increases, the resulting images show perceptual distortions in the edges and flat regions due to a blurring effect. Therefore, we have chosen the JPEG2000 compression ratio of 1:10 for the proposed method to achieve the highest possible visual quality in the reconstructed image derived from the thumbnail version.

4.2.3. Set of Quantization Step Δ

To determine the appropriate quantization step Δ used in QIM-DM [36,37] within the 2D-DCT frequency domain, we examine the parameter settings detailed in Sections 4.2.1 and 4.2.2. We will also consider a variable range for Δ, specifically from 40 to 80.

This test evaluates how the quantization step Δ affects the imperceptibility requirement by analyzing the watermarked color image I_w. It computes the visual distortion using the WPSNR metric specified in (13) and assesses visual quality through the Feature Similarity Index Method (FSIM) [52], which ranges from 0 to 1, where an FSIM value of 1 signifies that the original and reference images are identical, and values closer to 0 indicate poor visual quality. Additionally, it measures color differences using the Normalized Color Difference (NCD) [53], where an NCD value of 0 denotes that the original and reference images are the same, and values further from 0 indicate significant color differences. For brevity, readers can refer to [52,53] for the mathematical definitions of FSIM and NCD, respectively. Figure 8 illustrates the experimental results regarding WPSNR, FSIM, and NCD for quantization step Δ values of 40, 60, and 80 across each trial. Table 5 presents the results of these experiments, detailing the median value, quartiles, whiskers, and the number of outliers, which facilitates a comprehensive analysis of the results shown in Figure 8.

Table 5. Median values, quartiles, whiskers, and the number of outliers for WPSNR, FSIM, and NCD metrics to a variable quantization step Δ, obtained in all images in the dataset.

	WPSNR (dB)				**FSIM**				**NCD**			
Step Δ	**Median**	**Quartiles**	**Whiskers**	**Outliers**	**Median**	**Quartiles**	**Whiskers**	**Outliers**	**Median**	**Quartiles**	**Whiskers**	**Outliers**
40	46.94	46.79 47.09	46.51 47.55	27	0.9997	0.9996 0.9998	0.9994 0.9999	72	0.02808	0.02411 0.03365	0.01287 0.04794	85
60	43.64	43.46 43.80	43.10 44.30	23	0.9993	0.9992 0.9995	0.9987 0.9998	66	0.03960	0.03405 0.04734	0.01693 0.06720	81
80	41.26	41.09 41.43	40.58 41.94	17	0.9988	0.9985 0.9990	0.9976 0.9997	63	0.05114	0.04413 0.06071	0.02092 0.08543	83

In Table 5 and Figure 8, we observe that visual distortion, visual quality, and color changes decrease as the quantization step Δ increases. A higher value of Δ enhances the robustness of the encoded watermark; however, we find that imperceptibility is compromised. This illustrates a trade-off between robustness and imperceptibility. To determine the appropriate quantization step Δ, we conducted an additional experiment measuring robustness against a rigorous attack, specifically JPEG lossy compression at a quality factor of 75 combined with random cropping of 600 × 600 pixels. This attack was applied to all watermarked color images in the dataset. The results are presented in Figure 9, which considers quantization step Δ values of 40, 60, and 80 for each trial.

https://doi.org/10.3390/a19010028

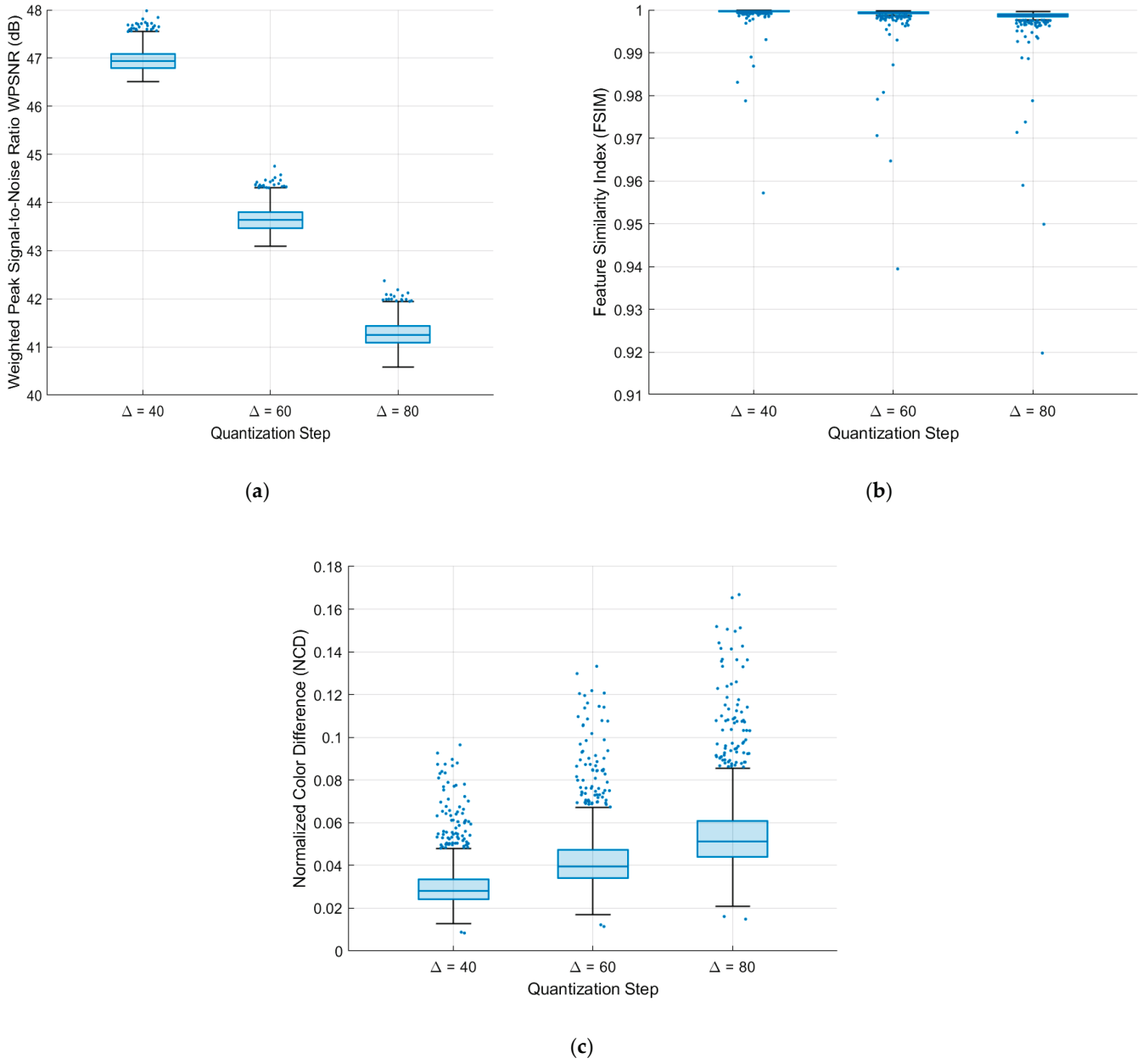


Figure 8. (**a**) WPSNR, (**b**) FSIM, and (**c**) NCD values for quantization steps Δ 40, 60, and 80, obtained in all images in the dataset.

In Figure 9, we show that a high value of Δ increases the robustness of the encoded watermark (*We*); however, as shown in Table 5 and Figure 8, the imperceptibility is diminished. As stated above, the result is a trade-off between the robustness and imperceptibility requirements. From Figure 9, for Δ = 40, the proposed scheme can correctly recover the thumbnail image of 1328 color images, while 407 thumbnails cannot be recovered. For Δ = 60, 1605 thumbnails were recovered correctly, and 130 could not be recovered. Finally, for Δ = 80, 1715 thumbnails were recovered correctly, and 20 could not be recovered. After this analysis, we determine that Δ = 60 is a suitable value for the quantization step of the proposed watermarking algorithm, preserving as much as possible a proper balance between imperceptibility and robustness.

https://doi.org/10.3390/a19010028

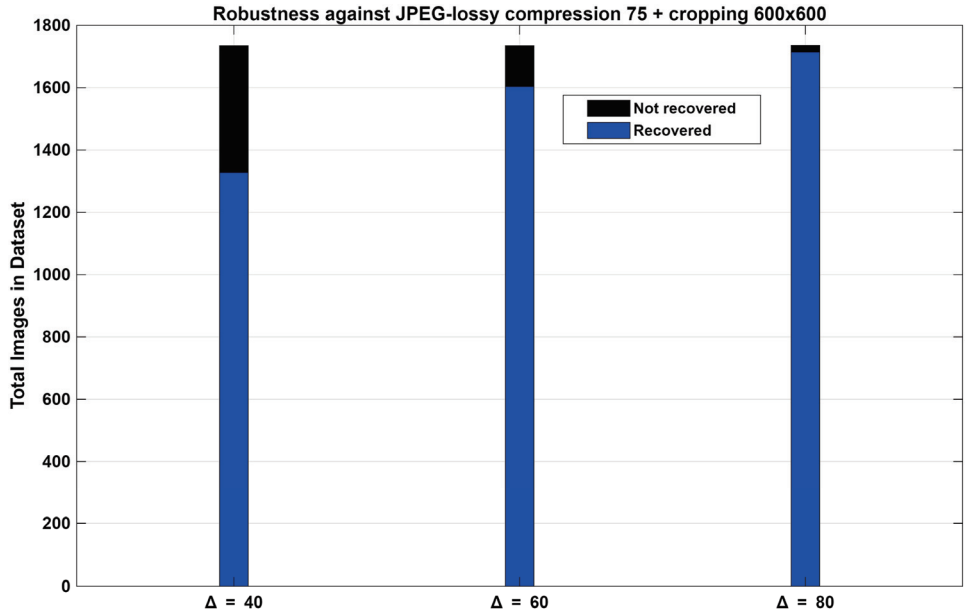


Figure 9. Robustness results of thumbnail recovery capability against aggressive combined distortion.

Table 6 shows the final values of all parameters that compose the proposed semi-fragile watermarking algorithm for high-resolution color images.

Table 6. Final values of all parameters of the proposed method.

Parameter	Value
Binary logo	32 × 32 pixels in size
Spatial resolution of thumbnail	VGA (640 × 480)
JPEG2000 compression ratio of thumbnail	Lossy compression, ratio 1:10
DCT block-segmentation to embedding procedure	3 × 3
DCT coefficients	AC (1, 2), AC (3, 1)
Image dataset	1735 high resolution color images in TIFF format with 4288 × 2848 pixels
Quantization step Δ	60

4.3. Imperceptibility Analysis

To evaluate the imperceptibility of the watermark provided by the proposed method, Figures 10 and 11 present two test images: the first featuring weak textures and the second showcasing strong textures. These images are used to assess visual quality alongside their respective results in terms of WPSNR, FSIM, and NCD. Typically, electronic devices like laptops display high-resolution image previews based on a predetermined percentage, enabling users to zoom in on specific areas using available software options. Accordingly, Figures 10 and 11 also include a 300% zoom of a small region from both the original and watermarked images, facilitating an evaluation of the degree of visual distortion caused by the proposed method.

As demonstrated in Figure 10a,c and Figure 11a,c, the watermark remains imperceptible under predefined viewing conditions on electronic devices such as laptops or smartphones, as confirmed by the WPSNR, FSIM, and NCD results. Conversely, Figures 10 and 11d illustrate the visual degradation that occurs when the image is zoomed in to 300% or more. This test is intended for illustrative purposes only, given that it is uncommon to zoom in beyond 100% on digital images displayed on devices like laptops, tablets, or smartphones.

https://doi.org/10.3390/a19010028

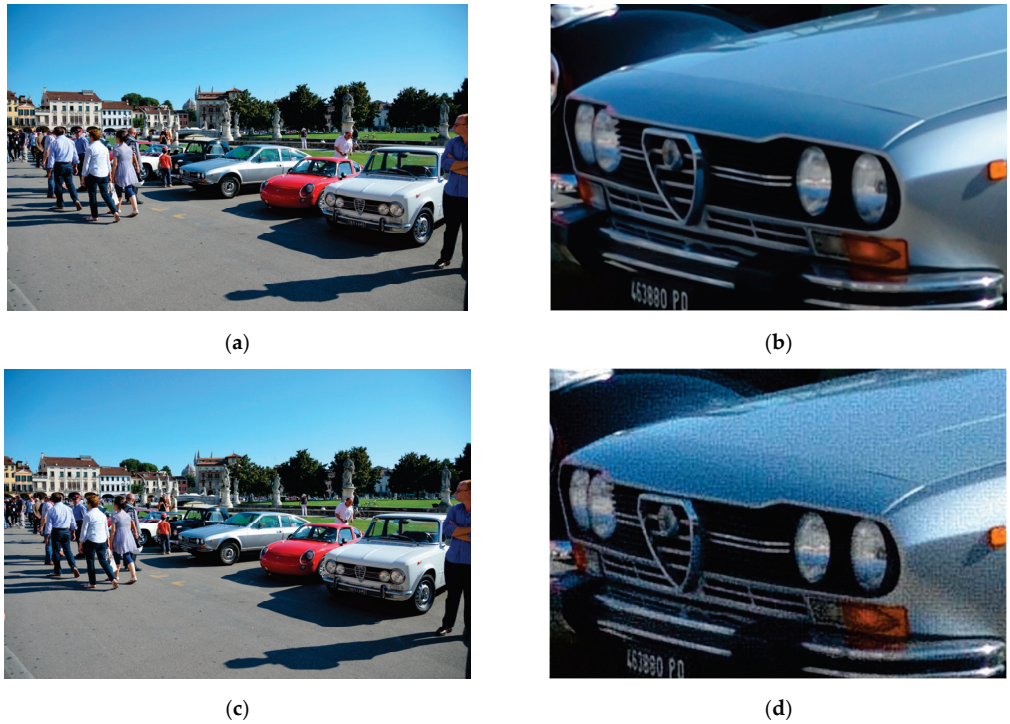


Figure 10. (**a**) Original image with weak textures. (**b**) Zoom at 300% of a small portion of the content of (**a**). (**c**) Watermarked image. (**d**) Zoom at 300% of a small portion of the content of (**c**). WPSNR, FSIM, and NCD results are in the last column.

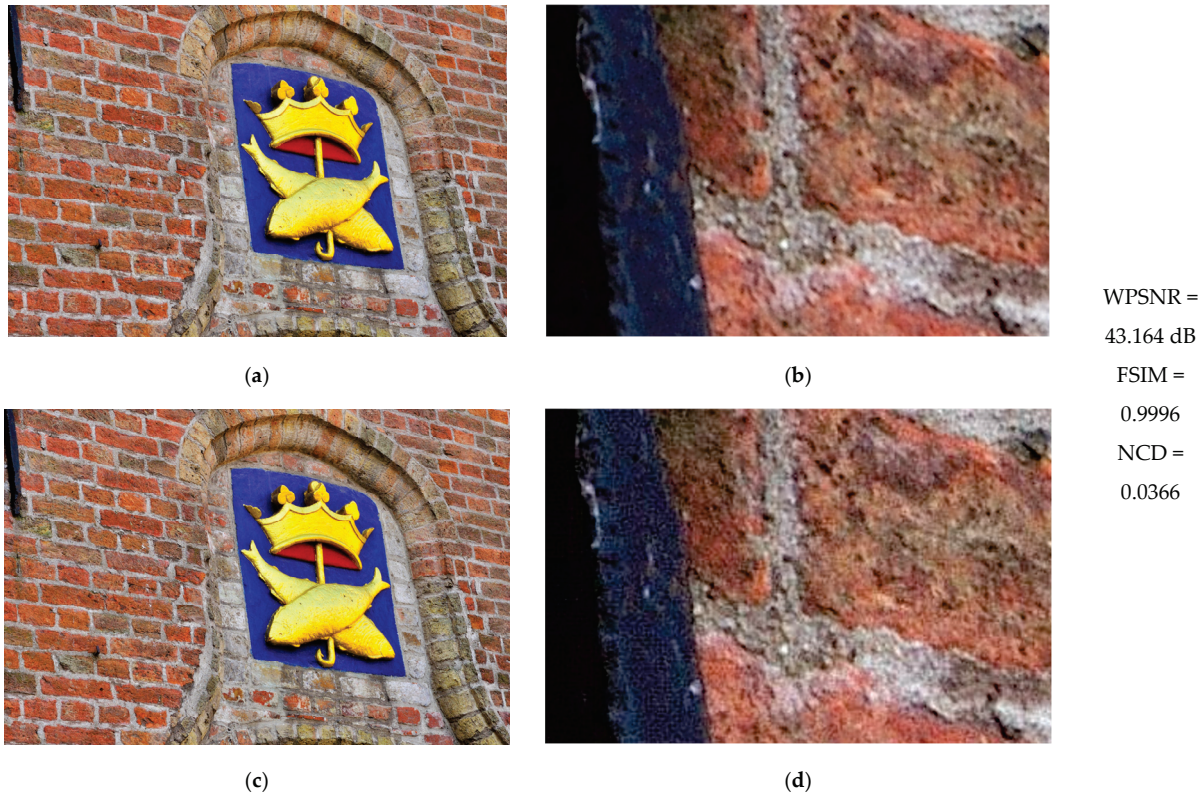


Figure 11. (**a**) Original image with strong textures. (**b**) Zoom at 300% of a small portion of the content of (**a**). (**c**) Watermarked image. (**d**) Zoom at 300% of a small portion of the content of (**c**). WPSNR, FSIM, and NCD results are in the last column.

https://doi.org/10.3390/a19010028

4.4. Robustness Analysis

To evaluate the robustness of the proposed method, we divide this test into two categories. The first category evaluates the robustness of the proposed semi-fragile watermarking algorithm in terms of its ability to recover the image thumbnail when the watermarked image is distorted by several image processing operations. The second one considers the capability to recover the binary logo after the watermarked image is attacked by a set of signal processing operations. Thus, Figure 12 illustrates the performance of the proposed algorithm regarding its robustness and its ability to recover the image thumbnail. In this sense, for non-watermarked images, no thumbnail has been recovered, confirming the non-ambiguity of the proposed method. Table 7 complements the results presented in Figure 12 for the rest of the distortions, with detailed purposes.

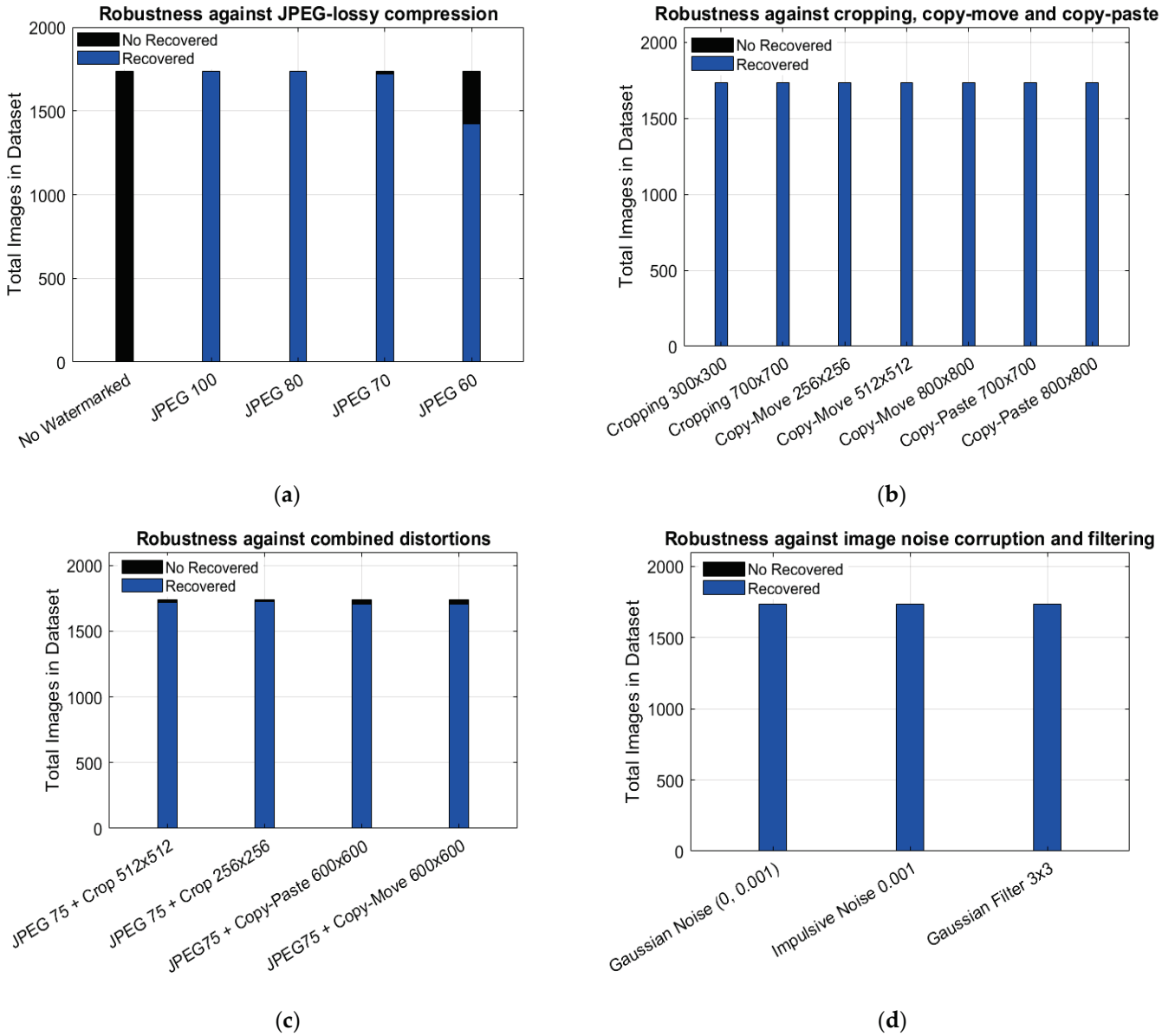


Figure 12. Robustness against: (**a**) Unwatermarked and JPEG lossy compression, (**b**) Cropping, copy–move and copy–paste, (**c**) combined distortions and, (**d**) image noise corruption and filtering.

https://doi.org/10.3390/a19010028

Table 7. Number of thumbnail images recovered and not recovered when watermarked images are distorted by several image processing operations.

Distortion	Thumbnail Recovered	Thumbnail Not Recovered	Distortion	Thumbnail Recovered	Thumbnail Not Recovered
JPEG 100	1735	0	Copy–paste 700 × 700	1735	0
JPEG 80	1735	0	Copy–paste 800 × 800	1733	2
JPEG 70	1719	16	JPEG75 + crop 512 × 512	1715	20
JPEG 60	1424	311	JPEG75 + crop 256 × 256	1726	9
Crop 300 × 300	1735	0	JPEG75 + copy–paste 600 × 600	1705	30
Crop 700 × 700	1735	0	JPEG75 + copy–move 600 × 600	1703	32
Copy–move 256 × 256	1735	0	Gaussian noise (0, 0.001)	1735	0
Copy–move 512 × 512	1735	0	Impulsive noise 0.001	1735	0
Copy–move 800 × 800	1734	1	Gaussian filter 3 × 3	1735	0

The proposal, which is a semi-fragile watermarking scheme, is illustrated in Figure 12 and Table 7. It demonstrates that thumbnail images can be recovered in nearly all attack scenarios; however, performance declines when the watermarked image undergoes JPEG 60 compression. These experimental results indicate that increasing the tolerance for each attack reduces the robustness of the proposed fragile watermarking algorithm. Additionally, it is important to note that the encoded watermark data bits, referred to as *We*, contain information from the entire JPEG2000 file that includes the thumbnail image, which makes recovery sensitive to any alterations.

On the other hand, Table 8 demonstrates the robustness of the proposed method, specifically in terms of the BCR defined previously in Equation (11). This metric evaluates the method's ability to recover the binary logo *Lb*, which has a pixel size of 32 × 32. The average BCR for images that are not watermarked is 0.4994. In the other attack scenarios, *Lb* remains visible to the unaided eye and is nearly fully recoverable.

Table 8. BCR of the recovered binary logo *Lb* when watermarked images are distorted by several image processing operations.

Distortion	BCR	Distortion	BCR
JPEG 100	1	Copy–paste 700 × 700	1
JPEG 80	1	Copy–paste 800 × 800	0.9995
JPEG 70	0.9998	JPEG75 + crop 512 × 512	0.9999
JPEG 60	0.9993	JPEG75 + crop 256 × 256	0.9999
Crop 500 × 500	1	JPEG75 + copy–paste 600 × 600	0.9998
Crop 800 × 800	1	JPEG75 + copy–move 600 × 600	0.9998
Copy–move 256 × 256	0.9999	Gaussian noise (0, 0.001)	1
Copy–move 512 × 512	0.9999	Impulsive noise 0.001	1
Copy–move 1000 × 1000	0.9996	Gaussian filter 3 × 3	1

4.5. Self-Recovery and Tamper Detection Analysis

The goal of this experiment is to visualize the performance of the proposed semi-fragile watermarking technique concerning self-recovery and tamper detection. For brevity, Figures 13–15 present the experimental results when the watermarked image undergoes manipulation through multiple copy–moves, extensive copy–pasting, and various combinations of both techniques. It is important to note that the color images used have a resolution of 4288 × 2848 pixels and a color bit depth of 24 bits per pixel. Based on the experimental findings, we would like to emphasize several key aspects of our proposal: (a) The proposed method extends beyond simply generating a tamper detection mask. (b) It possesses self-recovery capabilities that include color information, maintaining the original

https://doi.org/10.3390/a19010028

spatial and bit depth resolutions. (c) The visual quality of the restored color image is preserved to the greatest extent possible, as indicated by the WPSNR, FSIM, and NCD metrics. These metrics demonstrate the advantages of employing a denoising convolutional neural network (DnCNN) and a very deep super-resolution neural network (VDSR). Although Figures 13–15 indicate that the tamper detection mask exhibits low precision, the primary aim of this work is to identify tampered regions, even with low accuracy. The mask serves primarily for illustrative purposes and as an additional feature. The restored color image allows the naked eye to discern the forgery content, which is a notable highlight of the proposed method.



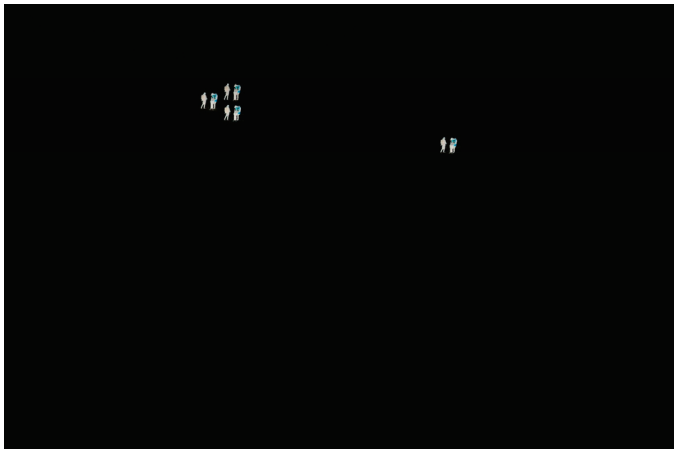
Watermarked image
WPSNR = 43.8686 dB, FSIM = 0.9985, NCD = 0.0244



Tampered version
Attack type: Multiple copy–move



Restored image from thumbnail applying a denoising convolutional neural network (DnCNN) and a very-deep super-resolution neural network (VDSR)
WPSNR = 41.4668 dB, FSIM = 0.9894, NCD = 0.0166



Tamper detection mask

Figure 13. Self-recovery and tamper detection performance against multiple copy–move.

https://doi.org/10.3390/a19010028



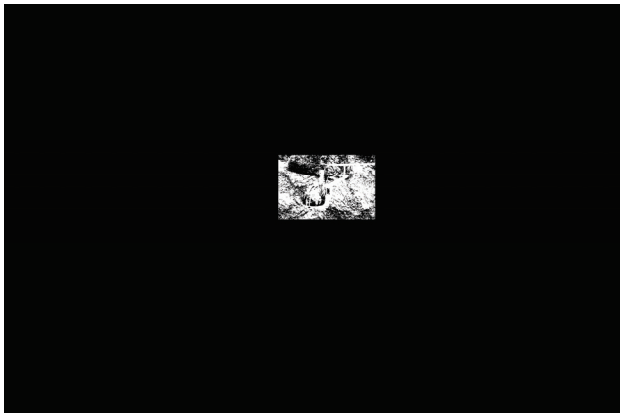
Watermarked image
WPSNR = 43.1375 dB, FSIM = 0.9996, NCD = 0.0422



Tampered version
Attack type: Copy–paste



Restored image from thumbnail applying a denoising convolutional neural network (DnCNN) and a very-deep super-resolution neural network (VDSR)
WPSNR = 31.2798 dB, FSIM = 0.9846, NCD = 0.1093



Tamper detection mask

Figure 14. Self-recovery and tamper detection performance against multiple copy–paste.



Watermarked image
WPSNR = 43.7993 dB, FSIM = 0.9991, NCD = 0.0298



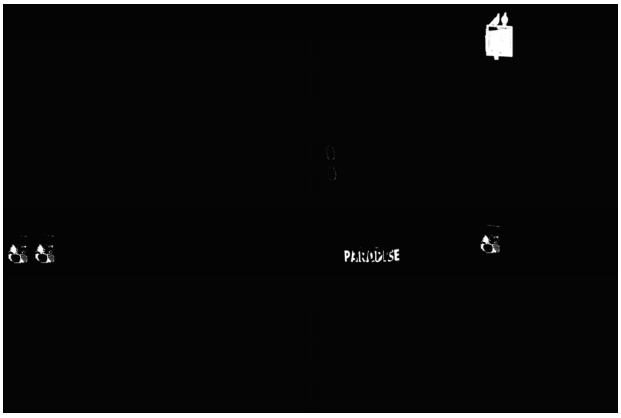
Tampered version
Attack type: Multiple combinations of copy–move and copy–paste

Figure 15. *Cont.*

https://doi.org/10.3390/a19010028



Restored image from thumbnail applying a denoising convolutional neural network (DnCNN) and a very-deep super-resolution neural network (VDSR)
WPSNR = 40.6878 dB, FSIM = 0.9956, NCD = 0.0283



Tamper detection mask

Figure 15. Self-recovery and tamper detection performance against multiple combinations of copy–move and copy–paste.

4.6. Security Analysis

The proposed semi-fragile watermarking algorithm consists of several components necessary for extracting all data bits required to recover and restore the thumbnail image. Key elements include the frequency domain used to embed the watermark, the size of the DCT blocks, the modulated AC coefficients, and the configuration of the convolutional encoder. The secret key (*SK*) enhances the security of the method by facilitating a pseudorandom walk through all b_i blocks, where $i = 1, \ldots A$. Thus, the total number of A blocks needed to embed two bits of the codified signal *We* within the color image I is indicated by Equation (3), as described in Section 3.2. The size of A varies depending on the image. For instance, if the minimum value of A in the dataset is approximately 995,000 and the maximum is approximately 1,103,000, there exists a difference of 108,000 options for the size of A, assuming the total number of DCT blocks for a 4288 × 2848 image resolution is $\lceil (4288/3) \cdot (2848/3) \rceil = 1{,}356{,}121$. Given that the seed *SK* for generating each pseudorandom walk sequence falls within the range of $[0, 2^{32} - 1]$, [54], there are numerous possible combinations ($C_{2^{32}-1}^{108,000}$) to estimate the correct sequence to the pseudorandom walk. Additionally, it is important to note that even if an adversary estimates *SK*, all the aforementioned configuration parameters must be known and properly configured to successfully recover and reconstruct the thumbnail image.

4.7. Performance Comparison

Table 9 presents a performance comparison that includes both the previously reported works in the state-of-the-art from [17–19,21,22,25,27] as well as our proposed method. This compendium was considered because it is the most relevant to our proposed semi-fragile watermarking algorithm. Then, the development of a homologated comparative analysis may facilitate a fair and equitable comparison. Thus, the comparison is constructed solely based on common criteria evaluated in a semi-fragile manner. When a criterion is not applicable, the designation "N/A" is assigned. Subsequently, we conduct a precise comparison based on Table 9. Obtaining an equitable comparison is difficult since, to the best of our knowledge, and according to an exhaustive literature review, we found that few methods of semi-fragile watermarking applied to color imaging; the majority focus their efforts on grayscale images with small spatial resolution. If there are several approaches to forgery and tamper detection

https://doi.org/10.3390/a19010028

designed as fragile [10–16], robust [23–27], and hybrid (robust/fragile) [28–35], in some cases, their aims are different from those of our proposal. In this way, a comparison with fragile algorithms [10–16] is not suitable by convention due to their absence of robustness against the image processing distortions investigated in this proposal. Conversely, a comparison with hybrid methods (robust/fragile) [28–35] is not appropriate because these works primarily aim to enhance the robustness of ownership authentication, while tamper detection and self-recovery are still handled by a fragile method. Comparatively, the robust modality [23–27] is unsuitable in terms of capabilities and design criteria when contrasting robust and semi-fragile watermarking modalities. However, we consider the related works reported in [25,27] because they focus on color images and are among the closest to our proposed method. From Table 8, we demonstrate that the proposals in [17,18,21] exhibit robustness against various signal processing distortions and possess the capability to detect alterations and recover lost content; however, these works are limited to grayscale images with a small spatial resolution of 512 × 512. The study in [19] is applicable to color images and shows robustness against a broad range of image processing distortions, including combined attacks; however, it primarily focuses on tamper detection with high accuracy. The works in [22,25] may align closely with our objectives and surpass our proposed method regarding robustness against common signal processing distortions; however, our proposal excels over [22] in its application to high-resolution color imaging, featuring color images relevant to real scenarios. Additionally, we outperform [25] in self-recovery, as [25] can only restore a halftone version of the luminance information, while our proposal recovers color images at the same resolution as the original. Finally, comparing our method with that of [27] indicates that our proposed method demonstrates greater robustness; furthermore, the method in [27] is restricted to RGBA images in Portable Network Graphics (PNG) format.

Table 9. Performance comparison.

Method	Domain	Spatial Resolution	Recover Information	Capabilities	Average Imperceptibility	Robustness
[17,18]	DCT IWT	-512 × 512 -8 bit/pixel	Luminance	-Tamper detection -Self recovery	PSNR = 42.46 dB SSIM = 0.9916	-Collage attack -Copy–paste -Text addition -JPEG lossy mode -Cropping -Rotation -Impulsive noise
[19]	QDFT	-512 × 512 -24 bit/pixel	N/A	-Tamper detection	PSNR = 40.12 dB SSIM = 0.9749	-Copy–paste -Sharpen -Gaussian, impulsive, and speckle noises -Histogram equalization -Contrast enhancement -Brightness -JPEG lossy mode -Median filtering -Combined attacks
[21]	IWT	-512 × 512 -8 bit/pixel	Luminance	-Tamper detection -Self recovery	PSNR = 54.73 dB WPSNR = 55.18 dB SSIM = 0.9987	-Collage attack -Copy–paste -Text addition -Cropping -Object manipulation -Impulsive noise

https://doi.org/10.3390/a19010028

Table 9. *Cont.*

Method	Domain	Spatial Resolution	Recover Information	Capabilities	Average Imperceptibility	Robustness
[22]	DWT	-512 × 512 -8 bit/pixel	Luminance	-Tamper detection -Self recovery - Ownership authentication	PSNR = 47.48 dB SSIM = 0.9981	-Copy–move -Copy–move -Splicing -Text addition -Impulsive and Gaussian noises -Mean and median filtering -Cropping -Rotation -Scaling -Translation
[25]	Distortion-free watermarking by double branch neural network	-Versatile resolution -24 bit/pixel	Halftone luminance	-Tamper detection -Self recovery -Ownership authentication	N/A	-Rotation -Scaling -Affine transformation -Translation -Cropping -JPEG lossy mode -Noise corruption -Histogram equalization -Blurring -Filtering -Sharpening -Combined attacks -Copy–move -Copy–paste
[27]	Secret sharing via distortion-free watermarking	-550 × 322 -512 × 512 -32 bit/pixel	Color	-Tamper detection -Self recovery	N/A	-Superimposing and Painting
Proposed Method	DCT	-4288 × 2848 -24 bit/pixel	Color	-Tamper detection -Self recovery -Ownership authentication	WPSNR = 43.64 dB FSIM = 0.9993 NCD = 0.03960	-JPEG lossy mode -Copy–move -Copy–paste -Collage attack -Impulsive noise -Gaussian noise -Cropping -JPEG75 + Copy–move -JPEG75 + Copy–paste -JPEG75 + Cropping -Gaussian filtering

5. Conclusions and Future Directions

In this paper, we propose a semi-fragile watermarking scheme for high-resolution color images. This scheme enhances conventional watermarking techniques, specifically the Quantization Index Modulation with Distortion Minimization (QIM-DM) in the Discrete Cosine Transform (DCT) domain, by improving payload capacity while maintaining a balance between imperceptibility and robustness. Our approach, in conjunction with the efficient JPEG2000 image compression algorithm, convolutional encoding, and supported

https://doi.org/10.3390/a19010028

by the DnCNN and VDSR neural networks, results in an effective multipurpose watermarking method. Tamper and forgery detection, along with recovery capabilities, present significant challenges in the field of digital image forensics and related areas. It is crucial to design algorithms that facilitate the recovery of forged content in real-world scenarios, and testing should extend beyond datasets with low spatial resolution. Additionally, assessing imperceptibility should involve metrics that reflect human visual system behavior, such as WPSNR and FISM, rather than relying solely on traditional PSNR and SSIM metrics. We evaluate the effectiveness of our proposed method by comparing its performance against existing state-of-the-art techniques. Our proposal preserves the original format and color resolution of the image dataset utilized. Future work will aim to further enhance the capacity, imperceptibility, and robustness achieved thus far, with a focus on significantly higher spatial resolutions and the development of a hybrid solution that combines both active and passive methodologies.

Author Contributions: Conceptualization, M.C.-H., A.C.-H. and J.C.S.-G.; methodology, M.C.-H. and A.C.-H.; software, M.C.-H. and A.C.-H.; validation, M.C.-H., A.C.-H., J.C.S.-G. and F.J.G.-U.; formal analysis, J.C.S.-G. and F.J.G.-U.; investigation, M.C.-H., A.C.-H. and F.J.G.-U.; resources, M.C.-H., F.J.G.-U. and J.C.S.-G.; data curation, M.C.-H.; writing—original draft preparation, M.C.-H.; writing—review and editing, A.C.-H., J.C.S.-G. and F.J.G.-U.; visualization, J.C.S.-G. and F.J.G.-U.; supervision, F.J.G.-U.; project administration, M.C.-H. and A.C.-H.; funding acquisition, M.C.-H., J.C.S.-G. and F.J.G.-U. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The original contributions presented in the study are included in the article; further inquiries can be directed to the corresponding author.

Acknowledgments: The authors thank the Instituto Politecnico Nacional (IPN), the Tecnologico de Monterrey, the Universidad Nacional Autonoma de México (UNAM), and the Secretaria de Ciencia, Humanidades, Tecnologia e Innovacion (SECIHTI) for their support during this research.

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

References

1. Barni, M.; Bartolini, F. Applications. In *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications*; CRC Press: Boca Raton, FL, USA, 2004. [CrossRef]
2. Bas, P.; Furon, T.; Cayre, F.; Doërr, G.; Mathon, B. A quick tour of watermarking techniques. In *Watermarking Security Fundamentals, Secure Design and Attacks*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 13–31. [CrossRef]
3. Cox, I.J.; Miller, M.L.; Bloom, J.A.; Fridrich, J.; Kalker, T. *Digital Watermarking and Steganography*, 2nd ed.; Morgan Kaufmann Publishers: San Francisco, CA, USA, 2009. [CrossRef]
4. Ji, L.; Kumar, S. Robust digital watermarking techniques for protecting copyright applied to digital data: A survey. *Turk. J. Comput. Math. Educ.* **2021**, *12*, 3819–3825. [CrossRef]
5. Chennamma, H.R.; Madhushree, B. A comprehensive survey on image authentication for tamper detection with localization. *Multimed. Tools Appl.* **2023**, *82*, 1873–1904. [CrossRef]
6. Madhushree, B.; Basanth Kumar, H.B.; Chennamma, H.R. An exhaustive review of authentication, tamper detection with localization and recovery techniques for medical images. *Multimed. Tools Appl.* **2024**, *83*, 39779–39821. [CrossRef]
7. Zheng, L.; Zhang, Y.; Thing, V.L. A survey on image tampering and its detection in real-world photos. *J. Vis. Commun. Image Represent.* **2019**, *58*, 380–399. [CrossRef]
8. Melman, A.; Evsutin, O. Methods for countering attacks on image watermarking schemes: Overview. *J. Vis. Commun. Image* **2024**, *99*, 104073. [CrossRef]
9. Awasthi, D.; Tiwari, A.; Khare, P.; Srivastava, V.K. A comprehensive review on optimization-based image watermarking techniques for copyright protection. *Expert Syst. Appl.* **2024**, *242*, 122830. [CrossRef]
10. Aminuddin, A.; Ernawan, F. AuSR1: Authentication and self-recovery using a new image inpainting technique with LSB shifting in fragile image watermarking. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 5822–5840. [CrossRef]

 https://doi.org/10.3390/a19010028

11. Azeroual, A.; Afdel, K. Real-time image tamper localization based on fragile watermarking and Faber-Schauder wavelet. *AEU-Int. J. Electron. Commun.* **2017**, *79*, 207–218. [CrossRef]
12. Adi, P.W.; Sugiharto, A.; Hakim, M.M.; Setiadi, D.R.I.M.; Winarno, E. Efficient fragile watermarking for image tampering detection using adaptive matrix on chaotic sequencing. *Intell. Syst. Appl.* **2025**, *26*, 200530. [CrossRef]
13. Bhalerao, S.; Ansari, I.A.; Kumar, A. A secure image watermarking for tamper detection and localization. *J. Ambient Intell. Hum. Comput.* **2021**, *12*, 1057–1068. [CrossRef]
14. Bouarroudj, R.; Souami, F.; Bellala, F.Z.; Zerrouki, N.; Harrou, F.; Sun, Y. Secure and reversible fragile watermarking for accurate authentication and tamper localization in medical images. *Comput. Electr. Eng.* **2025**, *123*, 110072. [CrossRef]
15. Singh, D.; Singh, S.K.; Udmale, S.S. An efficient self-embedding fragile watermarking scheme for image authentication with two chances for recovery capability. *Multimed. Tools Appl.* **2023**, *82*, 1045–1066. [CrossRef]
16. Xia, X.; Zhang, S.; Wang, K.; Gao, T. A novel color image tampering detection and self-recovery based on fragile watermarking. *J. Inf. Secur. Appl.* **2023**, *78*, 103619. [CrossRef]
17. Sivasubramanian, N.; Konganathan, G. A novel semi fragile watermarking technique for tamper detection and recovery using IWT and DCT. *Computing* **2020**, *102*, 1365–1384. [CrossRef]
18. Benrhouma, O. Cryptanalysis and improvement of a semi-fragile watermarking technique for tamper detection and recovery. *Multimed. Tools Appl.* **2023**, *82*, 22149–22174. [CrossRef]
19. Ouyang, J.; Huang, J.; Wen, X.; Shao, Z. A semi-fragile watermarking tamper localization method based on QDFT and multi-view fusion. *Multimed. Tools Appl.* **2023**, *82*, 15113–15141. [CrossRef]
20. Nanammal, V.; Rajalakshmi, S.; Remya, V.; Ranjith, S. ViTU-net: A hybrid deep learning model with patch-based LSB approach for medical image watermarking and authentication using a hybrid metaheuristic algorithm. *Comput. Biol. Med.* **2025**, *194*, 110393. [CrossRef]
21. Amrullah, A.; Ernawan, F.; Raffei, A.F.M.; Chuin, L.S. TDSF: Two-phase tamper detection in semi-fragile watermarking using two-level integer wavelet transform. *Eng. Sci. Technol. Int. J.* **2025**, *61*, 101909. [CrossRef]
22. Vaidya, S.P.; Kandala, R.N.V.P.S.; Chandra Mouli, P.V.S.S.R.; Zaini, H.G.; Jaffar, A.; Paramasivam, P.; Ghoneim, S.S.M. A robust fragile watermarking approach for image tampering detection and restoration utilizing hybrid transforms. *Sci. Rep.* **2025**, *15*, 17645. [CrossRef]
23. Tahaoglu, G.; Ustubioglu, B.; Ulutas, G.; Ulutas, M.; Nabiyev, V.V. Robust Copy-Move Forgery Detection Technique Against Image Degradation and Geometric Distortion Attacks. *Wirel. Pers. Commun.* **2023**, *131*, 2919–2947. [CrossRef]
24. Ying, Q.; Hu, X.; Zhang, X.; Qian, Z.; Li, S.; Zhang, X. RWN: Robust Watermarking Network for Image Cropping Localization. In Proceedings of the 2022 IEEE International Conference on Image Processing (ICIP), Bordeaux, France, 16–19 October 2022; pp. 301–305. [CrossRef]
25. Arevalo-Ancona, R.E.; Cedillo-Hernandez, M. Robust zero-watermarking based on dual branch neural network for ownership authentication, auxiliary information delivery and tamper detection. *Egypt. Inform. J.* **2025**, *30*, 100650. [CrossRef]
26. Ramirez-Rodriguez, A.E.; Arevalo-Ancona, R.E.; Perez-Meana, H.; Cedillo-Hernandez, M.; Nakano-Miyatake, M. AISMSNet: Advanced Image Splicing Manipulation Identification Based on Siamese Networks. *Appl. Sci.* **2024**, *14*, 5545. [CrossRef]
27. Lee, C.-W. A distortion-free authentication method for color images with tampering localization and self-recovery. *Signal Process. Image Commun.* **2024**, *124*, 117116. [CrossRef]
28. Li, M.; Yue, Y. Security Analysis and Improvement of Dual Watermarking Framework for Multimedia Privacy Protection and Content Authentication. *Mathematics* **2023**, *11*, 1689. [CrossRef]
29. Al-Otum, H.M.; Ellubani, A.A.A. Secure and effective color image tampering detection and self restoration using a dual watermarking approach. *Optik* **2022**, *262*, 169280. [CrossRef]
30. Laishram, D.; Manglem Singh, K. A watermarking scheme for source authentication, ownership identification, tamper detection and restoration for color medical images. *Multimed. Tools Appl.* **2021**, *80*, 23815–23875. [CrossRef]
31. Pal, P.; Jana, B.; Bhaumik, J. An Image Authentication and Tampered Detection Scheme Exploiting Local Binary Pattern Along with Hamming Error Correcting Code. *Wirel. Pers. Commun.* **2021**, *121*, 939–961. [CrossRef]
32. Sharma, S.; Zou, J.J.; Fang, G. A Novel Multipurpose Watermarking Scheme Capable of Protecting and Authenticating Images with Tamper Detection and Localisation Abilities. *IEEE Access* **2022**, *10*, 85677–85700. [CrossRef]
33. Sinhal, R.; Ansari, I.A. Multipurpose Image Watermarking: Ownership Check, Tamper Detection and Self-recovery. *Circuits Syst. Signal Process.* **2022**, *41*, 3199–3221. [CrossRef]
34. Bei, Y.-L.; Zhu, X.-R.; Zhang, Q.; Qiao, S. A Robust Image Watermarking Algorithm Based on Content Authentication and Intelligent Optimization. In Proceedings of the 5th International Conference on Control and Computer Vision (ICCCV 2022), New York, NY, USA, 19–21 August 2022; Association for Computing Machinery: New York, NY, USA, 2022; pp. 164–170. [CrossRef]

 https://doi.org/10.3390/a19010028

35. Meng, B.; Yuan, X. Robust and fragile watermarking for medical imaging. In Proceedings of the Sixteenth International Conference on Signal Processing Systems (ICSPS 2024), Kunming, China, 15–17 November 2024; SPIE: Washington, DC, USA, 2024; p. 135595. [CrossRef]
36. Chen, B.; Wornell, G.W. Quantization index modulation: A class of provably good method for digital watermarking and information embedding. *IEEE Trans. Inf. Theor.* **2001**, *47*, 1423–1443. [CrossRef]
37. Cedillo-Hernandez, M.; Cedillo-Hernandez, A.; Nakano-Miyatake, M.; Perez-Meana, H. Improving the management of medical imaging by using robust and secure dual watermarking. *Biomed. Signal Process. Control* **2020**, *56*, 101695. [CrossRef]
38. Odenwalder, J.P. Optimal Decoding of Convolutional Codes. Ph.D. Dissertation, University of California, Los Angeles, CA, USA, 1970.
39. Viterbi, A. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory* **1967**, *13*, 260–269. [CrossRef]
40. Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; Zhang, L. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. *IEEE Trans. Image Process.* **2017**, *26*, 3142–3155. [CrossRef]
41. Grubinger, M.; Clough, P.; Müller, H.; Deselaers, T. The IAPR TC-12 Benchmark: A New Evaluation Resource for Visual Information Systems. In Proceedings of the OntoImage 2006 Language Resources for Content-Based Image Retrieval, Genoa, Italy, 22 May 2006; Volume 5, p. 10. Available online: http://www.lrec-conf.org/proceedings/lrec2006/workshops/W02/RealFinalOntoImage2006-2.pdf (accessed on 3 November 2025).
42. Kim, J.; Lee, J.K.; Lee, K.M. Accurate Image Super-Resolution Using Very Deep Convolutional Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1646–1654. [CrossRef]
43. He, K.; Zhang, X.; Ren, S.; Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 13–16 December 2015; pp. 1026–1034. [CrossRef]
44. Board, C.A. IEEE Standard for Camera Phone Image Quality. In *IEEE Stdandard 1858-2016 (Incorporating IEEE Stdandard 1858-2016/Cor. 1-2017)*; IEEE: New York, NY, USA, 2017; pp. 1–146. [CrossRef]
45. Taubman, D.S.; Marcellin, M.W. JPEG2000: Standard for interactive imaging. *Proc. IEEE* **2002**, *90*, 1336–1357. [CrossRef]
46. Taubman, D.S.; Marcellin, M.W. Image Compression Fundamentals, Standards and Practice. In *The Springer International Series in Engineering and Computer Science*; Springer: New York, NY, USA, 2001; ISBN 978-0-7923-7519-7. [CrossRef]
47. Gonzalez, R.C.; Woods, R.E. *Digital Image Processing*, 4th ed.; Pearson Education: New York, NY, USA, 2018; p. 1022p.
48. Liang, Q.; Xiang, S. Invertible Color-to-Grayscale Conversion Using Lossy Compression and High-Capacity Data Hiding. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 7373–7385. [CrossRef]
49. Fragoso-Navarro, E.; Garcia-Ugalde, F.; Cedillo-Hernandez, M. Improved protection for colour images via invertible colour-to-grey, image distortion, and visible-imperceptible watermarking. *IET Image Process.* **2024**, *18*, 3854–3863. [CrossRef]
50. Dang-Nguyen, D.-T.; Pasquini, C.; Conotter, V.; Boato, G. RAISE—A Raw Images Dataset for Digital Image Forensics; ACM Multimedia Systems; MMSys '15. In Proceedings of the 6th ACM Multimedia Systems Conference, Portland, OR, USA, 18–20 March 2015; pp. 219–224. [CrossRef]
51. Moldovanu, S.; Michis, F.A.; Moraru, L. An invisible DWT watermarking algorithm using noise removal with application to dermoscopic images. *J. Phys. Conf. Ser.* **2021**, *1730*, 012010. [CrossRef]
52. Sara, U.; Akter, M.; Uddin, M.S. Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. *J. Comput. Commun.* **2019**, *7*, 8–18. [CrossRef]
53. Chang, H.-A.; Chen, H.H. Stochastic Color Interpolation for Digital Cameras. *IEEE Trans. Circuits Syst. Video Technol.* **2007**, *17*, 964–973. [CrossRef]
54. Control Random Number Generator, Matlab Help Center. Available online: https://mathworks.com/help/matlab/ref/rng.html (accessed on 17 December 2025).

