SA-FGSM: A Simulated Annealing-Enhanced Hybrid White-Box Adversarial Attack Framework

Djawhara Benchaira, Foudil Cherif
LESIA Laboratory, Department of Computer Science, University of Biskra, Biskra 07000, Algeria
E-mail: djawhara.benchaira@univ-biskra.dz, cherif.foudil@univ-biskra.dz

Keywords: Adversarial Attacks, Adversarial Robustness, Deep Learning Security, Simulated Annealing, Metaheuristic Optimization, White-Box Attacks, Robustness Evaluation

Received: August 18, 2025

The reliable evaluation of adversarial defenses is a critical challenge in deep learning security, often hindered by evaluation methods, such as Projected Gradient Descent (PGD), that can fail by getting trapped in local optima. This limitation can lead to a significant overestimation of a model's true robustness. In this work, we introduce the Simulated Annealing-Fast Gradient Sign Method (SA-FGSM), a novel two-phase hybrid attack designed to overcome this specific weakness. SA-FGSM first employs Simulated Annealing to perform a global, stochastic exploration of the perturbation space to find promising attack regions, followed by a gradient-based step for finalization. We conduct a comprehensive evaluation on CIFAR-10 and CIFAR-100 against state-of-the-art adversarially trained models (ResNet-18 and WideResNet-28-10), showing that SA-FGSM achieves a mean attack success rate of 83.6% compared to 51.9% for a suite of strong baselines including FGSM, MI-FGSM, PGD, and APGD. Furthermore, we demonstrate that SA-FGSM finds qualitatively superior perturbations, evidenced by a statistically significant reduction in both average ℓ_2 norm and perceptual distortion as measured by LPIPS (Learned Perceptual Image Patch Similarity), achieving 58.3% lower perceptual distance than gradient-based baselines. Analysis of the proposed attack variants identifies SA-FGSM-Swift as a particularly compelling option, offering state-of-the-art success rates at a fraction of the computational cost of stronger baselines. Our findings suggest that the robustness of even top-tier defenses may be overestimated and highlight the necessity of incorporating global search heuristics into standard evaluation protocols.

Povzetek: Za večjo varnost globokega učenja SA-FGSM združi simulirano ohlajanje in gradientni korak. S pomočjo nasprotniških napadov realneje oceni robustnosti modelov.

1 Introduction

Deep Neural Networks (DNNs) have become the state-of-the-art technology for a wide range of applications, from image classification to autonomous navigation. However, their widespread deployment is critically hindered by a fundamental vulnerability to adversarial examples: meticulously crafted, often imperceptible perturbations to inputs that cause the model to produce an incorrect output [28]. The existence of such examples necessitates the development of robust defense mechanisms and, just as critically, rigorous and reliable methods for their evaluation.

The security and reliability of a given defense are contingent on the strength of the attacks used to evaluate it. Current evaluation benchmarks are dominated by gradient-based white-box attacks, such as the Fast Gradient Sign Method (FGSM) [18], its iterative variant, Projected Gradient Descent (PGD) [24], and adaptive variants like Auto-PGD (APGD) [13]. While computationally efficient, these hill-climbing methods are fundamentally limited by the complex and non-convex nature of DNN loss landscapes, becoming trapped in poor local optima and failing to find adversarial examples that may exist elsewhere in the search space. This creates a false sense of security by systematically overestimating the model's true robustness.

To address this critical limitation, we introduce Simulated Annealing-Fast Gradient Sign Method (SA-FGSM), a novel hybrid adversarial attack that transcends purely gradient-based approaches. SA-FGSM operates through a strategic two-phase process: first, Simulated Annealing performs a global, stochastic search to identify promising perturbation regions while probabilistically escaping local minima through its temperature-guided acceptance mechanism. Second, a targeted gradient-based finalization step leverages the efficiency of gradient information to craft the final adversarial example from this superior starting position. To systematically explore the trade-offs between search intensity and speed, we introduce and analyze four distinct variants of the algorithm—Core, Swift, Thorough, and Adaptive—and identify SA-FGSM-Swift as a particularly compelling method. It offers state-of-the-art attack success rates at a fraction of the computational cost of other high-performing attacks, as quantified by our newly defined Efficiency Score metric.