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/a19010028

algorithms

MDPI

Article

Bayesian Optimisation and Adaptive Evolutionary Algorithms for Higher-Order Fuzzy Models with Application on Wind Speed Prediction

Panagiotis Korkidis and Anastasios Dounis *

Department of Biomedical Engineering, University of West Attica, 12243 Athens, Greece; p.korkidis@uniwa.gr
* Correspondence: aidounis@uniwa.gr

Abstract

To cope with the highly stochastic nature of wind speed, we explored the development of a predictive methodology. Considering an absence of studies pertaining to wind speed prediction that utilise state-of-the-art fuzzy models, the proposed approach adopted a novel higher-order Takagi–Sugeno–Kang fuzzy model intermixed with variational mode decomposition. The novelty of the predictive fuzzy model arises from the enhancement of rule consequents to include generalised terms and the incorporation of model complexity into the training scheme. To optimise the model, two approaches are considered: an adaptive differential evolution and a surrogate-based optimisation algorithm. The evolutionary approach employed two populations and a dual mutation scheme. The surrogate-based optimisation employed a Bayesian framework by fitting a Gaussian process model to the objective function. The latter approach yielded accurate predictive results while rapidly reducing the training time of the fuzzy model. A sequential wrapper-based algorithm was developed to effectively determine the feature space. The variational mode decomposed wind speed data were predicted individually, using an associated optimised fuzzy model. The proposed method was applied to a real-world wind speed dataset with exceptional approximation results. Comparisons with several artificial intelligence models highlighted the effectiveness and statistical significance of the methodology.

Keywords: wind speed prediction; Takagi–Sugeno–Kang model; adaptive differential evolution; Bayesian optimisation; variational mode decomposition; wrapper feature selection

1. Introduction

1.1. On Wind's Nature and Wind Energy's Importance

The movement of air masses, what is experienced as wind, plays a significant role in the Earth's climate. Wind carries a significant amount of kinetic energy, referred to as wind energy, which can be harnessed using wind turbines. This movement of air masses is primarily driven by variations in barometric pressure, mainly caused by the uneven heating of the various regions of the Earth's surface by the solar radiation. Due to the Earth's surface curvature, at higher latitudes, the Sun's ray travels a longer path through the atmosphere, which in turn results in varying amounts of solar energy being absorbed across regions. The temperature difference creates pressure gradients, which generate winds as air moves from areas of high to areas of low pressure. The main characteristics of wind are its speed and direction. The wind's speed, as a natural phenomenon, is continuously changing;

thus, the need to analyse its temporal evolution is essential in designing and optimising renewable energy systems, since the wind power of wind turbines is proportional to the cube of wind speed.

The goals outlined in the 2030 Agenda for Sustainable Development (https://sdgs.un.org/2030agenda (accessed on 10 September 2025)) are vital for both humanity and the planet. In particular, the seventh goal emphasises the importance of ensuring access to affordable, reliable, sustainable, and modern energy for all, with a strong focus on significantly increasing the global share of renewable energy. Among renewable energy options, wind energy stands out as one of the most promising and rapidly advancing sources. Its clean and abundant nature has attracted growing interest. This energy source has the potential to substantially reduce greenhouse gas emissions. Additionally, wind energy offers the advantage of widespread availability and suitability for remote areas, e.g., islands with limited access to conventional power grids. However, wind speed's highly stochastic and intermittent nature makes prediction an elusive topic, which has consequently drawn significant attention from the research community.

1.2. Related Work

According to the methodologies adopted in the literature, as well as the models of choice, four major classes of research can be distinguished: physical, statistical, intelligent, and hybrid methods [1]. As far as we are concerned, intelligent and hybrid methods can be considered as two sides of the same coin, named intelligent methods, and thus in the following analysis will be treated as a unified group.

Physical methods require precise and reliable observations and refer to numerical weather prediction, using explicit mathematical formulas and physical information. Despite the fact that physical methods are quite interpretable and accurate, their computational cost limits their adoption for short-term wind speed prediction. The remaining classes share a common principle; they are all based on data-driven models, i.e., models that learn and make predictions, given observations.

The first class is associated with classical statistical approaches of time series forecasting, such as ARMA and ARIMA, together with their variations [2], linear regression statistical methods, Bayesian linear models, Markov models, etc. The results generated by these approaches alone are satisfactory, but they cannot learn highly nonlinear and noisy data. It should, though, be mentioned that models from this class are quite often adopted in hybrid methods. In [3], the authors combined a nonlinear autoregressive model with a general regression neural network for the prediction of wind speed at two wind farms. Wang et al. [4] proposed a novel wind speed prediction model by combining statistical approaches, deep learning methods, and multi-objective optimisation. Zhang et al. in [5] studied the problem by decomposing the data into linear, nonlinear, and noise components. A radial basis function network is employed by the authors to cope with the nonlinear part of the data, and a Bayesian-based autoregressive moving average model with the associated linear model.

Intelligent methods, built primarily on artificial and computational intelligence models, have an inherent ability to incorporate the complex characteristics of data and display enhanced capabilities in learning nonlinear wind characteristics and patterns. Such approaches include neural networks, from shallow [6,7] to deep models [8–10], fuzzy systems [11,12], support vector machines [13,14], Gaussian processes [15–17], and ensemble models [18–20]. The latter class of methods serves as a landscape for developing new synergies along classical and modern models, and novel approaches, combined with data processing techniques, as well as optimisation methods. This interplay between ele-

 https://doi.org/10.3390/a19010046

ments of intelligent methods, e.g., data decomposition methods, evolutionary optimisation methods, and various models, leads to the generation of creative and high-performance methodologies. Decomposition of data, using SSA, is utilised in [21], and a combination with a structural learning algorithm of a neural network is proposed for wind speed prediction. Singular spectrum analysis is also used in [22], with an ANFIS to serve as the predictive model. The adaptation of ANFIS as a candidate predictive model is also considered in [23]. In the latter, authors studied wind speed and direction prediction from the perspective of an evolutionary trained adaptive-network-based fuzzy inference system, and compared the results with neural networks and a hybrid RBF-SVM model. An interesting adaptation of ANFIS is proposed in [24], where it serves as an estimator of the wind speed's Weibull probability density function, given the fact that wind speed data best follow a Weibull distribution. A novel methodology is studied in [25], where the original wind speed time series data are denoised using wavelets and then decomposed into several components using the extreme-point symmetric mode decomposition. An LS-SVM is the model of choice, and its parameters, e.g., kernel and regularisation, are computed using an evolutionary algorithm, called the fractional-order beetle swarm algorithm. A comprehensive survey for general time series decomposition methods is given in [26]. Further studies of wind speed predictive methodologies based on decomposition methods can be found in [19,27–29]. In the context of time series prediction, [30] studied the integration of a recurrent fuzzy neural network and variational mode decomposition for stock market index forecasting. Variational mode decomposition is also utilised in [31], intermixed with support vector machines and multi-objective optimisation for wind speed prediction. Wavelet decomposition is adopted in [32] for time series forecasting from multidisciplinary fields. Empirical wavelet transform is combined with variational mode decomposition and a deep LSTM network in [33], for multi-step cutterhead torque prediction. Taking a step back, it is necessary to mention that a few papers incorporate physical methods, e.g., numerical weather prediction, along with deep architectures [34] and Gaussian processes [15]. Finally, since wind speed models are based on observations, which in turn might be noisy and uncertain, [27,35,36] proposed a framework not only for point, but interval predictions as well, so that uncertainty is quantified.

1.3. Motivation and Contributions

Having examined the vista of wind speed prediction methods, several common issues have become evident. The research gaps identified, along with the solutions proposed in this study, are outlined below.

- There is a significant absence of studies utilising optimised, state-of-the-art fuzzy models. Even when fuzzy models are employed, they often consist of vanilla versions, lacking enhancements. Thus, this study proposes a novel fuzzy system, specifically a Takagi–Sugeno–Kang model with generalised rule consequents. The model's complexity is integrated into the training algorithm rather than being established a priori. According to the neuro-fuzzy modeling survey [37], there is a lack of research that combines TSK fuzzy systems with Bayesian optimisation; hence, this study proposes the incorporation of surrogate models for accurate and time-efficient model training. This involves adjusting its antecedent parameters, regularisation, and the complexity of the consequents using Bayesian optimisation. Furthermore, a hybrid evolutionary training method for the fuzzy model is explored, which incorporates a modern version of differential evolution with adaptive parameters and two populations.
- The selection of feature space, i.e., the input selection to the model, is not systematically studied; most research studies lack a clear justification for the choice of inputs used in

https://doi.org/10.3390/a19010046

their models. Thus, this study develops a sequential algorithm based on a wrapper-based approach to systematically select inputs for the model. This method minimises the generalisation error, allowing for the determination of the necessary number of model inputs without added complexity.

- Most methodologies rely on deep and/or machine learning models to perform the prediction task. While such models yield accurate results, they are frequently considered as black boxes. Thus, this study employs a fuzzy model as the prediction tool to address the existing gaps in research while providing a basis for analysing the results generated by simpler models, rather than defaulting to more complex deep learning approaches. Since fuzzy systems possess an intrinsic interpretability, encoded in terms of fuzzy rules, this paper offers a starting point for further studies toward interpretable models.

A Takagi–Sugeno–Kang model with generalised rule consequents has not been previously studied in the context of wind speed prediction. Furthermore, the application of this version of adaptive differential evolution for training fuzzy systems is missing in the literature. Finally, wrapper-based algorithms are rarely studied when developing wind speed prediction methods. We believe that this research provides valuable innovations in the field of intelligent methods, particularly data-driven algorithms for training fuzzy systems with applications to wind speed prediction.

1.4. Method Description

A description of the method follows: The wind speed data undergo pre-processing, which involves the replacement of any missing values, and are subsequently normalised. Then, the data are granulated into training and testing sets. A candidate feature space of the predictive models is generated. Using a sequential algorithm, specifically a wrapper-based method, we selected a subset of the candidate feature space as the input space, such that the performance on test observations is the best. A novel Takagi–Sugeno–Kang fuzzy system is employed to play the role of the foundational model. We proposed a generalised form of functionals for each fuzzy rule's consequent, thereby enhancing the complexity and degrees of freedom within the fuzzy model. A hybrid method of fuzzy model training was then considered: The antecedents' parameters were optimised using a modern version of differential evolution, which incorporated both elite and non-elite populations while self-adapting its evolution operators' parameters, and the consequents of the fuzzy rules were computed using least squares. Additionally, due to the fact that evolutionary algorithms can be time-consuming for optimisation, we proposed Bayesian optimisation, a surrogate model-based optimisation method, as an alternative method to achieve time-efficient and accurate hybrid training of the models. The methodology is based on the use of variational mode decomposition, chosen as the decomposition method providing the best results in our example. The original data were decomposed on a set of mode functions, with each fuzzy model corresponding to a particular mode function as its predictive model. The structure of each fuzzy model's rule functionals was encoded in the optimisation algorithm, during the training phase, along with a regularisation parameter and the fuzzy model antecedents' parameters. The final prediction was obtained by appropriately combining the individual predictions of each fuzzy model. To validate the results, comparisons were provided with a well-known and widely used fuzzy inference system, the Wang-Mendel method, an adaptive-network-based fuzzy inference system, an automated selection method of machine learning models, as well as a vanilla version of a Takagi–Sugeno–Kang system.

The remainder of this paper is organised as follows: A statement of the problem under study is given in Section 2, together with a concise discussion on preliminary notions. The

 https://doi.org/10.3390/a19010046

proposed methodology is described in Section 3, where the algorithms associated with the proper development of the prediction models are analysed in detail. Numerical studies are illustrated in Section 4, demonstrating the final methodology's performance, as measured by the regression metrics applied to a real-world wind speed dataset provided from the National Observatory of Athens, Greece. Moreover, comparisons among the proposed models and vanilla TSK fuzzy models, WM fuzzy models, an adaptive-network-based fuzzy inference system, and an automated selection methodology of machine learning models are provided to validate the results. A systematic discussion of the assumptions and the limitations of the methods considered can be found in Section 5. Finally, Section 6 presents the conclusions and outlines future research goals.

2. Preliminaries

This section provides a brief overview of the methods used in this paper. It includes a statement of the problem, as well as a concise description of the models utilised for comparisons.

2.1. Problem Formulation

Time series consist of ordered temporal observation sequences, such as $\{x_1, ..., x_{\mathcal{T}}\}$. The task of time series prediction involves predicting a future realisation of the variable under study based on past time observations. Mathematically, this problem can be viewed as a mapping from the feature space $\mathcal{X}$, given by the lagged values of the variable under study, $(x(t-D) \dots x(t-1)) \in \mathcal{X} \subseteq \mathcal{R}^D \rightarrow x(t) \in \mathcal{R}$. The space $\mathcal{X}$ is generated by selecting the appropriate lags, $m \leq D$, forming the model's necessary predictors.

The cornerstone of each intelligent model lies in learning the unknown mapping between the m-tupled input space and the output space, i.e., the previous values of the variable and its future realisation. The models were developed in a supervised learning setting, utilising the available dataset $\mathcal{D} = (\mathcal{X}, y)$, where $y \in \mathcal{Y} := x_{t \in \mathcal{T}}(t)$ denotes the future values, and $\mathcal{T}$ denotes the observations' time index. The training of the models was conducted on the training set $\mathcal{D}^{tr}$, and their performance was evaluated by testing predictions on both the training and, more importantly, the testing set.

2.2. Fuzzy Systems

Fuzzy systems have been adopted in numerous fields of science and engineering. Endowed with a universal approximation property [38,39], fuzzy systems can infer complex unknown maps. Their formalism is described by a precise logic of imprecision and approximate reasoning [40]. Albeit fuzzy systems initially gained popularity due to their ability to encode human knowledge in the form of if-then rules, they nowadays represent a vibrant research area within the artificial intelligence canvas, where researchers continually explore new methods and applications, enhancing the theoretical foundations and practical implementations of fuzzy systems. In an era where data is readily available, a data-driven approach to generating fuzzy systems is absolutely relevant.

A fuzzy system contains four interconnected components: fuzzification, rule base, inference, and defuzzification. An input to the system, after being fuzzified, activates a fuzzy rule to a certain degree. Each fuzzy rule generates, through inference, an output fuzzy set Y^l. These fuzzy sets are aggregated via a sequence of t-conorms, i.e., $\biguplus$, to generate the aggregated output fuzzy set Y, given by Equation (1).

$$\mathrm{Y} = \biguplus_{l=1}^{\mathbb{M}} Y^l \tag{1}$$

https://doi.org/10.3390/a19010046

The output fuzzy set Y is then defuzzified to produce the fuzzy system's output. The general form of a fuzzy rule of a Mamdani system with m inputs in $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 ... \times \mathcal{X}_m$ and output $y \in \mathcal{Y}$, is given by Equation (2).

$$R^l : \text{If } x_1 \text{ is } \mathcal{A}_1^l \text{ and } \ldots \ x_m \text{ is } \mathcal{A}_m^l, \text{ then } y \text{ is } G^l \tag{2}$$

where $\mathcal{A}_i^l \subset \mathcal{X}_i$, $i = 1, ..., m$, and $G^l \subset \mathcal{Y}$ are fuzzy sets of inputs and outputs, respectively. Fuzzy rules can be presumed as local models. By interpolating these local models, generalised representations are attained that effectively learn nonlinear relationships.

2.2.1. Wang-Mendel Fuzzy Inference System

The method of Wang-Mendel [41] is a widely adopted method of generating Mamdani fuzzy systems given observations. Since fuzzy systems suffer from the *curse of dimensionality*, manifested in the face of rule explosion, the WM method effectively reduces the number of fuzzy rules in the final inference system. This results in a simpler and more computationally efficient model. The parametric space optimised in the simplest version of Wang-Mendel's method is limited to the number of fuzzy rules; hence, a WM-based model can be considered a vanilla version of a data-driven fuzzy system. The procedure of generating fuzzy rules from the given data is analysed in the following steps:

Step 1: Generate fuzzy partitions for each input and output variable of the system, to cover the associated universes of discourse;
Step 2: For all $t \in \mathcal{T}^{tr} \subset \mathcal{T}$, i.e., all observations in $\mathcal{D}^{tr}$, generate a fuzzy rule. This involves the computation of membership degrees of the input vector to each fuzzy partition. For each observation, the fuzzy set with maximum membership is kept as the antecedent and consequent part of the inputs and output;
Step 3: Compute the degree of each fuzzy rule generated $\prod_{i=1}^{m} \mu_{\mathcal{A}_i^{l*}}(x_i)\mu_{G^{l*}}(y)$, where $\mu_{\mathcal{A}_i^{l*}}(x_i)$, and $\mu_{G^{l*}}(y)$ are the maximum memberships of fuzzy sets for each input and output, respectively;
Step 4: Remove all possible conflicting fuzzy rules to establish the final fuzzy rule base. Since the cardinality of the generated rule base is the same as the number of instances in $\mathcal{D}^{tr}$, rules with the same antecedent exist. Amongst these rules, the ones considered to form the final rule base are those with the maximum degree;
Step 5: Determine the fuzzy system's parameters: fuzzy connectives t-norms and t-conorms, fuzzy implication, aggregation and defuzzification operators;
Step 6: Test the fuzzy inference system's performance on a set of test observations.

2.2.2. Adaptive-Network-Based Fuzzy Inference System

ANFIS, proposed by Jang in [42], is one of the most successful fuzzy models due to its exceptional performance in regression tasks. It provides an efficient supervised learning algorithm for training TSK models by combining the theories of neural networks and fuzzy systems. It includes a five-layer structure as schematically illustrated in Figure 1.

Two sets of trainable parameters are involved: the membership functions' parameters in the antecedent parts of the rules, and the linear rules' consequent parameters. In the framework of neuro-fuzzy modeling, the training emerges from the neural networks literature and involves a gradient descent-based method for the determination of the antecedent parameters. Training is evolving on a hybrid basis; the forward pass, where the consequents' parameters are computed in batch mode, via least squares, and the backward pass, where gradient descent updates the antecedents' parameters.

https://doi.org/10.3390/a19010046

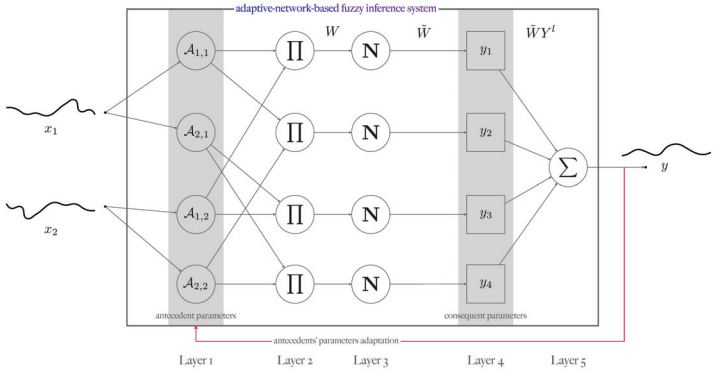


Figure 1. Layer representation of ANFIS.

2.3. Variational Mode Decomposition

Variational mode decomposition, introduced by Dragomiretskiy and Zosso in [43], is an adaptive and non-recursive decomposition method, noted for its enhanced robustness to noise and its capability to achieve precise separation of components [1]. It decomposes a time series into an ensemble of band-limited intrinsic mode functions, denoted by $u_k(t)$, with limited bandwidth and extracts the central frequency ω_k of the corresponding mode function. Each mode fluctuates around the central frequency. Intrinsic mode functions' main characteristic in variational mode decomposition is that each one exhibits a cosine function wave shape, featuring slowly varying and positive envelopes, and an instantaneous frequency that changes slowly in a nondecreasing manner. Consequently, the sub-signals $u_k(t)$ are appropriately combined to reproduce the original time series. The task is reflected in solving Equation (3), i.e., the variational optimisation problem given by:

$$\min_{u_k,\omega_k} \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2, \text{ s.t. } \sum_k u_k = f \tag{3}$$

where $\delta(.)$ represents the Dirac delta function and f denotes the original signal. The constrained problem is subsequently transformed into an unconstrained one via an augmented Lagrangian. Furthermore, a quadratic penalty term is added to ensure signal reconstruction under Gaussian noise [30], as shown in Equation (4).

$$\begin{aligned} \mathcal{L}(u_k, \omega_k, \varrho) = \varepsilon \sum_k \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \\ + \left\| f(t) - \sum_k u_k(t) \right\|_2^2 + \left\langle \varrho(t), f(t) - \sum_k u_k(t) \right\rangle \end{aligned} \tag{4}$$

where ϱ and ε denote the Lagrange multiplier and penalty parameter, respectively. The authors of the original paper solved the latter defined problem by using an iterative numerical method named alternating direction method of multipliers.

2.4. Evolutionary and Bayesian Optimisation

Evolutionary algorithms are a subclass of metaheuristics that explore the search space in a stochastic manner, aiming to find a global minimum on a minimisation problem. They are gleaned from models of biological evolution. Their conceptual base is to evolve a population of candidate solutions by utilising *variation* and *selection* operators. The former are utilised to bring diversity to the population for search space exploration, whilst the

https://doi.org/10.3390/a19010046

latter are used to guide the search towards the best solutions. EAs can be used for any problem for which we can at least evaluate the function to be minimised.

The evolutionary process is influenced by the algorithm's parameters. Modern evolutionary computational methods are enhanced by self-adjust capabilities; adapting the parameters which govern the variation and selection operators, as such, providing a robust and automated tool for many challenging optimisation problems.

Evaluating the function that needs to be minimised is crucial when using evolutionary algorithms. However, there are problems associated with this evaluation, such as being difficult or impractical, due to computational resources, time-consuming experiments, and a lack of explicit expressions. In such cases, a learning model can be adopted as a surrogate-based approximation of the original function. In this light, Bayesian optimisation [44,45] offers a valuable approach. Bayesian optimisation treats the objective function as a black box, and by fitting a surrogate model, it effectively reduces the number of evaluations needed. Based on the Bayesian paradigm, from where its name is inherited, Bayesian optimisation captures prior belief about the function to be minimised using a prior distribution, and utilises the surrogate model's response to update the posterior distribution. It is important to mention that Bayesian optimisation emerged from the very need to optimise machine learning models. Compared to evolutionary algorithms, it does not require function evaluations on a population, thus allowing more efficient use of resources.

2.5. Feature Selection

Feature selection involves choosing a subset of features $\mathcal{X}$ based on specific criteria. By reducing the number of features in the model, dimensionality reduction is achieved, resulting in lower computational complexity. According to [1], the two main frameworks for feature selection are the filter and wrapper methods. In this study, a wrapper approach was adopted. In its general formulation, the wrapper approach consists of using the predictive performance of a learning model to evaluate the relative effectiveness of subsets of features. The simple principle is to start with a small number of features and sequentially add those that increase the model's predictive performance. This is referred to as a forward selection approach to the wrapper method, i.e., features are gradually added to increasingly larger subsets.

3. Proposed Method

We proceed with a thorough analysis of the predictive (foundational) model, the optimisation, as well as the feature selection scheme. Finally, we describe the complete predictive methodology for generating the *evolutionary* fuzzy TSK and the *Bayesian*-optimised TSK models in combination with the variational mode decomposition, for wind speed prediction.

3.1. The Foundational Model

Let $\mathcal{D}^{tr} = (\mathcal{X}^{tr}, y^{tr})$ be the training set, where $\mathcal{X}^{tr} = [x(t-D) \; ... \; x(t-1)]_{t \in \mathcal{T}^{tr}}$ and $y^{tr} \in \mathcal{Y} = x_{t \in \mathcal{T}^{tr}}(t)$. For a more convenient and compact notation, the elements of $\mathcal{X}^{tr}$, i.e., the inputs, will be denoted as x_i, where $i = 1, ..., m$, and the elements of $\mathcal{Y}$ as y.

Let a TSK fuzzy model with m inputs of $|\mathcal{T}|$ observations, where the training set evolves in $\mathcal{T}^{tr}$ and the testing set in $\mathcal{T}^{*}$. The fuzzy model's l^{th} rule [46] can be written as in Equation (5).

$$R^l : \text{If } x_1 \text{ is } \mathcal{A}_1^l \text{ and } \ldots \; x_m \text{ is } \mathcal{A}_m^l, \text{ then } y^l = g^l(x_1, ..., x_m | \theta_{\text{cons}}^l) \tag{5}$$

where $\mathcal{A}_i^l$ are fuzzy sets of the i^{th} input's fuzzy partition, related to the l^{th} rule. The rule index l runs from unity to the total number of fuzzy rules $\mathbb{M}$. The consequents are now

 https://doi.org/10.3390/a19010046

given by the function g; a combination of the model's inputs and the parameter vector θ^l_{cons}. The fuzzy sets (Equation (6)) map the space $\mathcal{X}_i$ into $[0,1]$ as:

$$\mathcal{A}^l_i = \{(x_i, \mu_{\mathcal{A}^l_i}(x_i))|\ x_i \in \mathcal{X}_i,\ \mu_{\mathcal{A}^l_i} : \mathcal{X}_i \to [0,1]\} \tag{6}$$

The fuzzy sets are generated via Gaussian membership functions (Equation (7)), given by:

$$\mu_{\mathcal{A}^l_i}(x_i) = \exp\left(-\frac{(x_i - c_{l,i})^2}{2\sigma^2_{l,i}}\right) \tag{7}$$

where $c_{l,i}$ is the Gaussian membership function's core associated with i^{th} input and l^{th} rule, and $\sigma_{l,i}$ the standard deviation. We opted to use Gaussian membership functions due to their defining characteristic of having only two adjustable parameters. This simplicity facilitated efficient manipulation in the optimisation scheme.

Equation (8), denotes the firing degree of the l^{th} rule, computed using product t-norms:

$$w^l = \prod_{i=1}^{m} \mu_{\mathcal{A}^l_i}(x_i) \tag{8}$$

If the normalised firing degrees (Equation (9)), i.e., the fuzzy basis functions, of the l^{th} rule are defined as:

$$\tilde{w}^l = w^l / \sum_{j=1}^{\mathbb{M}} w^j \tag{9}$$

then the output of the Takagi–Sugeno–Kang fuzzy model can be written in terms of the fuzzy basis functions expansion (Equations (10)–(12)).

$$y_{\text{TSK}} = \frac{\sum_{l=1}^{\mathbb{M}} \prod_{i=1}^{m} \mu_{\mathcal{A}^l_i}(x_i) y^l}{\sum_{l=1}^{\mathbb{M}} \prod_{i=1}^{m} \mu_{\mathcal{A}^l_i}(x_i)} \tag{10}$$

$$= \frac{\sum_{l=1}^{\mathbb{M}} w^l g^l(x_1, ..., x_m|\theta^l_{\text{cons}})}{\sum_{l=1}^{\mathbb{M}} w^l} \tag{11}$$

$$= \sum_{l=1}^{\mathbb{M}} \tilde{w}^l g^l(x_1, ..., x_m|\theta^l_{\text{cons}}) \tag{12}$$

We enhanced the functional form of the fuzzy rules' consequents by incorporating nonlinear terms to increase the model's degrees of freedom. At this point, let us introduce the parameter $\mathcal{I} \in \{1,2,3,4\} \subset \mathbb{Z}$, which indicates the potential functional forms that the rules' consequent may have. The potential forms of $g^l_{\mathcal{I}}$ are mathematically described in Equations (13)–(16), according to the value of $\mathcal{I}$.

$$g^l_1(x_1, ..., x_m|\theta^l_{\text{cons}}) = b^0_l + \sum_i b^1_{l,i} x_i \tag{13}$$

$$g^l_2(x_1, ..., x_m|\theta^l_{\text{cons}}) = b^0_l + \sum_i b^1_{l,i} x_i + \sum_i b^2_{l,i} x^2_i \tag{14}$$

$$g^l_3(x_1, ..., x_m|\theta^l_{\text{cons}}) = b^0_l + \sum_i b^1_{l,i} x_i + \sum_i b^2_{l,i} x^2_i + \sum_{\substack{i,j \\ i \neq j}} b^3_{l,ij} x_i x_j \tag{15}$$

$$g^l_4(x_1, ..., x_m|\theta^l_{\text{cons}}) = b^0_l + \sum_i b^1_{l,i} x_i + \sum_{\substack{i,j \\ i \neq j}} b^3_{l,ij} x_i x_j \tag{16}$$

https://doi.org/10.3390/a19010046

A question arises regarding how to choose the value of the parameter $\mathcal{I}$. This will become clear once we have discussed model training in terms of the optimisation task. Let us take a step back and consider the parameters of the model and its training. If we assume that $\mathcal{I}$ is set to one, the parameter vector of the l^{th} rule is $\theta^l_{\text{cons}} = (b^0_l, b^1_{l,i})_{i=1\ldots m}$, containing the coefficients of the zero and first orders. If $\mathcal{I}$ is set to two, then $\theta^l_{\text{cons}} = (b^0_l, b^1_{l,i}, b^2_{l,i})_{i=1\ldots m}$, i.e., the zero, first and second order coefficients. This continues for the other potential functionals as well, based on the value assigned to $\mathcal{I}$. The set of the model's consequent parameters is $\theta = \bigcup_{l=1:\mathbb{M}} \theta^l_{\text{cons}}$. The consequents' parameters computation is reflected in the minimisation of the matrix equation, given by Equation (17).

$$\min_{\theta} \|\mathbb{A}\theta - y\|^2 + \lambda\|\theta\|^2 \tag{17}$$

which arises through the use of training observations. The parameter $\lambda \in \mathcal{R}^+$ is a regularising parameter introduced for computational stability on ill-posed problems and robustness to overfitting.

To this end, one can generate uniform fuzzy partitions on each $\mathcal{X}_i$, by specifying the cores and standard deviations of the membership functions and determining θ by solving Equation (17), given a predefined regularisation parameter. When the consequents are selected to be linear, this approach can be regarded as a vanilla version of the Takagi–Sugeno–Kang model, where the focus is solely on tuning the consequents' parameters. If nMFs number of membership functions is considered for m inputs, then a TSK model with linear consequents has a total of $2(\text{nMFs})m + (m+1)(\text{nMFs})^m$ parameters. TSK fuzzy models can, in practice, achieve, under certain conditions, comparative accuracy with other modeling approaches, such as radial basis functions, mixture of experts, and ensemble regression models [47].

3.2. The Optimisation Scheme

To date, the tunable parameters of the model are primarily associated with the consequents. However, a refinement is important to enhance the model's predictive performance. This refinement encompasses the parameters of the antecedents, as they play a vital role in generating the fuzzy partitions for the inputs. Consequently, the set of adjustable model parameters is augmented with each membership function's parameters, which in this case are the core and standard deviations.

Let $\mathbb{P}$ represent a newly defined set containing all parameters associated with both consequents and antecedents. Let us introduce a further notation θ_{ant}, including all the cores and standard deviations of the membership functions. To formally address the problem, we sought to identify the optimal set of parameters θ^*_{ant} emerging by solving the problem described in Equation (18).

$$\theta^*_{\text{ant}} = \arg\min_{\theta_{\text{ant}}} \mathbb{E}\Big[\mathcal{L}(y_{\text{TSK}}((\mathcal{X}^{tr}, y^{tr})|\mathbb{P}), y^{tr})\Big] \tag{18}$$

where $\mathcal{L}$ is the loss function, measuring the model's predictive performance, e.g., a squared-loss, an absolute-loss, etc. Furthermore, y_{TSK} is the model's output, trained on $\mathcal{D}^{tr}$ and parameterised by $\mathbb{P}$. This is equivalent to minimising the average training loss, given by Equation (19) [48].

$$\frac{1}{|\mathcal{T}^{tr}|} \sum_{t \in \mathcal{T}^{tr}} \mathcal{L}\left[y_{\text{TSK}}((\mathcal{X}^{tr}, y^{tr})_t|\mathbb{P}), y^{tr}_t\right] \tag{19}$$

where $|\mathcal{T}^{tr}|$ is the cardinality of training observations' set.

https://doi.org/10.3390/a19010046

To do so, a hybrid learning algorithm is considered; a least-squares approach to compute parameters θ, and an adaptive algorithm to optimise the cores and standard deviations of the antecedents' membership functions. Two optimisation algorithms have been studied and proposed: a modern/adaptive version of differential evolution and Bayesian optimisation.

3.2.1. Adaptive Differential Evolution Approach

Differential evolution [49] is considered one of the most powerful evolutionary optimisation algorithms. Compared to other EAs, differential evolution is parameterised by a minimum number of control parameters; the mutation scaling F, the crossover probability p_{cr}, and the population size n_p. One of the most significant strengths of differential evolution is the so-called contour matching [50]. This involves adapting the population to ensure that once promising regions of the search landscape are identified, they are automatically explored, maximising the efficiency of the search.

To perform the optimisation of the predictive model, the contemporary variation in differential evolution, proposed in [51], is adopted. Considering that the performance of differential evolution depends on the type of mutation and crossover method, along with their associated control parameters [50,52,53], the elite guidance mechanism, along with the dual mutation scheme, and adaptive parameters, are beneficial for the optimisation task.

In the proposed predictive methodology, the set of parameters to be optimised consists of the cores and standard deviations of each input fuzzy partition's membership functions. For a fully automated optimisation scheme, we have considered augmenting the set of antecedents' parameters with the regularisation parameter λ and the parameter $\mathcal{I}$ which specifies the functional form of the model's rule consequents. This new set is denoted as ϑ. It should be clarified that ϑ refers to the optimisable set formed by gathering each membership function's parameters $(c_{l,i}, \sigma_{l,i})$, the regularisation parameter λ, and $\mathcal{I}$. By including the parameter $\mathcal{I}$ in the optimisation process, it becomes mixed-integer, since this parameter takes values in $\mathbb{Z}$.

The candidate solutions are represented by a population of individuals, referred to as Θ_i. For every generation g, the population of the parameters is illustrated as the matrix $\mathcal{P}^g$, shown in Equation (20).

$$\mathcal{P}^g = \{\Theta_i^g\}_{i=1\ldots n_p} \in \mathcal{R}^{M \times n_p} \tag{20}$$

where $\Theta_i^g \in \mathcal{R}^M$. Each column of the above-defined matrix represents the set of the model's parameters, composing the optimisable set. Parameter M stands for the search space dimension, and is computed by $M = 2m(\text{nMFs}) + 2$, in which m is the input dimension and nMFs is the number of membership functions on each dimension (We assumed that each input is fuzzified using an equal number of membership functions). In the representation used the last two entries of Θ_i^g, $\forall i$, correspond to the regularisation and $\mathcal{I}$ parameters, i.e., $\vartheta_{M-1}^i = \lambda$ and $\vartheta_M^i = \mathcal{I}$.

A concise analysis of the optimisation algorithm [51] follows. To develop an algorithm that performs on both exploration and exploitation levels is not a trivial task [52]. To improve this trade-off between exploring and exploiting, the method suggests a synergy of elite and non-elite members among $\Theta_i^g \in \mathcal{P}^g$ and an adaptive mutation mechanism. In each generation g the individuals of $\mathcal{P}^g$ are separated into two groups according to their fitness: the elite population group $\mathcal{EP}^g$ and $\mathcal{NEP}^g$, such that $\mathcal{EP}^g \cup \mathcal{NEP}^g = \mathcal{P}^g$. The elite population consist of $n_{\mathcal{EP}^g}$ individuals.

https://doi.org/10.3390/a19010046

The generating process of mutant vectors v_i^{g+1} is governed by Equation (21), and it is based on a dual scheme.

$$v_i^{g+1} = \begin{cases} \Theta_{r_1}^g + \mathrm{F}(\Theta_{r_2}^g - \bar{\Theta}_{r_3}^g) + \mathrm{F}(\Theta_{r_4}^g - \bar{\Theta}_{r_5}^g) & \text{if } r_{[0,1]} \leq MP_g \\ \Theta_{\text{best}}^g + \mathrm{F}(\Theta_{r_2}^g - \bar{\Theta}_{r_3}^g) + \mathrm{F}(\Theta_{r_4}^g - \bar{\Theta}_{r_5}^g) & \text{otherwise} \end{cases} \tag{21}$$

where $i = 1, ..., n_p$, and $\Theta_{r_1}, \Theta_{r_2}, \Theta_{r_4}$ are picked from $\mathcal{EP}^g$ with $r_1, r_2, r_4 \in \{1, 2, ..., n_{\mathcal{EP}^g}\} \setminus \{i\}$, all mutually different, while $\bar{\Theta}_{r_3}, \bar{\Theta}_{r_5}$ are selected from $\mathcal{NEP}^g$, also mutually different. The number $r_{[0,1]}$ is generated from sampling the uniform distribution on $[0, 1]$. Vector Θ_{best}^g denotes the individual displaying the best fitness in the current generation.

Furthermore, $MP_g : \mathbb{Z}^+ \to [0.5, 1]$, plays the role of selection probability for the mutation operator, indicating whether the algorithm will perform as DE/rand/2 or DE/best/2. Figure 2 visually illustrates the idea. Note that the right axis corresponds to the random number generation.

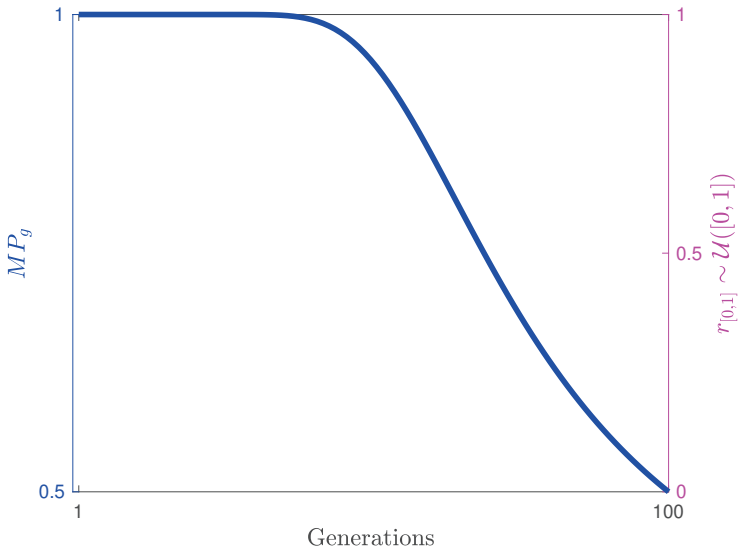


Figure 2. Selection probability function MP_g for the dual mutation scheme. The maximum number of generations on the figure is 100.

The total number of search directions per generation in the search space is given by Equation (22) [54], and it is approximately:

$$\mathcal{O}(n_p^{2n_v}) \tag{22}$$

where n_v is the number of differentials used to generate the mutation vector. Thus, by including two differentials into the differential evolution algorithm, the number of search directions is increased. The crossover operator (Equation (23)), which is utilised to produce the trial vectors u_i^{g+1}, is described as:

$$u_{ji}^{g+1} = \begin{cases} v_{ji}^{g+1} & \text{if } r_{[0,1]} < p_{\text{cr}} \text{ or } j = j_{\text{rand}} \\ \vartheta_{ji}^g & \text{otherwise} \end{cases} \tag{23}$$

where again $r_{[0,1]} \sim \mathcal{U}([0, 1])$, and the integer number j_{rand} is randomly picked from $[0, M]$. In the incorporated differential evolution variant, proposed by Li et al. [51], adaptive mutation scaling and probability of crossover are incorporated, based on history learning.

https://doi.org/10.3390/a19010046

The adaptation modification concerns assigning different control parameters for each individual of the population, as described by Equations (24) and (25).

$$\mathrm{F}_i^{g+1} = \begin{cases} \mathrm{F}_i^g & \text{if } \mathrm{NES}_i < \mathrm{MNES} \\ \sim \mathcal{U}([\mathrm{F}^{\min}, \mathrm{F}^{\max}]) & \text{otherwise} \end{cases} \tag{24}$$

$$p_{\mathrm{cr},i}^{g+1} = \begin{cases} p_{\mathrm{cr},i}^g & \text{if } \mathrm{NES}_i < \mathrm{MNES} \\ \sim \mathcal{U}([p_{\mathrm{cr}}^{\min}, p_{\mathrm{cr}}^{\max}]) & \text{otherwise} \end{cases} \tag{25}$$

where MNES refers to the maximum number of evolution stagnation, and NES_i the number of evolution stagnation of the i^{th} individual. Thus, the algorithm monitors each individual's evolution history. If on a subsequent generation the i^{th} individual is better than its corresponding one of the previous generation, then NES_i is set to zero. Otherwise, it means that the variation operators failed to generate a superior individual, hence NES_i should be increased. If NES_i exceeds the maximum number of evolution stagnation, then the control parameters are adapted by sampling the uniform distribution within their bound limits.

The pseudocode is given in Algorithm 1.

Algorithm 1 Adaptive evolutionary computation for optimisation

Given $\mathcal{E}(x_1 ... x_M)$, i.e., the objective function
task minimise $\mathcal{E}(x_1 ... x_M)$ with respect to $(x_1 ... x_M)$
define a_i, b_i, lower and upper bounds for x_i, n_p, MaxGen, percentage of n_p members in the elite population, MP_g, $p_{\mathrm{cr}}^{\min}$, $p_{\mathrm{cr}}^{\max}$, $\mathrm{F}^{\min}$, $\mathrm{F}^{\max}$, MNES
$g \leftarrow 1$
$\mathrm{BestObj}^g \leftarrow \infty$
generate initial population of candidate solutions
$\mathcal{P}^g \leftarrow \mathrm{INITIALISE}(a, b, M, n_p)$
$\{\mathcal{EP}^g, \mathcal{NEP}^g, \mathrm{BestObj}^g\} \leftarrow \mathrm{COMPOBJ}(\mathcal{P}^g)$
initialise control parameters and number of evolution stagnation
$\mathrm{F}_i^g \leftarrow 0.5$, $p_{\mathrm{cr},i}^g \leftarrow 0.9$, and $\mathrm{NES}_i^g \leftarrow 0$ for all individuals in $\mathcal{P}^g$
for $g = 2$ to MaxGen **do**
    for all individuals in $\mathcal{P}^g$ **do**
        $\mathcal{P}_{\mathrm{mutation}}^g \leftarrow \mathrm{MUTATE}(\mathcal{EP}^g, \mathcal{NEP}^g, MP_g, \mathrm{F}_i^g)$
        if bound constraints are violated **then**
            $\mathcal{P}_{\mathrm{mutation}}^g \leftarrow \mathrm{BOUNDS}(\mathcal{P}_{\mathrm{mutation}}^g, a_i, b_i)$
        end if
        $\mathcal{P}_{\mathrm{trial}}^g \leftarrow \mathrm{CROSSOVER}(\mathcal{P}^g, \mathcal{P}_{\mathrm{mutation}}^g, p_{\mathrm{cr,i}}^g)$
    end for
    $\{\mathrm{BestObj}^g, \mathcal{P}^g\} \leftarrow \mathrm{SELECTION}(\mathcal{P}^{g-1}, \mathcal{P}_{\mathrm{trial}}^g)$
    $\{\mathcal{EP}^g, \mathcal{NEP}^g, \mathrm{BestObj}^g\} \leftarrow \mathrm{COMPOBJ}(\mathcal{P}^g)$
    for all individuals in $\mathcal{P}^g$ **do**
        $\{\mathrm{F}_i^g, p_{\mathrm{cr},i}^g\} \leftarrow \mathrm{ADAPTATION}(\mathcal{P}^{g-1}, \mathcal{P}^g, \mathrm{NES}_i^g, \mathrm{MNES}, \mathrm{F}^{\min}, \mathrm{F}^{\max}, p_{\mathrm{cr}}^{\min}, p_{\mathrm{cr}}^{\max}, \mathrm{F}_i^{g-1}, p_{\mathrm{cr},i}^{g-1})$
    end for
end for

3.2.2. Bayesian Optimisation Approach

The evolutionary algorithm, although it provides accurate results, can be time-consuming. This is due to its fundamental principle, which lies in the evolution of a candidate solution's population. We were interested in both predictive accuracy and training efficiency; therefore, Bayesian optimisation seemed like a natural choice. The goal is to

https://doi.org/10.3390/a19010046

create a surrogate model, e.g., a Gaussian process, for the unknown objective function that evaluates the fuzzy model's predictive performance. This will enable the surrogate model to understand the landscape of the objective function we need to minimise. In Bayesian optimisation, two considerations should be considered: the kernel function and the acquisition function. The former dictates the form of functions that the Gaussian process can learn, while the latter offers a metric for determining which point should be next evaluated.

The Matérn family of kernels commonly used in machine learning includes the Matérn 3/2 kernel and the Matérn 5/2 kernel [55]. In the developed framework, the covariance matrix is computed using an automatic relevance determination Matérn 5/2 kernel, which is described by Equation (26).

$$\mathcal{K}_{\text{Matérn } 5/2}(x, x') = \varsigma\Big(1 + \frac{\sqrt{5}|x - x'|}{L} + \frac{5|x - x'|^2}{3L^2}\Big)\exp\Big(-\frac{\sqrt{5}|x - x'|}{L}\Big) \tag{26}$$

where $\ell = (\ell_1, ..., \ell_M)$, are the lengthscale parameters for each dimension in the search space, and $L = \text{diag}(\ell_1, \ell_2, ..., \ell_M)$. Parameter ς corresponds to the standard deviation. The rationale of choosing this kernel instead of the prevalent squared exponential is based on the fact that the latter generates extremely smooth functions, which may be unrealistic for real-world problems [56]. Furthermore, it has been proven that this kernel performs better in practical optimisation, compared to the squared exponential [57].

The acquisition function α, adopted in the proposed methodology, is known as expected improvement. We formalise the problem, following the analysis in [58]. Let the optimisation based on the Bayesian framework be described by a sequence of steps, and consider the n^{th} step. Thus, the optimisation has collected a set $\mathcal{D}^{\vartheta} = (\vartheta_i, \mathcal{E}(\vartheta_i))$, where $i = 1 \dots n$, where n is the maximum number of evaluations. Regarding the notation, $\mathcal{D}^{\vartheta}$ denotes the data set of the fuzzy model's parameters, which consists of the antecedents, the regularisation, and $\mathcal{I}$ parameters. Moreover, $\mathcal{E}$ denotes the objective function to be minimised, parameterised by ϑ. Note that this set is augmented at each iteration with the new parameters to be explored. The surrogate model, which approximates the objective function $\mathcal{E}$, admits a posterior distribution $p(\mathcal{E}|\mathcal{D}_n^{\vartheta}) = \mathcal{GP}(\mathcal{E}; \mu, \mathcal{K})$, where μ and $\mathcal{K}$ are the posterior mean and posterior covariance functions, respectively. Let η be the lowest objective function value available at the n^{th} step, given by Equation (27).

$$\eta = \min_{i \in \{1 \dots n-1\}} \mathcal{E}(\vartheta_i) \tag{27}$$

Consider the loss described by Equation (28).

$$\mathcal{L}_{\text{EI}}(\mathcal{D}_{n+1}^{\vartheta}) = \min\{\eta, \mathcal{E}(\vartheta_n)\} \tag{28}$$

The expectation of the loss, under the $\mathcal{GP}$ posterior, is:

$$\mathbb{E}\big(\mathcal{L}_{\text{EI}}(\mathcal{D}_{n+1}^{\vartheta})\big) = \int \min\{\eta, \mathcal{E}(\vartheta_n)\} p(\mathcal{E}(\vartheta_n)|\mathcal{D}_n^{\vartheta}) \partial\mathcal{E}(\vartheta_n) \tag{29}$$

$$= \eta + \int \min\{0, \mathcal{E}(\vartheta_n) - \eta\} p(\mathcal{E}(\vartheta_n)|\mathcal{D}_n^{\vartheta}) \partial\mathcal{E}(\vartheta_n) \tag{30}$$

$$\begin{aligned} = \eta &+ \int_{-\infty}^{\eta} (\mathcal{E}(\vartheta_n) - \eta) p(\mathcal{E}(\vartheta_n)|\mathcal{D}_n^{\vartheta}) \partial\mathcal{E}(\vartheta_n) \\ &+ \int_{\eta}^{\infty} 0 \times p(\mathcal{E}(\vartheta_n)|\mathcal{D}_n^{\vartheta}) \partial\mathcal{E}(\vartheta_n) \end{aligned} \tag{31}$$

$$= \eta + \int_{-\infty}^{\eta} (\mathcal{E}(\vartheta_n) - \eta) p(\mathcal{E}(\vartheta_n)|\mathcal{D}_n^{\vartheta}) \partial\mathcal{E}(\vartheta_n) \tag{32}$$

https://doi.org/10.3390/a19010046

Thus, the expectation of the loss is associated with the expected improvement over the current best, which is η. The search space location, on where the objective is to be sampled is given by the minimiser of $\mathbb{E}\big(\mathcal{L}_{\mathrm{EI}}(\vartheta_n)\big)$, which is essentially the minimiser of the expected improvement acquisition function defined in Equation (33).

$$\alpha_{\mathrm{EI}}(\vartheta_n) = \int_{-\infty}^{\eta} (\mathcal{E}(\vartheta_n) - \eta) p(\mathcal{E}(\vartheta_n)|\mathcal{D}_n^{\vartheta}) \partial \mathcal{E}(\vartheta_n) \tag{33}$$

Taking into account that $p(\mathcal{E}(\vartheta_n)|\mathcal{D}_n^{\vartheta}) = \mathcal{N}(\mathcal{E}(\vartheta_n); \mu(\vartheta_n), \mathcal{K}(\vartheta_n))$, and Φ denotes the cumulative distribution function of the Gaussian distribution, the acquisition function can be expressed analytically as:

$$\begin{aligned} \alpha_{\mathrm{EI}}(\vartheta_n) = &-\mathcal{K}(\vartheta_n)\mathcal{N}(\eta; \mu(\vartheta_n), \mathcal{K}(\vartheta_n)) \\ &+ (\mu(\vartheta_n) - \eta)\Phi(\eta; \mu(\vartheta_n), \mathcal{K}(\vartheta_n)) \end{aligned} \tag{34}$$

The latter expression illustrates that when the posterior mean is low, or the posterior covariance is large, the acquisition function decreases, indicating highly promising regions for the optimiser to move within the search space. This relationship highlights the importance of the posterior mean in driving exploitation, while the posterior covariance fosters essential exploration.

A pseudocode of Bayesian optimisation is presented in Algorithm 2.

Algorithm 2 Bayesian optimisation algorithm

Given objective function $\mathcal{E}$, $\mathcal{GP}$ kernel function $\mathcal{K}$, acquisition function α and numIterations
$\mathcal{D}_i^{\vartheta} \leftarrow \emptyset, i \leftarrow 1$
while $i \leq$ numIterations **do**
  if i=1 **then**
    initialise: $\vartheta_i \leftarrow \mathcal{U}, y_i \leftarrow \mathcal{E}(\vartheta_i)$ ▷ *Compute $\mathcal{E}$ on a uniformly random feasible point*
    $\mathcal{D}_i^{\vartheta} = (\vartheta_i, y_i)$
    $\mathcal{GP} \leftarrow \mathrm{FITGP}(\mathcal{K}, \mathcal{D}_i^{\vartheta})$
    store the current minimum
    $\vartheta^* \leftarrow \vartheta_i, y^* \leftarrow y_i$
  else
    $\vartheta_i \leftarrow \arg\min \alpha(\vartheta_i, \mathcal{GP})$
    $y_i \leftarrow \mathcal{E}(\vartheta_i)$
    if $y_i < y^*$ **then**
      store the new minimum
      $\vartheta^* \leftarrow \vartheta_i, y^* \leftarrow y_i$
    end if
  end if
  $\mathcal{D}_i^{\vartheta} \leftarrow \mathcal{D}_i^{\vartheta} \cup (\vartheta_i, \mathcal{E}(\vartheta_i))$
  refit the $\mathcal{GP}$ **model**: $\mathcal{GP} \leftarrow \mathrm{FITGP}(\mathcal{K}, \mathcal{D}_i^{\vartheta})$
  $i \leftarrow i + 1$
end while
return Best parameters ϑ^*

3.3. The Feature Selection Scheme

Choosing the *best* number of lags is crucial; it refers to the specific count that minimises the prediction error of the models on unseen data, by generating a particular input space. The wrapper/sequential method utilised in this study, as a method of selecting the best number of lags, generating the input space $\mathcal{X}$, is given in Algorithm 3.

https://doi.org/10.3390/a19010046

The custom algorithm utilised a vanilla VMD-TSK fuzzy model as the predictive model. We considered a candidate lag space ranging from 1 to 20. Since fuzzy systems have difficulty dealing with high-dimensional data, the maximum number of features—specifically, the maximum number of inputs to the models—was limited to five [59].

Two inputs were assumed as the starting point, i.e., the first two lags that will generate the input space will be selected by training $\binom{20}{2} = 190$ models, and monitoring their performance. The function COMBINE creates unique combinations of lags from the first two argument sequences, e.g., $\{1,2\},...,\{1,20\},...,\{2,3\},...,\{2,20\},...,\{19,20\}$. Furthermore, it appropriately increases the model counter. The function GENERATE created candidate feature/input spaces $\mathfrak{X}^i$, given observations and lag combinations. The computation of the best lag combination and the associated input space was utilised via the ARGMIN function. The developed wrapper-based sequential algorithm is provided in Algorithm 3.

Algorithm 3 Wrapper/Sequential algorithm for lag selection

Given the fuzzy model, time series data x, maxNumFeatures, and a set of candidate lags $\{1,...,20\}$
$S \leftarrow \emptyset$ ▷ *initialise the set of features*
maxNumFeatures $\leftarrow 5$
for $k = 1$ to maxNumFeatures **do**
    group$\{k\} \leftarrow \{1,...,20\}$
end for
model $\leftarrow 1$
lags(model) $\leftarrow$ COMBINE(group$\{1\}$, group$\{2\}$, model)
$\mathbb{X} = \{\mathfrak{X}^i\} \leftarrow$ GENERATE(x, lags(model))
for all $\mathfrak{X}^i \in \mathbb{X}$ **do**
    Train the model on $\mathfrak{X}^i$ and obtain its predictive performance on test data, e.g., create matrix $\mathcal{E}_{\text{RMSE}^*}$,
end for
$\{\mathcal{E}^{\text{ind}}_{\text{rmse}^*}, S, \text{ind}\} \leftarrow \text{ARGMIN}(\mathcal{E}_{\text{RMSE}^*})$ ▷ *update the set of features S, the corresponding lags ind and save the best observed metric*
for $k = 3$ to maxNumFeatures **do**
    group$\{k\} \leftarrow \{1,...,20\} \setminus \{\text{ind}\}$
    lags(model) $\leftarrow$ COMBINE(ind, group$\{k\}$, model)
    $\mathbb{X}_{\text{new}} = \{\mathfrak{X}^i_{\text{new}}\} \leftarrow$ GENERATE(x, lags(model))
    Train the model on the generated candidate sets $\mathfrak{X}^i_{\text{new}}$ and compute $\mathcal{E}^{\text{new}}_{\text{RMSE}^*}$
    $\{\mathcal{E}^{\text{new}}_{\text{RMSE}^*}, S^{\text{new}}, \text{ind}_{\text{new}}\} \leftarrow \text{ARGMIN}(\mathcal{E}^{\text{new}}_{\text{RMSE}^*})$
    if $\exists\ \text{ind}_{\text{new}}$ such that $\mathcal{E}^{\text{new}}_{\text{RMSE}^*} < \mathcal{E}_{\text{RMSE}^*}$ **then**
        ind $\leftarrow$ ind $\cup$ ind$_{\text{new}}$
        $\mathcal{E}_{\text{rmse}^*} \leftarrow \mathcal{E}^{\text{new}}_{\text{rmse}^*}$
        $S \leftarrow S_{\text{new}}$
    end if
end for

3.4. The Complete Methodology

The proposed methodology for predicting wind speed can be summarised in the following steps:

- **Data pre-processing**: We began by addressing any missing values in the time series data and subsequently rescaled the dataset within the range of $[0,1]$.
- **Feature selection**: We implemented the sequential/wrapper-based algorithm to identify the most effective lags for constructing the feature space of the models.
- **Data decomposition**: We utilised variational mode decomposition to break down the data into several mode functions.

https://doi.org/10.3390/a19010046

- **Train the models**: For each mode function, we generated feature spaces and trained the models using the corresponding data. The optimisation task has been utilised in terms of the evolutionary or the Bayesian optimisation algorithms.
- **Test the models**: Predictions on both training and test datasets were generated by aggregating the predictions derived from the models associated with each mode function. We then assessed the predictive performance using regression metrics.
- **Comparisons**: We evaluated the predictive performance of the proposed model in relation to other machine learning models, using their respective predictions for comparison. Finally, the Diebold-Mariano statistical test has been performed.

A flowchart of the proposed methodology, concluding the latter mentioned, is illustrated in Figure 3.

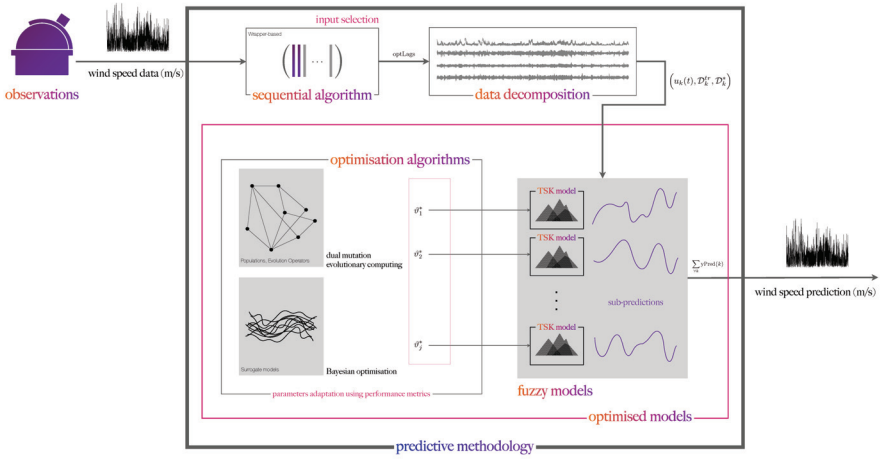


Figure 3. Flowchart of the proposed predictive scheme.

4. Numerical Studies

The numerical studies feature the proposed evolutionary and Bayesian optimised fuzzy models, alongside comparisons with ANFIS, WM fuzzy inference system, and a vanilla TSK. Additionally, we have included a comparison with a powerful automated methodology for selecting and training machine learning models. The Wang-Mendel model is a quite interpretable fuzzy inference system of the Mamdani class. The vanilla version of the Takagi–Sugeno–Kang model utilised uniform fuzzy partitions along each input dimension. Both of these models demonstrate high interpretability due to their rule-based nature and the semantic properties at a fuzzy partition level. ANFIS has been included as a comparison model due to its excellent performance in regression problems. Consequently, the automated methodology for generating machine learning models is highly novel, yielding a blend of models that achieve the highest observed performance. This strategy is preferable to relying on a single machine learning model when evaluating the proposed optimised predictive models. Therefore, the comparison models represent a wide group of predictive models, from simple (non-optimised) to fully optimised models.

All experiments focused on one-step-ahead predictions, meaning that based on past observations, the models generated a forecast of the next wind speed realisation.

4.1. Data Analysis

The wind speed data, provided by the National Observatory of Athens, Greece, pertain to the year 2019 and do not correspond to a typical reference year. This choice is intentional; a typical year is designed to represent average conditions, not actual weather. It is smoother

https://doi.org/10.3390/a19010046

than the actual year; hence, it fails to capture noise and extremes. The main goal was to develop and test models on a highly variable dataset; thus, wind speed data from an actual year serves this purpose. The data were organised in a table format, with each column representing the year, month, day, hour, and wind speed in meters per second (m/s).

The sampling was conducted hourly, meaning that every hour we obtained a single wind speed observation. The statistical characteristics of the data under study are given in Table 1. The distribution of the wind speed data is right-skewed, since there is a long tail on the right side. This is further confirmed by the mean being larger than the median of the data, and additionally by the fact that wind speed data follow a Weibull distribution.

Table 1. Statistical characteristics of the wind speed data.

Characteristic	Value
Mean	3.2188
Standard Deviation	2.2892
Kurtosis	4.2813
Skewness	1.2023
Median	2.5365

A visual representation of the data follows. Figure 4 demonstrates the temporal evolution of wind speed. The left figure illustrates the annual fluctuations in wind speed for the year 2019, while the one on the right focuses on selected two-month intervals (from 1 January to 28 February), providing a detailed examination of the wind's variable nature. Moreover, a histogram and a parallel plot are featured in Figure 5 to enhance the understanding of the dataset.

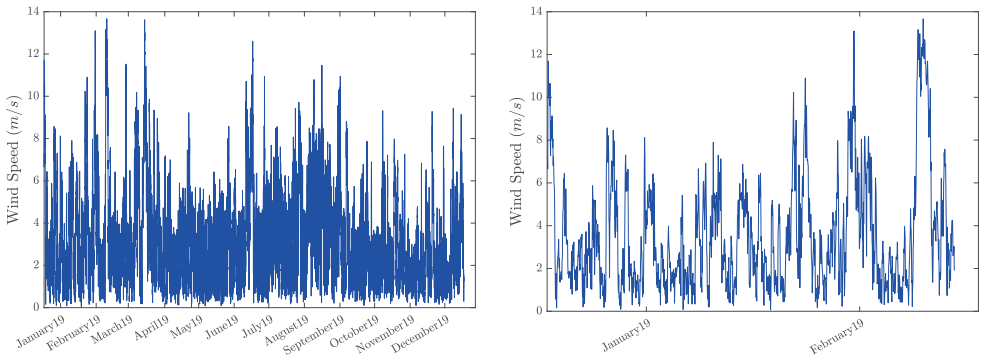


Figure 4. Temporal evolution of wind speed for the year 2019. Note that every *month-tick* on the horizontal axis corresponds to the middle of each month, not its beginning.

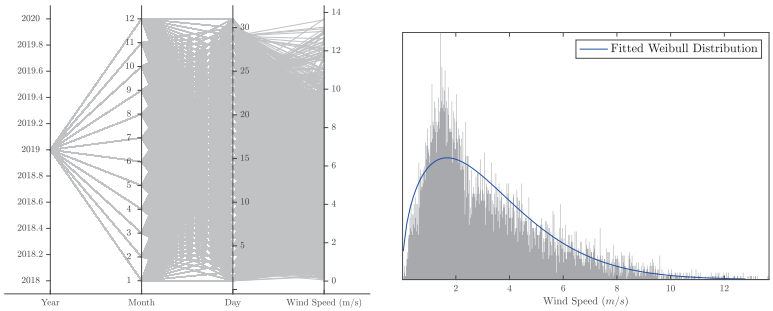


Figure 5. Parallel plot and histogram with a distribution fit for the wind speed data.

https://doi.org/10.3390/a19010046

Regarding pre-processing, the missing values were replaced using linear interpolation. It is important to note that the potential presence of noise in the data has not been examined. In other words, the data were treated as noiseless, despite their high variability. We believe that this approach effectively tests the models' ability in the prediction of highly volatile data. Finally, to train the models, the data were rescaled into $[0, 1]$.

4.2. Performance Metrics

The quantification of the developed models' performance was reflected in the face of two metrics: RMSE and MAPE, described by Equations (35) and (36), respectively.

$$\mathcal{E}_{\text{RMSE}} = \left[\frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \left(y(t) - y_{\text{model}}(t) \right)^2 \right]^{\frac{1}{2}} \tag{35}$$

$$\mathcal{E}_{\text{MAPE}} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \left| \frac{y(t) - y_{\text{model}}(t)}{y(t)} \right| \tag{36}$$

where $y(t)$ is the target of the dataset $\mathcal{D}$, $|\mathcal{T}|$ is the number of examples, and $y_{\text{model}}(t)$ represents the output of the prediction model. Models with lower $\mathcal{E}_{\text{RMSE}}$ result in more accurate predictions. $\mathcal{E}_{\text{MAPE}}$ metric is used because it is independent of scale. Additionally, a percentage-based performance forecast provides readers with better insights into the models.

The coefficient of determination, given by Equation (37), has also been considered. It provides a descriptive statistic that measures the proportion of the variance of the dependent variable $y(t)$ that is predictable from the independent variables. It is given by:

$$r^2 = 1 - \frac{\sum_{t \in \mathcal{T}} \left(y(t) - y_{\text{model}}(t) \right)}{\sum_{t \in \mathcal{T}} \left(y(t) - \bar{y}(t) \right)} \tag{37}$$

where $\bar{y}(t)$ denotes the mean value of the dataset's target. The latter metric indicates the extent to which the output is predictable, i.e., how well the model fits the observed data.

4.3. Data Split

The training observations, as illustrated in Figure 6, constitute 70% of the total data set, while the remaining 30% is reserved for testing, allowing for effective validation and evaluation of performance on unseen data. The training data begin on 1 January 2019, and end on the eleventh hour of 13 September 2019.

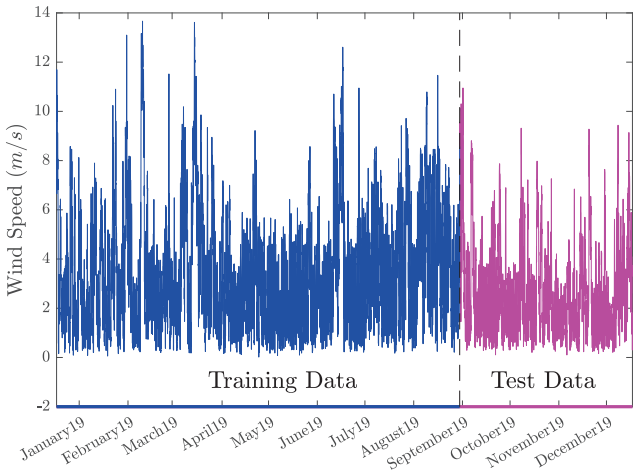


Figure 6. Training $\mathcal{T}^{\text{tr}}$, and testing $\mathcal{T}^*$ time spans for wind speed data.

https://doi.org/10.3390/a19010046

4.4. Input Selection

As far as feature selection is concerned, the wrapper-based algorithm has been employed to identify the time lags that will be used to create the input space for the predictive models. The algorithm is based on a Takagi–Sugeno–Kang model incorporating variational mode decomposition. After rescaling the data into the range $[0, 1]$, they were decomposed using the VMD algorithm into nine intrinsic mode functions.

A vanilla TSK model with linear rule consequents was trained on each of the subseries data, i.e., each intrinsic mode function, and was subsequently tested on the corresponding test set. The candidate lags were utilised to generate the input space of each sub-model, which was trained on the associated subseries of the decomposed data. All of the above-mentioned fuzzy models used uniform fuzzy partitions on all input dimensions and generated rules via grid partitioning. The fuzzy partitions were computed using two Gaussian membership functions. Moreover, product t-norms have been utilised. The regularising parameter λ was set to 0.001.

The selection of the optimal lag was determined by the lag combination that yielded the minimum $\mathcal{E}_{\text{RMSE}}$ on the test set. By applying the feature selection algorithm, a two-dimensional input space, that was generated by the lags $x(t-1)$ and $x(t-2)$ for $t \in \mathcal{T}$, was considered.

The decomposition of the original time series of wind speed data (Not rescaled into $[0, 1]$) is illustrated in Figure 7. The variational mode decomposition algorithm generates several intrinsic mode functions, the number of which is predefined. In this study, a number of nine mode functions have been selected, providing a highly accurate reconstruction of the original time series.

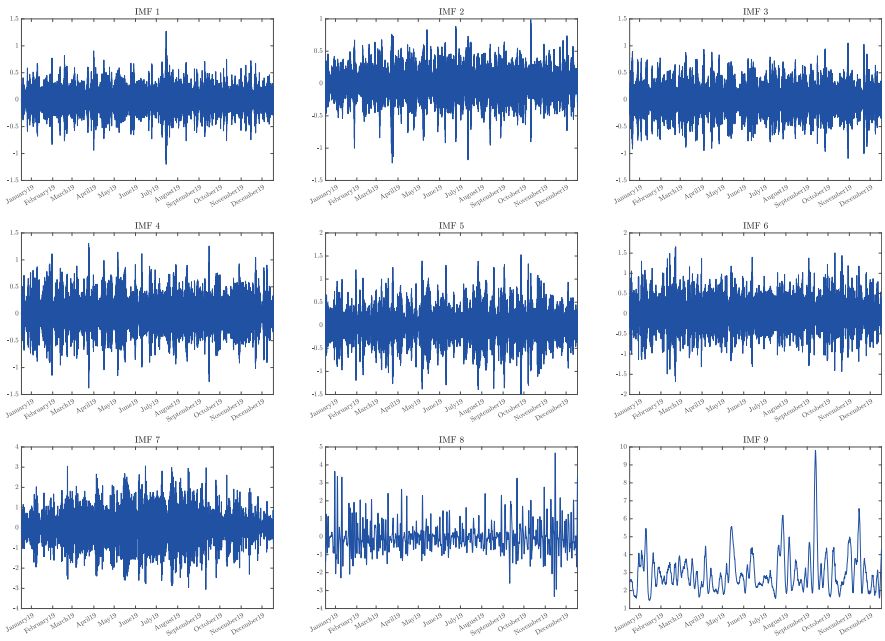


Figure 7. The nine intrinsic functions u_k emerged from the variational mode decomposition of the original time series data.

4.5. Data Decomposition

The core principle of the present methodology involves applying a predictive model to each sub-time series and using the computed lags from the wrapper-based algorithm to form the input space for each sub-model.

https://doi.org/10.3390/a19010046

Algorithm 4 outlines the above-mentioned. Functions SUBDATA, OPTIMISE, and PRED correspond to generating training and testing inputs for each sub-model, optimising each sub-model, and obtaining predictions on unseen data using the optimised sub-model, respectively. The inputs to the algorithm are given by the type of optimiser used, e.g., the adaptive differential evolution or the Bayesian optimisation scheme, and the optLags, which are delivered by the wrapper-based algorithm. The training and testing percentages are necessary for the generation of $\mathcal{D}_k^{\text{tr}}$ and $\mathcal{D}_k^*$, from the k^{th} mode function u_k.

Algorithm 4 VMD-based prediction

Given predictive model (predModel), optimiser `type` (evolutionary or Bayesian), time series data x, maxNumIMFs, optLags, training and testing data percentages, objective function $\mathcal{E}_{\text{RMSE}}$
$u_k \leftarrow \text{VMD}(x, \text{maxNumIMFs})$ ▷ u_k is the k^{th} IMF
$k \leftarrow 1$
while $k \leq$ maxNumIMFs **do**
    $\{\mathcal{D}_k^{\text{tr}}, \mathcal{D}_k^*\} \leftarrow \text{SUBDATA}(u_k, \text{optLags}, 70\%)$
    $\text{optModel}\{k\} \leftarrow \text{OPTIMISE}(\mathcal{D}_k^{\text{tr}}, \texttt{type}, \text{predModel})$
    $\text{yPredTest}\{k\} \leftarrow \text{PRED}(\text{optModel}\{k\}, \mathcal{D}_k^*)$
    $k \leftarrow k + 1$
end while
aggregate the sub-models' prediction
$\text{yPredTest} = \sum_{\forall k} \text{yPredTest}\{k\}$

It is essential to mention that Algorithm 4 is not model-specific; it can be incorporated with an arbitrary choice of predictive model. Moreover, it is acceptable to employ different models for each sub-series, allowing a framework for further experiments.

4.6. Implementation Details

Regarding the implementation details of the models, the following were considered: In the vanilla TSK model, two Gaussian-shaped membership functions were used to granulate the inputs $x(t-1)$ and $x(t-2)$ into uniform fuzzy partitions. The fuzzy rules were generated through grid partitioning, using product t-norms. The regularising parameter was set to 0.001. The consequences of the fuzzy rules were considered to be linear.

For the WM fuzzy inference system, to ensure comparable results, the number of Gaussian membership functions was increased to nine for each input, and these inputs remained identical across all models. The output has also been uniformly fuzzified using nine Gaussian membership functions. The fuzzy connectives were implemented using product t-norms, and Larsen implication was considered. The defuzzification of the aggregated output fuzzy set was carried out using the centroid method. Both vanilla TSK and WM models were trained in a batch mode, i.e., a forward pass of the training data.

Regarding the adaptive-network-based fuzzy inference system, the exact settings used for the vanilla TSK model have been considered. However, a hybrid training method has been employed; a gradient descent algorithm to optimise the parameters of the antecedents, and a least squares approach to compute the parameters of the fuzzy rules' consequents. The number of iterations was set to thirty.

The automated selection method for choosing and training machine learning models was based on ASHA [60]. This method has been implemented using MATLAB's version 2024a fitrauto function. We chose ASHA optimiser since it offered a faster procedure compared to the Bayesian, which can also be utilised with this function. Linear, as well as nonlinear, machine learning models were considered by the latter automated approach, together with their parameter optimisation.

https://doi.org/10.3390/a19010046

In the proposed models, and as far as the evolutionary TSK model was concerned, the following holds: the population size was determined to be five times the problem's dimensionality, which was $M = 2m(\text{nMFS}) + 2 = 10$, since the selection of two inputs ($m = 2$), two membership functions (nMFs $= 2$), each with two optimisable parameters, the regularisation parameter λ and the parameter $\mathcal{I}$, which indicated the form of the fuzzy rules' consequents. The percentage of elite individuals among each $\mathcal{P}^g$ was set to 30% of the population size. With respect to the adaptation scheme for the mutation scaling and the crossover probability, the maximum number of evolution stagnation was set to 3. The initial mutation scaling F_i^0 was set to 0.5 for all individuals in the population. Similarly, the probability of crossover was initialised at 0.9 for every individual in the initial population. Furthermore, the minimum and maximum values for each F_i were set as $\text{F}^{\min} = 0.1$ and $\text{F}^{\max} = 0.5$, respectively. The minimum and maximum values of crossover probability were set to $p_{\text{cr}}^{\min} = 0.5$ and $p_{\text{cr}}^{\max} = 1$. The number of evolution stagnation, i.e., NES_i, was initialised to zero for all individuals in the initial population. The maximum number of generations was set to thirty. All the optimised sub-models feature nonlinear fuzzy rule consequents, which is signified by the value of the parameter $\mathcal{I}$ being three. This was expected since including linear, squared, and cross-product terms of the model's inputs in each fuzzy rule's consequents increases complexity, enhancing the model's capability to achieve more accurate results.

The Bayesian optimised TSK model has been implemented utilising the automatic relevance determination Matérn 5/2 kernel along with the expected improvement acquisition function. The dimensionality of the search space was identical to the evolutionary TSK scheme. The maximum number of optimisation iterations was set to thirty. The outcomes of the Bayesian optimisation demonstrated a high degree of accuracy, closely aligning with the results obtained from the evolutionary TSK model. However, each optimised sub-model featured different forms of fuzzy rule consequents. The Bayesian optimisation scheme did not always favor the most complex type of consequent.

4.7. Results

Tables 2–4 showcase the predictive performance results of the proposed models compared to alternative approaches, for both training and testing data. All methods incorporated variational mode decomposition to ensure accurate approximation results. The performance metric values in the tables refer to average values obtained by running the models several times, since all models, except the vanilla TSK and the Wang-Mendel fuzzy inference system, were trained in an iterative stochastic manner.

In all tables, we have referred to the models as EVOLUTIONARY TSK when adaptive differential evolution optimisation has been used, and as BAYESIAN TSK if the model has been optimised using the surrogate-based algorithm. Moreover, VANILLA TSK refers to the method where the foundational model on each sub-series of the decomposed data was a Takagi–Sugeno–Kang with linear rule consequents and uniform fuzzy partitions. As far as fuzzy inference systems are concerned, a WM FUZZY SYSTEM was considered. Additionally, the adaptive-network-based fuzzy inference system is referred to as ANFIS, and an automated methodology of selecting/optimising machine learning models is referred to as AUTOMATED ML METHOD.

Alongside the tables, a selection of representative plots is presented to visually illustrate the performance of the models. These plots include zoomed-in sections of both the training and testing regions of the time series, regression plots, and error histograms alongside fitted distributions. Figures 8 and 9 are associated with the VANILLA TSK, Figures 10 and 11 with the WM FUZZY SYSTEM, Figures 12 and 13 with ANFIS,

 https://doi.org/10.3390/a19010046

and Figures 14 and 15 are associated with the AUTOMATED ML METHOD. Furthermore, Figures 16 and 17 are associated with the BAYESIAN TSK, while Figures 18 and 19 with the EVOLUTIONARY TSK.

NOTE: The zoomed-in sections, shown in Figures 8, 10, 12, 14, 16 and 18, correspond to, the 5th until the 20th day of May for the training regions, and from the 18th to the 29th day of October for the testing. The labels in the x-axis follow an *hour-day-month* format.

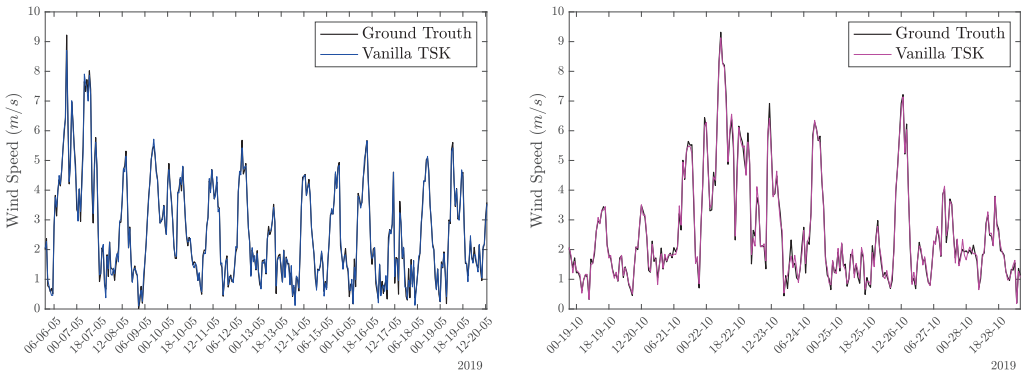


Figure 8. Vanilla TSK prediction results on wind speed data. The (**left**) plot illustrates a snapshot of the approximation on the training data, while the (**right**) plot zooms into the approximation on the test region.

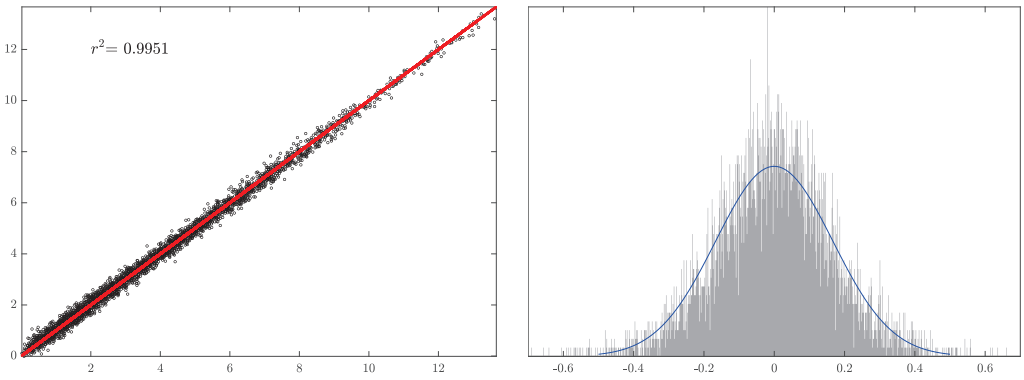


Figure 9. Vanilla TSK prediction results on wind speed data. The (**left**) plot illustrates a regression plot, where the ground truth is highlighted in red colour. The (**right**) plot shows the error histogram, along with a normal density function fit.

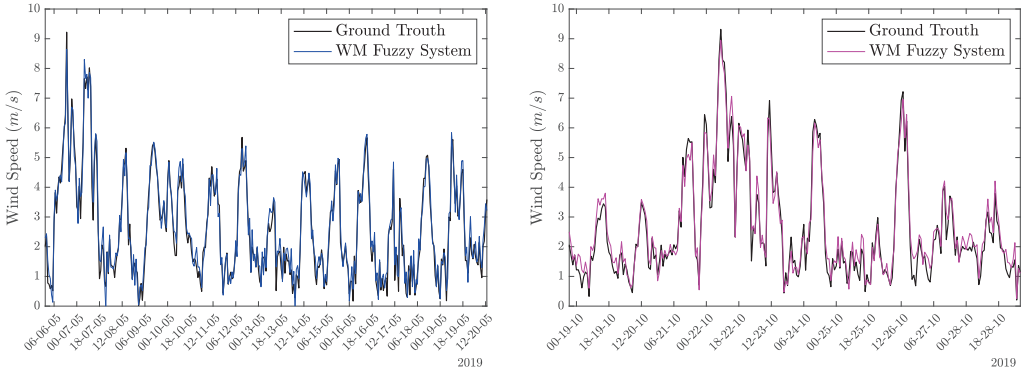


Figure 10. WM fuzzy system prediction results on wind speed data. The (**left**) plot illustrates a snapshot of the approximation on the training data, while the (**right**) plot zooms into the approximation on the test region.

https://doi.org/10.3390/a19010046

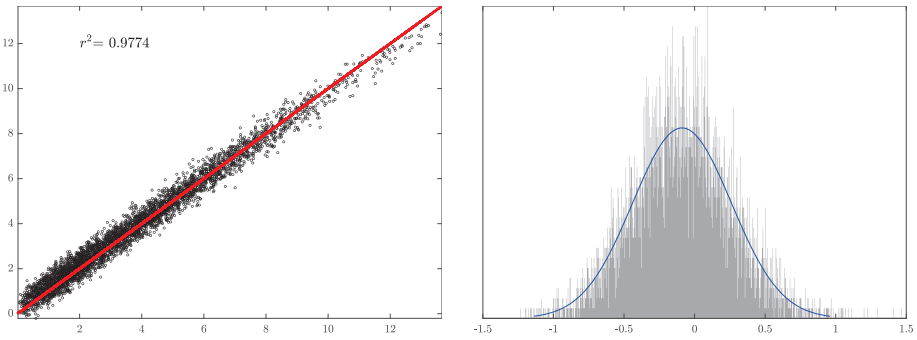


Figure 11. WM fuzzy system prediction results on wind speed data. On the (**left**): regression plot, and on the (**right**): error histogram along with a normal density function fit.

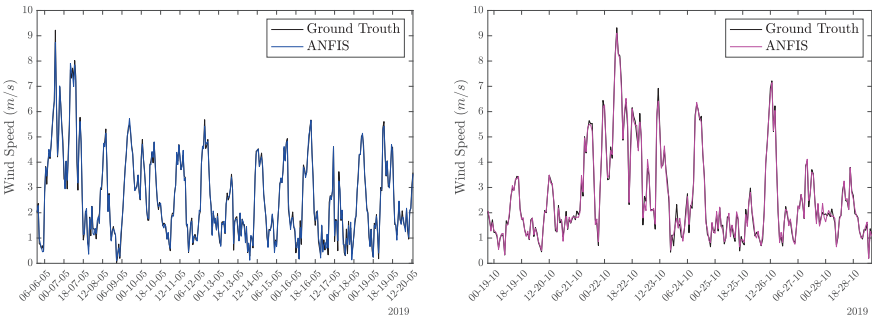


Figure 12. ANFIS prediction results on wind speed data. The (**left**) plot illustrates a snapshot of the approximation on the training data, while the (**right**) plot zooms into the approximation on the test region.

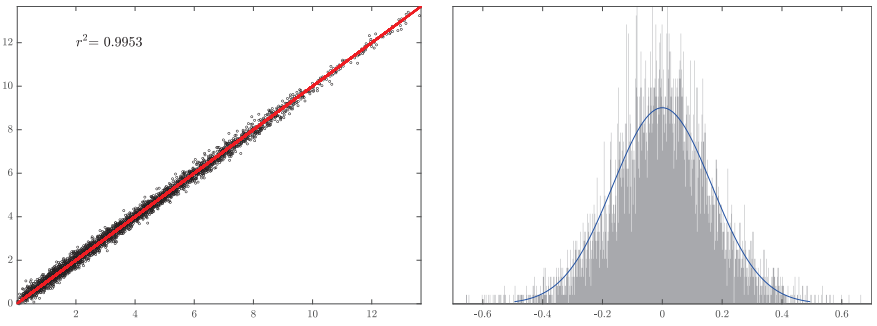


Figure 13. ANFIS prediction results on wind speed data. On the (**left**): regression plot, and on the (**right**): error histogram along with a normal density function fit.

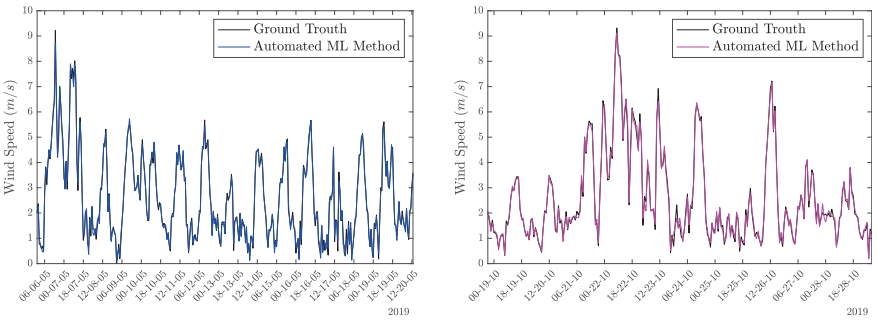


Figure 14. Automated ML method prediction results on wind speed data. The (**left**) plot illustrates a snapshot of the approximation on the training data, while the (**right**) plot zooms into the approximation on the test region.

https://doi.org/10.3390/a19010046

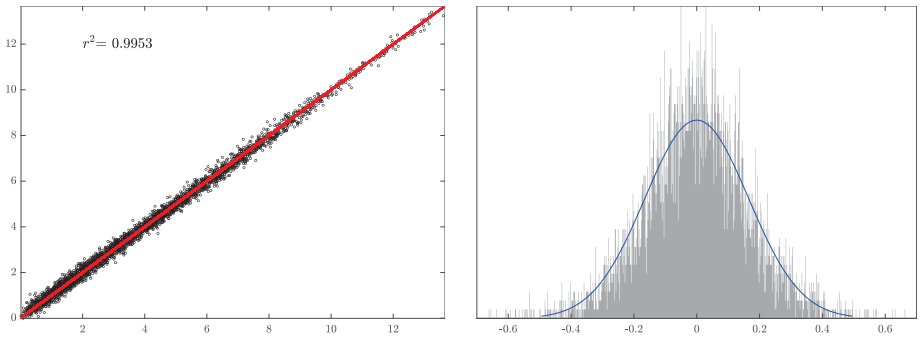


Figure 15. Automated ML method prediction results on wind speed data. On the (**left**): regression plot, and on the (**right**): error histogram along with a normal density function fit.

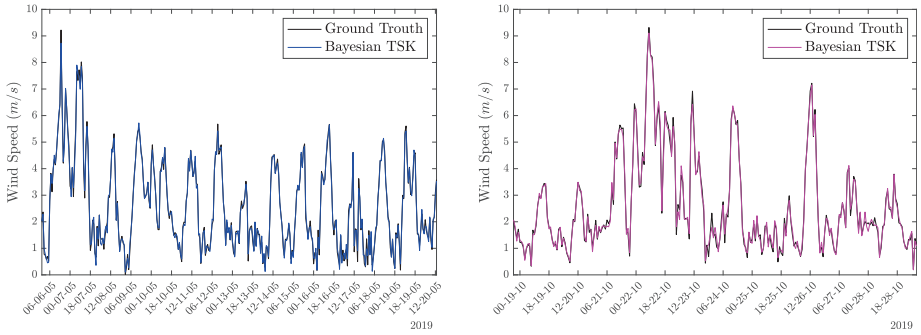


Figure 16. Bayesian TSK method prediction results on wind speed data. The (**left**) plot illustrates a snapshot of the approximation on the training data, while the (**right**) plot zooms into the approximation on the test region.

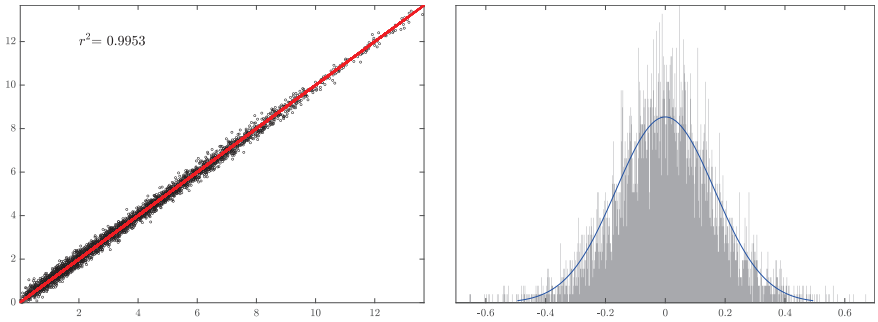


Figure 17. Bayesian TSK prediction results on wind speed data. The (**left**) plot illustrates a regression plot, where the ground truth is highlighted in red colour. The (**right**) plot shows the error histogram along with a normal density function fit.

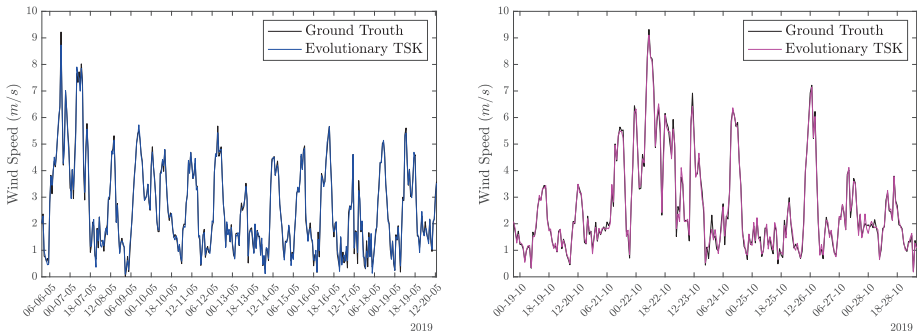


Figure 18. Evolutionary TSK method prediction results on wind speed data. The (**left**) plot illustrates a snapshot of the approximation on the training data, while the (**right**) plot zooms into the approximation on the test region.

https://doi.org/10.3390/a19010046

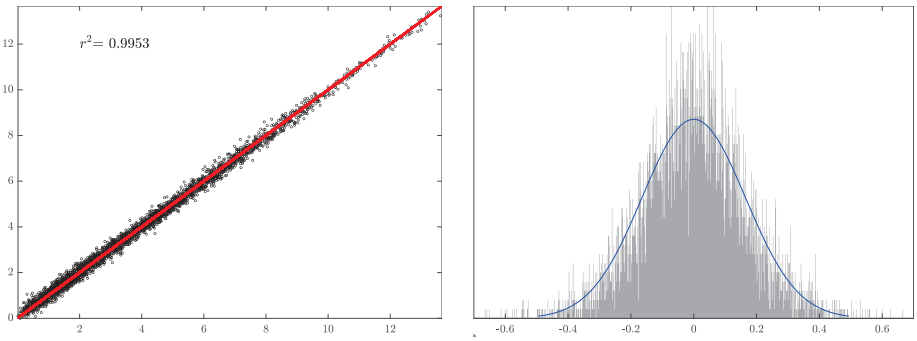


Figure 19. Evolutionary TSK method prediction results on wind speed data. On the (**left**): regression plot, and on the (**right**): error histogram along with a normal density function fit.