Our evaluation targets state-of-the-art adversarially robust models from the RobustBench library [13], specif-

ically the `Rebuffi2021Fixing_R18_cutmix_ddpm` and `Rebuffi2021Fixing_28_10_cutmix_ddpm` models [26], ensuring our assessment reflects genuine advances against modern defenses rather than improvements over weak baselines.

Through comprehensive and statistically rigorous evaluation, we demonstrate that SA-FGSM significantly outperforms strong baseline attacks. Our method achieves a mean attack success rate of 83.6%, a 31.7 percentage point improvement over the baseline average—a result that is highly statistically significant ($p < 0.0001$). Beyond higher success rates, SA-FGSM generates qualitatively superior adversarial examples with significantly lower average ℓ_2 norms, indicating more efficient perturbations that find direct paths to decision boundaries.

Our specific contributions are:

- A novel hybrid adversarial attack that combines global metaheuristic search with gradient-based optimization to overcome local optima limitations.
- Comprehensive empirical validation demonstrating substantial improvements over established baselines across multiple robust model architectures.
- Introduction and analysis of four algorithm variants (Core, Swift, Thorough, and Adaptive), with SA-FGSM-Swift identified as offering optimal efficacy-efficiency trade-offs.
- Evidence that current gradient-based evaluation frameworks may systematically underestimate vulnerability, highlighting the need for more comprehensive robustness assessment methodologies.
- Demonstration that SA-FGSM finds qualitatively superior adversarial examples, characterized by statistically significantly lower average ℓ_2 norms and a 58.3% reduction in perceptual distortion (LPIPS) compared to baselines, indicating more efficient perturbations that are less perceptually detectable.

This paper is structured as follows: Section 2 reviews related work in adversarial attacks. Section 3 formally details the proposed SA-FGSM algorithm and its variants. Section 4 describes our rigorous experimental methodology and evaluation metrics. Section 5 presents the empirical results, including perceptual quality and computational analysis. Section 6 discusses practical implications, transferability, limitations, and suggests directions for future research. Finally, Section 7 concludes the paper.

2 Related work

The field of adversarial attacks has experienced rapid evolution, with research primarily focused on the white-box setting where the attacker has full knowledge of the model's architecture and parameters. We situate our work within this context, focusing on methods designed to generate ℓ_∞-norm bounded perturbations. Recent comprehensive surveys have identified fundamental limitations in current gradient-based approaches and highlighted the growing need for alternative optimization strategies [37, 44].

2.1 Gradient-based white-box attacks

The majority of influential white-box attacks leverage the model's gradient to efficiently find adversarial directions. The foundational method in this category is the Fast Gradient Sign Method (FGSM) [18], which performs a single step in the direction of the sign of the loss function's gradient with respect to the input image. Its simplicity and speed are compelling, but its effectiveness is limited, as it often finds suboptimal solutions. To address this, iterative methods were introduced. Projected Gradient Descent (PGD) is arguably the most important and widely recognized attack, representing a strong and universal first-order adversary [24]. PGD takes multiple smaller steps in the gradient sign direction, projecting the resulting perturbation back onto the allowed ℓ_∞ ball after each step. Due to its strength, robustness against PGD has become the de facto standard for evaluating adversarial defenses. However, as a hill-climbing algorithm, PGD remains a local search method and is still susceptible to getting trapped in local optima, particularly in the complex loss landscapes of robustly trained models.

Recent work has explicitly identified fundamental issues with gradient-based optimization in adversarial settings. Liu et al. [22] demonstrate that gradient calculations suffer from "wrong blocking" and "over transmission" problems in ReLU networks, leading to misleading directional guidance. Wang et al. [32] propose replacing traditional Jacobian gradients with integrated gradients to address these directional guidance issues, showing improvements over standard gradient methods. Several enhancements to PGD have been proposed to overcome local optima limitations. The Momentum Iterative FGSM (MI-FGSM) incorporates a momentum term into the iterative process [14]. By accumulating a velocity vector in promising directions, momentum helps the attack escape shallow local minima and find more stable, transferable adversarial examples. More recently, Wang et al. [31] introduced Global Momentum Initialization to address the "gradient elimination and local momentum optimum dilemma" through global search before attack initialization, improving attack success rates by 6.4% across various defense mechanisms.