Table 2. $\mathcal{E}_{\text{RMSE}}$ for wind speed prediction on $\mathcal{D}^{\text{tr}}$ and $\mathcal{D}^{*}$.

Model	Training Data	Testing Data
Vanilla TSK	1.667×10^{-1}	1.390×10^{-1}
WM Fuzzy System	3.607×10^{-1}	3.669×10^{-1}
ANFIS	1.648×10^{-1}	1.368×10^{-1}
Evolutionary TSK	1.647×10^{-1}	1.371×10^{-1}
Bayesian TSK	1.647×10^{-1}	1.368×10^{-1}
Automated ML Method	1.648×10^{-1}	1.368×10^{-1}

Table 3. $\mathcal{E}_{\text{MAPE}}$ for wind speed prediction on $\mathcal{D}^{\text{tr}}$ and $\mathcal{D}^{*}$.

Model	Training Data	Testing Data
Vanilla TSK	7.0033	6.5292
WM Fuzzy System	15.9329	19.6175
ANFIS	6.9154	6.4252
Evolutionary TSK	6.9028	6.4344
Bayesian TSK	6.9226	6.4232
Automated ML Method	6.9402	6.4306

Table 4. Coefficient of determination for wind speed prediction on $\mathcal{D}^{\text{tr}}$ and $\mathcal{D}^{*}$.

Model	Training Data	Testing Data
Vanilla TSK	0.9951	0.9945
WM Fuzzy System	0.9774	0.9620
ANFIS	0.9953	0.9947
Evolutionary TSK	0.9953	0.9947
Bayesian TSK	0.9953	0.9947
Automated ML Method	0.9953	0.9947

Table 5 presents the average run times, obtained by simulating the predictive models several times. Run time refers to the time required for training on $\mathcal{D}^{\text{tr}}$ and inference on $\mathcal{D}^{*}$. Training refers to data rescaling, variational mode decomposition, and the tuning of parameters, given the optimisation algorithm. Inference refers to applying the trained models to the unseen data.

https://doi.org/10.3390/a19010046

Table 5. Total run time: training on $\mathcal{D}^{\text{tr}}$ and inference on $\mathcal{D}^{*}$.

MODEL	RUN TIME
EVOLUTIONARY TSK	15′:42″.567
BAYESIAN TSK	0′:40″.664
AUTOMATED ML METHOD	1:14′:15″.611

4.8. Statistical Test

To formally evaluate the statistical significance of the predictive performance exhibited by the model-generated forecasts, the Diebold-Mariano test [4,10,14] was employed. Statistical tests are quite essential for determining whether the observed differences in forecasting are likely due to random chance. Since we exclusively focused on one-step-ahead predictions, the Diebold-Mariano statistic was computed by Equation (38).

$$DM = \frac{\bar{d}}{\sqrt{\frac{2\pi \hat{f}_d(0)}{|\mathcal{T}|}}} \tag{38}$$

where $\bar{d}$ denotes the mean value of the difference between the squared errors of the two comparing forecasts, and $\hat{f}_d(0) = \frac{1}{2\pi}\hat{\gamma}_d(0)$, where $\hat{\gamma}_d(0) = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}(d_t - \bar{d})^2$.

The null hypothesis $\mathcal{H}_0$ postulates that the two forecasts possess equal accuracy. Rejecting $\mathcal{H}_0$ provides statistical evidence that the forecasts display a significant difference in accuracy, and this differentiation is not attributable to pure randomness. Under the null hypothesis, the DM statistic is asymptotically $\mathcal{N}(0,1)$ distributed. Thus, if $|DM| > 1.96$, the null hypothesis is rejected, i.e., at the 5% significance level, the errors' difference is not zero mean.

Table 6 showcases the results from the Diebold-Mariano statistical test. The forecasts generated by the proposed models successfully rejected the null hypothesis when compared to both the vanilla and WM-based forecasts. In the case of forecasts generated from ANFIS and the AUTOMATED ML method, the null hypothesis failed to be rejected, suggesting that these methods yielded forecasts of comparable accuracy. It is important to note that this finding should not be viewed as a disadvantage; rather, it illustrates that the proposed approaches were as competent as the established ANFIS and the very powerful method of designing machine learning models, demonstrating their effectiveness in generating prominent forecasts.

Table 6. Diebold-Mariano statistical test results.

	EVOLUTIONARY TSK		**BAYESIAN TSK**	
	DM	***p*-Value**	***DM***	***p*-Value**
VANILLA TSK	8.12	5.57×10^{-16}	8.69	4.24×10^{-18}
WM FUZZY SYSTEM	44.59	0	44.63	0

Finally, in comparison to a zero-order ANFIS, i.e., constant fuzzy rule consequents, both of the proposed models generated forecasts that rejected $\mathcal{H}_0$. Specifically, compared to the Bayesian TSK a DM statistic of 4.10 was obtained, with a p-value of 4.09×10^{-5}. Compared to the evolutionary TSK model, the DM statistic was computed to be 3.70 with a p-value of 2.13×10^{-4}.

https://doi.org/10.3390/a19010046

5. Discussion

Since wind speed prediction is quite a challenging task, the application of multiresolution analysis is fundamental to achieving accurate and reliable results. While fewer intrinsic mode functions may reduce prediction accuracy, increasing the level of decomposition can significantly improve prediction quality, albeit at the cost of added complexity/computational burden from incorporating and training an additional model. Thus, the accuracy level of the prediction was contingent upon the level of decomposition chosen.

In terms of the predictive model, a novel fuzzy TSK model was proposed. The novelty of this study arises from incorporating generalised rule consequents. The complexity of the model was encoded into the optimisation scheme; hence, an automated methodology for complexity determination has emerged. However, some limitation of the model needs to be discussed.

Fuzzy systems suffer from the curse of dimensionality: the problem is increased exponentially in volume associated with adding more dimensions to the input space; hence, their applicability is limited in relatively low-dimensional feature spaces. In this study, a grid partition was utilised to generate the fuzzy rules, focusing on two inputs that were determined using a wrapper-based algorithm.

Even without explicit constraints, fuzzy systems uphold intrinsic interpretability due to their clear rule-based structure and localised reasoning. This distinguishes them from black-box learning models. When modeling complex systems using fuzzy systems, the known accuracy-interpretability trade-off arises [37]. High predictive accuracy and interpretability are conflicting objectives; improving the interpretability of fuzzy models often degrades their overall performance, and vice versa [61]. In this study, the choice to prioritise predictive accuracy over full interpretability is intentional. While the resulting fuzzy system functions as a Takagi–Sugeno–Kang model with explicit rules and membership functions, the optimised fuzzy partitions may not necessarily align with conventional linguistic labels. However, fuzzy systems may yield interpretability to emerge much more naturally, considering some attention in the optimisation process, while this is not the case for neural or machine learning models. Comparing fuzzy systems with deep models, it is evident that in the former, interpretability forms the main basis, while in the latter, any interpretability study is conducted a posteriori, using SHAP and Local Interpretable Model-Agnostic Explanations (LIME) methods, which are the most dominant across the field of black-box machine learning interpretability [62].

Regarding the optimisation scheme, evolutionary algorithms can be quite time-consuming when dealing with a large parametric space. Even though the adopted version of differential evolution is adaptive in terms of updating its hyperparameters, evaluating candidate solutions involves computing the objective function related to the model's prediction performance. Clearly, if training the model is time-intensive, the application of evolutionary optimisation approaches may be limited. Another issue to consider is the complexity of the optimised model resulting from the optimisation algorithm. Since the optimisation relies on the training data, it is evident that the model may lead to overfitting.

6. Conclusions

In this study, the problem of wind speed prediction has been studied by exploring the development of a complete methodology. The approach utilised a higher-order Takagi–Sugeno–Kang model with a generalised form of rule consequents, which were not limited to linear. The TSK model served as the foundational predictive model, integrated into a variational mode decomposition framework, and generated predictions for each sub-

 https://doi.org/10.3390/a19010046

series of the decomposed data. The final prediction was obtained by aggregating the sub-predictions of all sub-models.

The training of the model was performed using either a modern version of adaptive differential evolution or a surrogate-based method, specifically Bayesian optimisation. Both optimisation techniques yielded identical and accurate results, as showcased via the adopted regression metrics. The optimisation process focused on optimally tuning the parameters of the antecedents, alongside the regularisation parameter. Furthermore, the functional form of the consequents was integrated into the optimisation procedure, thereby enabling an automated determination of model complexity.

A sequential wrapper-based algorithm has been used to determine the significant lags, which generated the feature space of the predictive models. Comparisons indicated that both proposed methods were as effective as established models in the artificial intelligence landscape.

In the context of practical applications, the Bayesian-optimised TSK model distinguishes itself due to its time-efficient training and scalable nature. This model has demonstrated the ability to generate accurate predictions in a rapid training time, thus providing a candidate effective solution for complex real-world challenges. In comparison to the evolutionary-based TSK model, this approach offered a notably faster training duration, while not always leading to the most complex fuzzy rule consequent, resulting in a simpler predictive model.

Future research points toward the following directions:

- Exploring new directions for the feature selection method, such as incorporating this task into the optimisation process;
- Exploring model selection methods, focusing on the use of simpler models;
- Exploring methods for evaluating uncertainty; generate predictions as intervals rather than point estimates.

Author Contributions: Conceptualisation, P.K. and A.D.; methodology, P.K. and A.D.; software, P.K.; validation, P.K. and A.D.; formal analysis, P.K.; investigation, P.K. and A.D.; resources, A.D.; writing—original draft preparation, P.K.; writing—review and editing, P.K. and A.D.; visualisation, P.K.; supervision, A.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data in this study were kindly provided, upon the authors' request, by the National Observatory of Athens, Greece. The authors do not have permission to share the data.

Acknowledgments: The authors would like to express their gratitude to the National Observatory of Athens, Greece, for the kind provision of the data used in this study. Additionally, the authors wish to express their heartfelt gratitude for the valuable suggestions provided by the anonymous reviewers, which greatly enhanced the quality of the paper.

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

References

1. Liu, H.; Chen, C. Data processing strategies in wind energy forecasting models and applications: A comprehensive review. *Appl. Energy* **2019**, *249*, 392–408. [CrossRef]
2. Lydia, M.; Suresh Kumar, S.; Immanuel Selvakumar, A.; Edwin Prem Kumar, G. Linear and non-linear autoregressive models for short-term wind speed forecasting. *Energy Convers. Manag.* **2016**, *112*, 115–124. [CrossRef]
3. Zhao, W.; Wei, Y.M.; Su, Z. One day ahead wind speed forecasting: A resampling-based approach. *Appl. Energy* **2016**, *178*, 886–901. [CrossRef]

https://doi.org/10.3390/a19010046

4. Wang, S.; Wang, J.; Lu, H.; Zhao, W. A novel combined model for wind speed prediction – Combination of linear model, shallow neural networks, and deep learning approaches. *Energy* **2021**, *234*, 121275. [CrossRef]
5. Zhang, Y.; Zhao, Y.; Kong, C.; Chen, B. A new prediction method based on VMD-PRBF-ARMA-E model considering wind speed characteristic. *Energy Convers. Manag.* **2020**, *203*, 112254. [CrossRef]
6. Gnana Sheela, K.; Deepa, S. Neural network based hybrid computing model for wind speed prediction. *Neurocomputing* **2013**, *122*, 425–429. [CrossRef]
7. Gao, Y.; Wang, J.; Zhang, X.; Li, R. Ensemble wind speed prediction system based on envelope decomposition method and fuzzy inference evaluation of predictability. *Appl. Soft Comput.* **2022**, *124*, 109010. [CrossRef]
8. Shang, Z.; Chen, Y.; Chen, Y.; Guo, Z.; Yang, Y. Decomposition-based wind speed forecasting model using causal convolutional network and attention mechanism. *Expert Syst. Appl.* **2023**, *223*, 119878. [CrossRef]
9. Hong, Y.Y.; Arce, C.J.E.; Huang, T.W. A Robust Hybrid Classical and Quantum Model for Short-Term Wind Speed Forecasting. *IEEE Access* **2023**, *11*, 90811–90824. [CrossRef]
10. Barjasteh, A.; Ghafouri, S.H.; Hashemi, M. A hybrid model based on discrete wavelet transform (DWT) and bidirectional recurrent neural networks for wind speed prediction. *Eng. Appl. Artif. Intell.* **2024**, *127*, 107340. [CrossRef]
11. Li, C.; Wang, L.; Zhang, G.; Wang, H.; Shang, F. Functional-type single-input-rule-modules connected neural fuzzy system for wind speed prediction. *IEEE/CAA J. Autom. Sin.* **2017**, *4*, 751–762. [CrossRef]
12. Zhao, J.; Guo, Z.H.; Su, Z.Y.; Zhao, Z.Y.; Xiao, X.; Liu, F. An improved multi-step forecasting model based on WRF ensembles and creative fuzzy systems for wind speed. *Appl. Energy* **2016**, *162*, 808–826. [CrossRef]
13. Kong, X.; Liu, X.; Shi, R.; Lee, K.Y. Wind speed prediction using reduced support vector machines with feature selection. *Neurocomputing* **2015**, *169*, 449–456. [CrossRef]
14. Tian, Z. Short-term wind speed prediction based on LMD and improved FA optimized combined kernel function LSSVM. *Eng. Appl. Artif. Intell.* **2020**, *91*, 103573. [CrossRef]
15. Chen, N.; Qian, Z.; Nabney, I.T.; Meng, X. Wind Power Forecasts Using Gaussian Processes and Numerical Weather Prediction. *IEEE Trans. Power Syst.* **2014**, *29*, 656–665. [CrossRef]
16. Zhang, C.; Wei, H.; Zhao, X.; Liu, T.; Zhang, K. A Gaussian process regression based hybrid approach for short-term wind speed prediction. *Energy Convers. Manag.* **2016**, *126*, 1084–1092. [CrossRef]
17. Zhang, Z.; Ye, L.; Qin, H.; Liu, Y.; Wang, C.; Yu, X.; Yin, X.; Li, J. Wind speed prediction method using Shared Weight Long Short-Term Memory Network and Gaussian Process Regression. *Appl. Energy* **2019**, *247*, 270–284. [CrossRef]
18. Liu, H.; Duan, Z.; Li, Y.; Lu, H. A novel ensemble model of different mother wavelets for wind speed multi-step forecasting. *Appl. Energy* **2018**, *228*, 1783–1800. [CrossRef]
19. Li, Y.; Shi, H.; Han, F.; Duan, Z.; Liu, H. Smart wind speed forecasting approach using various boosting algorithms, big multi-step forecasting strategy. *Renew. Energy* **2019**, *135*, 540–553. [CrossRef]
20. Qu, Z.; Zhang, K.; Mao, W.; Wang, J.; Liu, C.; Zhang, W. Research and application of ensemble forecasting based on a novel multi-objective optimization algorithm for wind-speed forecasting. *Energy Convers. Manag.* **2017**, *154*, 440–454. [CrossRef]
21. Mi, X.; Zhao, S. Wind speed prediction based on singular spectrum analysis and neural network structural learning. *Energy Convers. Manag.* **2020**, *216*, 112956. [CrossRef]
22. Moreno, S.R.; dos Santos Coelho, L. Wind speed forecasting approach based on Singular Spectrum Analysis and Adaptive Neuro Fuzzy Inference System. *Renew. Energy* **2018**, *126*, 736–754. [CrossRef]
23. Khosravi, A.; Koury, R.; Machado, L.; Pabon, J. Prediction of wind speed and wind direction using artificial neural network, support vector regression and adaptive neuro-fuzzy inference system. *Sustain. Energy Technol. Assess.* **2018**, *25*, 146–160. [CrossRef]
24. Asghar, A.B.; Liu, X. Estimation of wind speed probability distribution and wind energy potential using adaptive neuro-fuzzy methodology. *Neurocomputing* **2018**, *287*, 58–67. [CrossRef]
25. Gao, Y.; Wang, B.; Chen, F.; Zhang, W.; Zhou, D.; Wu, F.; Chen, D. Multi-step wind speed prediction based on LSSVM combined with ESMD and fractional-order beetle swarm optimization. *Energy Rep.* **2023**, *9*, 6114–6134. [CrossRef]
26. Duarte, F.S.; Rios, R.A.; Hruschka, E.R.; de Mello, R.F. Decomposing time series into deterministic and stochastic influences: A survey. *Digit. Signal Process.* **2019**, *95*, 102582. [CrossRef]
27. Zhang, H.; Wang, J.; Qian, Y.; Li, Q. Point and interval wind speed forecasting of multivariate time series based on dual-layer LSTM. *Energy* **2024**, *294*, 130875. [CrossRef]
28. Wang, J.; Zhang, W.; Wang, J.; Han, T.; Kong, L. A novel hybrid approach for wind speed prediction. *Inf. Sci.* **2014**, *273*, 304–318. [CrossRef]
29. Zhang, C.; Wei, H.; Zhao, J.; Liu, T.; Zhu, T.; Zhang, K. Short-term wind speed forecasting using empirical mode decomposition and feature selection. *Renew. Energy* **2016**, *96*, 727–737. [CrossRef]

 https://doi.org/10.3390/a19010046

30. Nasiri, H.; Ebadzadeh, M.M. Multi-step-ahead stock price prediction using recurrent fuzzy neural network and variational mode decomposition. *Appl. Soft Comput.* **2023**, *148*, 110867. [CrossRef]
31. Wang, J.; Cheng, Z. Wind speed interval prediction model based on variational mode decomposition and multi-objective optimization. *Appl. Soft Comput.* **2021**, *113*, 107848. [CrossRef]
32. Yang, S.; Liu, J. Time-Series Forecasting Based on High-Order Fuzzy Cognitive Maps and Wavelet Transform. *IEEE Trans. Fuzzy Syst.* **2018**, *26*, 3391–3402. [CrossRef]
33. Shi, G.; Qin, C.; Tao, J.; Liu, C. A VMD-EWT-LSTM-based multi-step prediction approach for shield tunneling machine cutterhead torque. *Knowl.-Based Syst.* **2021**, *228*, 107213. [CrossRef]
34. Ding, M.; Zhou, H.; Xie, H.; Wu, M.; Nakanishi, Y.; Yokoyama, R. A gated recurrent unit neural networks based wind speed error correction model for short-term wind power forecasting. *Neurocomputing* **2019**, *365*, 54–61. [CrossRef]
35. Niu, Y.; Wang, J.; Zhang, Z.; Yu, Y.; Liu, J. A combined interval prediction system based on fuzzy strategy and neural network for wind speed. *Appl. Soft Comput.* **2024**, *155*, 111408. [CrossRef]
36. Li, Q.; Wang, J.; Zhang, H. A wind speed interval forecasting system based on constrained lower upper bound estimation and parallel feature selection. *Knowl.-Based Syst.* **2021**, *231*, 107435. [CrossRef]
37. Shihabudheen, K.; Pillai, G. Recent advances in neuro-fuzzy system: A survey. *Knowl.-Based Syst.* **2018**, *152*, 136–162. [CrossRef]
38. Wang, L.X.; Mendel, J. Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Trans. Neural Netw.* **1992**, *3*, 807–814. [CrossRef]
39. Kosko, B. Fuzzy systems as universal approximators. In Proceedings of the [1992 Proceedings] IEEE International Conference on Fuzzy Systems, San Diego, CA, USA, 8–12 March 1992; pp. 1153–1162. [CrossRef]
40. Zadeh, L.A. Is there a need for fuzzy logic? *Inf. Sci.* **2008**, *178*, 2751–2779. [CrossRef]
41. Wang, L.X.; Mendel, J. Generating fuzzy rules by learning from examples. *IEEE Trans. Syst. Man Cybern.* **1992**, *22*, 1414–1427. [CrossRef]
42. Jang, J.S. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man Cybern.* **1993**, *23*, 665–685. [CrossRef]
43. Dragomiretskiy, K.; Zosso, D. Variational Mode Decomposition. *IEEE Trans. Signal Process.* **2014**, *62*, 531–544. [CrossRef]
44. Garnett, R. *Bayesian Optimization*; Cambridge University Press: Cambridge, UK, 2023.
45. Frazier, P.I. A Tutorial on Bayesian Optimization. *arXiv* **2018**, arXiv:1807.02811. [CrossRef]
46. Takagi, T.; Sugeno, M. Fuzzy identification of systems and its applications to modeling and control. *IEEE Trans. Syst. Man Cybern.* **1985**, *SMC-15*, 116–132. [CrossRef]
47. Wu, D.; Lin, C.T.; Huang, J.; Zeng, Z. On the Functional Equivalence of TSK Fuzzy Systems to Neural Networks, Mixture of Experts, CART, and Stacking Ensemble Regression. *IEEE Trans. Fuzzy Syst.* **2020**, *28*, 2570–2580. [CrossRef]
48. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; The MIT Press: Cambridge, MA, USA, 2016.
49. 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]
50. Das, S.; Suganthan, P.N. Differential Evolution: A Survey of the State-of-the-Art. *IEEE Trans. Evol. Comput.* **2011**, *15*, 4–31. [CrossRef]
51. Li, Y.; Wang, S.; Yang, B. An improved differential evolution algorithm with dual mutation strategies collaboration. *Expert Syst. Appl.* **2020**, *153*, 113451. [CrossRef]
52. Zhang, Y.; Chen, G.; Cheng, L.; Wang, Q.; Li, Q. Methods to balance the exploration and exploitation in Differential Evolution from different scales: A survey. *Neurocomputing* **2023**, *561*, 126899. [CrossRef]
53. Opara, K.R.; Arabas, J. Differential Evolution: A survey of theoretical analyses. *Swarm Evol. Comput.* **2019**, *44*, 546–558. [CrossRef]
54. Engelbrecht, A.P. *Computational Intelligence: An Introduction*, 2nd ed.; Wiley Publishing: Hoboken, NJ, USA, 2007.
55. Rasmussen, C.; Williams, C. *Gaussian Processes for Machine Learning*; Adaptive Computation And Machine Learning; MIT Press: Cambridge, MA, USA, 2005.
56. Snoek, J.; Larochelle, H.; Adams, R.P. Practical Bayesian Optimization of Machine Learning Algorithms. *arXiv* **2012**, arXiv:1206.2944. [CrossRef]
57. Xu, Z.; Wang, H.; Phillips, J.M.; Zhe, S. Standard Gaussian Process Can Be Excellent for High-Dimensional Bayesian Optimization. *arXiv* **2024**, arXiv:2402.02746. [CrossRef]
58. Hennig, P.; Osborne, M.; Kersting, H. *Probabilistic Numerics: Computation as Machine Learning*; Cambridge University Press: Cambridge, UK, 2022.
59. Wu, D.; Yuan, Y.; Huang, J.; Tan, Y. Optimize TSK Fuzzy Systems for Regression Problems: Minibatch Gradient Descent With Regularization, DropRule, and AdaBound (MBGD-RDA). *IEEE Trans. Fuzzy Syst.* **2020**, *28*, 1003–1015. [CrossRef]
60. Li, L.; Jamieson, K.; Rostamizadeh, A.; Gonina, E.; Hardt, M.; Recht, B.; Talwalkar, A. A System for Massively Parallel Hyperparameter Tuning. *arXiv* **2020**, arXiv:1810.05934. [CrossRef]

https://doi.org/10.3390/a19010046

61. Zhou, S.M.; Gan, J.Q. Low-level interpretability and high-level interpretability: A unified view of data-driven interpretable fuzzy system modelling. *Fuzzy Sets Syst.* **2008**, *159*, 3091–3131. [CrossRef]
62. Linardatos, P.; Papastefanopoulos, V.; Kotsiantis, S. Explainable AI: A Review of Machine Learning Interpretability Methods. *Entropy* **2021**, *23*, 18. [CrossRef] [PubMed]

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/a19010046

algorithms

MDPI

Article

Hybrid Evolutionary Multi-Objective Method for Automatic Design of a Lightweight CNN Architecture Applied to Coronary Stenosis Classification

Miguel-Angel Gil-Rios [1], Ivan Cruz-Aceves [2,*], Arturo Hernandez-Aguirre [3], Erick-G. G.-de-Paz [3] and Juan-Manuel Lopez-Hernandez [4]

[1] Departamento de Tecnologías de la Información e Innovación Digital, Universidad Tecnológica de León, León 37670, Mexico; mgil@utleon.edu.mx
[2] SECIHTI—Centro de Investigación en Matemáticas, A.C., Jalisco S/N, Col. Valenciana, Guanajuato 36000, Mexico
[3] Centro de Investigación en Matemáticas, A.C., Jalisco S/N, Col. Valenciana, Guanajuato 36000, Mexico; artha@cimat.mx (A.H.-A.); erick.giles@cimat.mx (E.-G.G.-d.-P.)
[4] División de Ingenierías (DICIS), Campus Irapuato-Salamanca, Universidad de Guanajuato, Carretera Salamanca-Valle de Santiago km 3.5 + 1.8 km, Comunidad de Palo Blanco, Salamanca 36885, Mexico; jmlopez@ugto.mx
* Correspondence: ivan.cruz@cimat.mx; Tel.: +52-473-732-7155 (ext. 4506)

Abstract

This paper presents a novel method based on a Hybrid Multi-Objective Evolutionary strategy for the automatic design of a lightweight convolutional neural network used for coronary stenosis classification. The hybrid methodology consists of two search stages, starting with the Multi-Objective Evolutionary Algorithm based on Decomposition, to generate a set of optimal solutions focused on the minimization of two objectives: the accuracy classification error and the number of learning parameters in the convolutional neural network. Subsequently, the Simulated Annealing algorithm is applied to improve a subset of the solutions produced in the previous step. After the method was complete, a convolutional neural network model consisting of 3498 learning parameters was found by the proposed hybrid strategy, which is a considerably low number compared with the other architectures reported in the literature. Consequently, the found model achieved the highest classification performance rate in terms of the Accuracy and Jaccard Similarity Coefficient metrics with values of 0.94 and 0.89, respectively, using a database consisting of 608 images of regions with positive and negative coronary stenosis cases. On a second test, the model was tested using a database consisting of 2788 instances of natural and synthetic images of coronary stenosis cases. Corresponding maximum classification rates of 0.97 and 0.93 for the Accuracy and Jaccard Similarity Coefficient metrics, respectively, were achieved. In addition, the average required time to classify a single instance was 0.009 seconds. The obtained results showed that the proposed method is feasible for the automatic design of lightweight convolutional neural networks that can be used as a part of decision-making systems in clinical practice.

Keywords: multi-objective; optimization; evolutionary; convolutional neural network; image processing; classification; coronary stenosis

1. Introduction

Coronary stenosis is a common disease which leads to heart attacks and eventually the death of patients, but it is not always accurately diagnosed by cardiology specialists [1]. X-ray tomography still remains the most widely adopted method to acquire coronary angiographies from patients with heart diseases [2]. However, the acquired X-ray coronary digital images are commonly low resolution with narrow regions or a low contrast of vessels and background, which makes it difficult for cardiologists to find affected arteries by an exhaustive visual image exploration. Figure 1 presents a coronary angiography involving a region with a stenosis case labeled by a cardiology specialist and a corresponding schematic representation.

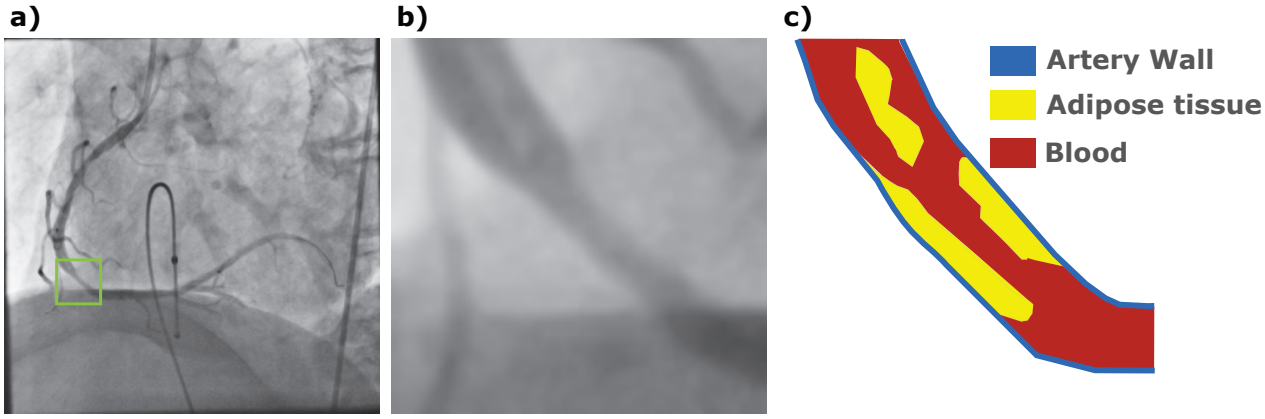


Figure 1. (**a**) A coronary angiography with a stenosis case labeled (green square) by a cardiology specialist. (**b**) The labeled region expanded. (**c**) A schematic representation of the corresponding coronary stenosis case.

Since the identification of coronary stenosis cases relies on the expertise and visual accuracy of cardiology specialists, to identify toxin or adipose tissue areas inside the arteries, the use of techniques and digital image processing methods is important to improve the detection of artery diseases. The automatic detection and classification of coronary stenosis cases from digital images of coronary angiographies have been reported in the literature using distinct approaches and methods. The use of feature extraction and machine learning classification techniques are some of the pioneer approaches [3]. For instance, the use of image processing techniques to enhance or segment arteries, which allows working with morphological features, has been used to differentiate healthy arteries from those indicating disease [4–7]. The combination of feature extraction and automatic feature selection techniques is also used to extract optimal feature subsets able to train specific classification machine learning models [8].

The problem has been also addressed by the use of deep learning techniques, which are a completely different approach when compared with manual feature extraction and machine learning techniques. The use of convolutional neural networks (CNN) for coronary stenosis detection and classification is the most common approach reported in deep learning-related literature [9,10]. One of main drawbacks of using traditional CNN architectures is the need for large image databases. This commonly leads to the use of highly complex CNN architectures, which could decrease its performance in terms of accuracy when small image databases are used for object detection or classification tasks [11]. To overcome the disadvantages of work with small image databases, the use of data augmentation and transfer learning techniques has been applied successfully. Despite the gains in classification accuracy provided by data augmentation, the absence of novel, empirical

https://doi.org/10.3390/a19010047

datasets poses challenges for clinical adoption [12,13]. In addition, as CNN architectural complexity increases, the difficulty of identifying the specific components most critical to the classification task scales accordingly [14].

The search for optimal CNN architectures is described in the literature as the Neural Architecture Search (NAS) problem. It is not a trivial task due to the high number of combinations that are possible to form with the distinct layer types. Furthermore, structural constraints must be addressed to prevent inconsistent layer connectivity or execution errors during the training phase. Beyond these constraints, representing CNN architectures across diverse structural formats—conforming to various computational techniques and algorithms—remains a significant challenge [15,16]. Consequently, numerous methodologies for the automated design of CNN models have emerged in the literature. For instance, Jiang et al. [17], utilize the Multi-Objective Particle Swarm Optimization along with the problem decomposition technique to generate CNN architectures. Similarly, Elsken et al. [18] propose a hill-climbing approach, which is validated using the CIFAR-10 image database [19]. More recently, Franco-Gaona et al. [20] employed an Estimation of Distribution Algorithm (EDA) to evolve optimal architectures for diverse image-processing tasks, including table detection and coronary stenosis classification. Parallel to these developments, substantial research has focused on the automated design of lightweight CNNs. Pham et al. [21] introduced a methodology for the creation of lightweight CNN architectures by optimizing inter-layer connections. Alternatively, proposals for the design of CNN architectures that are able to run on mobile and embedded devices with important hardware limitations have been reported [22,23]. The use of model compression techniques have been also applied to reduce the resource consumption cost of CNN models [24,25].

While CNNs have revolutionized medical image recognition, their application is significantly hindered by the scarcity of large-scale, annotated datasets. Unlike general-purpose image databases such as ImageNet, which served as the foundation for modern CNN architectures, medical databases often lack the volume and diversity required for robust training. In this context, high-complexity architectures including ResNet50 [26], InceptionV3 [27] and VGG-16 [28] yield suboptimal classification results when trained on limited image databases. Similarly, lightweight models such as MobileNetV2 [29] and EfficientNetB0 [30], despite their efficiency on large-scale image databases, exhibit a considerable decline in performance when applied to constrained medical image databases. Even when the use of transfer learning increases classification rates, there is a need to augment the original images because of large differences in the number of instances in which transfer learning is supported [31]. However, while data augmentation can be beneficial, some strategies could incorporate unrealistic variances or narrow relevant features that are critical to the analysis [32]. In addition, since data augmentation is a standard technique to improve model generalization, its application in medical imaging presents unique challenges that can compromise diagnostic integrity or violate critical data relationships [33,34].

This paper proposes a novel Hybrid Multi-Objective Evolutionary method to automatically design lightweight CNNs for classifying coronary stenosis cases. The use of multi-objective strategies such as the Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA-D) [35] is relevant because it avoids the problem of assigning weights to the distinct goals involved. In addition, since the MOEA-D produces a set of trade-off solutions among the objectives involved, an additional search-refined stage driven by the Simulated Annealing (SA) method is performed [36]. In this stage, a subset containing the 3 best solutions achieved by the MOEA-D is extracted in order to perform a second search

 https://doi.org/10.3390/a19010047

stage driven by the SA algorithm, which decreases the architecture complexity and, at the same time, tries to maintain or increase the classification accuracy.

The main contribution of this paper is a methodology that is able to automatically design lightweight specialized CNN architectures focused on the classification of positive and negative coronary stenosis cases using small image databases. It involves important design constraints to assure the creation of consistent CNN models. The proposed method conceptualizes the NAS problem as a multi-objective optimization problem with two goals: minimizing the classification error rate and the architecture complexity. A high classification accuracy is essential to guarantee the quality of the model architecture to be used in clinical practice. On the other hand, the use of a lightweight architecture is adequate to achieve optimal classification rates using small image datasets. In order to measure the classification rate, the Accuracy error metric was used during the training stage. Furthermore, to assure the obtained results, additional metrics such as Precision, Recall, F1-Score and Jaccard Similarity Coefficient were also used on posterior independent tests. In addition, the number of learning parameters was used as a complexity measure. For the experiments, two distinct image databases of coronary stenosis cases were used to confirm the obtained results. The first image database is composed of negative and positive coronary stenosis patches. On consecutive experiments, a second independent database, involving natural and synthetic images of coronary stenosis cases, was used. In the conducted tests, the proposed method performs better than the other compared techniques from the literature in almost all metrics. Finally, based on the presented results, the proposed methodology was adequate to produce CNN models that can be used in clinical practice as part of information systems for computer-aided diagnosis.

2. Background

2.1. Convolutional Neural Networks

The convolutional neural network (CNN) is a deep learning technique which has been applied successfully in distinct types of problems involving image classification and object detection tasks [37]. CNNs are based on traditional artificial neural networks (ANNs), inheriting their learning capacity but with substantial differences. The ANN is a general-purpose model where every neuron is connected to every other neuron in the next layer. It was designed primarily for classification and pattern recognition tasks over plain data more than spatial information. In contrast, the input data on a CNN is formed by the original image information involving its size and pixel intensities in each color band. The accuracy of a CNN relies in its internal structure or architecture, which is formed by combining and linking distinct layer types. From a general perspective, a CNN model focused on classification tasks can be viewed as a three-component artifact. It starts with an input layer, followed by a set of internal layers, and it ends with an output layer which involves the distinct expected classes to produce a weighted value according to the most similar computed class. Figure 2 illustrates a top-level overview of a CNN model for classification tasks.

The input layer corresponds to the pixel values of the original image, whose shape is a matrix of size $w \times h$, where w and h correspond to the image width and height, respectively, for a grayscale image. Consequently, for data processing and feature extraction, distinct layer types are used as described below.

https://doi.org/10.3390/a19010047

- *Convolution Layer.* This is the core processing layer in a CNN. It is designed to perform the convolution operation, which in the spatial domain of a discrete $2D$ image is expressed as follows in practical implementations:

$$G(x,y) = \sum_{m=-a}^{a} \sum_{n=-b}^{b} I(x+m, y+n)K(m,n), \tag{1}$$

where I is the input image, K is the kernel expressed as a matrix of size $(2a+1) \times (2b+1)$, G is the convolution operation response, and (x,y) correspond to the coordinates of the pixel in the image. Finally, m and n are the kernel K indices. The kernel height and width are defined as $(2a+1)$ and $(2b+1)$, respectively.
In the convolution operation, a small matrix (or filter) is slid over the original image to produce an output in which certain features are enhanced or segmented, depending on the kernel size and its values. In CNNs, the convolution layer is governed by four main parameters: the kernel size, the number of filters to be created, the stride size and the padding type. It is important to mention that the stride and the padding parameters are related and must be established coherently. Since the stride defines the step length in which the kernel is passed over the image pixels, the output size will be affected. For instance, if the input size in pixels is 128×128 and the stride parameter is established with a value of 2, the output size will be 64×64. As a consequence, a padding of type 'valid' will not modify the output. However, if the padding value is established as 'same', the ouput image will have the original size filling with zeros around the output image.
- *Maxpooling Layer.* A maxpooling layer is focused in the dimensional reduction of the input data by discriminating low values. Figure 3 illustrates how a maxpooling layer discriminates information from an input matrix to produce a new one of lower size.
- *Normalization Layer.* Since the convolution layers produce outputs with out-of-range values, normalization layers rescale them into a valid range. In addition, normalized values help to accelerate the overall training process of the CNN [38].
- *Rectification Layer.* Rectification layers such as Rectified Linear Unit (ReLU) and Leaky Rectified Linear Unit (LeakyReLU) are intended to introduce non-linear capabilities to the CNN. The main difference between ReLU and LeakyReLU layers is the way they manage negative input values. A ReLU layer will set negative values to zero. Otherwise, the output will be the same. This function is expressed as follows:

$$f(x) = max(0, x), \tag{2}$$

where x is the value of each element in the input matrix or vector. On the other hand, the LeakyReLU layer transforms negative values into a small positive fraction of the input. The LeakyReLU is a less aggressive rectification method compared with ReLU, since neurons with zero value can be inactive permanently and are considered dead neurons. The LeakyReLU function can be expressed as follows:

$$f(x) = \max(\alpha x, x), \tag{3}$$

where α is a small constant value, for instance 0.01.
- *Flatten Layer.* This layer flattens the input. If the input is a matrix of size 4×4, the flatten layer will convert it into a vector of 16 elements, as illustrated in Figure 4. Flatten layers are necessary in CNNs because they allow the CNN to be connected by dense layers.

https://doi.org/10.3390/a19010047

- *Dense Layer.* The dense layer or fully connected layer is an essential part of a CNN that will be used for classification purposes. This layer connects every one of its neurons with every neuron of the preceding layer. As a consequence, dense layers allow the learning of complex relationships and patterns.

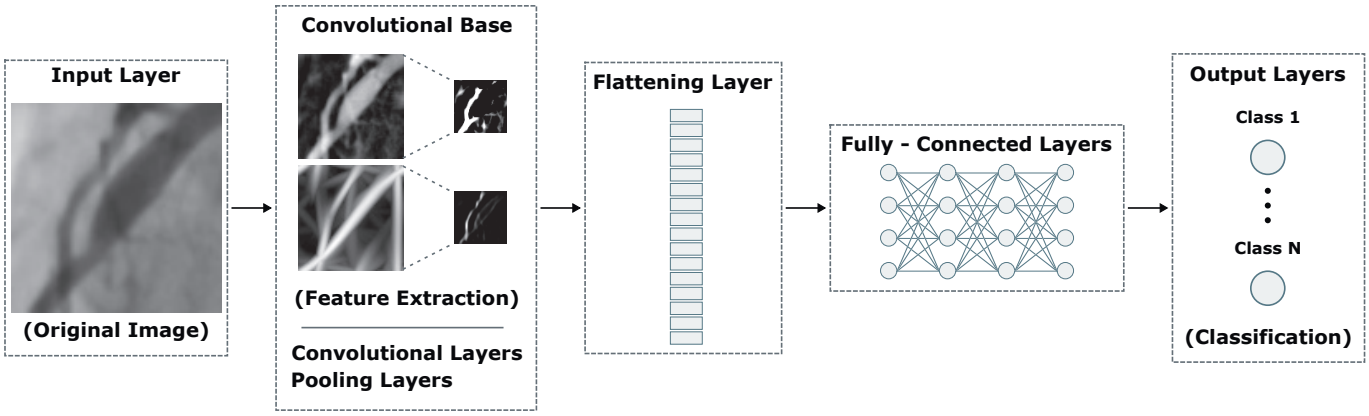


Figure 2. Top-level overview of a CNN model for classification tasks.

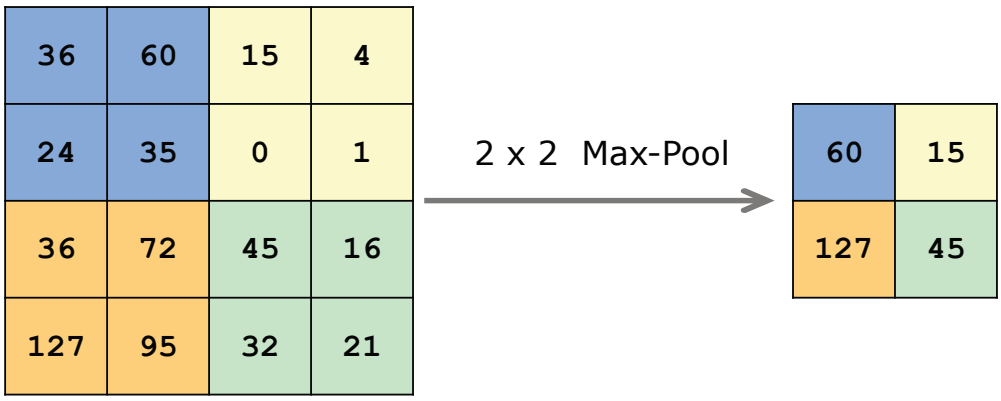


Figure 3. Representation of the max-pooling operation. Colors are only representative for the groups of values that are considered for the dimensional reduction operation.

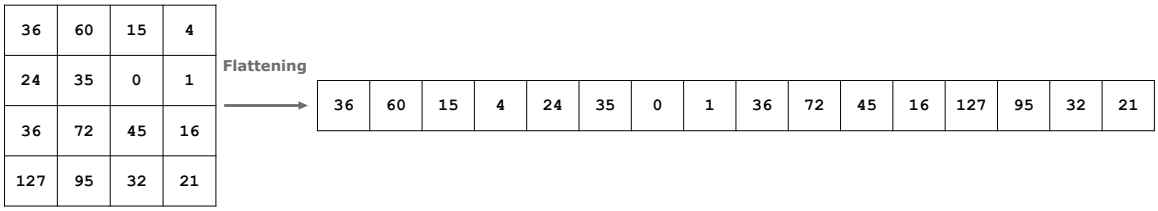


Figure 4. Representation of a matrix flattening of size 4×4.

2.2. Multi-Objective Evolutionary Algorithm Based on Decomposition

The Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA-D) is a strategy focused on the search of solutions that accomplish distinct objectives simultaneously [39]. The MOEA-D decomposes a multi-objective problem (MOP) into a set of interconnected single-objective subproblems, which are solved simultaneously. A key difference from other multi-objective evolutionary strategies is the use of neighborhoods to create new populations instead of elitist selection mechanisms based on the Pareto dominance. The MOEA-D selects individuals from the closest neighborhood to each solution, and then traditional operators such as crossover and mutation are applied to create the individuals for the new population. Figure 5 illustrates the set of the best trade-off solutions

https://doi.org/10.3390/a19010047

produced by the MOEA-D, which was focused on the minimization of the ZDT1 function of Zitzler et al. [40] involving two objectives.

An inherent trait in multi-objective optimization strategies is the generation of diverse solutions involving a trade-off among the objectives. The solution marked in a red square in the upper-left position of Figure 5 achieves the minimum value for the first objective with a high cost in the second one. In contrast, the solution marked in the red square at the bottom-right achieves the worst and best cost for the first and second objective, respectively. As a consequence, the solution that has the closest distance to the origin, in terms of the Euclidean definition, was found near the center of all the solution sets, which have equilibrated fitness values for the two involved objectives. The main strategy of the MOEA-D to solve multi-objective problems is the decomposition process, which decomposes the problem into a set of interconnected single-objective scalarized subproblems. The problem decomposition starts by defining a number N of weight vectors defined as follows:

$$\Lambda = \{\lambda_1, \lambda_2, ..., \lambda_N\}, \tag{4}$$

where Λ is the set of weight vectors λ_i, which are distributed over the solution space (that is formed the first time by evaluating the initial population). Consequently, each element λ_i corresponds to the solution (fitness value) for a unique subproblem whose scalarization can be computed using the Tchebicheff formula [41], which is expressed as follows:

$$\min g^{tch}(x|\lambda) = \max_{j=1}^{m}\{\lambda_j|f_j(x) - z_j^*|\}, \tag{5}$$

where $f_j(x)$ is the fitness value for a solution x, and z_j^* is an ideal point for subproblem j computed as follows:

$$z^* = (z_1^*, z_2^*, \ldots, z_m^*), \tag{6}$$

where $z_j^* = \min_{i=1}^{N}(f_j(x))$. By applying the Equations (5) and (6), the MOEA-D pseudocode is described in detail in Algorithm 1.

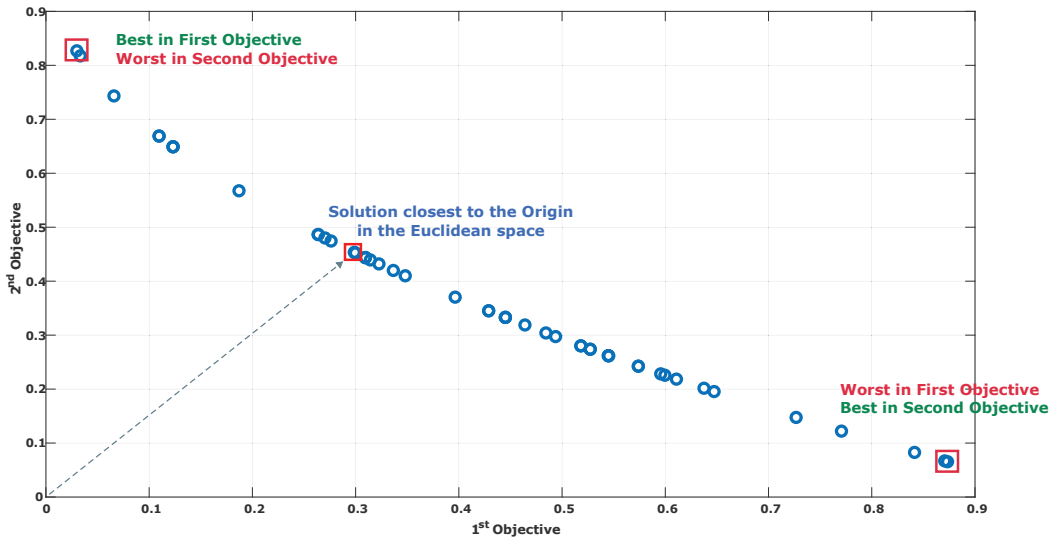


Figure 5. Set of trade-off solutions produced by the MOEA-D in the minimization of the multi-objective ZDT1 problem.

https://doi.org/10.3390/a19010047

Algorithm 1: MOEA-D Pseudocode.

Input:
N /* The number of subproblems considered in MOEA-D */
A /* Uniform spread of N weight vectors $\lambda^1, \ldots \lambda^N$ */
T /* Number of weight vectors in the neighborhood of each weight vector */
begin
  Set $EP = \phi$
  Set $gen = 0$ /* Current generation. */
  Set $gen_max = 100$ /* Maximum number of generations */
  Set $X = \{x_1, \ldots, x_N\}$ /* Generate the initial population X randomly */
  /* Evaluate each individual (solution) to obtain the fitness vectors */
  for *i=1 to N* **do**
    $F(x_i) =$ evaluate(x_i)
  end
  /* Create a set of N weight vectors distributed uniformly over the objective space: */
  $\Lambda = \{\lambda_1, \ldots, \lambda_N\}$
  /* Define neighborhoods: */
  for *each* λ_i **do**
    Find the T closest neighbors to form neighborhood set $B(i)$
  end
  /* Initialize ideal point considering the m objectives involved in the problem: */
  $z^* = (z_1^*, \ldots, z_m^*)$, where $z_j^* = \min_{i=1}^{N}\{f_j(x_i)\}$
  /* Loop until max generations are reached (or other stop criteria) */
  while $gen < gen_max$ **do**
    for $i = 1$ *to* N **do**
      Select k parents (for instance, $k = 2$) from neighborhood $B(i)$
      /* Generate new individuals applying genetic operators */
      Set $x_new =$ crossover$(B(i)_{k_1}, (B(i)_{k_2})$
      Set $x_new =$ mutate(x_new)
      Set $X_i = x_new$
      /* Evaluate solution to obtain its fitness */
      Set $F(y) =$ evaluate(x_new)
      /* Update the ideal point for each objective : */
      for $j = 1$ *to* m **do**
        Set $z_j^* = \min(z_j^*, F_j(m))$
      end
      /* For each subproblem j in $B(i)$ */
      for $j = 1$ *to* m **do**
        /* Calculate the max fitness of the offspring y for subproblem j */
        $g^{tch}(y|\lambda_j, z^*) = \max_{k=1}^{m}\{\lambda_{jk}|f_k(y) - z_k^*|\}$
        /* Calculate the max fitness of the current solution x_j for subproblem j */
        $g^{tch}(x_j|\lambda_j, z^*) = \max_{k=1}^{m}\{\lambda_{jk}|f_k(x_j) - z_k^*|\}$
        if $g^{tch}(y|\lambda_j, z^*) \leq g^{tch}(x_j|\lambda_j, z^*)$ **then**
          /* Replace the solution */
          Set $x_j = y$
          /* Replace the objective vector */
          Set $F(x_j) = F(y)$
          /* Add the solution to EP if no vector in EP dominate $F(y)$ */
          if *NOT dominate*$(EP, F(y))$ **then**
            Set $EP = \{EP, y\}$
          end
        end
      end
    end
    Set $gen = gen + 1$
  end
end
Output: EP: The set of trade-off solutions.

https://doi.org/10.3390/a19010047

2.3. Simulated Annealing

Simulated Annealing (SA) is a search strategy which was abstracted from the controlled cooling process used in metallurgy. SA starts with a temperature factor and an initial solution that will be transformed iteratively until the method ends. On each iteration, the temperature factor is decreased constantly, and it is used along with an energy difference value to compute an acceptance probability, which is computed as follows:

$$P(e, e', T) = \begin{cases} 1 & \text{if } e' < e, \\ \exp(-(e' - e)/T) & \textit{otherwise}, \end{cases} \tag{7}$$

where T is the current temperature, and e and e' are the corresponding function objective values of the current and the new solution, respectively. The SA pseudocode is described in Algorithm 2.

Algorithm 2: Simulated Annealing Pseudocode

```
Input:
T_max /* Max temperature value */
T_min /* Min temperature value */
T_step /* Temperature variation from T_max to T_min */
begin
    Set s = s_0 /* Generate a random solution */
    Set T_current = T_max
    while T_current > T_min do
        /* Select a random element */
        s_new = selectRandom(s)
        /* Compute P(E(s), E(s_new), T_current) using the Equation (7) */
        if P(E(s), E(s_new), T_current) ≥ generateRandom(0, 1) then
            Set s = s_new
        end
        Set T_current = T_current − T_step
    end
end
Output: The final state s
```

2.4. Performance Evaluation Metrics

The classification performance can be measured using distinct metrics which relies on the fraction of the four distinct results that can be obtained in a binary classification process:

- *True Positive (TP):* the number of instances that are classified as positive and correspond to real positive cases.
- *True Negative (TN):* the number of instances that are classified as negative and correspond to real negative cases.
- *False Positive (FP):* the number of instances that are classified as positive and correspond to real negative cases.
- *False Negative (FN):* the number of instances that are classified as negative and correspond to real positive cases.

By using the *TP*, *TN*, *FP* and *FN* classification results, different classification performance metrics such as Accuracy (Acc), Precision, Recall, Jaccard Coefficient of Similarity (JSC) and F1-Score can be computed as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

https://doi.org/10.3390/a19010047

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$JSC = \frac{TP}{TP + FP + FN} \tag{11}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{12}$$

Additionally, the complexity of a CNN can be measured by its number of learning parameters. It is important to note that the complexity is related to the performance of a CNN in terms of hardware resources usage such as RAM, CPU/GPU, the required time for the training and classification stages, and the classification performance in terms of Accuracy or another metric [42].

3. Proposed Method

The proposed method involves two main stages related to the NAS problem. In the first stage, a set of solutions is produced by the MOEA-D strategy. Each solution is a trade-off between the error classification accuracy during the training and validation stages and the architecture complexity, which is measured using the number of learnable parameters of the CNN. In the second stage, a subset of optimal solutions, found by the MOEA-D, are selected and improved independently using the SA algorithm to finally select the optimal solution. The difficulty of designing an optimal CNN architecture is due to needing the correct number and combination of distinct layer types involving their own setup parameters. Since the multi-objective evolutionary algorithms such as the MOEA-D work with the concepts of individual and population, each solution (or individual) is represented as a vector of values considering the distinct variables involved in the problem. Therefore, a vector of 7 elements was used for the representation of a basic CNN architecture by considering the most common CNN layer types. Figure 6 illustrates the design approach used for the representation of distinct CNN layer types.

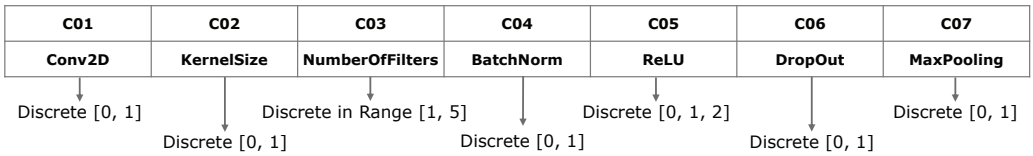


Figure 6. Codification of distinct layer types of the CNN using a vector of 7 elements identified as cells *C*01 to *C*07.

According to Figure 6, distinct CNN layers are represented as a vector of values requiring important considerations, which are described as follows. The first cell stores a discrete value to decide if a convolution layer will be present (1) or not (0). The second cell stores the convolution kernel size, which is constrained to sizes of 3×3 if the *C*02 value is 0 or 5×5 otherwise [28]. Cell *C*03 governs the number of convolution filters. The cell stores a discrete value in the range $[1, 5]$ to generate the power of 2 values, which can be $2, 4, 8, 16$ and 32 convolution filters [43]. Cell *C*04 indicates if a normalization layer will be added (1) or not (0). Cell *C*05 controls the activation and type of the rectified linear unit layer, where 0 means that a ReLu layer will not be added, 1 adds a ReLu layer, and 2 adds a LeakyReLU layer. Cell *C*06 activates the addition of a dropout layer if the value is 1. If the value is 0, the dropout layer will not be added. Cell *C*07 is a discrete value indicating if a maxpooling layer will be added (1) or not (0). The size of a pooling layer can be established

https://doi.org/10.3390/a19010047

arbitrarily. However, CNN architectures such as VGG and ResNet have demonstrated that applying a half uniform downsampling strategy produces optimal classification results, decreasing the computational load, memory and CPU/GPU resources [44].

By considering distinct pooling sizes (64×64, 32×32, 16×16, 8×8, 4×4 and 2×2), the previous structure can be repeated 6 times to form the full representation of the CNN architecture as a vector of 42 elements, as illustrated in Figure 7.

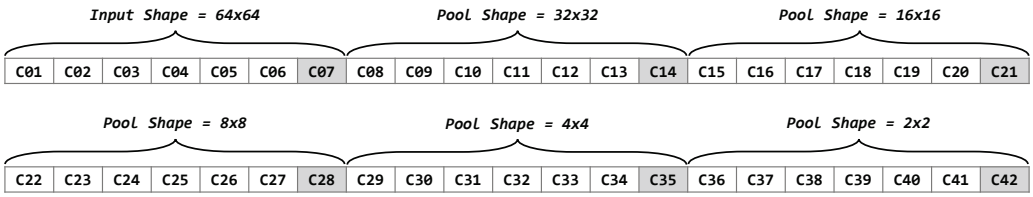


Figure 7. Codification of the full representation of a uniform downsampling CNN architecture using a vector of 42 elements.

In addition, several constraints were added to the architecture representation as follows. It was strategic to set the parameter stride of convolutional layers with a value of 2 in order to add stabilization and performance improvement [26,45]. Even when the CNN architecture representation is intended to decrease on each block of seven cells, the insertion of a pooling layer that downscales the inputs to subsequent layers depends on the activation of cells $C07$, $C14$, $C21$, $C28$ and $C35$. If some of those cells are not active (has a value of 1), the inputs remain in the last downscale size from previous layers. In addition, cell $C42$ does not have an effect on the process, but it is present to keep the CNN representation structure. This design strategy allows adding a diversity of solutions because the CNN architectures are not constrained to a single form. After a solution is produced, it is important to finish the CNN architecture by adding the corresponding flatten and dense layers, which are responsible for the classification task. Since the input of these layer types depends of the output of the previous layer, they were not considered as part of the solution representation.

Based on the representation described before, the proposed method for the NAS stage can be implemented accordingly. Figure 8 illustrates the hybrid multi-objective evolutionary method schema, which is focused on the automatic design of a lightweight CNN for the classification of positive and negative coronary stenosis cases.

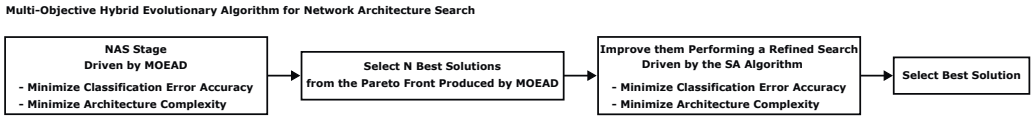


Figure 8. Proposed method steps for the automatic design of CNN architectures.

According to Figure 8, the hybrid metaheuristic starts with an initial CNN architecture search driven by the MOEA-D method, which returns a set of trade-off solutions known as the Pareto front. In step 2, a subset of N solutions is selected from the Pareto front to be improved. For this paper, we selected 3 solutions closest to the origin in terms of the Euclidean distance. In step 3, the SA algorithm is used to improve the previously selected solutions.

It is important to mention that since the SA method is single-objective, a solution is accepted as the best only if the two involved objectives (Accuracy classification error and number of learning parameters) are minimized simultaneously, which is equivalent to assigning a balanced weight to each of them. Finally, in step 4, the best solution (closest to the origin of the Euclidean space) is taken in accordance with the results obtained by the SA algorithm.

https://doi.org/10.3390/a19010047

4. Results and Discussion

For the experiments, two distinct publicly available image databases of coronary stenosis cases were used. The first image database contains 608 images corresponding to regions of coronary angiograms, which were properly validated by cardiology experts [46]. Each image size is 64×64 pixels. The number of positive and negative cases is 304 for each. Figure 9 illustrates a sample of 14 images containing positive and negative coronary stenosis cases.

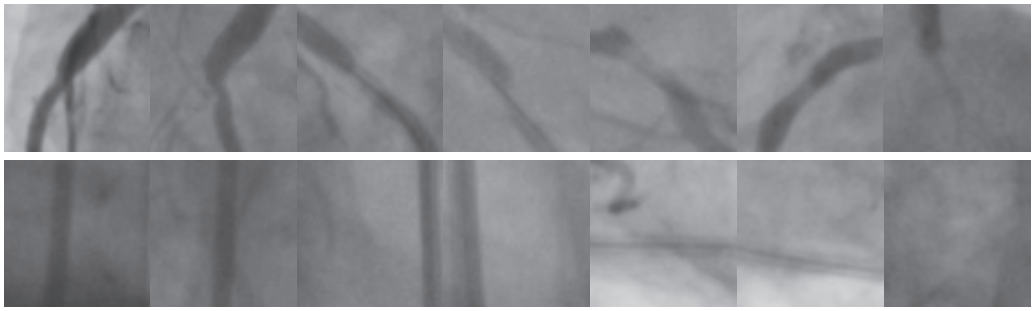


Figure 9. Image samples taken from the Stenosis608 [46] image database. The first row presents positive coronary stenosis cases. Consequently, the second row corresponds to negative coronary stenosis cases.

The second image database involves real and synthetic images corresponding to coronary stenosis regions [47]. The size of each image is 32×32 pixels. Figure 10 illustrates a sample of 21 images containing natural and synthetic coronary regions of positive and negative stenosis cases.

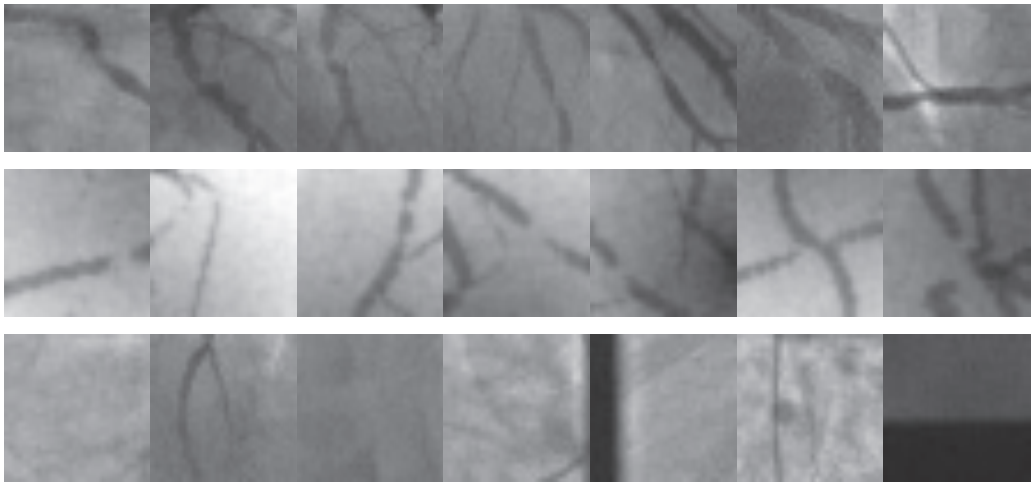


Figure 10. Image samples taken from the Antczak [47] image database. The first row contains natural images of coronary stenosis positive cases. The second row corresponds to synthetic images of positive cases. The third row corresponds to natural negative cases.

All the experiments were carried out on a computer with an Intel Core i7-9700K CPU with 3.60 GHz and 16 GB of RAM. It also contains an NVidia Titan RTX GPU with 24 GB of DDR6 VRAM, 4608 CUDA Cores and 576 Tensor Cores. The algorithms and search strategies were written in Matlab R2024a software.

For the NAS task, driven by the MOEA-D and the SA strategies, the Stenosis608 database was split using the holdout strategy with a proportion of 80–20% for training and testing, respectively. In addition, a cross-validation strategy with $k = 10$ was used on the training stage. It is important to mention that only the training partition was used for the

https://doi.org/10.3390/a19010047

NAS process. In addition, the MOEA-D implementation is derived from the Kalami [48] code base. The corresponding setup parameters required for the MOEA-D and the CNN training are described in Table 1.

Table 1. Parameter values for the MOEA-D and CNN training, which are used in the first stage of the proposed method for the NAS process.

MOEA-D	CNN Training Options
Population Size: 50	Max. Epochs: 20
Max. Generations: 100	Learning Rate: 1×10^{-3}
T (number of neighbors): 8	Optimizer: Adam
Crossover Points: 1	Validation Frequency: 30
Mutation Probability: 0.3	Shuffle: Every epoch
	Data Augmentation: Not used

Consequently, for the second step, the value of N, which corresponds to the number of taken solutions from the Pareto front produced by the MOEA-D, was established as 3. The corresponding parameter values for the SA and the CNN training options are described in detail in Table 2.

Table 2. Parameter values for the SA and CNN training, which are used in the second stage of the proposed method for the NAS process.

SA	CNN Training Options
T_{max}: 1.00	Max. Epochs: 200
T_{min}: 0.10	Learning Rate: 1×10^{-3}
T_{step}: 4.5×10^{-3}	Optimizer: Adam
	Validation Frequency: 30
	Shuffle: Every epoch
	Data Augmentation: Not used

As described in Table 2, the number of epochs for the CNN models training was increased considerably due to the single-solution nature of the SA method. Consequently, it is important to mention that all of the previous parameter values were established as a trade-off between the required time to obtain a response and an adequate fitness value.

After the NAS stage was concluded, the results show that the hybrid approach improved almost all of the solutions for all of the metaheuristics, as described in Table 3.

Based on data described in Table 3, the hybrid approach improves the CNN architecture in almost all cases. By comparing columns related with the initial learning parameters (ILPs) and final learning parameters (FLPs), there is evidence related to decreasing the number of learning parameters in almost all of the CNN models. However, due to the nature of the SA algorithm, in which the worst solutions are accepted in the first iterations and many goals are involved, the final result could be affected when an accepted solution improves only one of the goals. For instance, the solutions produced by the Hybrid MOPSO slightly improve the CNN complexity and the validation accuracy. However, its performance is below those of the other methods. Consequently, the first solution of the NSGA-II starts with a CNN architecture consisting of 1026 learnable parameters and a validation accuracy of 0.79. When the SA method ends, the number of parameters was improved because the value decreases to 646. In addition, the validation accuracy also decreases to 0.74 instead of increasing or keeping the initial value. In contrast, the second solution found by the MOEA-D algorithm was improved considerably, since it started with 11,910 learnable parameters and a validation accuracy of 0.90. Once the SA algorithm concluded, the number

https://doi.org/10.3390/a19010047

of learnable decreased to 3438 and the validation accuracy increased to 0.92, indicating that the final solution improved the two goals simultaneously. To complement the previous data, Figure 11 illustrates the Pareto front found by the MOEA-D.

Table 3. Performance comparison in terms of accuracy and number of CNN learnable parameters for the Hybrid MOPSO, Hybrid NSGA-II and Hybrid MOEA-D algorithms. Data columns are related to the method name, the number of solutions, the number of initial learning parameters (ILPs) before the improvement step, the number of final learning parameters (FLPs) after the SA improvement step, the initial validation accuracy (IVA) before the improvement step, and the final validation accuracy (FVA) after the SA improvement step.

Method	Solution	ILP	FLP	IVA	FVA
Hybrid MOPSO	1	14710	6700	0.87	0.87
	2	11700	4380	0.84	0.85
	3	17680	5090	0.86	0.88
Hybrid NSGA-II	1	1026	**646**	0.79	0.74
	2	13474	6618	0.81	0.90
	3	158	6030	0.64	0.90
Hybrid MOEA-D *	1	7166	2528	**0.90**	0.87
	2	11910	3498	**0.90**	**0.92**
	3	13070	3970	0.85	**0.92**

* Proposed method.

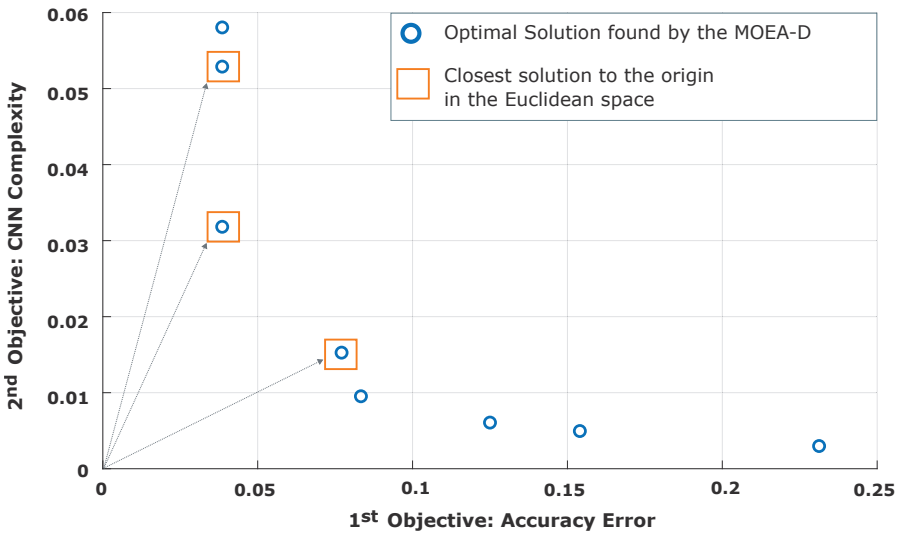


Figure 11. Representation of the Pareto front involving the optimal solutions found by the MOEA-D.

Figure 12 illustrates the CNN architecture for the best solution found by the proposed method.

The CNN architecture found by the proposed hybrid evolutionary multi-objective method is formed by 15 processing layers plus the classification one. As illustrated in Figure 12, the resulting architecture makes use of all common layer types, starting with a convolution in the second layer, whose corresponding kernel size is 5×5. Because of the stride value, which was established in 2 as a constant, an initial input downscale is performed, transforming the input size of 64×64 to 32×32 pixels, increasing computational efficiency and showing that the selected strategy was adequate. Consequently, a normalization layer and leaky ReLU layers were added, respectively. The fifth layer type is a convolution layer with a kernel size of 3×3 and 16 filters, which is followed by a ReLU layer. The seventh layer is of convolution type involving a kernel size of 3×3 and

https://doi.org/10.3390/a19010047

16 filters. At this point, and because of the fifth layer, the image data size was decreased to 8×8 pixels. Layers 8, 9 and 10 correspond to normalization, leaky ReLU and dropout types, respectively. The eleventh layer is a maxpool with no padding to assure that all input pixels are covered (including edges) in the input feature map. The cost of this consideration is that the image size is not reduced. However, this behavior is compensated by the stride value in the convolution layers. The twelfth layer corresponds to a convolution using 16 filters. Because of the previous maxpool layer and the fact that in the last part, the maxpool size is 2×2, a kernel size of 1×1 is used, which decreases the input size from 8×8 to 4×4. The thirteenth layer transforms the input from the last layer, whose size is $4 \times 4 \times 16$, to a data vector of 256 values. Consequently, a fully connected layer consisting of *two* neurons (the number of classes) is added at the fourteenth position. Finally, an activation softmax, plus the classification layer, are added.

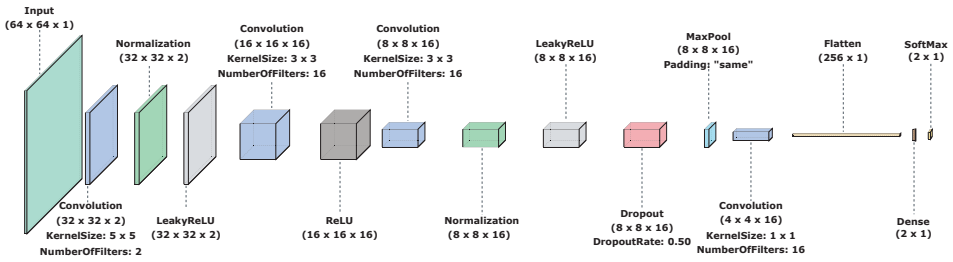


Figure 12. CNN lightweight architecture designed automatically by the proposed method based on the MOEA-D and the SA algorithm.

To assure model stability, the CNN architecture formed by each strategy was trained using 488 random images (≈80%) from the Stenosis608 image database. The remaining 120 instances (≈20%) were used for testing on a later step. In this test, only training images were involved, using a cross-validation with $k = 10$ on 30 independent trials. The number of epochs was established in 1000 to perform an exhaustive training of the models to increase its reliability and efficiency in terms of the Accuracy metric. The corresponding results are described in Table 4.

Table 4. Statistical analysis of the training performance in terms of the Accuracy metric and a cross-validation process with $k = 10$ for the proposed method and distinct evolutionary-based methods on 30 independent trials, using 488 images from the Stenosis608 image database. In addition, the number of learning parameters for each model is described in the Learners column, which is expressed in units of millions (M) and thousands (K).

Method	Min	Max	Median	Avg	StdDev	Learners
MOPSO [17]	0.81	0.86	0.84	0.83	0.02	14.6 K
NSGA-II [49]	0.80	0.89	0.87	0.85	0.03	9.6 K
MOEAD [35]	0.80	0.89	0.85	0.84	0.03	12.6 K
Hybrid MOPSO	0.81	0.91	0.86	0.86	0.03	6.7 K
Hybrid NSGA-II	0.85	**0.94**	0.88	0.89	0.04	6.6 K
Proposed Method	**0.90**	**0.94**	**0.92**	**0.92**	**0.01**	**3.5 K**

Based on data described in Table 4, the proposed method achieved the highest classification rates in terms of the Accuracy metric. However, there is evidence of the impact of multi-objective search strategies, since the number of learning parameters in the models decreases significantly to the order of thousands. Consequently, all models were stable, since the standard deviation of the classification accuracy was low for all cases. In order

https://doi.org/10.3390/a19010047

to measure reliability, the previous models were tested against additional classification metrics such as Precision, Recall, F1-Score and Jaccard Similarity Coefficient, as described in Section 2.4, using the trained model with the highest classification rate from the previous test. The corresponding results are presented in Table 5.

Table 5. Comparison of classification performance rates achieved by the proposed method and distinct CNN models from the literature, using 120 testing instances from the Stenosis608 image database. The classification performance was measured in terms of the Accuracy (Acc), Precision (Pres), Recall, F1-Score (F1) and Jaccard Similarity Coefficient (JSC). Additionally, the number of learning parameters for each model is described in the Learners column and expressed in units of millions (M) and thousands (K).

Method	Acc	Pres	Recall	F1	JSC	Learners
ResNet50 [26]	0.81	0.85	0.75	0.80	0.67	25 M
Inception-v3 [27]	0.72	0.74	0.67	0.70	0.54	23.8 M
VGG-16 [28]	0.84	0.87	0.80	0.84	0.72	138 M
CNN-16C [47]	0.86	0.91	0.80	0.85	0.74	131 K
MobNetV2 [29]	0.78	0.80	0.75	0.78	0.63	2.2 M
EffNetB0 [30]	0.87	0.88	0.85	0.86	0.76	4 M
Franco et al. [20]	0.92	0.93	0.90	0.92	0.84	19 M
MOPSO [17]	0.86	0.88	0.84	0.86	0.75	14.6 K
NSGA-II [49]	0.89	0.92	0.86	0.89	0.80	9.6 K
MOEAD [35]	0.89	0.93	0.84	0.88	0.79	12.6 K
Hybrid MOPSO	0.91	1.00	0.82	0.90	0.82	6.7 K
Hybrid NSGA-II	**0.94**	**1.00**	0.88	**0.94**	**0.89**	6.6 K
Proposed method	**0.94**	0.97	**0.92**	**0.94**	**0.89**	**3.5 K**

According to the results presented in Table 5, there is evidence related to the effect of the solution improvement stage performed by the SA algorithm, in which some of the best solutions produced by the multi-objective techniques are taken and improved individually. However, even when some of the compared methods achieved high rates in metrics such as Accuracy, Recall and F1-Score, in clinical practice, they fail to accurately classify real positive disease cases, which is critical. As a consequence, the use of the Jaccard Similarity Coefficient metric was important, because it inflicts a high penalty to positive cases classified as negative (False Negatives). By considering this, the proposed method achieved the best classification results compared to the other methods.

To assure the reliability of the designed CNN architecture, a second test was performed using the Antczak image database, which contains natural and synthetic artery coronary stenosis images. For the first test, only natural images were considered, forming a dataset of 244 instances in a balanced way between positive and negative coronary stenosis cases. The dataset was split into 194 and 50 instances of training and testing, respectively. In Table 6, a statistical analysis of the training accuracy, based on the classification performance, is presented. Each model was trained using the cross-validation strategy with $k = 10$ on 30 independent trials.

Consequently, each architecture was tested by taking its corresponding model, which achieved the highest validation classification Accuracy in the previous test. The pretrained models were tested using the image testing partition consisting of 50 images. The obtained results are presented in Table 7.

https://doi.org/10.3390/a19010047

Table 6. Statistical analysis of the training performance in terms of the Accuracy metric and a cross-validation process with $k = 10$ for the proposed method and distinct evolutionary-based methods on 30 independent trials, using 194 natural images from the Antczak [47] image database. In addition, the number of learning parameters for each model is described in the Learners column, which is expressed in units of millions (M) and thousands (K).

Method	Min	Max	Median	Avg	StdDev	Learners
ResNet50 [26]	0.72	0.79	0.75	0.75	0.02	25 M
Inception-v3 [27]	0.66	0.70	0.68	0.68	**0.01**	23.8 M
VGG-16 [28]	0.79	0.82	0.80	0.80	**0.01**	138 M
CNN-16C [47]	0.80	0.84	0.82	0.82	**0.01**	131 K
MobNetV2 [29]	0.72	0.78	0.74	0.74	**0.01**	2.2 M
EffNetB0 [30]	0.74	0.78	0.76	0.76	**0.01**	4 M
Franco et al. [20]	0.81	0.86	0.83	0.83	0.02	19 M
MOPSO [17]	0.63	0.71	0.66	0.66	0.02	14.6 K
NSGA-II [49]	0.75	0.79	0.77	0.77	**0.01**	9.6 K
MOEAD [35]	0.78	0.81	0.80	0.80	**0.01**	12.6 K
Hybrid MOPSO	0.80	0.85	0.82	0.82	0.01	6.7 K
Hybrid NSGAII	0.84	0.89	0.86	0.86	0.01	6.6 K
Proposed method	**0.86**	**0.90**	**0.89**	**0.88**	**0.01**	**3.5 K**

Table 7. Comparison of classification performance rates achieved by the proposed method and distinct CNN models from the literature using the testing set, which consists of 50 images. The tests were performed using the the model with the maximum Accuracy rate on validation according to the previous statistical analysis. The classification performance was measured in terms of the Accuracy (Acc), Precision (Pres), Recall, F1-Score (F1) and Jaccard Similarity Coefficient (JSC) metrics. In addition, the number of learning parameters for each model is described in the Learners column and expressed in units of millions (M) and thousands (K).

Method	Acc	Pres	Recall	F1	JSC	Learners
ResNet50 [26]	0.77	0.80	0.73	0.76	0.61	25 M
Inception-v3 [27]	0.68	0.70	0.64	0.67	0.50	23.8 M
VGG-16 [28]	0.80	0.86	0.72	0.78	0.64	138 M
CNN-16C [47]	0.82	0.86	0.76	0.81	0.68	131 K
MobNetV2 [29]	0.74	0.75	0.72	0.74	0.58	2.2 M
EffNetB0 [30]	0.76	0.78	0.72	0.75	0.60	4 M
Franco et al. [20]	0.84	0.87	0.80	0.83	0.71	19 M
MOPSO [17]	0.68	0.70	0.64	0.67	0.50	14.6 K
NSGA-II [49]	0.78	0.79	0.76	0.78	0.63	9.6 K
MOEAD [35]	0.80	0.83	0.76	0.80	0.66	12.6 K
Hybrid MOPSO	0.82	0.90	0.72	0.80	0.67	6.7 K
Hybrid NSGA-II	**0.88**	**0.95**	0.80	0.87	0.77	6.6 K
Proposed method	**0.88**	0.88	**0.88**	**0.88**	**0.79**	**3.4 K**

Based on the results presented in Table 7, the proposed method achieved the highest rates in almost all metrics. It is difficult for complex CNN architectures to achieve correct training and classification rates with image databases having a low number of instances. Consequently,

https://doi.org/10.3390/a19010047

the design of lightweight CNN architectures was adequate because they achieve superior classification results with datasets containing a low number of instances. Finally, a dataset containing 2788 images was formed. Since the Antczak coronary stenosis database contains 1394 and 122 natural images corresponding to negative and positive cases, respectively, the database was equilibrated by adding 1272 synthetic images of positive cases using the author's method to generate them. Similar to the previous test, the image dataset was split in a proportion of ≈80–20% for training and testing, which corresponds to 2240 and 548 images, respectively. Using the images from training partition, each model was trained using the cross-validation strategy with $k = 10$ on 30 independent trials. Based on the obtained results in each trial, a corresponding statistical analysis is presented in Table 8.

Table 8. Statistical analysis of the training performance in terms of the Accuracy metric and a cross-validation process with $k = 10$ for the proposed method and distinct evolutionary-based methods on 30 independent trials. The training partition is used, which consists of 2240 images from the Antczak [47] image database, involving natural and synthetic coronary stenosis cases. In addition, the number of learning parameters for each model is described in the Learners column and expressed in units of millions (M) and thousands (K).

Method	Min	Max	Median	Avg	StdDev	Learners
ResNet50 [26]	0.80	0.87	0.82	0.83	0.02	25 M
Inception-v3 [27]	0.81	0.85	0.83	0.83	**0.01**	23.8 M
VGG-16 [28]	0.86	0.92	0.90	0.89	0.02	138 M
CNN-16C [47]	**0.92**	0.95	**0.94**	**0.94**	**0.01**	131 K
MobNetV2 [29]	0.84	0.87	0.85	0.85	**0.01**	2.2 M
EffNetB0 [30]	0.90	0.95	0.93	0.93	0.02	4 M
Franco et al. [20]	0.90	0.95	0.93	0.93	**0.01**	19 M
MOPSO [17]	0.86	0.89	0.87	0.87	**0.01**	14.6 K
NSGA-II [49]	0.88	0.92	0.91	0.90	**0.01**	9.6 K
MOEAD [35]	0.88	0.92	0.91	0.90	**0.01**	12.6 K
Hybrid MOPSO	0.90	0.94	0.92	0.92	0.01	6.7 K
Hybrid NSGAII	0.91	0.96	**0.94**	**0.94**	**0.01**	6.6 K
Proposed method	**0.92**	**0.97**	**0.94**	**0.94**	**0.01**	**3.5 K**

Based on the data presented in Table 8, the increase in the number of instances improved the training validation accuracy considerably for all CNN architectures. However, the CNN model found by the proposed method overpassed or was equal to the rest of them in all measurements.

According to the data described in Table 9, the CNN architectures produced by all the compared strategies increased their classification rate under all of the measured metrics. This is an important fact in traditional CNN architectures such as ResNet50 and VGG-16 because they tend to improve classification rates by increasing the size of the training image dataset. Consequently, lightweight models such as MobileNetV2 [29] and EfficientNetB0 [30] also increased on all classification metrics considerably, using the testing set. However, the proposed methodology, based on the MOEA-D and the SA methods, performs considerably better than the other compared techniques in all metrics, which provides evidence about the CNN model's reliability when analyzing distinct coronary stenosis image databases.

https://doi.org/10.3390/a19010047

Table 9. Comparison of classification performance rates achieved by the proposed method and distinct CNN models from the literature using the testing set, which consists of 548 images. The tests were performed using the the model with the maximum Accuracy rate on validation according to the previous statistical analysis. The classification performance was measured in terms of the Accuracy (Acc), Precision (Pres), Recall, F1-Score (F1) and Jaccard Similarity Coefficient (JSC) metrics. In addition, the number of learning parameters for each model is described in the Learners column and expressed in units of millions (M) and thousands (K).

Method	Acc	Pres	Recall	F1	JSC	Learners
ResNet50 [26]	0.80	0.83	0.77	0.80	0.66	25 M
Inception-v3 [27]	0.78	0.77	0.79	0.78	0.64	23.8 M
VGG-16 [28]	0.86	0.87	0.84	0.86	0.75	138 M
CNN-16C [47]	0.89	0.91	0.88	0.89	0.81	131 K
MobNetV2 [29]	0.81	0.83	0.77	0.80	0.67	2.2 M
EffNetB0 [30]	0.89	0.92	0.86	0.89	0.80	4 M
Franco et al. [20]	0.92	0.93	0.90	0.92	0.84	19 M
MOPSO [17]	0.81	0.83	0.79	0.81	0.68	14.6 K
NSGA-II [49]	0.89	0.91	0.86	0.88	0.79	9.6 K
MOEAD [35]	0.87	0.89	0.84	0.87	0.76	12.6 K
Hybrid MOPSO	0.92	0.93	0.90	0.92	0.86	6.7 K
Hybrid NSGA-II	0.93	0.94	0.91	0.92	0.86	6.6 K
Proposed method	**0.97**	**0.97**	**0.96**	**0.97**	**0.93**	**3.4 K**

In order to explore details about the image features that allow the proposed CNN model to classify positive and negative coronary stenosis cases, the Grad-CAM technique was used [50]. Figure 13 presents distinct images from the Stenosis608 database, consisting of 21 positive coronary stenosis cases with relevant features marked in a heatmap by the Grad-CAM method.

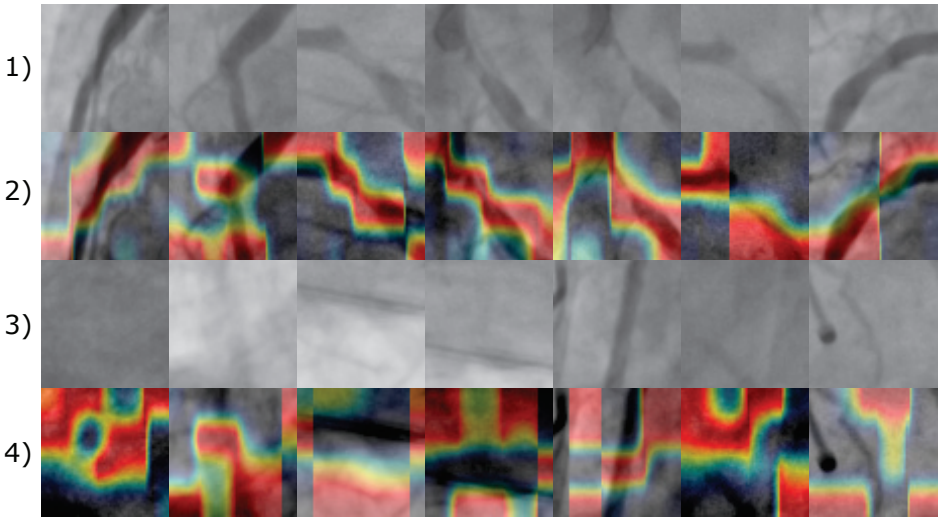


Figure 13. Image samples of the Stenosis608 image database. First and third rows illustrates the original images of positive and negative coronary stenosis cases, respectively. The second and fourth rows display the corresponding Grad-CAM heatmaps, where a color gradient from red to blue represents feature importance; red regions highlight the most influential features, while blue denotes areas of minimal contribution to the classification.

https://doi.org/10.3390/a19010047

According to Figure 13, the image features leading the CNN model to produce a positive classification result are almost related to the pixels corresponding to arteries. Consequently, for negative stenosis classification results, the model involves features from arteries and non-artery pixels. In Figure 14, the corresponding Grad-CAM results from the Antczak image database are also presented.

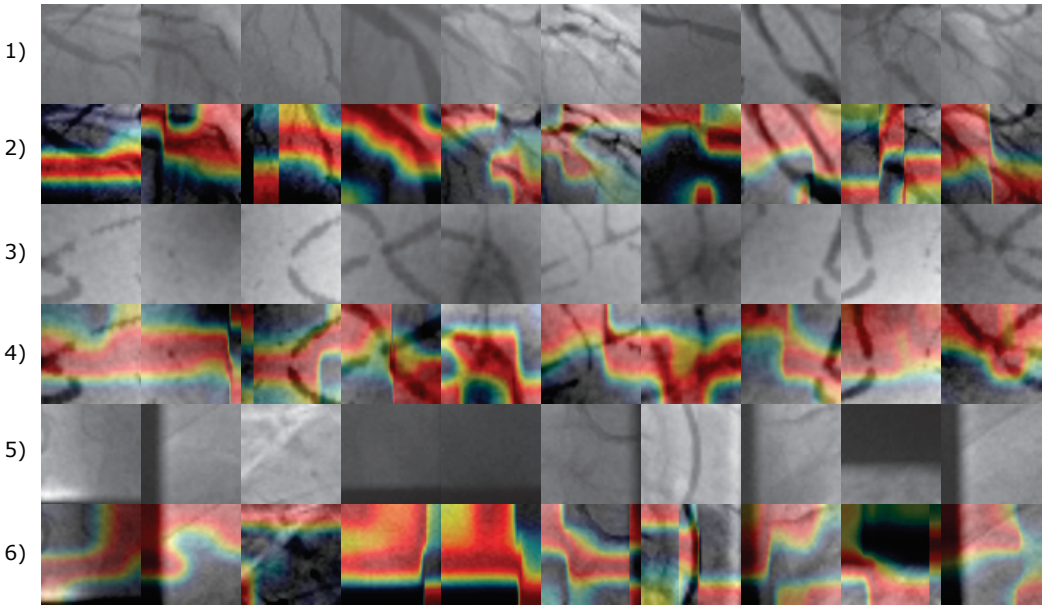


Figure 14. Image samples of the Antczak image database involving natural and synthetic images. The first row illustrates the original images of positive coronary stenosis cases. The second row illustrates the corresponding Grad-CAM heatmaps. The third and fourth rows illustrate the original and the corresponding Grad-CAM heatmap images for the synthetically generated instances of positive coronary stenosis cases. The fifth and sixth rows corresponds to natural images of negative coronary stenosis cases with their respective Grad-CAM heatmap. The heatmap is represented by a color gradient from red to blue denoting feature importance. Red regions highlight the most influential features, while blue denotes areas of minimal contribution to the classification.

Based on the images presented in Figure 14, the Grad-CAM shows evidence that artery pixels were also relevant to the identification of features related to positive coronary stenosis cases. Consequently, non-artery information form the image was relevant to the identification of negative coronary stenosis cases.

Finally, the required time for each stage of the experiments was interesting. For instance, in the initial search step, training a single solution consumed an average of ≈85 s: when multiplied by the number of individuals (50) and by the number of generations (100), the total required time was ≈5 days. For the second step, after the MOEA-D method produced the solution set, the SA takes ≈2.30 h to improve the *three* best solutions found in the previous step. The required time in this step was considerably low when compared with the initial search step. Moreover, once the model is trained, the classification of a single image takes an average time of 0.009 s, which is a considerably low value. In Appendix A, the corresponding Matlab code to build, train and perform a single test of the CNN architecture of the proposed method is presented.

5. Conclusions

In this paper, a novel method based on a Hybrid-Evolutionary Multi-Objective strategy for the automatic design of lightweight CNN architectures was presented. The main contribution is a methodology that can be applied for the automatic design of CNN models that are

https://doi.org/10.3390/a19010047

able to learn from small datasets without the need for data augmentation or transfer learning mechanisms. This aspect is relevant in distinct contexts such as medicine, where the image databases are scarce due to ethical and legal restrictions. The proposed method starts with an exhaustive search stage driven by the MOEA-D strategy over distinct combinations of layer types in order to produce an optimal CNN architecture. In the process, all produced models are validated to be coherent between the distinct layers that are added and connected. For instance, two consecutive ReLU layers did not improve the model, so the second was removed from the solution. Another type of inconsistency can occur if all of the convolution activation values of an encoded solution have a value of 0, which indicates the absence of convolution operations in the CNN model. In this case, the solution is corrected by adding a convolution layer. After the first stage of the search process concluded, the *three* best solutions are selected from the Pareto front and improved independently using the SA algorithm. The hybrid approach produced a CNN architecture consisting of 15 layers and 3498 learning parameters, which is considered lightweight when compared with other such as ResNet50 or VGG-16 in which the number of learners is 25 and 138 million, respectively. As for the model architecture's effectiveness, a public database consisting of 608 images corresponding to regions of coronary angiographies was used. In the test stage, we achieved classification rates of 0.94, 0.97, 0.92, 0.94 and 0.89 in terms of the Accuracy, Precision, Recall, F1-Score and Jaccard Similarity Coefficient, which achieved superior results to those of the other techniques from the literature. In a consecutive step, an independent image database consisting of natural and synthetic images was used to test the achieved model architecture. The test started using an image dataset of 244 instances, containing only natural images. The proposed method achieved classification rates of 0.88, 0.88, 0.88, 0.88 and 0.79 for Accuracy, Precision, Recall, F1-Score and Jaccard Similarity Coefficient metrics, respectively. In the last test, natural and synthetic images were used to form an image bank of 2788 instances. Using this image database, the CNN model achieved classification rates of 0.97, 0.97, 0.96, 0.97 and 0.93 under the same metrics. Moreover, once the model is trained, the required average time to classify a single image was 0.009 seconds. The main disadvantage of the proposed method is the required time to achieve a solution. The NAS stage is very time consuming, considering that in the population-based search methods, each individual (solution) corresponds to a CNN architecture that must be trained independently. Regarding this point, it is important to mention that the proposed method is intended to work with small image databases, which is a common fact in medical context. Consequently, its use on large image databases could considerably increase the required time to produce an optimal result. Another disadvantage of the representation of CNN architectures in the proposed method is the design of only feed-forward architectures. As a consequence, future work could be focused on performing non-adjacent connections between the distinct layers of the model and implementing a strategy to decrease the required time during the NAS stage. However, according to the presented results, the proposed method could be applied in the automatic design of lightweight CNN architectures for other medical databases involving retina, chest, lung or other human organ images. Finally, based on the obtained results, it can be concluded that the proposed method can be applied to design lightweight CNN models to assist in clinical practice as a part of information systems for computer-aided diagnosis and decision-making support.

Author Contributions: Conceptualization, M.-A.G.-R. and I.C.-A.; Formal analysis, M.-A.G.-R. and I.C.-A. and E.-G.G.-d.-P.; Investigation, A.H.-A. and E.-G.G.-d.-P.; Methodology, M.-A.G.-R., I.C.-A. and J.-M.L.-H.; Project administration, A.H.-A. and J.-M.L.-H.; Software, M.-A.G.-R., A.H.-A. and J.-M.L.-H.; Validation, M.-A.G.-R., I.C.-A. and E.-G.G.-d.-P.; Visualization, M.-A.G.-R., E.-G.G.-d.-P. and J.-M.L.-H.; Writing—original draft, M.-A.G.-R. and I.C.-A. All authors have read and agreed to the published version of the manuscript.

https://doi.org/10.3390/a19010047

Funding: This research received no external funding.

Data Availability Statement: Research images and achieved results are accessible at: https://github.com/mgil-utleon/lightweight_cnn_coronary_stenosis, accessed on 1 January 2026.

Acknowledgments: The authors thank Centro de Investigación en Matemáticas A.C. along with the Instituto de Innovación, Ciencia y Emprendimiento para la Competitividad para el Estado de Guanajuato (IDEA GTO), under the project "Supercómputo como impulsor de colaboraciones Academia-Industria"; Universidad Tecnológica de León. This work was supported by SECIHTI under Project IxM-SECIHTI No. 3097-7185.

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

Abbreviations

The following abbreviations are used in this manuscript:

Acc	Accuracy Metric
Avg	Average
BatchNorm	CNN Batch Normalization Layer
EDA	Estimation Distribution Algorithm
IVA	Initial Validation Accuracy
Conv2D	CNN Convolution 2-D Layer for single color channel images
CNN	Convolutional Neural Network
CPU	Central Unit Processing
F1	F1-Score
FLP	Final Learning Parameters
FVA	Final Validation Accuracy
GA	Genetic Algorithm
GHz	Gigahertz
ILP	Initial Learning Parameters
JSC	Jaccard Similarity Coefficient
Max	Maximum
Min	Minimum
MOEA-D	Multi-Objective Evolutionary Algorithm based on Decomposition
MOPSO	Multi-Objective Particle Swarm Optimization
NAS	Neural Architecture Search
NSGA-II	Non-dominated Sorting Genetic Algorithm version 2
Pres	Precision
RAM	Random Access Memory
ReLU	Rectified Linear Unit
SA	Simulated Annealing
StdDev	Standard Deviation

Appendix A. Matlab Code for the CNN Model Obtained with the Hybrid MOEAD-SA Algorithm

The corresponding code is available at: https://github.com/mgil-utleon/lightweight_cnn_coronary_stenosis, accessed on 4 January 2020.

```
% Prepare the training dataset:
dir_training = 'path/to/your/training/dir/';
dataset_training = imageDatastore(dir_training, ...
                                  "IncludeSubfolders", true, ...
                                  "LabelSource", "foldernames");
```

 https://doi.org/10.3390/a19010047

```
% Split the training dataset for validation during training:
[imgs_train, imgs_val] = splitEachLabel(dataset_training, ...
                                        0.9, ...
                                        0.1, ...
                                        'randomized');

% Prepare the testing dataset:
dir_testing = 'path/to/your/testing/dir/';

% Load an image to extract its size:
img = imread([dir_training, 'Positive/xyz_Positive.pgm']);
image_size = size(img);

% Set the max train epochs:
max_train_epochs = 1000;

% Configure training parameters:
cnn_train_options = trainingOptions('adam', ...
                                    'InitialLearnRate',    0.001, ...
                                    'MaxEpochs',       max_train_epochs, ...
                                    'MiniBatchSize',       16, ...
                                    'Shuffle',             'every-epoch', ..
                                    'ValidationData',      imgs_val, ...
                                    'ValidationFrequency', 30, ...
                                    'Verbose',             true);

% Build the CNN model found by the proposed method:
cnnmodel=[
            imageInputLayer(image_size), ...
            convolution2dLayer(5, 2, 'Padding', 'same', 'Stride', 2), ...
            batchNormalizationLayer(), ...
            leakyReluLayer(), ...
            convolution2dLayer(3, 16, 'Padding', 'same', 'Stride', 2), ...
            reluLayer(), ...
            convolution2dLayer(3, 16, 'Padding', 'same', 'Stride', 2), ...
            batchNormalizationLayer(), ...
            leakyReluLayer(), ...
            dropoutLayer(0.5), ...
            maxPooling2dLayer([2, 2], 'Padding', 'same'), ...
            convolution2dLayer(1, 16, 'Padding', 'same', 'Stride', 2), ...
            flattenLayer(), ...
            fullyConnectedLayer(2), ...
            softmaxLayer(), ...
            classificationLayer()
        ];

% Uncomment next line to show the full model information:
%analyzeNetwork(cnnmodel(1:numel(cnnmodel)-1));
```


https://doi.org/10.3390/a19010047

```
% Train the CNN model:
[cnnmodel_trained, t_info] = trainNetwork(imgs_train, ...
                                          cnnmodel, ...
                                          cnn_train_options);

% Test the model with an image:
predicted_label = classify(cnnmodel_trained, img);
```

References

1. Kay, F.U.; Canan, A.; Kukkar, V.; Hulsey, K.; Scanio, A.; Fan, C.; Hallam, K.A.; Gulsun, M.A.; Wels, M.; Schoebinger, M.; et al. Diagnostic accuracy of on-premise automated coronary CT angiography analysis based on Coronary Artery Disease Reporting and Data System 2.0. *Radiology* **2025**, *315*, e242087. [CrossRef]
2. Athanasiou, L.S.; Fotiadis, D.I.; Michalis, L.K. *Atherosclerotic Plaque Characterization Methods Based on Coronary Imaging*; Academic Press: Cambridge, MA, USA, 2017.
3. Chowdhary, C.L.; Acharjya, D. Segmentation and Feature Extraction in Medical Imaging: A Systematic Review. *Procedia Comput. Sci.* **2020**, *167*, 26–36. [CrossRef]
4. Tessmann, M.; Vega-Higuera, F.; Fritz, D.; Scheuering, M.; Greiner, G. Multi-scale feature extraction for learning-based classification of coronary artery stenosis. In Proceedings of the Medical Imaging 2009: Computer-Aided Diagnosis, Lake Buena Vista, FL, USA, 7–12 February 2009; Karssemeijer, N., Giger, M.L., Eds.; International Society for Optics and Photonics, SPIE: Bellingham, WA, USA, 2009; Volume 7260. [CrossRef]
5. Cervantes-Sanchez, F.; Cruz-Aceves, I.; Hernandez-Aguirre, A. Automatic detection of coronary artery stenosis in X-ray angiograms using Gaussian filters and genetic algorithms. *AIP Conf. Proc.* **2016**, *1747*, 020005. [CrossRef]
6. Sameh, S.; Azim, M.A.; AbdelRaouf, A. Narrowed Coronary Artery Detection and Classification using Angiographic Scans. In Proceedings of the 2017 12th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 19–20 December 2017; pp. 73–79. [CrossRef]
7. Wan, T.; Feng, H.; Tong, C.; Li, D.; Qin, Z. Automated identification and grading of coronary artery stenoses with X-ray angiography. *Comput. Methods Programs Biomed.* **2018**, *167*, 13–22. [CrossRef] [PubMed]
8. Gil-Rios, M.A.; Cruz-Aceves, I.; Hernandez-Aguirre, A.; Moya-Albor, E.; Brieva, J.; Hernandez-Gonzalez, M.A.; Solorio-Meza, S.E. High-Dimensional Feature Selection for Automatic Classification of Coronary Stenosis Using an Evolutionary Algorithm. *Diagnostics* **2024**, *14*, 268. [CrossRef]
9. Öztürk, S. *Convolutional Neural Networks for Medical Image Processing Applications*, 1st ed.; CRC Press: Boca Raton, FL, USA, 2022. [CrossRef]
10. Tu, L.; Deng, Y.; Chen, Y.; Luo, Y. Accuracy of deep learning in the differential diagnosis of coronary artery stenosis: A systematic review and meta-analysis. *BMC Med. Imaging* **2024**, *24*, 243. [CrossRef]
11. Maji, P.; Mullins, R. On the Reduction of Computational Complexity of Deep Convolutional Neural Networks. *Entropy* **2018**, *20*, 305. [CrossRef]
12. Chlap, P.; Min, H.; Vandenberg, N.; Dowling, J.; Holloway, L.; Haworth, A. A review of medical image data augmentation techniques for deep learning applications. *Med. Imaging Radiat. Oncol.* **2021**, *65*, 545–563. [CrossRef]
13. Garcea, F.; Serra, A.; Lamberti, F.; Morra, L. Data augmentation for medical imaging: A systematic literature review. *Comput. Biol. Med.* **2023**, *152*, 106391. [CrossRef]
14. Murdoch, W.J.; Singh, C.; Kumbier, K.; Abbasi-Asl, R.; Yu, B. Definitions, methods, and applications in interpretable machine learning. *Proc. Natl. Acad. Sci. USA* **2019**, *116*, 22071–22080. [CrossRef] [PubMed]
15. 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]
16. Weng, Y.; Zhou, T.; Liu, L.; Xia, C. Automatic Convolutional Neural Architecture Search for Image Classification Under Different Scenes. *IEEE Access* **2019**, *7*, 38495–38506. [CrossRef]
17. Jiang, J.; Han, F.; Ling, Q.; Wang, J.; Li, T.; Han, H. Efficient network architecture search via multiobjective particle swarm optimization based on decomposition. *Neural Netw.* **2020**, *123*, 305–316. [CrossRef]
18. Elsken, T.; Metzen, J.H.; Hutter, F. Simple And Efficient Architecture Search for Convolutional Neural Networks. *arXiv* **2017**, arXiv:1711.04528. [CrossRef]
19. Krizhevsky, A. Learning Multiple Layers of Features from Tiny Images. 2009. Available online: https://www.cs.utoronto.ca/~kriz/learning-features-2009-TR.pdf (accessed on 17 November 2025).

https://doi.org/10.3390/a19010047

20. Franco-Gaona, E.; Avila-Garcia, M.S.; Cruz-Aceves, I. Automatic Neural Architecture Search Based on an Estimation of Distribution Algorithm for Binary Classification of Image Databases. *Mathematics* **2025**, *13*, 605. [CrossRef]
21. Pham, H.; Guan, M.; Zoph, B.; Le, Q.; Dean, J. Efficient Neural Architecture Search via Parameters Sharing. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Proceedings of Machine Learning Research; Dy, J., Krause, A., Eds.; PMLR: Cambridge, MA, USA, 2018; Volume 80, pp. 4095–4104.
22. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.
23. Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; Le, Q.V. MnasNet: Platform-Aware Neural Architecture Search for Mobile. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
24. Zhang, D.; Chen, K.; Zhao, Y.; Yang, B.; Yao, L.; Jensen, C.S. Design Automation for Fast, Lightweight, and Effective Deep Learning Models: A Survey. *arXiv* **2022**, arXiv:2208.10498. [CrossRef]
25. Hafiz, A.M. A Survey on Light-Weight Convolutional Neural Networks: Trends, Issues and Future Scope. *J. Mob. Multimed.* **2023**, *19*, 1277–1297. [CrossRef]
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27-30 June 2016; pp. 770–778. [CrossRef]
27. 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 IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
28. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556. [CrossRef]
29. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *arXiv* **2019**, arXiv:1801.04381. http://arxiv.org/abs/1801.04381. [CrossRef]
30. Tan, M.; Le, Q.V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2020**, arXiv:1905.11946. . [CrossRef]
31. Ovalle-Magallanes, E.; Avina-Cervantes, J.G.; Cruz-Aceves, I.; Ruiz-Pinales, J. Transfer Learning for Stenosis Detection in X-ray Coronary Angiography. *Mathematics* **2020**, *8*, 1510. [CrossRef]
32. Kirichenko, P.; Ibrahim, M.; Balestriero, R.; Bouchacourt, D.; Vedantam, S.R.; Firooz, H.; Wilson, A.G. Understanding the detrimental class-level effects of data augmentation. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 17498–17526.
33. Stacke, K.; Eilertsen, G.; Unger, J.; Lundström, C. A Closer Look at Domain Shift for Deep Learning in Histopathology. *arXiv* **2019**, arXiv:1909.11575. [CrossRef]
34. Castro, D.C.; Walker, I.; Glocker, B. Causality matters in medical imaging. *Nat. Commun.* **2020**, *11*, 3673. [CrossRef]
35. Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [CrossRef]
36. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [CrossRef] [PubMed]
37. LeCun, Y.; Boser, B.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.; Jackel, L.D. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
38. Sun, J.; Cao, X.; Liang, H.; Huang, W.; Chen, Z.; Li, Z. New Interpretations of Normalization Methods in Deep Learning. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 5875–5882. [CrossRef]
39. Zhou, A.; Zhang, Q.; Zhang, G. A multiobjective evolutionary algorithm based on decomposition and probability model. In Proceedings of the 2012 IEEE Congress on Evolutionary Computation, Brisbane, QLD, Australia, 10–15 June 2012; pp. 1–8. [CrossRef]
40. Zitzler, E.; Deb, K.; Thiele, L. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evol. Comput.* **2000**, *8*, 173–195. [CrossRef]
41. Kobayashi, Y.; Aranha, C.; Sakurai, T. Dynamic adaptation of decomposition vector set size for MOEA/D. In Proceedings of the GECCO'21: Genetic and Evolutionary Computation Conference Companion, Lille France, 10–14 July 2021; pp. 181–182. [CrossRef]
42. Zhao, H.; Yan, L.; Hou, Z.; Lin, J.; Zhao, Y.; Ji, Z.; Wang, Y. Error Analysis Strategy for Long-Term Correlated Network Systems: Generalized Nonlinear Stochastic Processes and Dual-Layer Filtering Architecture. *IEEE Internet Things J.* **2025**, *12*, 33731–33745. [CrossRef]
43. Zhao, Y.; Cui, J.; Li, Z.; Jiao, S.; Jiajia, L. CUTHERMO: Understanding GPU Memory Inefficiencies with Heat Map Profiling. *arXiv* **2025**, arXiv:2507.18729. [CrossRef]

https://doi.org/10.3390/a19010047

44. Zafar, A.; Aamir, M.; Mohd-Nawi, N.; Arshad, A.; Riaz, S.; Alruban, A.; Dutta, A.K.; Almotairi, S. A Comparison of Pooling Methods for Convolutional Neural Networks. *Appl. Sci.* **2022**, *12*, 8643. [CrossRef]
45. Jost-Tobias, S.; Dosovitskiy, A.; Brox, T.; Riedmiller, M. Striving for Simplicity: The All Convolutional Net. *arXiv* **2015**, arXiv:1412.6806. [CrossRef]
46. Gil-Rios, M.A.; Cruz-Aceves, I.; Hernandez-Aguirre, A.; Hernandez-Gonzalez, M.A.; Solorio-Meza, S.E. Improving Automatic Coronary Stenosis Classification Using a Hybrid Metaheuristic with Diversity Control. *Diagnostics* **2024**, *14*, 2372. [CrossRef]
47. Antczak, K.; Liberadzki, Ł. Stenosis Detection with Deep Convolutional Neural Networks. In Proceedings of the MATEC Web of Conferences, Mallorca Island, Spain, 14–17 July 2018; EDP Sciences: Ulis, France, 2018; Volume 210, p. 04001. [CrossRef]
48. Kalami-Heris, M. MOEA/D in MATLAB. 2015 Available online: https://la.mathworks.com/matlabcentral/fileexchange/52873-multi-objective-evolutionary-algorithm-based-on-decomposition-moea-d (accessed on 16 December 2025).
49. 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]
50. Selvaraju, R.R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; Batra, D. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Int. J. Comput. Vis.* **2019**, *128*, 336–359. [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/a19010047

algorithms

MDPI

Article

Optimizing Facial Muscle Activation Features for Emotion Recognition: A Metaheuristic Approach Using Inner Triangle Points

Erick G. G. de Paz [1], Ivan Cruz-Aceves [2,*], Arturo Hernandez-Aguirre [1] and Miguel-Angel Gil-Rios [3]

[1] Centro de Investigación en Matemáticas (CIMAT), A.C., Jalisco S/N, Col. Valenciana, Guanajuato 36000, Guanajuato, Mexico; erick.giles@cimat.mx (E.G.G.d.P.); artha@cimat.mx (A.H.-A.)
[2] SECIHTI-Centro de Investigación en Matemáticas (CIMAT), A.C., Jalisco S/N, Col. Valenciana, Guanajuato 36000, Guanajuato, Mexico
[3] Departamento de Tecnologías de la Información e Innovación Digital, Universidad Tecnológica de León, Blvd. Universidad Tecnlógica 225, Col. San Carlos, León 37670, Guanajuato, Mexico; mgil@utleon.edu.mx
* Correspondence: ivan.cruz@cimat.mx

Abstract

Facial Expression Recognition (FER) is a critical component of affective computing, with deep learning models dominating performance metrics. In contrast, geometric approaches based on the Facial Action Coding System (FACS) offer explainability through using triangles aligned to facial landmarks. The notable points of these triangles capture the deformation of muscles. However, restricting the feature extraction to notable points may be suboptimal. This paper introduces a novel method for optimizing the extraction of features by searching for optimal inner points in 22 facial triangles applying three metaheuristics: Differential Evolution (DE), Particle Swarm Optimization (PSO), and Convex Partition (CP). This results in a set of 59 geometric-based descriptors that capture muscle deformation more accurately. The proposed method was evaluated using five machine learning classifiers on two benchmark databases: the Karolinska Directed Emotional Faces (KDEF) and the Japanese Female Facial Expression (JAFFE). Experimental results demonstrate significant performance improvements. The combination of DE with a Multi-Layer Perceptron (MLP) achieved an accuracy of 0.91 on the KDEF database, while Support Vector Machine (SVM) optimized via CP attained an accuracy of 0.81 on the JAFFE database. Statistical analysis confirms that optimized descriptors yield higher accuracy than previous geometric methods.

Keywords: evolutionary algorithms; explainable artificial intelligence; facial expression recognition; geometric features; inner triangle points; machine learning

1. Introduction

Facial expressions are the most direct non-verbal channel for communicating human emotion, a capability that has been a determinant factor for the survival and evolution of humans throughout history [1]. These expressions are grounded in physiological responses to environmental stimuli [2] and involve the coordinated movement of a complex array of approximately 43 facial muscles [3]. The accurate classification of these movements is critical for applications ranging from human–computer interaction to clinical monitoring

of neurological disorders [4,5]. For instance, recent studies have demonstrated the utility of analyzing regional facial information for the automatic diagnosis of facial palsy [6].

The foundational framework for analyzing facial movement is the Facial Action Coding System (FACS) proposed by Ekman and Friesen [7], which decomposes expressions into specific Action Units (AUs) corresponding to muscle contractions. While this categorical approach is standard, researchers have also explored dimensional models, such as the circular model of affect proposed by Barrett and Russell [8], which positions emotions on axes of pleasantness and activation. However, modeling the multifaceted nature of emotion computationally remains a significant challenge [9].

Regarding automated recognition, while recent advancements in Convolutional Neural Networks (CNNs) have achieved high accuracy [10], they often lack explainability. Conversely, geometric approaches that track the deformation of specific facial regions preserve the anatomical context of the analysis [11]. Standard geometric methods rely on the detection of 68 facial landmarks, a process extensively reviewed by Ouanan et al. [12].

To bridge the gap between these raw points and muscular mechanics, recent research has attempted to infer muscle activity directly from visual data. Benli and Eskil [13] utilized optical flow to track landmark displacement as a proxy for muscle features. Similarly, hybrid approaches have combined computer vision with wearable electromyography (EMG) to map muscle synergies to observable kinematics [14,15]. Building on the concept of non-intrusive muscle estimation, Arrazola et al. [5] proposed measuring the deformation of triangles anchored by the *centroid* to estimate muscle activation, providing a robust set of features for emotion recognition. Building on this, Aguilera et al. [11] demonstrated that the centroid is not always the most discriminative point. They improved performance by selecting the best *notable point* (e.g., Incenter, Fermat point, Nagel point) for each triangle from a finite set of candidates [11]. To address this limitation, the present work proposes a new method to find the optimal inner point for each triangle in a continuous search space. Metaheuristic optimization algorithms—such as Differential Evolution (DE), Particle Swarm Optimization (PSO), and Convex Partition (CP)—are used for determining the precise linear combination of vertices that maximizes the discriminative power of the geometric features. This approach generalizes previous geometric works using inner point location as a tunable parameter, thereby enhancing the performance of standard machine learning classifiers.

The emphasis on explainability in this work may justify the difference in accuracy between the proposal (which achieves 0.91 in KDEF and 0.81 in JAFFE) and the state-of-the-art deep learning methods ad hoc for each database (0.96 in KDEF and 0.98 in JAFFE) [10]. In critical application scenarios, such as clinical diagnosis aids for neurological disorders or forensic psychological assessment, the ability to validate a decision provided with explainability at the muscle level is vital, with a minimum acceptable performance threshold typically set close to 0.80 [6]. Our method fulfills this by mapping feature deformation directly to specific muscle groups, allowing results to be interpreted through the lens of anatomical dynamics rather than uninterpretable pixel-level correlations.

Section 2 briefly introduces the databases, the methodology for constructing geometric descriptors, and the optimization metaheuristics algorithm, along with evaluated machine learning classifiers. Section 3 details how metaheuristics are applied for tuning 22 FACS-consistent triangles from 68 landmarks. Section 4 reports the performance of tested strategies and utilizes the non-parametric *Friedman test* for statistical analysis and visual resources to discuss results. Finally, Section 5 summarizes the successful deployment of the evolutionary metaheuristics, confirming that the proposed methodology yields state-of-the-art results among geometric FER approaches.

 https://doi.org/10.3390/a19010057

2. Materials and Methods

This section briefly describes the databases, the geometric reference framework, the machine learning, and the metaheuristic applied to FER.

2.1. Databases

Two widely recognized databases in the field of affective computing are utilized, targeting the classification of seven basic emotions: Anger, Disgust, Fear, Happiness, Sadness, Surprise, and Neutral.

1. **KDEF (Karolinska Directed Emotional Faces):** Developed by Lundqvist et al. (1998) [16], this database consists of 4900 images of human facial expressions. The set features 70 amateur actors (35 females and 35 males). For this study, only the frontal view images are utilized to ensure consistency with the geometric landmark detection (980 images).
2. **JAFFE (Japanese Female Facial Expression):** Created by Lyons et al. (1998) [17], this database contains 213 images of 7 facial expressions posed by 10 Japanese female models.

Figure 1 shows the proportion of emotions for both databases. In both cases, emotion labels are uniformly distributed, i.e., well-balanced.

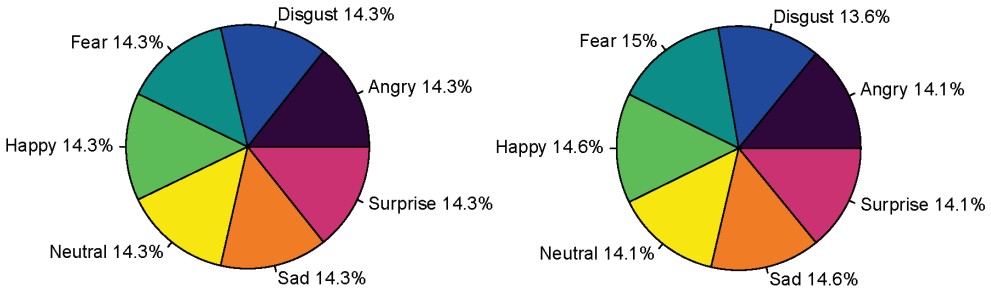


Figure 1. Proportions of emotion labels in databases: (**a**) KDEF and (**b**) JAFFE.

2.2. Face Landmarks and Facial Muscle Representation

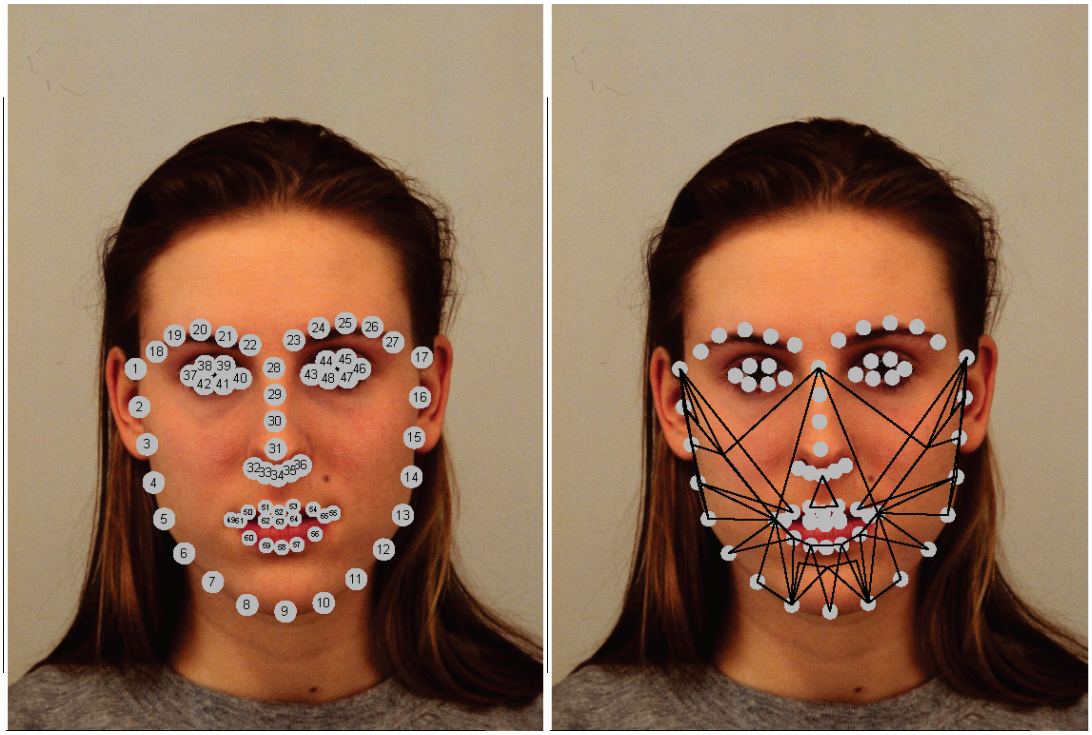
Facial landmarks are key points that localize distinctive features of the face (e.g., eyes, eyebrows, mouth, and jawline). This work is based on the 68-point annotation scheme proposed by Kazemi and Sullivan [18], which utilizes an ensemble of regression trees to estimate face alignment. Kazemi's geometric approach aligns with the Facial Action Coding System (FACS) [7], which states that expressions are the result of specific movements of facial muscles called Action Units (AU). Significant AUs are listed in Table 1 for further reference.

Formally, for any given picture, let $\mathbb{L} = \{L_1, \ldots, L_{68}\} \subset \mathbb{R}^2$ be a collection of Kazemi's face landmarks, i.e., a set of points aligned to specific parts of the face. According to [5,11], constructing triangles based on the vertex set $\mathbb{L}$ constitutes a robust muscle representation for emotion recognition. Table 2 lists the vertices of 22 triangles manually defined to be consistent with Table 1 based on the FACS theory proposed by [7]. Figure 2 illustrates (a) the enumerated Kazemi's Face Landmarks and (b) the proposed triangles.

https://doi.org/10.3390/a19010057

Table 1. Kazemi's landmarks associated with emotional AUs [11].

Emotion	AU [7]	Description	Facial Muscle	Kazemi's Landmarks
Anger	4	Eyebrow lowered	Corrugator supercilii	19, 20, 21, 24, 25, 26
	5	Upper lid raised	Levator palpebrae superioris	38, 39, 44, 45
	7	Eyelid tightened	Orbicularis oculi	42, 41, 48, 47
	23	Lips tightened	Orbicularis oris	57, 58, 59
Disgust	9	Nose wrinkled	Levator labii	29, 30
	10	Upper lip raised	Levator labii	51, 52, 53
	25	Lips parted	Depressor labii	50, 54, 60, 56
Sadness	1	Inner eyebrow raised	Frontalis medialis	20, 25
	4	Eyebrow lowered	Corrugator supercilii	19, 20, 21, 24, 25, 26
	15	Lip corners depressed	Depressor anguli oris	49, 51, 61, 65
Fear	1	Inner eyebrow raised	Frontalis medialis	20, 25
	2	Outer eyebrow raised	Frontalis lateralis	18, 22, 23, 27
	4	Eyebrow lowered	Corrugator supercilii	19, 21, 24, 26
	5	Upper lid raised	Levator palpebrae superioris	38, 39, 44, 45
	20	Lips stretched	Risorius	52, 58, 63
Surprise	1	Inner eyebrow raised	Frontalis medialis	20, 25
	2	Outer eyebrow raised	Frontalis lateralis	18, 22, 23, 27
	5	Upper lid raised	Levator palpebrae superioris	38, 39, 44, 45
	26	Jaw dropped	Masseter	8, 9, 10
Happiness	6	Cheeks raised	Orbicularis oculi	5, 13
	12	Lip corners pulled up	Zygomaticus major	49, 61, 65, 55



(**a**) (**b**)

Figure 2. Visual representation of the feature extraction basis in image AF05NES from KDEF. (**a**) The 68 enumerated Kazemi's facial landmarks. (**b**) The corresponding mesh of 22 triangles designed to capture FACS-based muscle movements.

https://doi.org/10.3390/a19010057

Table 2. Definition of the 22 facial triangles utilized for muscle representation. Each triangle T_i is constructed using three vertex landmarks (L_i, L_j, L_k) from the Kazemi set or an inner point $\dot{T}_i$ from a previously defined triangle T_i.

$T_1 = (L_{48}, L_{59}, L_5)$	$T_2 = (L_{59}, L_{58}, L_7)$	$T_3 = (L_{58}, L_{56}, L_8)$	$T_4 = (L_{56}, L_{55}, L_9)$	$T_5 = (L_{54}, L_{55}, L_{11})$	$T_6 = (\dot{T}_2, \dot{T}_3, L_9)$
$T_7 = (\dot{T}_1, \dot{T}_2, L_7)$	$T_8 = (L_{54}, L_{12}, L_{15})$	$T_9 = (L_{48}, L_1, L_4)$	$T_{10} = (L_{53}, L_{13}, L_{16})$	$T_{11} = (\dot{T}_9, L_{14}, L_{16})$	$T_{12} = (\dot{T}_{10}, L_{15}, L_{16})$
$T_{13} = (L_{49}, L_0, L_3)$	$T_{14} = (\dot{T}_{12}, L_0, L_2)$	$T_{15} = (\dot{T}_{13}, L_0, L_1)$	$T_{16} = (\dot{T}_{12}, L_{48}, L_{27})$	$T_{17} = (\dot{T}_9, L_{54}, L_{27})$	$T_{18} = (L_{50}, L_{52}, L_{33})$
$T_{19} = (L_3, L_4, L_{48})$	$T_{20} = (L_{13}, L_{12}, L_{54})$	$T_{21} = (L_{54}, L_{10}, L_9)$	$T_{22} = (L_{48}, L_6, L_7)$		

To identify different expressions, the deformation of triangles is measured as the Euclidean distance between inner points of a pair of triangles, which represent relevant muscle groups. As fundamental geometric descriptors, the specific pairs of triangles utilized are listed in Table 3. Based on these distances, a feature vector of 59 descriptors is computed for each image. The following subsection details how these inner points are defined.

Table 3. List of the 59 geometric descriptors employed for classification. The descriptors are computed as Euclidean distances between specific pairs of points.

(L_0, L_{16})	(L_{21}, L_{39})	(L_{22}, L_{42})	(L_{21}, L_{22})	(L_{20}, L_{23})	(L_{19}, L_{24})	(L_{18}, L_{25})	(L_{17}, L_{26})
(L_{37}, L_{41})	(L_{44}, L_{46})	(L_{38}, L_{40})	(L_{43}, L_{47})	(L_{17}, L_{18})	(L_{18}, L_{19})	(L_{19}, L_{20})	(L_{20}, L_{21})
(L_{26}, L_{25})	(L_{25}, L_{24})	(L_{24}, L_{23})	(L_{23}, L_{22})	$(L_{39}, \dot{T}_{15})$	$(\dot{T}_{15}, L_{48})$	$(L_{42}, \dot{T}_{16})$	$(\dot{T}_{16}, L_{54})$
$(\dot{T}_{14}, \dot{T}_{12})$	$(\dot{T}_{12}, L_{49})$	$(\dot{T}_{11}, \dot{T}_9)$	$(\dot{T}_9, L_{53})$	$(L_1, \dot{T}_8)$	$(\dot{T}_8, L_{48})$	$(L_{15}, \dot{T}_7)$	$(\dot{T}_7, L_{54})$
$(\dot{T}_7, \dot{T}_{19})$	$(\dot{T}_8, \dot{T}_{18})$	$(L_3, \dot{T}_{18})$	$(\dot{T}_{18}, L_{48})$	$(L_{13}, \dot{T}_{19})$	$(\dot{T}_{19}, L_{54})$	(L_{48}, L_{54})	(L_{62}, L_{66})
$(L_{48}, \dot{T}_{17})$	$(L_{54}, \dot{T}_{17})$	$(L_{48}, \dot{T}_2)$	$(L_{54}, \dot{T}_2)$	(L_{33}, L_{51})	$(L_{48}, \dot{T}_0)$	$(L_{54}, \dot{T}_4)$	$(L_{48}, \dot{T}_{21})$
$(\dot{T}_{21}, L_7)$	$(L_{54}, \dot{T}_{20})$	$(\dot{T}_{20}, L_9)$	$(\dot{T}_0, L_6)$	$(\dot{T}_4, L_{10})$	$(\dot{T}_6, L_8)$	$(\dot{T}_5, L_8)$	(L_{17}, L_{36})
(L_{26}, L_{45})	(L_{19}, L_{37})	(L_{24}, L_{44})					

Note: $\dot{T}_i$ denotes the optimized inner point of triangle T_i.

2.3. Inner Points

The discriminative quality of the 59 descriptors depends heavily on the definition of the inner points within each triangle. Formally, for a triangle T defined by three vertices $\{L_i, L_j, L_k\}$, any inner point $\dot{T} \in T$ can be expressed by a linear combination as follows:

$$\dot{T} = w_i L_i + w_j L_j + w_k L_k \text{ , where } w_i, w_j, w_k > 0 \text{ and } w_i + w_j + w_k = 1 \quad (1)$$

The existing literature provides standard approaches for defining these points:

1. Arrazola et al. [5] defines each inner point in T as the centroid of the triangle, i.e., $w_i = w_j = w_k = 1/3$;
2. Aguilera et al. [11] utilizes the trajectory metaheuristic of Simulated Annealing to select the best option among four notable triangle centers (Incenter, Centroid, Nagel point, or Torricelli point).

In this context, the main idea of this paper is applying continuous optimization metaheuristics to directly tune the weights w_i, w_j, and w_k. This allows for an efficient search of the optimal feature extraction point to maximize the performance of machine learning methods.

2.4. Machine Learning Methods

The task of facial expression recognition is easily formulated as a supervised classification problem. Formally, for both databases, the i-th image is encoded as a vector of 59 descriptors $X_i \in \mathbb{R}^{59}$ (Table 3) associated with a known emotion label $Y_i \in \{\text{Anger}, \text{Disgust}, \text{Sadness}, \text{Fear}, \text{Surprise}, \text{Happiness}, \text{Neutral}\}$. To solve this multi-class problem in $\mathbb{R}^{59}$, the algorithms listed below are applied and evaluated.

https://doi.org/10.3390/a19010057

- **Support Vector Machine (SVM) [19]:** A powerful classifier that constructs hyperplanes in a high-dimensional space to separate classes. A linear kernel is used to handle relationships in the feature space. The Support Vector Machine classifier is established using the `SVC` class from the `sklearn.svm` library. The model is configured with the following parameters:
  1. `kernel = 'linear'`: Specifies the use of a linear kernel to create the decision boundary;
  2. `random_state = 42`: Controls the pseudo-random number generation for shuffling data to ensure reproducibility.
- **Multilayer Perceptron (MLP) [20]:** A feedforward artificial neural network, capable of learning non-linear function approximations for the emotion descriptors. The artificial neural network is constructed using the `MLPClassifier` function from the `sklearn.neural_network` library. The specific architecture and training behavior are governed by the following parameters:
  1. `hidden_layer_sizes = (100, 50)`: Defines a network with two hidden layers containing 100 and 50 neurons, respectively;
  2. `max_iter = 500`: Sets the maximum number of training iterations (epochs) to 500;
  3. `activation = 'relu'`: Specifies the Rectified Linear Unit function for the hidden layers;
  4. `solver = 'adam'`: Utilizes the Adam optimizer for weight adjustment;
  5. `random_state = 42`: Fixes the seed for random weight initialization.
- **Decision Tree (DT) [21]:** A non-parametric method that learns simple decision rules inferred from the data features. It provides high interpretability regarding which geometric distances are most critical for classification. The classification model is built using the `DecisionTreeClassifier` class from the `sklearn.tree` sublibrary. The model is defined by the following hyperparameters explicitly passed during initialization:
  1. `criterion = 'gini'`: Uses Gini impurity to measure the quality of splits;
  2. `max_depth = 5`: Limits the tree expansion to a maximum depth of 5 to prevent overfitting;
  3. `random_state = 42`: Ensures the stochastic elements of the estimator are reproducible.
- **Linear Discriminant Analysis (LDA) [22]:** A statistical method that finds a linear combination of features that characterizes or separates two or more classes of objects or events, assuming Gaussian densities. The Linear Discriminant Analysis model is instantiated using the `LinearDiscriminantAnalysis` class from the `sklearn.discriminant_analysis` library. No specific hyperparameters are explicitly defined, implying the model relies entirely on the default configuration provided by the library
- **XGBoost (XGB) [23]:** An efficient implementation of gradient-boosted decision trees. It is designed for speed and performance, operating by combining an ensemble of weak prediction models to produce a strong learner. The XGB model is implemented using the R library `xgboost 1.7.11` with its default parameters.

To measure the effectiveness of these methods, a 5-fold cross-validation is performed using `scikit-learn 1.4.1`. The performance is evaluated by computing the mean ac-

 https://doi.org/10.3390/a19010057

curacy across all folds, which represents the average proportion of correctly classified instances across the experimental folds. The **mean accuracy** is formally expressed as

$$\text{Mean Accuracy} = \frac{1}{K}\sum_{k=1}^{K}\left(\frac{\sum_{i=1}^{C} c_{ii}}{\sum_{i=1}^{C}\sum_{j=1}^{C} c_{ij}}\right)_k \tag{2}$$

where

- K is the number of folds ($K = 5$);
- C is the total number of emotion classes ($C = 7$);
- c_{ij} is the (i, j) entry of the confusion matrix).

2.5. Optimization Metaheuristics

Consider that inner triangle points are parametric, so different geometric descriptors can be computed, and the performance of any machine learning algorithm is consequently affected. In order to improve the classification, several optimization metaheuristics are applied to discover the parameters that maximize the accuracy of each machine learning method. The optimization metaheuristics listed below share remarkable characteristics: (1) initially, they explore uniformly the space of parameters; (2) they are population-based and avoid local optima; and (3) they can be tuned with similar parameters. As shown below, their parameters are similarly tuned. For example, the population is set to 200 and the number of iterations to 50.

- **Differential Evolution (DE) [24].** A population-based metaheuristic that optimizes a problem by iteratively trying to improve a candidate solution. It uses mechanisms analogous to biological evolution, such as mutation, crossover, and selection, and is particularly effective for continuous search spaces. Algorithm 1 is the basic algorithmic schema of DE. The **DEoptim** [25] library implements Differential Evolution (DE) for global optimization.

Algorithm 1 Differential Evolution (DE)

Require: Objective function accuracy, Population size NP
Ensure: Best solution V_{best}
Initialize population $P = \{V_1, \ldots, V_{NP}\}$
while termination criteria not met **do**
  for $i = 1, \ldots, NP$ **do**
    Generate trial vector U_i using mutation and crossover
    if $f(U_i) \leq f(V_i)$ **then**
      $V_i \leftarrow U_i$
    end if
  end for
end while

The following parameters are used:

1. **Maximum Iterations (`itermax`)**: The maximum number of generations/iterations for the DE algorithm, set to `50` via the `control` list.
2. **Population Size (`NP`)**: The number of population members, set to `200` via the `control` list.
3. **Crossover Probability (`CR`)**: Controls the fraction of parameter values copied from the mutant vector to the trial vector. It regulates the diversity of the population and is set to `0.5`

https://doi.org/10.3390/a19010057

4. **Differential Weighting Factor (`F`)**: A scaling factor applied to the difference vector during mutation. It controls the magnitude of the perturbation and is set to `0.8`.
5. *Approximate function evaluations for the given parameters:* $FE = 1 \times 10^4$.

- **Particle Swarm Optimization (PSO) [26]:** A computational method that solves a problem by having a population of candidate solutions, here named particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Algorithm 2 is the basic algorithmic schema of PSO.

Algorithm 2 Particle Swarm Optimization (PSO)

Require: Objective function accuracy, Swarm size s
Ensure: Best solution V_g (global best)
Initialize particles V_i and velocities δ_i for $i = 1, \ldots, s$
while termination criteria not met **do**
  for $i = 1, \ldots, s$ **do**
    Update velocity δ_i using local best V_i' and global best V_g
    Update position $V_i \leftarrow V_i + \delta_i$
    if $f(V_i)$ is better than $f(V_i')$ **then**
      Update local best $V_i' \leftarrow V_i$ and global best V_g
    end if
  end for
end while

The utilized **pso** library implements Particle Swarm Optimization (PSO) via the `psoptim` [27] function. The experiments reported in this paper follow the parameters suggested by [28], which are summarized below.

1. **Swarm Size (`s`)**: The number of particles in the swarm, set to `200` via the `control` list.
2. **Local Exploration Constant (`c.p`)**: Controls the particle's tendency to return to its best-found position. The default value is $0.5 + \log(2)$.
3. **Global Exploration Constant (`c.g`)**: Controls the particle's attraction to the swarm's best-found position. The default value is $0.5 + \log(2)$.
4. *Approximate function evaluations for the given parameters:* $FE = 1 \times 10^4$.

- **Convex Partition (CP) [29]:** A Bayesian optimization approach, which utilizes a regression tree structure to partition the search space into convex regions, allowing for efficient exploration of high-dimensional continuous spaces. Algorithm 3 is the basic algorithmic schema of CP.

Algorithm 3 Convex Partition (CP)

Require: An objective function $f : D \to \mathbb{R}$
Ensure: A domain partition $\mathbb{G}$ focused on promising subsets
$\mathbb{G} = \{G_1 = D\}$
for $i = 2, \ldots, t$ **do**
  Select a promising* subset G from the collection $\mathbb{G}$
  Note: promising subset to contain extreme values of f, previously not observed
  Remove G from $\mathbb{G}$
  Split G into $\{A, B\}$
  Insert $\{A, B\}$ into the collection $\mathbb{G}$
end for

The **CPoptim** [30] library is used for executing the optimization process with the following parameters:

https://doi.org/10.3390/a19010057

1. **Maximum Function Evaluations (**`maxFE`**)**: The maximum number of times the objective function will be evaluated, set to `10,000`.
2. **Sample Size (**`sampleSize`**)**: The size of the sample/population used in the search, set to `200`.
3. *Approximate function evaluations for the given parameters:* $FE = 1 \times 10^4$.

3. Proposed Method

This section details the proposed methodological framework for the Facial Expression Recognition (FER) task. The experimental pipeline can be summarized into three main stages: (1) data acquisition and preprocessing, where facial landmarks are extracted from standard databases; (2) feature extraction, where a geometric representation based on triangular regions is utilized; and (3) optimization and classification, where metaheuristic algorithms tune the geometric features to maximize the performance of various machine learning classifiers. The following subsections explain (1) how metaheuristics can be used for producing efficient descriptors and (2) details about the experiment designed for discovering the optimal combination of metaheuristics and machine learning methods.

3.1. Encoding Solutions for Metaheuristics

To implement optimization metaheuristics, firstly a proper representation of inner points consistent with Equation (1) is defined. Formally,

Definition 1. *For any pair* $u_1, u_2 \in [0,1] \subset \mathbb{R}$, $\dot{T}(u_1, u_2)$ *denotes an indexed* ***inner point*** *of the triangle* T *whose vertices are* L_i, L_j, L_k. *Formally,* $\dot{T}(u_1, u_2) = w_i L_i + w_j L_j + w_k L_k$, *where* $w_i = \min(u_1, u_2)$, $w_j = \max(u_1, u_2) - \min(u_1, u_2)$, $w_k = 1 - \max(u_1, u_2)$

The transformation above is not bijective, but it possesses a remarkable property:

Theorem 1. *If* $U_1 = u_1, U_2 = u_2 \sim$ *Uniform*$(0,1)$, *then* $\dot{T}(u_1, u_2)$ *is a random observation uniformly distributed inside* T.

Proof. To prove that this transformation results in a uniform distribution within the triangle, consider the following: (1) $\min(u_1, u_2)$ and $\max(u_1, u_2)$ are order statistics of uniform variables; (2) the Jacobian of the transformation of these order statistics to w_1, w_2 results in a flat Dirichlet distribution; and (3) the flat Dirichlet distribution samples uniformly on a simplex (a triangle in this case). See [31] for more details. □

This property allows us to uniformly explore the space of inner points without additional constraints on u_1, u_2. Therefore, for each of the 22 triangles (Table 2), an inner point is determined by two parameters $u_1, u_2 \in [0,1]$. Consequently, the full configuration of 22 inner points is encoded as a real vector $V \in [0,1]^{44}$. For a fixed machine learning method at time, the optimization problem is formulated as maximizing the mean accuracy of a nested 5-fold cross-validation as an objective function $f : [0,1]^{44} \to [0,1]$ (Equation (2)) by tuning the parameter vector V.

3.2. Experimental Design

To identify the most efficient combination of optimization metaheuristics (MHs) and machine learning (ML) methods, every triad of Database $BD \in \{KDEF, JAFFE\}$, Metaheuristic $MH \in \{DE, PSO, CP\}$, and Classifier $ML \in \{SVM, MLP, LDA, DT, XGB\}$ is evaluated once. For each triad (DB, MH, ML), the optimal vector of inner points $V \in [0,1]^{44}$ is estimated via the following procedure:

https://doi.org/10.3390/a19010057

1. Compute Kazemi's face landmark for each image in the database BD using the *Dlib 2.0* library, and keep landmarks alongside the corresponding emotion labels.
2. Define the objective function f that takes a vector V as a single parameter and returns the mean accuracy of the model ML trained and tested by a 5-fold cross-validation schema. In detail: (i) Compute the triangles and descriptors listed in Tables 2 and 3, utilizing the kept landmarks and parameters V to define the inner points and geometric descriptors. (ii) Perform a 5-fold cross-validation schema for the ML model using *Scikit-learn 1.4.1*. The model is trained and tested five different times, ensuring every data point tested has not been used for training in the current iteration. (iii) Return the mean accuracy across the 5 folds (Equation (2)).
3. Apply the optimization metaheuristic MH to maximize the previously defined objective function f. The feasible search space is restricted to $[0, 1]^{44}$.

4. Results and Discussion

The computational experiments were executed on a system running the Windows 11 Pro operating system, featuring an Intel Core i5-6500, 3.2 GHz. processor, and 16 GB of RAM. The core machine learning analysis, including the implementation of the classifiers and optimization procedures, was performed in Python 3.14, utilizing the scikit-learn library version 1.4.1. The overall experimental workflow, which included the coordination of the Python scripts and the generation of final summary reports and statistical analyses, was conducted by scripts written in the statistical computing language R, version 4.3. The resulting optimal accuracy for each combination (BD, ML, MH) is reported and discussed in this section. Tables 4 and 5 report the mean accuracy and its corresponding standard deviation estimated by a 5-fold cross-validation for each pair (ML, MH) for both databases. Table 6 organizes running time in minutes for each tested combination (ML, MH), additionally it reports the total time by column (machine learning method) and row (metaheuristic). To contrast the time efficiency of each method, Table 7 shows the average time taken in milliseconds for classifying an image.

Table 4. Optimum mean accuracy (nested 5-fold cross-validation) reached by different metaheuristics for different machine learning methods.

Database	Metaheuristic	MLP	SVM	LDA	DT	XGB
KDEF	DE	**0.9173**	0.8629	0.8730	0.7151	0.8190
	PSO	0.9034	0.8774	0.8887	0.7276	0.8017
	CP	0.8914	0.8978	0.9004	0.7370	0.7990
JAFFE	DE	0.8024	0.7981	0.7159	0.5675	0.7606
	PSO	0.8178	0.7895	0.7243	0.5570	0.7298
	CP	0.8196	**0.8326**	0.7546	0.5492	0.7236

Table 5. Standard deviation of accuracy estimated by a 5-fold cross-validation for each combination of machine learning methods and optimization metaheuristics.

Database	Metaheuristic	MLP	SVM	LDA	DT	XGB
KDEF	DE	0.0295	0.0284	0.0176	0.0396	0.0288
	PSO	0.0236	0.0307	0.0217	0.0286	0.0262
	CP	0.0230	0.0369	0.0288	0.0204	0.0273
JAFFE	DE	0.0429	0.0500	0.0315	0.0326	0.0393
	PSO	0.0315	0.0477	0.0497	0.0419	0.0427
	CP	0.0302	0.0413	0.0805	0.0488	0.0502

https://doi.org/10.3390/a19010057

Table 6. Execution time in minutes for each trained combination.

Database	Metaheuristic	MLP	SVM	LDA	DT	XGB	*Total*
KDEF	DE	1290	1148	521	908	1030	4897
	PSO	1283	1147	520	896	1031	4877
	CP	1303	1181	552	926	1040	5002
	Total	3876	3476	1593	2730	3101	**14,776**
JAFFE	DE	1277	1153	528	895	1034	4887
	PSO	1283	1167	519	901	1017	4887
	CP	1288	1169	543	931	1050	4981
	Total	3848	3489	1590	2727	3101	**14,755**

Table 7. Average testing classification time per image in milliseconds.

Database	Metaheuristic	MLP	SVM	LDA	DT	XGB
KDEF	DE	0.00513	0.03431	0.00154	0.00135	0.02044
	PSO	0.00512	0.03554	0.00148	0.00132	0.01682
	CP	0.00511	0.03677	0.00142	0.00130	0.01812
JAFFE	DE	0.01213	0.01722	0.00512	0.00574	0.01165
	PSO	0.01206	0.01741	0.00494	0.00543	0.01240
	CP	0.01199	0.01760	0.00477	0.00512	0.01061

To statistically compare each combination, the non-parametric Friedman test is selected because the distribution of performance metrics did not satisfy the necessary assumptions for parametric testing, specifically regarding the lack of normality and homoscedasticity (homogeneity of variance) in samples (See Table 5). The Friedman test operates on the rank of the performance metrics rather than the raw values, making it robust against outliers and non-normal distributions, thus providing a reliable statistical basis for determining if significant differences exist between the methods. Figure 3 illustrates the results of the Friedman test through bar charts representing the sum of ranks. Figure 3a displays the comparison among machine learning methods. The test yielded a p-value of 3×10^{-4}, which is well below the significance threshold ($\alpha = 0.05$). This allows us to reject the null hypothesis (H_0) that all classifiers are equivalent, confirming that the choice of classifier significantly impacts performance. As observed in the chart, MLP and SVM obtained the highest rank sums, statistically outperforming LDA, DT, and XGB. Conversely, Figure 3b compares the optimization metaheuristics. Here, the test returned a p-value of 0.8357. In this case, we fail to reject the null hypothesis, indicating that there is no statistically significant difference between the performance of DE, PSO, and CP. The statistical analysis confirms that the selection of the machine learning classifier is critical (with MLP and SVM being superior choices), while the choice of optimization metaheuristic (among DE, PSO, and CP) has a negligible impact on the final model performance. Similarly, Figure 4 illustrates Friedman tests for contrasting optimizing–training time between (a) machine learning methods and (b) metaheuristics. In both cases, the hypothesis of equality is rejected: (a) MLP and SVM are significantly slower, and (b) CP is significantly slower than DE and PSO. Although metaheuristics perform approximately the same number of function evaluations 1×10^4, CP is a surrogate-based technique, which implies additional time to estimate a latent Bayesian model at each iteration [29].

https://doi.org/10.3390/a19010057

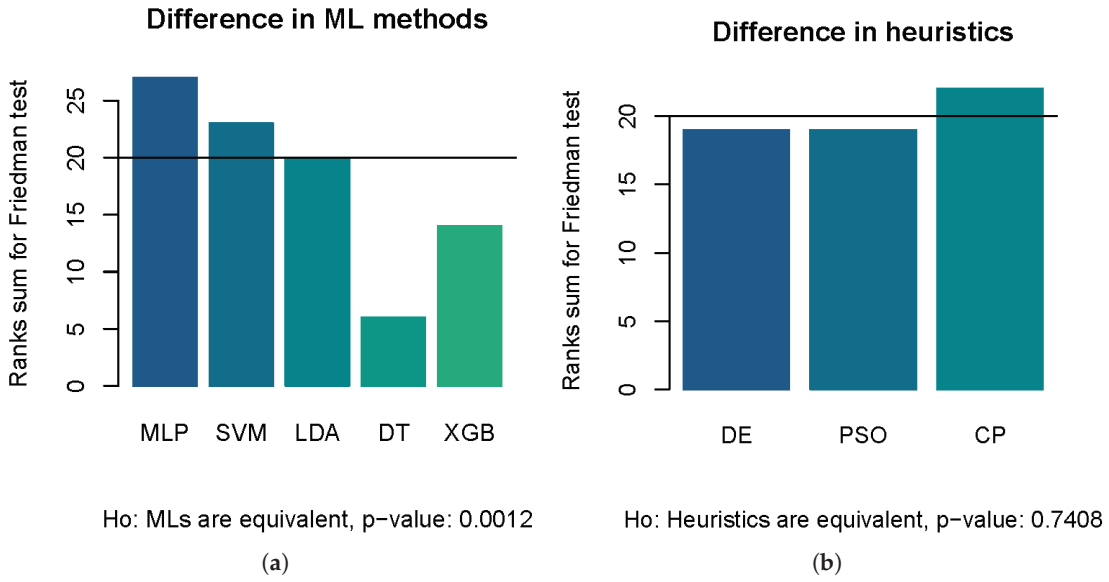


Figure 3. Bar charts for ranks sums of Friedman tests. Each chart reports a *p*-value for a hypotheses: (**a**) All machine learning methods are equivalent in mean accuracy (5-fold cross-validation), (**b**) All optimization metaheuristics are equivalent in mean accuracy (5-fold cross-validation). The horizontal line marks the expected ranks sum if contrasted variables were equivalent.

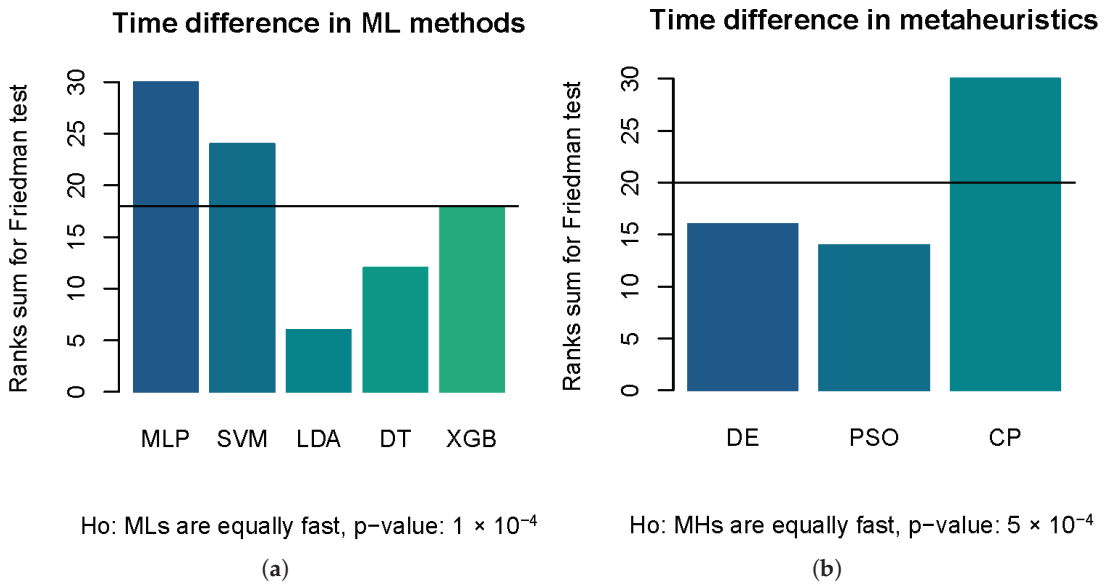


Figure 4. Bar charts for ranks sums of Friedman tests. Each chart reports a *p*-value for a hypotheses: (**a**) All machine learning methods are equally efficient in optimizing-training minutes. (**b**) All optimization metaheuristics are equally efficient in optimizing–training minutes. The horizontal line marks the expected ranks sum if contrasted variables were equivalent.

Table 4 indicates that the Multi-layer Perceptron (MLP) + Differential Evolution (DE) achieved the highest performance on the KDEF database, achieving an accuracy of 0.9173. Similarly, the combination of Support Vector Machine (SVM) and Convex Partition (CP) demonstrated robust stability on the JAFFE dataset, achieving an accuracy of 0.8326. In contrast, the Decision Tree (DT) classifier produced the lowest results, with an accuracy of 0.54 on the JAFFE dataset. The Friedman test supports that the best experimental results were obtained by utilizing the MLP+DE (Accuracy: 0.9173) and SVM+CP (Accuracy:

https://doi.org/10.3390/a19010057

0.8326), for KDEF and JAFFE respectively. Hence, further analysis is focused on these best models. Figure 5 shows how the mean accuracy converges for MLP and SVM during the execution of the three tested metaheuristics. These convergence charts visually exhibit the superiority of combinations MLP+DE and SVM+CP. In all cases, the metaheuristic search stops when the trend gets stuck.

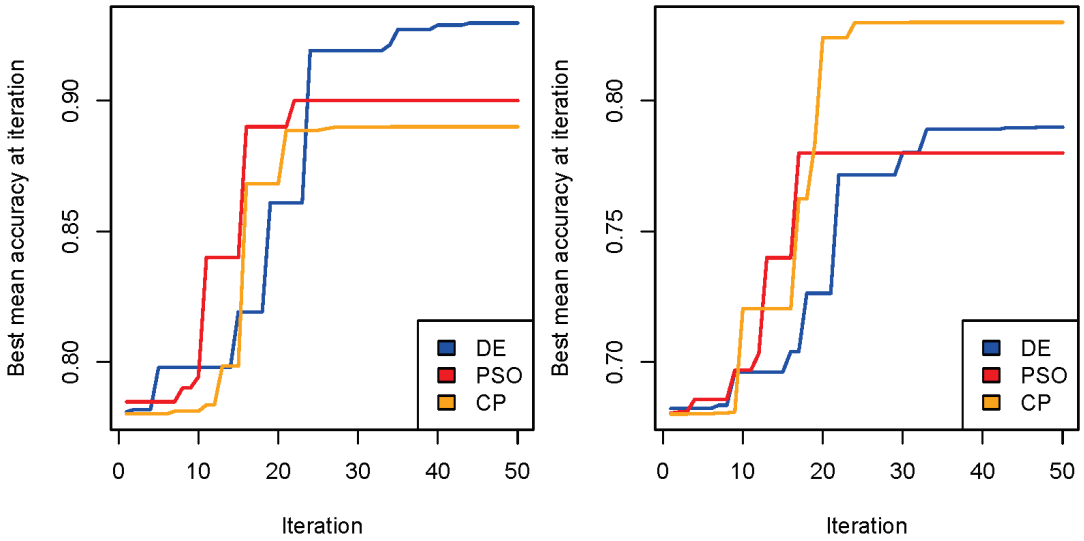
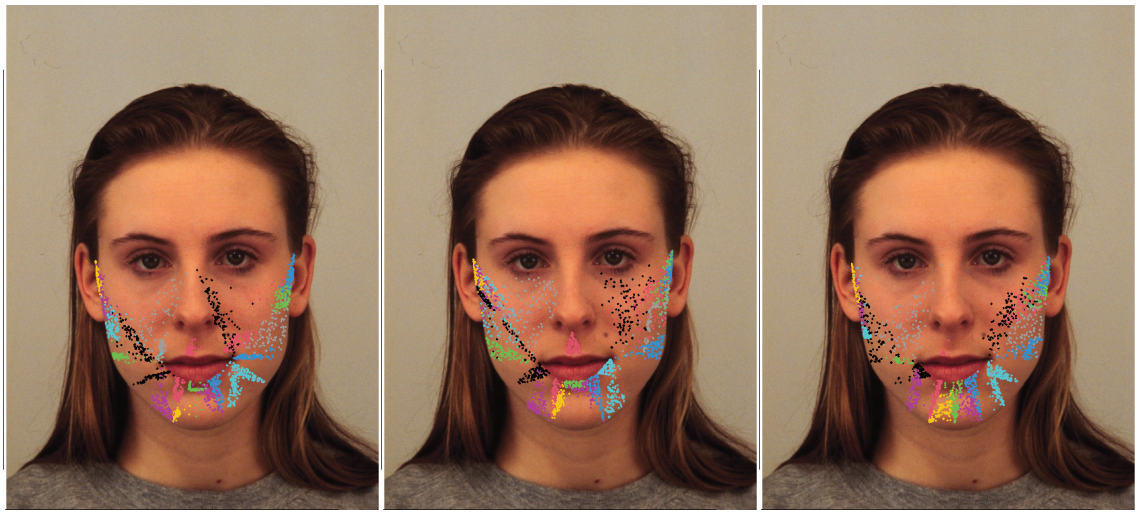


Figure 5. Convergence of the mean accuracy (nested 5-fold cross-validation) during the tuning of parameter V for (**a**) MLP+DE for KDEF and (**b**) SVM+CP for JAFFE.

The results presented validate the hypothesis that optimizing the geometric landmark points within facial triangles significantly improves emotion recognition performance. By relaxing the constraints imposed by previous methods—which relied on fixed points like centroids [5] or a discrete selection of triangle centers [11]—the continuous optimization approach allows the model to adapt to the specific deformation dynamics of each muscle group. The superiority of the Multi-Layer Perceptron (MLP) combined with Differential Evolution (DE), achieving an accuracy of 0.9173 on the KDEF dataset, suggests that this classifier is the best suited to interpret the complex geometric relationships discovered by the metaheuristic search. Comparing these findings to the state-of-the-art, the proposed method outperforms the fixed-center approaches reported by Aguilera-Hernández et al. [11] This indicates that the *most informative* point on a facial triangle is often an arbitrary location best found through stochastic search rather than geometric intuition. Figure 6 illustrates the best 100 sets of inner points (coded by V) found for (a) MLP+DE, (b) SVM+CP, and (c) LDA+CP. It is possible to visually state that points are not focused at the center of triangles. The distribution of points allows us to identify areas that are intuitively relevant for emotions recognition:

- **Mouth and Chin Region:** There is a heavy concentration of points around the lips, corners of the mouth, and the chin/jawline. This region is critical for expressing emotions and it is associated with AUs 20, 25, and 26.
- **Cheeks and Nasolabial Folds.:** Points are distributed across the cheeks and along the nasolabial folds (the lines running from the sides of the nose to the corners of the mouth). These areas are associated with AUs 9, 6, and 12.

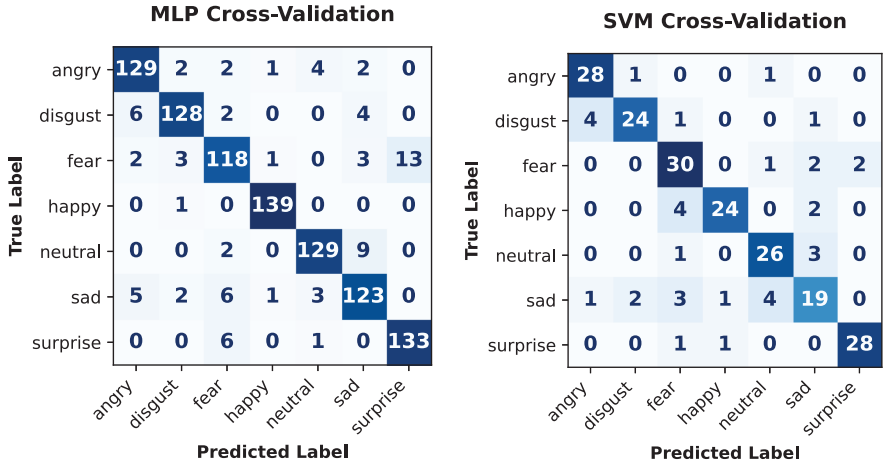
https://doi.org/10.3390/a19010057



(**a**) (**b**) (**c**)

Figure 6. Different sets of inner points corresponding to the best 100 vectors V found for (**a**) MLP+DE, (**b**) SVM+CP, and (**c**) LDA+CP. To identify each triangle, inner points are illustrated with different color.

Figure 7 shows two confusion matrices for (a) MLP+DE on the KDEF database and (b) SVM+CP on the JAFFE database, respectively. Matrix (a) demonstrates superior results, marked by exceptionally high true positive counts across all emotions and very minimal confusion, with errors primarily isolated to instances like *Angry/Disgust* and *Neutral/Sad* misclassifications. In contrast, matrix (b) shows a moderate-to-high overall score; it particularly struggles with discrimination among *Fear*, *Sad*, and *Surprise* categories. The geometric features derived from the optimization process are more robust and distinctive on the KDEF dataset.



(**a**) (**b**)

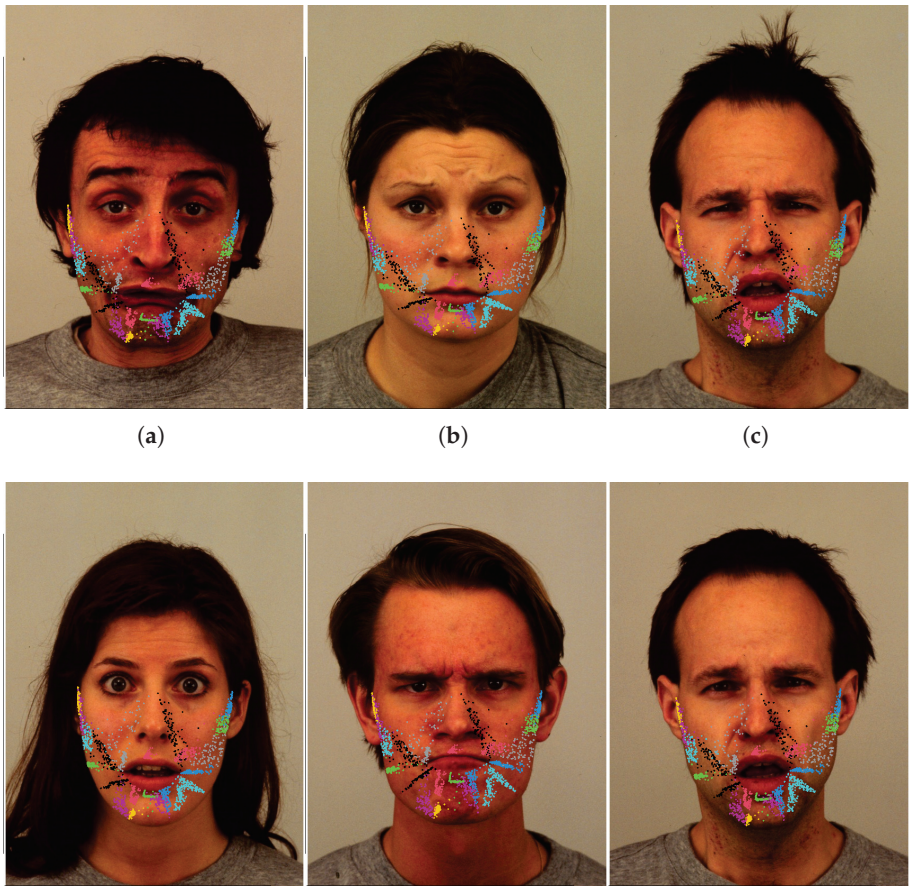
Figure 7. Confusion matrices for the best models reported in Table 4: (**a**) MLP+DE for KDEF and (**b**) SVM+CP for JAFFE.

Figure 8 shows six cases of misclassification produced by the best model MLP+DE. The misclassifications shown, such as *Fear* being confused with *Surprise* or *Anger* with *Disgust*, strongly support the argument that these specific emotions are inherently ambiguous and difficult to classify, even for a high-performing model. This difficulty arises because the facial expressions share significant overlaps in their underlying muscle movements (Action Units): for example, both Fear and Surprise often involve wide eyes and an open

https://doi.org/10.3390/a19010057

mouth. Those errors are not mere methodological failures, but rather a reflection of the subtle individual variations and the blurring lines between emotion categories. Consider that these specific expressions would likely confuse even human observers due to their high structural similarity. The inner points added to the plots visually indicate where the model is focused, so these points may constitute a form of explainable AI (XAI). They reveal where the MLP+DE model is attending on the face—typically highlighting critical facial regions like the cheeks, mouth and chin, i.e., AUs [2]: 6, 9, 12, 20, 25, 26, etc. Observing the scatter of these points directly may help understand which specific facial features are driving the misclassification. This crucial insight then guides the design of future models by allowing developers to reinforce weights on truly discriminative features or, conversely, to ignore confounding noise in ambiguous regions.

To contrast the proposed method with state-of-the-art results, Table 8 lists the accuracy scores of ten different methodologies applied to FER, divided into Deep Learning, Hybrid, and Geometric approaches. To visually compare these methodologies, Figure 9 presents bar charts of accuracy scores listed in Table 8. In the charts for KDEF and JAFFE, the proposed method, along with one or two Deep Learning models, appears in the third quartile (indicated by a vertical line). Notably, the proposed method is consistently the only geometric approach positioned in the third quartile. These results indicate that metaheuristic optimization allows geometric features to achieve competitive accuracy, despite registering lower accuracy than Deep Learning approaches.



(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

Figure 8. Six misclassification cases predicted by the best model (MLP+DE). Each case is listed as 'Expected Emotion' vs. 'Predicted Emotion': (**a**) fear vs. surprise, (**b**) sad vs. neutral, (**c**) fear vs. disgust, (**d**) surprise vs. fear, (**e**) angry vs. disgust, and (**f**) fear vs. disgust.

https://doi.org/10.3390/a19010057

Table 8. Accuracy reported for state-of-the-art methods.

Approach	Reference	Method	KDEF	JAFFE
Deep Learning	[32]	Deep Convolutional Neural Networks (CNNs)	0.81	0.94
Deep Learning	[10]	Attention mechanism + CNNs	-	0.98
Deep Learning	[33]	HOG with deep learning-based features	0.96	-
Deep Learning	[32]	GoogLeNet (pretrained model by Google)	0.47	0.71
Hybrid	[34]	CNN-based Hybrid Design + Image Preprocessing	0.71	0.71
Hybrid	[35]	LRSE features + ProCRC	0.79	-
Hybrid	[36]	Recurrent + Convolutional Neural Networks	-	0.79
Geometric	[37]	HOG based on handcrafted landmarks	0.90	0.77
Geometric	[5]	Inner Triangle points (Centroid)	0.81	0.71
Geometric	[11]	Inner Triangle points (Notable points)	0.87	0.81
Geometric	Proposed method	Optimized Inner Triangle points	0.91	0.83

HOG: Histogram of Oriented Gradients, LRSE: Low-Rank Sparse Error (LRSE), ProCRC: Probabilistic Collaborative Representation-based Classifier.

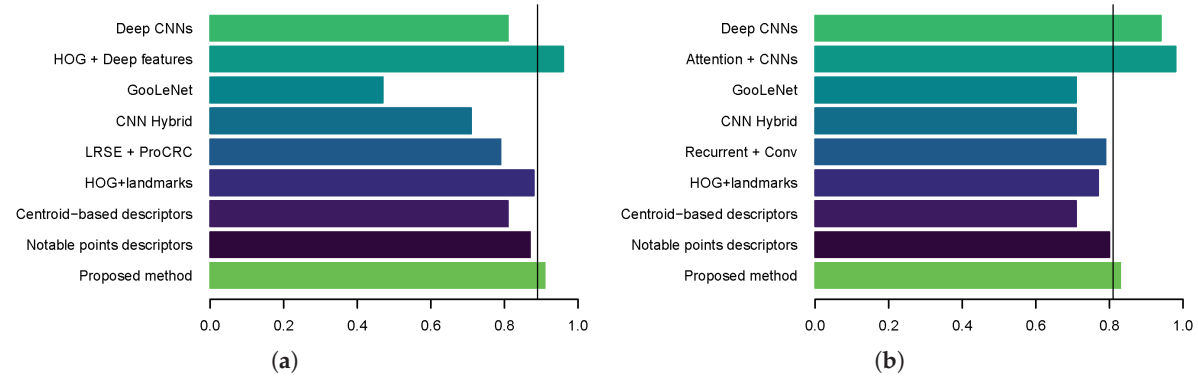


Figure 9. Bar charts of accuracy for (**a**) KDEF and (**b**) JAFFE. In both charts, the 3rd quartile is marked by a vertical line.

While Figure 9 suggests that methodologies based on Deep Learning capture more discriminating information, the proposed method maintains the crucial advantage of modeling muscle action directly through continuously optimized geometric landmarks. This provides a white-box solution that explicitly validates the psychological plausibility of the Facial Action Coding System (FACS). The complexity of these attention-based deep learning models comes at the cost of explainability, whereas the proposed method remains fully transparent; it achieves competitive performance using inner points, making it a robust, computationally efficient, and highly explainable tool for emotion recognition.

In this context, it is relevant to compare the best proposed model (MLP+DE) against another explainable geometric approach based on inner points, for instance, Arrazola et al. [5], which uses centroids as inner points. To visually contrast these last two referred methods, Figures 10 and 11 show collections of images correctly classified and misclassified, respectively. In both figures, blue points locate the optimized inner points (the best proposed model MLP+DE), and red points mark the centroids. For both classes of points, the convex hull is traced with the corresponding color. The blue convex hull associated with the best proposed model covers more area than the red convex hull associated with the centroids proposed by [5]. In addition, the blue points seem to be more uniformly distributed over their corresponding convex hull.

https://doi.org/10.3390/a19010057

Considering the traced convex hulls in Figures 10 and 11, and the triangles shown in Figures 6 and 8, the proposed approach constitutes a framework to generate more powerful interpreting tools than those offered by geometric approaches merely based on central points, such as [5,11].

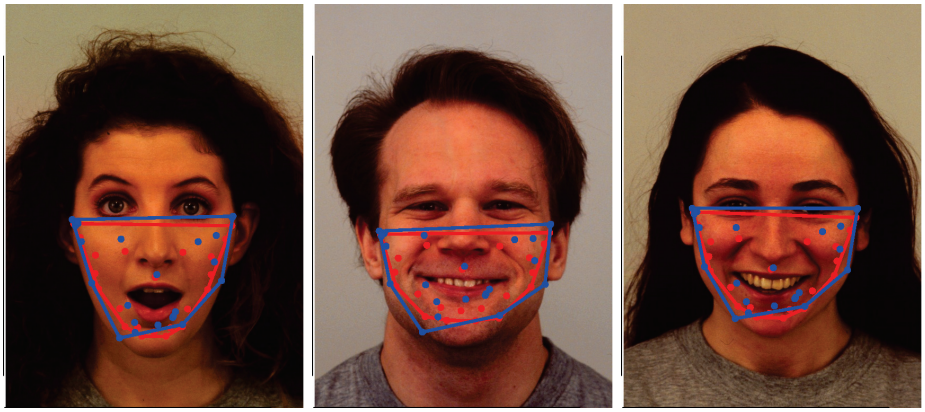


Figure 10. Gallery of properly classified images with inner points and convex hull traced for the best proposed model MLP+DE (blue), and the model based on centroids [5] (red).



Figure 11. Gallery of misclassified images with inner points and convex hull traced for the best proposed model MLP+DE (blue), and the model based on centroids [5] (red).

The proposed method based on metaheuristics is evaluated against the other strategies based on inner points under identical conditions (the same databases, 5-fold cross-validation, and classifiers) to ensure a fair comparison. Table 9 presents the mean accuracy for these three approaches across various machine learning models. The baseline strategies involve (1) using the centroid as proposed by [5], and (2) selecting the best of several notable points (centroid, incenter, Nagel, or Torricelli) as proposed by [11]. As shown, the proposed method consistently outperforms the others, demonstrating that metaheuristic optimization of inner points significantly enhances classification accuracy.

Table 9. Mean accuracy comparison of inner-point methods under identical experimental conditions.

Inner Points Strategy	Database	MLP	SVM	LDA	DT	XGB
Centroid [5]	KDEF	0.78	0.81	0.80	0.65	0.69
	JAFFE	0.62	0.71	0.62	0.43	0.51
Notable point [11]	KDEF	0.87	0.84	0.84	0.71	0.75
	JAFFE	0.77	0.81	0.71	0.50	0.58
Proposed (optimized)	KDEF	0.91	0.89	0.90	0.73	0.81
	JAFFE	0.81	0.83	0.75	0.54	0.72

https://doi.org/10.3390/a19010057

To provide a visual and explainable relation between the optimized inner points and Action Units (AUs) listed in Table 1, Figure 12 illustrates the facial landmarks where the radius of each point is proportional to the sum of the weights w assigned to it during the optimization process. This weighting scheme reflects the discriminative importance of each landmark, analogous to the loading vectors in Principal Component Analysis (PCA), which reveal the contribution of each dimension to the total variance [38]. In this context, the size of a landmark illustrates its relevance to the classification task. The points are numbered for reference. For instance, points 20 and 25 around the mouth are related to the Lip Corner Puller (AU12) and the Risorius (AU20). Similarly, points 5, 8, and 10 are associated with Jaw Drop (AU26), while points 1, 2, 17, and 18 capture the activation of the Inner Eyebrow Raiser (AU1) and the Outer Eyebrow Raiser (AU2).

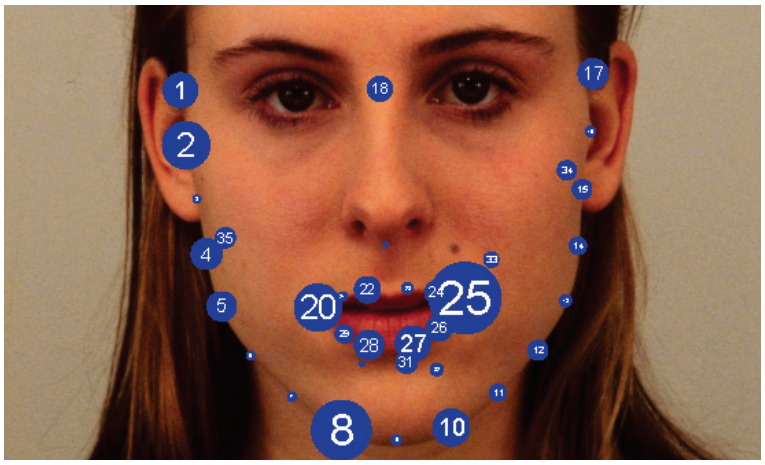


Figure 12. Spatial distribution of landmark relevance. The radius of each numbered point represents the cumulative weight assigned during the optimization of inner triangle points. A larger radius indicates higher discriminative power.

5. Conclusions

The research introduces a novel approach to facial expression recognition that significantly enhances the performance of geometric-based methods through the optimization of inner triangle points. The key contributions include the following:

1. Superior performance compared to traditional geometric and hybrid approaches;
2. Validation of the method's effectiveness across multiple machine learning classifiers and optimization algorithms;
3. Achievement of state-of-the-art results slightly lower than deep learning FER approaches, with accuracy of 0.91 (KDEF) and 0.83 (JAFFE).

Compared to discrete geometric centers, the proposed method provides a more precise model of muscle activation, offering a high-performance, explainable alternative for FER tasks. Continuously optimizing inner points provides a more accurate representation of muscle movements, bridging the gap between raw facial landmarks and precise muscle activation modeling. Excluding deep learning approaches, the proposal improves recognition accuracy and may also help with explainability and interpretability. Currently, the inner points merely help locate relevant areas of the face. To be valuable to non-expert users, the inner points should be interpreted automatically, explaining which AUs and motions are associated. Creating a user-friendly interface that justifies each recognized emotion could contribute to the affective computing community. This paper is a first step in that direction.

https://doi.org/10.3390/a19010057

Author Contributions: Conceptualization I.C.-A., E.G.G.d.P., A.H.-A., and M.-A.G.-R.; Investigation I.C.-A., E.G.G.d.P., A.H.-A., and M.-A.G.-R.; Formal analysis I.C.-A., E.G.G.d.P., A.H.-A., and M.-A.G.-R.; Writing—original draft, E.G.G.d.P. and I.C.-A. All authors have read and agreed to the published version of the manuscript.

Funding: Author thanks SECIHTI and CIMAT. This work was supported by SECIHTI under project IxM-SECIHTI No. 3097-7185.

Institutional Review Board Statement: Ethical review and approval were waived for this study due to the use of pre-existing and publicly available datasets (JAFFE and KDEF) collected by third parties for research purposes.

Informed Consent Statement: Not applicable. This study involved the secondary analysis of publicly available datasets (JAFFE and KDEF) where informed consent was obtained by the original investigators at the time of data collection.

Data Availability Statement: The data presented in this study are available in (1) KDEF https://kdef.se/ (accessed on 1 November 2025), (2) JAFFE http://www.kasrl.org/jaffe.html (accessed on 1 November 2025).

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

Abbreviations

The following abbreviations are used in this manuscript:

CNNs	Convolutional Neural Networks
CP	Convex Partition
DE	Differential Evolution
DT	Decision Tree
FACS	Facial Action Coding System
FER	Facial Expression Recognition
HOG	Histogram of Oriented Gradients
LDA	Linear Discriminant Analysis
MLP	Multilayer Perceptron
PSO	Particle Swarm Optimization
SVM	Support Vector Machine
XGB	XGBoost

References

1. Frank, M. Facial expressions. In *International Encyclopedia of the Social & Behavioral Sciences*; Elsevier: Amsterdam, The Netherlands, 2001; pp. 5230–5234.
2. Ekman, P. An argument for basic emotions. *Cogn. Emot.* **1992**, *6*, 169–200. [CrossRef]
3. Waller, B.M.; Gray, J.J.; Burrows, A.M. Selection for universal facial emotion. *Emotion* **2008**, *8*, 435–439. [CrossRef] [PubMed]
4. Wingenbach, T.S. Facial EMG investigating the interplay of facial muscles and emotions. In *Social and Affective Neuroscience of Everyday Human Interaction: From Theory to Methodology*; Springer: Cham, Switzerland, 2020.
5. Solis-Arrazola, M.A.; Sanchez-Yañez, R.E.; Garcia-Capulin, C.H.; Rostro-Gonzalez, H. Enhancing image-based facial expression recognition through muscle activation-based facial feature extraction. *Comput. Vis. Image Underst.* **2024**, *240*, 103927. [CrossRef]
6. Parra-Dominguez, G.; Garcia-Capulin, C.; Sanchez-Yanez, R. Automatic facial palsy diagnosis as a classification Problem Using Regional information extracted from a photograph. *Diagnostics* **2022**, *12*, 1528. [CrossRef] [PubMed]
7. Ekman, P.; Friesen, W.V. *Facial Action Coding System: A Technique for the Measurement of Facial Movement*; Consulting Psychologists Press: Palo Alto, CA, USA, 1978.
8. Barrett, L.F.; Russell, J.A. Core affect, prototypical emotional episodes, and other things called emotion: Dissecting the elephant. *J. Personal. Soc. Psychol.* **1999**, *76*, 805–819.
9. Ong, D.C.; Zaki, J.; Goodman, N.D. Computational models of emotion inference in theory of mind: A review and roadmap. *Top. Cogn. Sci.* **2019**, *11*, 338–357. [CrossRef] [PubMed]

 https://doi.org/10.3390/a19010057

10. Li, J.; Jin, K.; Zhou, D.; Kubota, N.; Ju, Z. Attention mechanism-based CNN for facial expression recognition. *Neurocomputing* **2020**, *411*, 340–350. [CrossRef]
11. Aguilera-Hernández, E.I.; Cruz-Aceves, I.; Hernández-Aguirre, A.; Moya-Albor, E.; Brieva, J. Image Facial Expression Recognition based on Active Muscles and their Notable Triangle Points. In Proceedings of the International Symposium on Biomedical Image Processing and Analysis, Pasto, Colombia, 19–21 November 2025.
12. Ouanan, H.; Ouanan, M.; Aksasse, B. Facial landmark localization: Past, present and future. In Proceedings of the 2016 4th IEEE International Colloquium on Information Science and Technology (CiSt), Tangier, Morocco, 24–26 October 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 487–493.
13. Benli, K.S.; Eskil, M.T. Extraction and selection of muscle based features for facial expression recognition. In Proceedings of the 2014 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1651–1656.
14. Perusquía-Hernández, M.; Dollack, F.; Tan, C.; Namba, S.; Ayabe-Kanamura, S.; Suzuki, K. Facial movement synergies and action unit detection from distal wearable electromyography and computer vision. *IEEE Access* **2020**, *8*, 146585–146600.
15. Bello, H.; Zhou, B.; Lukowicz, P. Facial muscle activity recognition with reconfigurable differential stethoscope-microphones. *Sensors* **2020**, *20*, 4904. [CrossRef] [PubMed]
16. Lundqvist, D.; Flykt, A.; Öhman, A. The Karolinska Directed Emotional Faces (KDEF). *CD ROM Dep. Clin. Neurosci. Psychol. Sect. Karolinska Institutet* **1998**, *91*, 630.
17. Lyons, M.; Akamatsu, S.; Kamachi, M.; Gyoba, J. Coding facial expressions with gabor wavelets. In Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 14–16 April 1998; IEEE: Piscataway, NJ, USA, 1998; pp. 200–205.
18. Kazemi, V.; Sullivan, J. One millisecond face alignment with an ensemble of regression trees. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1867–1874.
19. Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
20. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
21. Breiman, L.; Friedman, J.H.; Olshen, R.A.; Stone, C.J. *Classification and Regression Trees*; Chapman & Hall/CRC: Boca Raton, FL, USA, 1984.
22. Fisher, R.A. The Use of Multiple Measurements in Taxonomic Problems. *Ann. Eugen.* **1936**, *7*, 179–188. [CrossRef]
23. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; KDD '16; pp. 785–794.
24. 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]
25. Mullen, K.; Ardia, D.; Gil, D.; Windover, D.; Cline, J. DEoptim: An R Package for Global Optimization by Differential Evolution. *J. Stat. Softw.* **2011**, *40*, 1–26. [CrossRef]
26. Kennedy, J.; Eberhart, R. Particle Swarm Optimization. In Proceedings of the ICNN'95-International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; IEEE: Piscataway, NJ, USA, 1995; Volume 4, pp. 1942–1948. [CrossRef]
27. Bendtsen, C. *pso: Particle Swarm Optimization*; R package version 1.0.4; R Foundation: Vienna, Austria, 2022. [CrossRef]
28. Clerc, M. *L'aléatoire Contrôlé en Optimisation*; ISTE Group: London, UK, 2015; Volume 1.
29. de Paz, E.G.; Vaquera-Huerta, H.; Albores-Velasco, F.J.; Bauer-Mengelberg, J.R.; Romero-Padilla, J.M. Convex Partition: A Bayesian Regression Tree for Black-box Optimisation. *Pattern Recognit. Lett.* **2025**, *196*, 344–350. [CrossRef]
30. de Paz, E.G.; Huerta, H.V.; Velasco, F.J.A.; Mengelberg, J.R.B.; Padilla, J.M.R. *CPoptim: Convex Partition Optimisation*; R package version 0.1.0; R Foundation: Vienna, Austria, 2024. [CrossRef]
31. Martino, L.; Luengo, D.; Miguez, J. *Independent Random Sampling Methods*, 1st ed.; Statistics and Computing; Springer International Publishing: Cham, Switzerland, 2018. [CrossRef]
32. Rashad, M.; Alebiary, D.; Aldawsari, M.; Elsawy, A.; AbuEl-Atta, A.H. FERDCNN: An efficient method for facial expression recognition through deep convolutional neural networks. *PeerJ Comput. Sci.* **2024**, *10*, e2272. [CrossRef] [PubMed]
33. Ahadit, A.B.; Jatoth, R.K. A novel multi-feature fusion deep neural network using HOG and VGG-Face for facial expression classification. *Mach. Vis. Appl.* **2022**, *33*, 55. [CrossRef]
34. Puthanidam, R.V.; Moh, T.S. A Hybrid Approach for Facial Expression Recognition. In Proceedings of the 12th International Conference on Ubiquitous Information Management and Communication, Langkawi, Malaysia, 5–7 January 2018; IMCOM '18. [CrossRef]

 https://doi.org/10.3390/a19010057

35. Sun, Z.; Hu, Z.P.; Wang, M.; Bai, F.; Sun, B. Robust Facial Expression Recognition with Low-Rank Sparse Error Dictionary Based Probabilistic Collaborative Representation Classification. *Int. J. Artif. Intell. Tools* **2018**, *27*, 1850001. [CrossRef]
36. Fan, Y.; Lu, X.; Li, D.; Liu, Y. Video-based emotion recognition using CNN-RNN and C3D hybrid networks. In Proceedings of the 18th ACM International Conference on Multimodal Interaction, Tokyo, Japan, 12–16 November 2016; ICMI '16; pp. 445–450. [CrossRef]
37. Kas, M.; Merabet, Y.E.; Ruichek, Y.; Messoussi, R. New framework for person-independent facial expression recognition combining textural and shape analysis through new feature extraction approach. *Inf. Sci.* **2021**, *549*, 200–220. [CrossRef]
38. Dillon, W.; Goldstein, M. *Multivariate Analysis: Methods and Applications*; Wiley Series in Probability and Statistics; Wiley: Hoboken, NJ, USA, 1984.

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/a19010057

algorithms

MDPI

Article

Efficient Deep Learning-Based M-PSK Detection for OFDM V2V Systems Using MobileNetV3

Luis E. Tonix-Gleason [1,†], José A. Del-Puerto-Flores [1,*], Fernando Peña-Campos [2,†], Dunstano del Puerto-Flores [3], Juan-Carlos López-Pimentel [1], Carolina Del-Valle-Soto [1] and Luis René Vela-Garcia [2]

[1] Facultad de Ingeniería, Universidad Panamericana, Álvaro del Portillo 49, Zapopan 45010, Mexico; ltonix@up.edu.mx (L.E.T.-G.); clopezp@up.edu.mx (J.-C.L.-P.); cvalle@up.edu.mx (C.D.-V.-S.)
[2] Department of Electrical Engineering, Communications Section, Cinvestav, Guadalajara 45019, Mexico; fernando.pena@cinvestav.mx (F.P.-C.); luis.vela@cinvestav.mx (L.R.V.-G.)
[3] Department of Mechanical-Electrical Engineering, CUCEI, University of Guadalajara, Guadalajara 44430, Mexico; d.delpuerto@academicos.udg.mx
* Correspondence: jpuerto@up.edu.mx
† These authors contributed equally to this work.

Abstract

This paper investigates M-PSK symbol detection in Orthogonal Frequency Division Multiplexing (OFDM) systems for wideband Vehicle-to-Vehicle (V2V) communications using lightweight convolutional neural networks. In doubly dispersive channels, Inter-Carrier Interference (ICI) degrades subcarrier orthogonality, rendering conventional equalization ineffective. Current ICI mitigation techniques face a trade-off between Bit-Error Rate (BER) performance and computational complexity, limiting their applicability in dynamic vehicular scenarios. To address this issue, a low-complexity MobileNetV3-based receiver is proposed, incorporating a signal-model-driven preprocessing stage that compensates for Doppler-induced phase distortions responsible for ICI. Simulation results show that the proposed receiver improves BER performance compared to conventional equalizers and recent neural-based schemes in the low-SNR regime (below 15 dB) while maintaining computational complexity close to linear least-squares detection.

Keywords: deep learning; DNN; ICI; neural network; OFDM; V2V

1. Introduction

In contemporary Intelligent Transportation Systems (ITS), the exchange of real-time information is fundamental. In this context, Vehicle-to-Infrastructure (V2I) and Vehicle-to-Vehicle (V2V) communication schemes must be efficiently integrated. However, these links typically operate over Doubly Selective Channels (DSCs), which arise from multipath propagation, environmental scatterers, and the high mobility of vehicles, thereby posing significant challenges to the design and performance of communication systems [1–4].

In V2V environments, vehicular motion induces significant Doppler spreads that lead to the loss of orthogonality among Orthogonal Frequency Division Multiplexing (OFDM) subcarriers, making Inter-Carrier Interference (ICI) particularly pronounced [5–8]. As a result, the induced ICI degrades the effectiveness of OFDM-based V2V systems and hinders fundamental receiver tasks, such as channel estimation, data detection, and error correction, which are required to mitigate their effects. These tasks typically entail high computational

complexity, exceeding $\mathcal{O}(N^3)$, thereby complicating their real-time implementation and operation. These limitations have motivated considerable research interest in developing approaches that reduce computational complexity while improving performance compared to conventional equalization and data detection schemes [9–11].

Linear detectors, including Least Squares (LS) and Linear Minimum Mean Square Error (LMMSE), have been extensively investigated [12,13]. Although these schemes are attractive due to their simplicity and suitability for hardware implementation, their Bit-Error Rate (BER) performance is often limited compared to nonlinear methods. This performance degradation is mainly attributed to the fact that linear detectors perform symbol detection without exploiting the discrete structure of the signal constellation [4,14], which leads to suboptimal decisions, particularly in V2V scenarios characterized by low signal-to-noise ratios. Despite these limitations, their low computational complexity and reduced processing requirements continue to make them appealing candidates for practical systems, thereby highlighting the need for improved schemes that preserve these advantages while enhancing performance.

Nonlinear detectors exploit the finite symbol alphabet to improve performance. Common examples include Ordered Successive Interference Cancellation (OSIC) [15] and Decision Feedback Equalization (DFE) [16,17], and Maximum Likelihood (ML) detection [14], which typically achieve lower BER than LMMSE; however, both linear and nonlinear methods experience significant performance degradation in vehicular channels due to ICI. Although nonlinear detectors provide strong performance, this is often achieved at the expense of high computational complexity, typically exceeding $\mathcal{O}(N^3)$, which hinders real-time implementation. To alleviate this complexity, suboptimal ML variants restrict the search to a lower-dimensional latent space or perform candidate sequence pruning.

Neural Networks (NN) constitute a natural bridge between linear and nonlinear approaches by offering a favorable trade-off between performance and computational complexity [18–20]. On the one hand, nonlinear activations enable the modeling of complex decision boundaries, leading to significant performance gains; on the other hand, the underlying operations largely reduce to matrix-vector products, which are highly amenable to hardware acceleration and efficient real-time execution. Within this framework, NN are built from two fundamental blocks: fully connected layers and convolutional layers. Convolutional layers are especially attractive because they reuse the same weights and focus on local regions, allowing for faster inference with a lower computational cost. In the context of equalizer design, this efficiency translates into MobileNetV3-based architectures that achieve low latency, good scalability, and strong performance, outperforming linear NN. Moreover, MobileNetV3 incorporates lightweight activation functions such as hard-swish and hard-sigmoid, along with depthwise-separable convolutions, which drastically reduce the number of parameters and operations compared to standard convolutions [21], further reinforcing its suitability for real-time vehicular applications.

Neural networks have recently been integrated into OFDM receiver stages in an effort to increase detection accuracy while maintaining practical deployment viability. Most NN-based methods for rapidly time varying multipath channels focus primarily on channel estimation [22–24]. While others like ComNet explicitly combine estimation and equalization [25], none of these cutting-edge techniques takes advantage of the channel matrix to perform a dedicated preprocessing step prior to the network input.

Contribution and Article Structure

This work addresses the problem of low-complexity symbol detection in OFDM-based V2V communication systems operating over doubly dispersive channels, where inter-

https://doi.org/10.3390/a19030210

carrier interference severely degrades performance. The main contributions of this paper are summarized as follows:

- **MobileNetV3-based neural equalization for V2V links:** A lightweight convolutional neural network based on MobileNetV3 is proposed for M-PSK detection in OFDM systems affected by ICI, adapting an architecture originally designed for mobile devices to the vehicular communications context.
- **Statistics-informed ICI-aware preprocessing:** A channel-prediction-based preprocessing stage is introduced to mitigate ICI prior to neural equalization, relieving the network from directly suppressing interference and allowing it to focus on detection.
- **Signal-model-guided phase distortion compensation:** The signal model proposed during the preprocessing stage compensates for Doppler-induced phase distortions, thereby facilitating MobileNetV3 to perform detection by efficiently exploiting phase–data correlations.
- **Favorable performance–complexity trade-off:** The receiver proposed achieves a lower bit-error rate than conventional linear, nonlinear, and recent neural-based approaches, while maintaining computational complexity close to linear least-squares detection.

Overall, the proposed framework provides an efficient and scalable detection solution for ICI-impaired V2V OFDM systems, bridging the gap between the high performance of nonlinear methods and the low complexity of conventional linear receivers.

The remainder of this paper is organized as follows. Section 2 presents the system model of the proposed OFDM framework. Section 3 describes the proposed receiver architecture based on MobileNetV3 for OFDM signal detection. Section 4 analyzes the computational complexity and presents the performance results of the proposed receiver. Additionally, Section 5 provides a discussion of the experimental results. Finally, the conclusions of this work are drawn in Section 6.

2. Model System

The model includes the OFDM modulation of the 802.11p standard. In this context, the signal $y[n]$ at the receiver side in complex baseband, following synchronization and cyclic prefix (CP) removal, is denoted by:

$$y[n] = \sum_{l=0}^{L-1} h[n,l]x[\langle n-l\rangle_N] + w[n], \tag{1}$$

using $\langle\cdot\rangle_N$ to denote N-modulus indexing, the transmitted signal at the n-th sample is denoted as $x[n]$. The channel's impulse response at time n for the l-th preceding sample is represented as $h[n,l]$, where L denotes the number of channel taps; whereas $w[n]$ signifies the complex Additive White Gaussian Noise (AWGN) characterized by zero mean and variance $\sigma_w^2 = N_0/2$, where N_0 denotes the noise power spectral density. The circular convolution, as delineated in (1), between the impulse response $h[n,l]$ and $x[n]$, can be reformulated in a matrix-vector representation as follows:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}, \tag{2}$$

where $\mathbf{H}$ denotes a $N \times N$ matrix, indexed by $0, 1, \ldots, N-1$ in each dimension, with its elements formulated from the Channel Impulse Response (CIR) coefficients as follows:

$$[\mathbf{H}]_{n,n'} = h\big[n, \langle n-n'\rangle_N\big], \tag{3}$$

https://doi.org/10.3390/a19030210

for $n, n' \in \{0, 1, \ldots, N-1\}$. The CIR is taken to be zero whenever $\langle n - n' \rangle_N > L - 1$. After CP removal, the received OFDM symbol in the frequency domain is obtained by applying the normalized Discrete Fourier Transform (DFT):

$$\mathbf{u} = \mathbf{G}\mathbf{s} + \mathbf{z}, \tag{4}$$

where $\mathbf{u}$, $\mathbf{s}$, and $\mathbf{z}$ are the DFTs of $\mathbf{y}$, $\mathbf{x}$, and $\mathbf{w}$, respectively. The channel frequency matrix is $\mathbf{G} = \mathbf{F}\mathbf{H}\mathbf{F}^H$, with $\mathbf{F}$ denoting the normalized DFT matrix. The data vector $\mathbf{s} = [\, s[0], s[1], \ldots, s[N_D - 1] \,]^T$ contains N_D M-PSK symbols, which can be written in terms of magnitude A and phases $\theta[n]$ as:

$$\mathbf{s} = |\alpha| [\, e^{j\theta[0]}, \, e^{j\theta[1]}, \cdots, \, e^{j\theta[N_D-1]} \,]^T, \tag{5}$$

where $e^{j\theta[n]}$ denotes the n-th transmitted complex value in the discrete frequency domain. In a typical OFDM receiver with ICI cancellation, the estimate of $\mathbf{G}$ is used to perform data detection via LMMSE, e.g., in the form (where σ is the noise-regularization term equivalent to $\sigma = 1/\mathrm{SNR}$ for unit-power symbols):

$$\hat{\mathbf{s}} = \left(\mathbf{G}\mathbf{G}^H + \sigma \mathbf{I}\right)^{-1} \mathbf{G}^H \mathbf{u}. \tag{6}$$

This structure is taken as a reference for the proposed NN-based detector to compute the MobileNetV3 input vector:

$$\mathbf{u}_d = \mathbf{G}^H \mathbf{u} = \mathbf{G}^H \mathbf{G}\mathbf{s} + \mathbf{G}^H \mathbf{z}, \tag{7}$$

where $(\cdot)^H$ denotes the Hermitian transpose. Here, $\mathbf{u}_d$ denotes the matched-filter preprocessed received vector, obtained as $\mathbf{u}_d = \mathbf{G}^H \mathbf{u}$ and used as the NN input. The operation $\mathbf{G}^H \mathbf{u}$ corresponds to a Matched-Filter (MF) preprocessing step with respect to the effective frequency-domain channel $\mathbf{G}$, since it correlates the received vector with the conjugate channel response. The matched filter does not cancel interference and therefore does not suppress ICI completely.

In doubly selective channels, $\mathbf{G}$ is generally non-diagonal, so $\mathbf{G}^H \mathbf{G}$ contains off-diagonal terms that represent residual ICI. Nevertheless, applying $\mathbf{G}^H$ partially compensates channel-induced phase distortions and tends to tighten the received constellation, as illustrated in Figure 1. Overall, this step reduces the learning burden on the neural network and enables a more streamlined architecture with fewer layers, while still leaving residual ICI that can limit performance at high SNR.

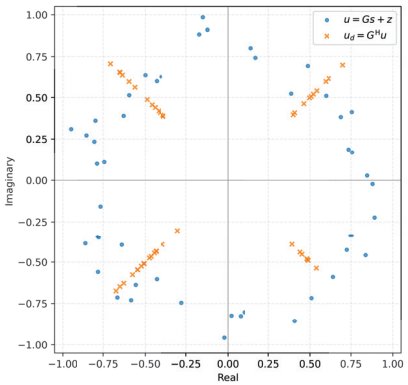


Figure 1. Received symbol constellation before and after preprocessing.

https://doi.org/10.3390/a19030210

3. Deep Learning MobileNetV3

CNNs effectively capture spatial correlations; however, their computational cost increases significantly with the number of feature extraction channels. MobileNetV3 addresses this limitation through the use of depthwise separable convolutions, enabling a more computationally efficient architecture.

In this work, MobileNetV3 is adapted for M-PSK detection in OFDM systems operating over doubly dispersive channels. Since standard CNN implementations operate in the real domain, the complex channel matrix $\mathbf{G} \in \mathbb{C}^{48\times 48}$ is first transformed into its polar representation. The magnitude and phase components are then arranged as two input channels forming the tensor $\chi \in \mathbb{R}^{48\times 48\times 2}$. Specifically,

$$\chi_0 = |\mathbf{G}| \tag{8}$$

$$\chi_1 = \angle \mathbf{G} \tag{9}$$

The input tensor is constructed by stacking these components along the channel dimension; i.e.,

$$\chi = \text{stack}(\chi_0, \chi_1). \tag{10}$$

The complete end-to-end processing chain—including preprocessing, network inference, and postprocessing—is illustrated in Figure 2 and described in detail in Algorithms 1 and 2. The network output is defined as $\hat{\boldsymbol{\rho}} \in \mathbb{R}^{1\times 48}$ and is projected onto the complex unit circle to generate the phasor $e^{j\hat{\boldsymbol{\rho}}}$. This phasor acts as a learned phase correction term and is applied elementwise, via the Hadamard operator $\odot$, to the Matched-Filter (MF) preprocessing output $\mathbf{u}_d$ defined in (7). The angular component of the resulting compensated signal provides the final phase estimate, which is expressed as:

$$\hat{\theta} = \arg\left(\mathbf{u}_d \odot e^{j\hat{\rho}}\right) \tag{11}$$

The training objective does not aim to directly regress $\hat{\rho}$, as this variable serves only as an intermediate latent representation produced by the network. Instead, the optimization is formulated over the final phase estimate defined in (11), which directly impacts the symbol decision process. In this way, the learning procedure is explicitly aligned with the system performance metric.

Specifically, the goal is to minimize the discrepancy between the estimated phase $\hat{\theta}$ and the true phase $\theta = \angle(s)$, where θ denotes the ground-truth phase of the transmitted symbol vector. To achieve this, it is necessary to define a loss function that properly quantifies this discrepancy in the angular domain.

A straightforward approach would be to employ a direct Mean-Squared Error (MSE) between angles,

$$e = \text{MSE}(\hat{\theta}, \theta), \tag{12}$$

however, this metric is not appropriate due to the inherent 2π periodicity of phase. When $\hat{\theta}$ and θ lie on opposite sides of the wrapping boundary (e.g., near $-\pi$ and $+\pi$), direct subtraction may yield an artificially large error, even though the phases are physically close. To properly account for the circular nature of phase, one could define the wrapped angular difference as

$$\Delta\theta = \text{wrap}(\hat{\theta} - \theta), \tag{13}$$

https://doi.org/10.3390/a19030210

and compute the mean-squared wrapped phase error over a batch of size N as

$$e_{\text{wrap}} = \frac{1}{N}\sum_{n=1}^{N}(\Delta\theta_n)^2. \tag{14}$$

However, in the proposed approach, an equivalent reformulation in the complex domain is adopted. Since $e^{j\theta}$ represents a unit phasor on the complex circle, the phase discrepancy can be measured as the Euclidean distance between the corresponding unit phasors. Under this formulation, the loss function is defined as:

$$\mathcal{L}_{\text{phasor}} = \frac{1}{N}\sum_{n=1}^{N}\left\|e^{j\hat{\theta}_n} - e^{j\theta_n}\right\|_2^2 \tag{15}$$

expanding the squared norm in the complex plane yields

$$\left\|e^{j\hat{\theta}} - e^{j\theta}\right\|_2^2 = 2\big(1-\cos(\hat{\theta}-\theta)\big), \tag{16}$$

so that the loss can be equivalently expressed as:

$$\mathcal{L}_{\text{phasor}} = \frac{2}{N}\sum_{n=1}^{N}\big(1-\cos(\hat{\theta}_n-\theta_n)\big). \tag{17}$$

This formulation is inherently 2π-periodic and directly penalizes the true circular distance between phases, avoiding ambiguities associated with angular discontinuities.

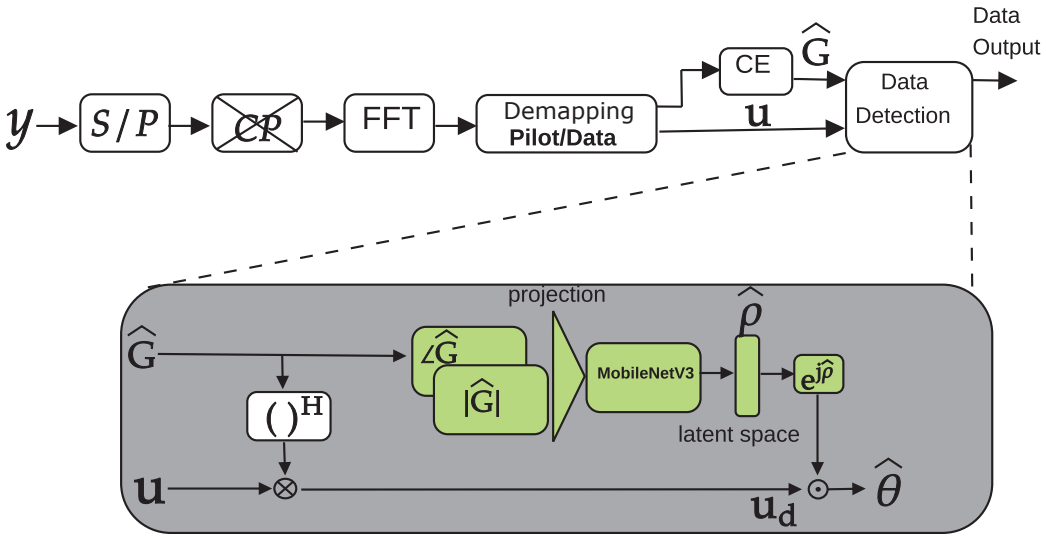


Figure 2. Block diagram of the proposed MobileNetV3 receiver.

Given the phase aware loss defined above, we can now describe how the proposed receiver is trained. Algorithm 1 details the end-to-end procedure used to generate training frames, apply the Matched-Filter (MF) preprocessing, form the real-valued channel representation χ, and forward it through MobileNetV3 to obtain the phase-correction vector $\hat{\rho}$. The correction is mapped to a unit phasor $e^{j\hat{\rho}}$ and applied elementwise to the MF output, yielding the phase estimate $\hat{\theta}$ in (11). The wrapped phase discrepancy between $\hat{\theta}$ and the ground truth θ is then evaluated with the proposed loss and used to update the network weights via backpropagation. Algorithm 2 follows the same pipeline at test time but omits the loss computation and weight updates.

https://doi.org/10.3390/a19030210

Algorithm 1 Training of the proposed MobileNetV3 receiver

Input: Number of frames N_f; subcarriers M=48; channel matrix $\mathbf{G} \in \mathbb{C}^{M\times M}$; constellation $\mathcal{A}$; SNR.
Output: Trained weights $\mathcal{W}$.

for $k = 1$ to N_f **do**
  Draw symbols $\mathbf{s} \in \mathcal{A}^M$.
  Generate noise $\mathbf{z} \sim \mathcal{CN}(0, \sigma^2\mathbf{I})$ according to SNR.
  Receive $\mathbf{u} = \mathbf{G}\mathbf{s} + \mathbf{z}$.
  MF preprocessing: $\mathbf{u}_d = \mathbf{G}^H\mathbf{u}$.
  Build network input from G:
  $\mathbf{Ø}_0 \leftarrow \text{norm}(|\mathbf{G}|) \in \mathbb{R}^{M\times M}$ ▷ magnitude channel normalized
  $\mathbf{Ø}_1 \leftarrow (\angle\mathbf{G} + \pi)/(2\pi) \in \mathbb{R}^{M\times M}$ ▷ phase channel normalized
  $\mathbf{Ø} \leftarrow \text{stack}(\mathbf{Ø}_0, \mathbf{Ø}_1) \in \mathbb{R}^{2\times M\times M}$.
  Forward pass: $\hat{\boldsymbol{\rho}} \leftarrow \text{MobileNetV3}(\mathbf{Ø}; \mathcal{W}) \in \mathbb{R}^M$.
  Apply phase correction: $\tilde{\mathbf{u}} = \mathbf{u}_d \odot e^{j\hat{\boldsymbol{\rho}}}$.
  Predict phases $\hat{\boldsymbol{\theta}} \leftarrow \angle(\tilde{\mathbf{u}})$ and target $\boldsymbol{\theta} \leftarrow \angle(\mathbf{s})$.
  Wrapped phase error: $\mathbf{d} \leftarrow \hat{\boldsymbol{\theta}} - \boldsymbol{\theta}$.
  $\mathbf{d}_w \leftarrow \text{atan2}(\sin\mathbf{d}, \cos\mathbf{d})$.
  Loss $\mathcal{L} \leftarrow \text{MSE}(\mathbf{d}_w)$.
  Update $\mathcal{W}$ using backpropagation.
end for

Algorithm 2 Inference mode of the proposed MobileNetV3 receiver

Input: Trained weights $\mathcal{W}$; channel matrix $\mathbf{G} \in \mathbb{C}^{M\times M}$; received vector $\mathbf{u} \in \mathbb{C}^M$.
Output: Detected symbols $\hat{\mathbf{s}}$.

Build network input from G:
  $\mathbf{Ø}_0 \leftarrow \text{norm}(|\mathbf{G}|), \quad \mathbf{Ø}_1 \leftarrow (\angle\mathbf{G} + \pi)/(2\pi)$
  $\mathbf{Ø} \leftarrow \text{stack}(\mathbf{Ø}_0, \mathbf{Ø}_1) \in \mathbb{R}^{2\times M\times M}$
MF preprocessing: $\mathbf{u}_d = \mathbf{G}^H\mathbf{u}$.
Forward pass: $\hat{\boldsymbol{\rho}} \leftarrow \text{MobileNetV3}(\mathbf{Ø}; \mathcal{W})$
Apply Phase correction: $\tilde{\mathbf{u}} = \mathbf{u}_d \odot e^{j\hat{\boldsymbol{\rho}}}$.
Phase Correction $\tilde{\mathbf{u}} = \mathbf{u}_d \odot e^{j\hat{\boldsymbol{\rho}}}$
Estimate $\hat{\boldsymbol{\theta}} \leftarrow \angle(\tilde{\mathbf{u}})$

Rather than explicitly learning to cancel ICI entirely, the preprocessing stage partially mitigates ICI through an initial matched-filter step. By reducing the complexity of the interference, the network can focus on correcting residual distortions, primarily due to noise, by leveraging phase consistency among the received signals.

The effectiveness of this preprocessing approach is clearly demonstrated in Figure 1:

- **Before preprocessing (Blue scatters)**: Symbols are widely scattered due to ICI and Gaussian noise, limiting accurate identification.
- **After preprocessing (Orange scatters)**: Symbols are clearly clustered around the most effective QPSK constellation points, suggesting a substantial reduction in interference. This simplifies the learning process for MobileNetV3 correction.

For reproducibility, Table 1 details the dataset construction and stored tensors used in training: the transmitted 4-QAM (QPSK) symbols, the matched-filtered noisy received vectors, the per-frame channel matrices, and the binary LOS/NLOS indicator that controls how channel realizations are selected across frames, together with the fixed SNR and modulation order. In particular, LOS and NLOS channel snapshots are alternated (toggled) during dataset generation to expose the network to both propagation conditions; however, this indicator is used only for bookkeeping and analysis and is *not* provided as an input feature. Therefore, the proposed model is not explicitly aware of whether the current

 https://doi.org/10.3390/a19030210

received frame corresponds to LOS or NLOS, and it must learn a unified mapping that generalizes across both channel regimes.

Table 1. Data Generation.

Name	Type/Shape	Description
`s_all`	$\mathbb{C}^{48\times 100000}$	Tx 4-QAM $\mathbf{s}$ (unit average power)
`Mf_all`	$\mathbb{C}^{48\times 100000}$	Matched-filtered observations $\mathbf{G}^H\mathbf{u}$
`G_all`	$\mathbb{C}^{48\times 48\times 100000}$	Channel matrices $\mathbf{G}$
`isLOS`	$\{0,1\}^{100000\times 1}$	$1 =$ LOS, $0 =$ NLOS
`SNR`	scalar	15 dB
`modorder`	scalar	4 (Gray-mapped QAM)

Table 2 complements this by listing all learning-related settings for the proposed MobileNetV3 receiver, including the backbone choice, tensor shapes, wrapped phase-MSE objective, optimizer and learning rate, batch/epoch schedule, validation protocol, numerical precision, and magnitude/phase normalization used to build the network input.

Table 2. Hyperparameters for custom OFDM MobileNetV3.

Hyperparameter	Value
Backbone	MobileNetV3-Small (`torchvision`)
Input shape	$2 \times 48 \times 48$
Output dim	48
Loss	Wrapped phase MSE on $\angle\left(\mathbf{u_d} \odot e^{j\hat{\boldsymbol{\theta}}}\right)$ vs. $\angle(\mathbf{s})$
Optimizer	AdamW
Learning rate	1×10^{-3}
Batch size	64
Max epochs	10
Validation split	0.1
Num. workers	16
Precision	‘‘`32-true`’’
χ_0 magnitude norm	Per-sample min–max to $[0,1]$
χ_1 phase norm	Map $[-\pi,\pi] \to [0,1]$
$\mathbf{u_d}$ preprocessing	Real/imag, no normalization
Target phase	$\angle(\mathbf{s}) \in [-\pi,\pi]$
Checkpointing	Top-3 by `val_loss`; optional warm start/resume

4. Results and Experiments

The MobileNetV3-based equalizer was assessed in simulation for computational complexity ($\mathcal{O}$), BER, and Block-Error Rate (BLER). Performance was evaluated against OFDM systems utilizing both linear and nonlinear detectors. A WINNER II V2V channel model was employed to simulate a doubly selective environment [26]. All performance experiments were conducted under ideal channel state information conditions. The complete channel and system configurations are detailed in Table 3.

The proper dataset SNR for the network training was determined by running a series of simulations under several fixed and mixed SNR values. The results in Figure 3 show that training with a wider range of SNR values (or with mixed-SNR combinations) does not improve the final BER performance. In contrast, training only at 15 dB achieves a comparable BER while converging faster and requiring fewer computational resources.

https://doi.org/10.3390/a19030210

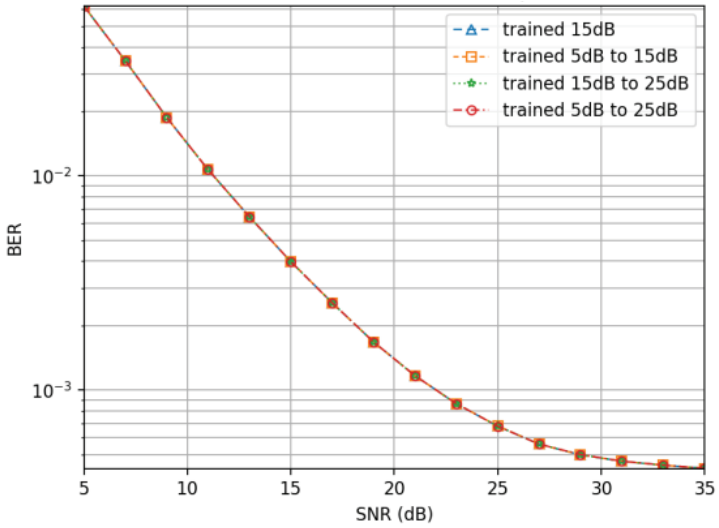


Figure 3. BER performance over different training SNR values.

The Big $\mathcal{O}$ Benchmark chart (Figure 4) presents the computing complexity of several equalization algorithms applied to OFDM signals. The x-axis represents N, indicating the number of symbols, while the y-axis measures 'Operations' in terms of real products, with values displayed on a logarithmic scale. The red dashed line at $N = 48$ delineates the particular complexity associated with the 802.11p standard parameters [27].

Table 3. Channel and system parameters.

Parameter (Units)	Value
Path Delay (μs)	$\{0, 0.1, 0.2, 0.3, 0.4, 1\}$
Path Power (dB)	$\{0, -3, -5, -7, -9, -15\}$
Bandwidth (MHz)	10
Taps number L	6
Doppler (kHz)	Jakes-0.5
Block length (N, N_D)	$\{64, 48\}$
Constellation size	4-PSK
Spectral efficiency (b/Hz/s)	1.5

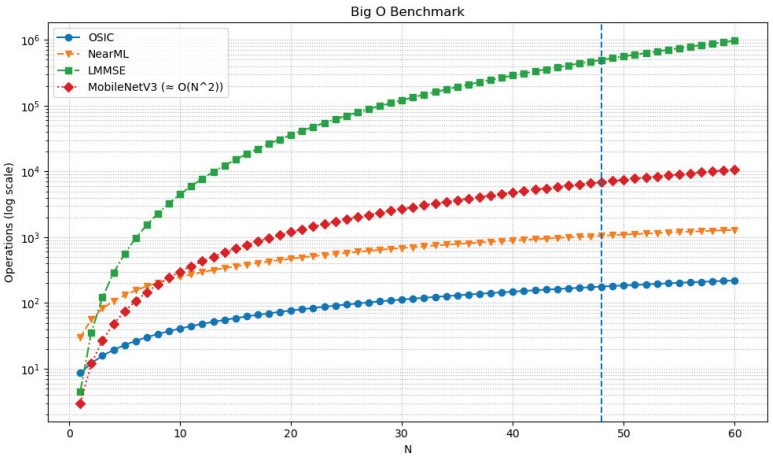


Figure 4. Complexity benchmark.

Figure 4 shows frame size (x-axis, $N = 48$) and operation count (y-axis, log scale). Key points follow:

https://doi.org/10.3390/a19030210

- The OSIC approach demonstrates a significant escalation in operational complexity as N increases. In contrast to prior observations, it exhibits the lowest complexity among the evaluated strategies. Concurrently, the growth trajectories of NearML stabilize as N escalates. Significantly, NearML exhibits a increased complexity compared to LMMSE.
- **MobileNetV3-Small complexity:** For an $N \times N$ input with 2 channels, each depthwise-separable block at layer ℓ with cumulative stride s_ℓ costs $\frac{N^2}{s_\ell^2}\left(k_\ell^2 C_\ell + C_\ell C_{\ell+1}\right)$ MACs, where k_ℓ is the kernel size and C_ℓ, $C_{\ell+1}$ are the input/output channel counts. Since MobileNetV3 uses fixed channel widths and small kernels (e.g., $k = 3$), the complexity grows mainly with the spatial size N^2. Early downsampling ($s_\ell > 1$) reduces the spatial size of feature maps (e.g., $48 \times 48 \to 24 \times 24$ for stride 2), so later layers operate on fewer pixels and the total operations drop significantly. Thus MobileNetV3 scales roughly with N^2 but is nearly constant for a fixed N (e.g., 48), unlike fully connected layers whose cost can grow quadratically with input size [21].
- Occupying the most complex is LMMSE, bearing the highest complexity, scaling as N^3 due to the need to invert a matrix.

The graph shows a comparison of both BER (Figure 5) and BLER (Figure 6) performance between PhaseNet (Figure 2) and classical equalization methods—MobileNetV3, ComNet, LMMSE, OSIC, and NearML—across various SNR values in an OFDM system. In the context of LTE, BLER is defined as the ratio of the number of erroneous blocks to the total number of received blocks. This metric is calculated using a Cyclic Redundancy Check (CRC) evaluation, where each transport block has an attached CRC. At the receiver side, the transport block is considered successfully decoded if the CRC calculated by the receiver matches the CRC sent by the transmitter. In BLER metrics, MobileNetV3 behaves pretty similar to the classical methods and does not show a significant improvement. However, its capability to being parallelized makes it a suitable approach for real-time implementation.

- **Low SNR Range (5–15 dB):** In this range, MobileNetV3 has lower BER than classical methods. The BER for MobileNetV3 decreases significantly, reaching values below 10^{-2} by 10 dB, whereas methods such as OSIC, LMMSE, and NearML remain closer to 10^{-1}.
- **Moderate SNR Range (15–20 dB):** As the SNR increases, MobileNetV3 slows its decreasing rate compared to traditional methods around 17 dB. ComNet shows only limited improvement, with a slow reduction in BER and an almost flat trend. By 20 dB, the MobileNetV3 BER approaches 10^{-3}, which is higher than the classical methods, it does not surpass the performance of the classical methods.
- **High SNR Range (20–25 dB):** At higher SNR values, the performance of MobileNetV3 flattens and stabilizes around 10^{-3}, while classical methods continue to decrease and achieve better results under noise. ComNet shows a similar stabilization but with a noticeable offset of about one order of magnitude, reaching BER values near 10^{-2}. Traditional detectors such as LMMSE and OSIC achieve slightly lower error rates, although the gap becomes minimal in this regime. Neural networks demonstrate their limitations beyond the trained SNR range but remain competitive with classical approaches.

https://doi.org/10.3390/a19030210

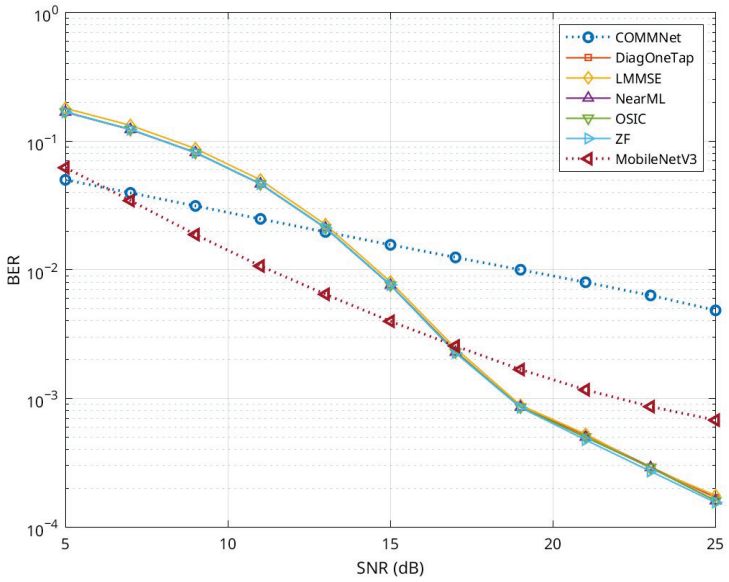


Figure 5. MobileNetV3 vs. classical methods in BER performance.

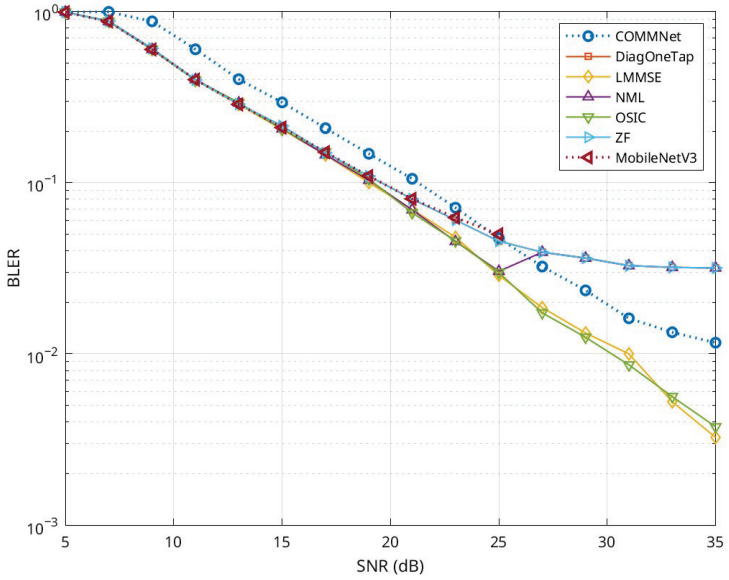


Figure 6. MobileNetV3 vs. classical methods in BLER performance.

We also extended the evaluation scope to higher-order constellations. Figures 7 and 8 report the BER and BLER performance, respectively, when the same receiver architecture is tested on QPSK, 8-PSK, and 16-QAM.

As the constellation order increases, constellation points become more densely packed for a fixed average symbol energy. This reduces the effective separation between decision regions, making symbol detection more sensitive to additive noise as well as to residual phase and amplitude errors. As a result, for the same SNR, higher-order constellations generally exhibit higher symbol error rates and, under Gray mapping, higher BER.

https://doi.org/10.3390/a19030210

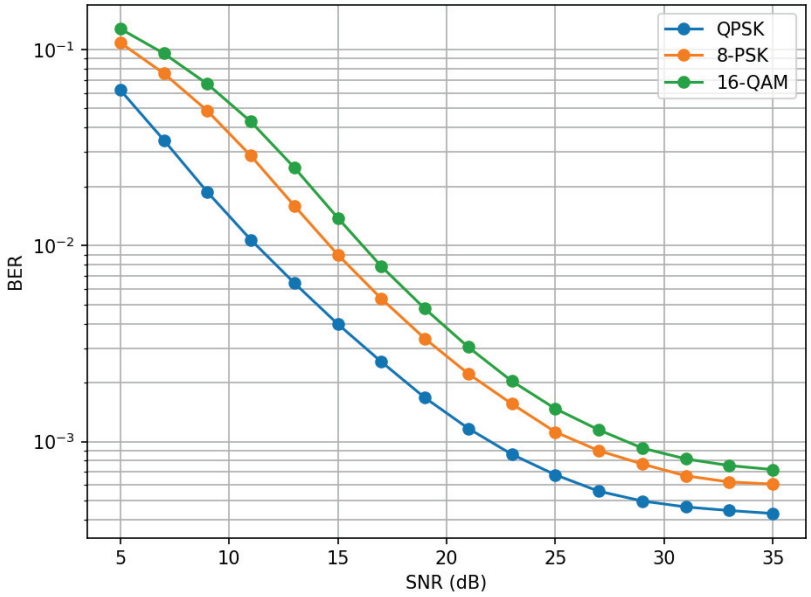


Figure 7. BER performance across modulation orders (QPSK, 8-PSK, 16-QAM).

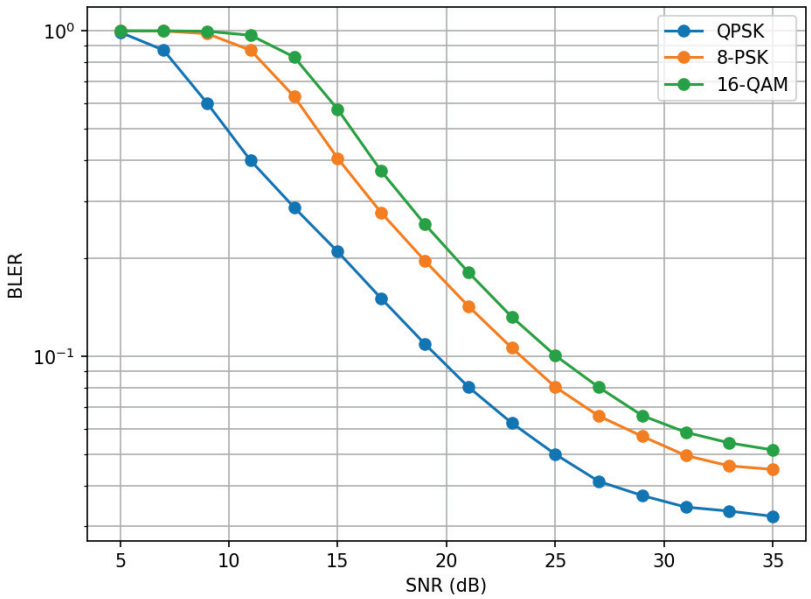


Figure 8. BLER performance across modulation orders (QPSK, 8-PSK, 16-QAM).

This trend is consistent with our results: compared to QPSK, the 8-PSK BER curve is shifted upward by approximately two orders of magnitude on average, and moving from 8-PSK to 16-QAM introduces roughly one additional order of magnitude. The BLER curves follow the same ordering, indicating that the network maintains the same relative behavior when evaluated on blocks. This is expected because BLER treats a block as incorrect if it contains even a single bit error.

5. Discussion

MobileNetV3 is an efficient neural network architecture designed to achieve fast inference by relying primarily on lightweight convolutional operations. Its depthwise separable convolution strategy is particularly well suited for embedded and real-time applications, as it significantly reduces computational cost while maintaining strong feature

https://doi.org/10.3390/a19030210

extraction capability. In this work, we explore a new application of MobileNetV3 in a regression setting, where the objective is to estimate a latent representation of the channel as a single vector. Conceptually, this can be interpreted as extracting a diagonal like summary of the channel matrix, while simultaneously incorporating the most informative features from the entire channel.

Beyond the architectural efficiency of MobileNetV3, the novelty of the proposed receiver lies in how channel knowledge is exploited and how the network output is integrated into the classical OFDM chain. Instead of directly estimating transmitted symbols or learning a full equalizer, the network processes the complex channel matrix $\mathbf{G} \in \mathbb{C}^{48\times 48}$ by forming the real-valued two-channel tensor $\chi = \text{stack}(|\mathbf{G}|, \angle\mathbf{G}) \in \mathbb{R}^{48\times 48\times 2}$ and regresses a phase-correction vector $\hat{\boldsymbol{\rho}} \in \mathbb{R}^{1\times 48}$. The model is trained with an MSE objective that minimizes the discrepancy between the corrected phase $\angle\left(\mathbf{u}_d \odot e^{j\hat{\rho}}\right)$ and the ideal phase $\angle\mathbf{s}$. During inference, $\hat{\boldsymbol{\rho}}$ is mapped to a unit-magnitude phasor $e^{j\hat{\rho}}$ and applied elementwise to the matched-filter output $\mathbf{u}_d$ (which already partially mitigates ICI), yielding the final phase estimate $\hat{\theta} = \arg\left(\mathbf{u}_d \odot e^{j\hat{\rho}}\right)$. This structured design targets residual phase distortions after model-based preprocessing, rather than replacing the detection or equalization pipeline.

Furthermore, a matched-filter preprocessing stage is explicitly introduced prior to the neural network to partially mitigate channel-induced distortions and tighten the received constellation, thereby reducing the learning burden of the network. In contrast to end-to-end architectures that implicitly learn estimation and equalization jointly, the proposed approach enforces a structured decomposition: the channel matrix is explicitly used to construct the preprocessing stage and the network input representation, while training is performed using a wrapped phase-MSE criterion to properly handle phase periodicity. This model-aware design differentiates the proposed receiver from existing NN-based OFDM detectors such as ComNet.

One limitation of the current design is that the network can only be trained over a limited SNR range, and it does not incorporate temporal context from previous OFDM frames. As a consequence, residual inter carrier interference cannot be fully eliminated when the interference originates from adjacent V2V frames. Nevertheless, the model effectively reduces BER levels over low SNR values and various channel impairments within the considered operating range.

In terms of BLER performance, the network behaves similarly to the classical methods baseline. This is a promising result as it indicates that the network is capable of learning and reproducing the behavior of a traditional signal processing method. It is also worth noting that neural networks are highly parallelizable, even on embedded platforms. With current state-of-the-art hardware accelerators such as NPUs, FPGAs, and lightweight AI inference engines, deploying MobileNet-based architectures in practical V2V systems is becoming increasingly feasible.

6. Conclusions

In this paper, a low-complexity detection scheme for M-PSK symbols in OFDM systems operating over doubly dispersive V2V channels is proposed. The MobileNetV3-based detector emerges as a suitable solution for mitigating the effects of severe ICI induced by Doppler spread in vehicular communication links. Unlike classical linear and nonlinear equalizers, which rely on computationally expensive matrix inversions or iterative interference cancellation, the proposed receiver introduces a lightweight preprocessing stage based on the signal model that compensates for Doppler-induced phase distortions. This preprocessing enables the neural network to focus on symbol detection by exploiting phase-data correlations rather than directly suppressing ICI. The proposed MobileNetV3-based

 https://doi.org/10.3390/a19030210

receiver is particularly effective in the low-to-mid SNR regime, where ICI degrades the performance of conventional OFDM systems. Simulation results show that the proposed receiver improves BER performance compared to linear and nonlinear detectors, as well as ComNet, in the low-SNR regime (below 15 dB). At high SNR values, a mild performance saturation is observed, which is consistent with the absence of temporal memory in the network architecture. Nevertheless, this trade-off is justified by the reduced computational complexity of the proposed solution.

Overall, the proposed receiver distinguishes itself from existing equalization and neural-based schemes through its compact architecture, inherent parallelism, and reduced computational burden. These characteristics make the MobileNetV3-based approach a strong candidate for real-time V2V receivers and hardware-constrained platforms such as FPGA- or ASIC-based embedded systems.

Author Contributions: Conceptualization, L.E.T.-G., J.A.D.-P.-F. and F.P.-C.; methodology, L.E.T.-G., J.A.D.-P.-F. and F.P.-C.; software, L.E.T.-G., J.A.D.-P.-F. and F.P.-C.; validation, D.d.P.-F., J.-C.L.-P., C.D.-V.-S. and L.R.V.-G.; formal analysis, L.E.T.-G., J.A.D.-P.-F. and F.P.-C.; investigation, D.d.P.-F., J.-C.L.-P., C.D.-V.-S. and L.R.V.-G.; resources, D.d.P.-F., J.-C.L.-P., C.D.-V.-S. and L.R.V.-G.; data curation, D.d.P.-F., J.-C.L.-P., C.D.-V.-S. and L.R.V.-G.; writing—original draft preparation, L.E.T.-G., J.A.D.-P.-F. and F.P.-C.; writing—review and editing, D.d.P.-F., J.-C.L.-P., C.D.-V.-S. and L.R.V.-G.; visualization, D.d.P.-F., J.-C.L.-P., C.D.-V.-S. and L.R.V.-G.; supervision, D.d.P.-F., J.-C.L.-P., C.D.-V.-S. and L.R.V.-G.; project administration, D.d.P.-F., J.-C.L.-P., C.D.-V.-S. and L.R.V.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The code used to run the simulations presented in this study is openly available through the following GitHub repository (accessed on 29 January 2026): https://github.com/Tonix22/PhdDegreeCode/tree/main/MatlabCode/ZeroFocingMobileNet.

Acknowledgments: The authors acknowledge at Universidad Panamericana.

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

References

1. *IEEE P1609.2.1/D3*; IEEE Approved Draft Standard for Wireless Access in Vehicular Environments (WAVE)—Certificate Management Interfaces for End Entities. IEEE: New York, NY, USA, 2026; pp. 1–259.
2. Hartenstein, H.; Laberteaux, L. A tutorial survey on vehicular ad hoc networks. *IEEE Commun. Mag.* **2008**, *46*, 164–171. [CrossRef]
3. Gutiérrez, C.A.; Contreras-Ponce, O.; Ornelas-Lizcano, J.C.; Pätzold, M.; Castillo-Soria, F.R. Polarimetric Modeling of Mobile Fading Channels. *IEEE Trans. Veh. Technol.* **2025**, *74*, 21–37. [CrossRef]
4. Del Puerto-Flores, J.A.; Peña-Campos, F.; Parra-Michel, R.; Del-Valle-Soto, C. Carrier Diversity Incorporation to Low-Complexity Near-ML Detection for Multicarrier Systems over V2V Radio Channel. *Sensors* **2021**, *21*, 6067. [CrossRef] [PubMed]
5. Acosta-Marum, G.; Ingram, M.A. Six time- and frequency- selective empirical channel models for vehicular wireless LANs. *IEEE Veh. Technol. Mag.* **2007**, *2*, 4–11. [CrossRef]
6. Nguyen, T.H.; Nguyen, T.H.; Yoon, T.; Jung, W.S.; Yoo, D.; Ro, S. An ICI Suppression Analysis Testbed for Harbor Unmanned Ground Vehicle Deployment. *IEEE Access* **2019**, *7*, 107757–107768. [CrossRef]
7. Del Puerto-Flores, J.A.; Castillo-Soria, F.R.; Gutiérrez, C.A.; Peña-Campos, F. Efficient Index Modulation-Based MIMO OFDM Data Transmission and Detection for V2V Highly Dispersive Channels. *Mathematics* **2023**, *11*, 2773. [CrossRef]
8. Ko, K.; Lim, S. Generalized Maximum Delay Estimation for Enhanced Channel Estimation in IEEE 802.11p/OFDM Systems. *Electronics* **2025**, *14*, 2404. [CrossRef]
9. Toland, K.; Taiwo, P.; Cole-Rhodes, A. Towards Equalization of Mixed Multi-user OFDM Signals Over a Doubly-Dispersive Channel. In *Proceedings of the 2023 57th Annual Conference on Information Sciences and Systems (CISS)*; IEEE: New York, NY, USA, 2023; pp. 1–5. [CrossRef]
10. Liu, X.; Anand, K.; Guan, Y.L.; Deng, L.; Fan, P.; Zhou, Z. BEM-PSP for Single-Carrier and SC-FDMA Communication Over a Doubly Selective Fading Channel. *IEEE Trans. Wirel. Commun.* **2020**, *19*, 3924–3937. [CrossRef]

https://doi.org/10.3390/a19030210

11. Peña-Campos, F.; Parra-Michel, R.; Kontorovich, V. MIMO Multicarrier Transmission Over Doubly Selective Channels with Virtual Trajectories Receiver. *IEEE Trans. Veh. Technol.* **2019**, *68*, 9330–9338. [CrossRef]
12. Ramadan, K.; Aqeel, I.; Hassan, E.S. Robust OTFS Detection via MMSE-DFE Equalization for ISAC in Doubly Dispersive Channels. *Mathematics* **2025**, *13*, 3545. [CrossRef]
13. Li, M.; Wang, W. Phase Noise Effects on OFDM Chirp Communication Systems: Characteristics and Compensation. *Information* **2024**, *15*, 221. [CrossRef]
14. Kim, K.H. Low-Complexity Suboptimal ML Detection for OFDM-IM Systems. *IEEE Wirel. Commun. Lett.* **2023**, *12*, 416–420. [CrossRef]
15. Vlachos, E.; Lalos, A.S.; Berberidis, K. Low-Complexity OSIC Equalization for OFDM-Based Vehicular Communications. *IEEE Trans. Veh. Technol.* **2017**, *66*, 3765–3776. [CrossRef]
16. Zhang, X.; Xing, L.; Wu, H.; Ji, B.; Zhang, G. Low-Complexity Decision Feedback Equalization for Single-Carrier Massive MIMO Systems. *IEEE Trans. Veh. Technol.* **2024**, *73*, 17316–17330. [CrossRef]
17. Velázquez, R.; Pissaloux, E.; Del-Valle-Soto, C.; Arai, M.; Valdivia, L.J.; Del Puerto-Flores, J.; Gutiérrez, C.A. Performance Evaluation of Active and Passive Haptic Feedback in Shape Perception. In *Proceedings of the 2019 IEEE 39th Central America and Panama Convention (CONCAPAN XXXIX)*; IEEE: New York, NY, USA, 2019; pp. 1–6. [CrossRef]
18. Mohammed, A.F.Y.; Sultan, S.M.; Patni, S. Collaborative Beamforming with DQN for Interference Mitigation in 5G and Beyond Networks. *Telecom* **2024**, *5*, 1192–1204. [CrossRef]
19. Aziz, M.A.; Rahman, M.H.; Tabassum, R.; Sejan, M.A.S.; Baek, M.S.; Song, H.K. A Hybrid Deep Learning Framework for OFDM with Index Modulation Under Uncertain Channel Conditions. *Mathematics* **2024**, *12*, 3583. [CrossRef]
20. Tonix-Gleason, L.E.; Del-Puerto-Flores, J.A.; Castillo-Soria, F.R.; Parra-Michel, R.; Campos, F.P. Neural Network Aided M-PSK Detection in 802.11P V2V OFDM Systems Under ICI Conditions. *IEEE Wirel. Commun. Lett.* **2025**, *14*, 3420–3424. [CrossRef]
21. Howard, A.; Sandler, M.; Chu, G.; Chen, L.C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. Searching for MobileNetV3. *arXiv* **2019**, arXiv:1905.02244. [CrossRef]
22. Gümüş, M.; Duman, T.M. Channel Estimation and Symbol Demodulation for OFDM Systems Over Rapidly Varying Multipath Channels with Hybrid Deep Neural Networks. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 9361–9373. [CrossRef]
23. Gizzini, A.K.; Chafii, M. Deep Learning Based Channel Estimation in High Mobility Communications Using Bi-RNN Networks. In *Proceedings of the ICC 2023—IEEE International Conference on Communications*; IEEE: New York, NY, USA, 2023; pp. 2607–2612. [CrossRef]
24. Almeida, I.; Guerreiro, J.; Dinis, R. On Deep Learning Hybrid Architectures for MIMO-OFDM Channel Estimation. *Electronics* **2025**, *14*, 4692. [CrossRef]
25. Gao, X.; Jin, S.; Wen, C.K.; Li, G.Y. ComNet: Combination of Deep Learning and Expert Knowledge in OFDM Receivers. *IEEE Commun. Lett.* **2018**, *22*, 2627–2630. [CrossRef]
26. MathWorks. WINNER II Channel Documentation. 2026. Available online: https://la.mathworks.com/help/comm/ug/winner-ii-channel.html (accessed on 26 February 2026).
27. *Unapproved Draft Std P802.11p /D11.0*; IEEE Draft Standard for Information Technology—Telecommunications and Information Exchange Between Systems Local and Metropolitan Area Networks Specific Requirements Part 11: Wireless Lan Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment: Wireless Access in Vehicular Environments. IEEE: New York, NY, USA, 2010.

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/a19030210

MDPI AG
Grosspeteranlage 5
4052 Basel
Switzerland
Tel.: +41 61 683 77 34

Algorithms Editorial Office
E-mail: algorithms@mdpi.com
www.mdpi.com/journal/algorithms



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 978-3-7258-9208-2