The Auto Conjugate Gradient (ACG) attack [39] represents a significant advancement in addressing ill-conditioned problems in adversarial optimization. By using conjugate gradient methods instead of steepest descent, ACG demonstrates that diversified search strategies can overcome local optima where traditional gradient methods fail. Recent extensions like ReACG [38] further improve upon this by automatically modifying search direction and step size control, achieving 0.4-0.9% improvement over APGD through enhanced output diversity. Our work shares

the goal of escaping local optima but employs a fundamentally different, gradient-free global search mechanism that addresses the root cause of gradient method limitations.

2.2 Optimization-based attacks

A second class of attacks formulates the search for an adversarial example as a formal optimization problem. The Carlini & Wagner (C&W) attack is the most prominent example [6]. The C&W attack aims to find the minimum possible perturbation (typically measured in the ℓ_2 norm) that induces misclassification, using a specialized objective function and a change-of-variables technique to handle the box constraints on the input. While extremely effective at finding low-norm perturbations and achieving high success rates, C&W attacks are computationally expensive, often requiring thousands of iterations, which makes them less practical for evaluating large models or datasets.

Recent advances in optimization-based attacks have explored multi-objective formulations. Williams and Li [34] propose SA-MOO, which treats the loss function and ℓ_2 norm as separate objectives in a bi-objective optimization framework using evolutionary computation principles. Similarly, Bui et al. [5] address the challenge of generating qualified and divergent adversarial examples by formulating adversarial generation as a multi-objective optimization problem with adaptive objective weighting. Advanced global optimization techniques have also emerged. Cheng et al. [10] leverage surrogate models as global function priors rather than relying on local gradients, providing theoretical regret bound analysis that demonstrates improvement when incorporating global information about the loss landscape. Bayesian optimization approaches like BayAtk [16] introduce pixel-level and region-based removal priors with adaptive dynamic weighting strategies for enhanced transferability. SA-FGSM, in contrast, is designed to significantly improve upon the effectiveness of PGD without incurring the prohibitive computational cost of C&W while maintaining the global search capabilities demonstrated by these advanced optimization approaches.

2.3 Stochastic and heuristic search methods

Recognizing the limitations of local search, extensive research has explored stochastic and heuristic optimization techniques, demonstrating their superiority over gradient-based methods across multiple domains. These approaches have been applied in both black-box and white-box settings, consistently showing improved performance in escaping local optima.

Evolutionary and genetic algorithms Evolutionary algorithms have established themselves as powerful alternatives to gradient-based methods. The landmark GenAttack [2] demonstrated that genetic algorithms require 2,126× fewer queries than gradient-based ZOO [9] on MNIST/CIFAR-10, establishing evolutionary approaches as fundamentally more query-efficient. Recent advances include EvolBA [30], which uses Covariance Matrix Adaptation Evolution Strategy (CMA-ES) with fractal-based initialization, demonstrating superior performance over gradient-based methods (HSJA [8], Boundary Attack [4]) by avoiding local optima through probabilistic acceptance mechanisms. Multi-stage evolutionary approaches have directly addressed the local optima problem. The Genetic Algorithm with Multiple Fitness Functions [35] divides evolution into exploration, exploitation, and stable stages with different fitness functions for each stage, explicitly overcoming the limitations of single-objective gradient descent.

Particle swarm optimization Particle Swarm Optimization (PSO) has shown remarkable success in adversarial settings. AdversarialPSO [25] achieves high success rates on CIFAR-10, MNIST, and ImageNet respectively, while requiring fewer queries than state-of-the-art gradient methods. MGRR-PSO [27] explicitly addresses local optima through multi-group random redistribution, achieving 100% success rate on MNIST/CIFAR-10, demonstrating that distributed population-based approaches can overcome gradient method limitations. Recent developments include MISPSO-Attack [45], which uses multiple initial solution strategies to avoid local optima, achieving 89.50% attack success rate on ImageNet through multi-swarm optimization.

Simulated annealing applications Direct applications of Simulated Annealing in adversarial contexts have validated its effectiveness. BESA: BERT-based Simulated Annealing [40] demonstrates significant improvements in attack success rate while maintaining low word substitution rates in the text domain. Neural Simulated Annealing [12] presents an advanced SA framework with learnable components, viewing SA from a reinforcement learning perspective for enhanced solution quality. These approaches demonstrate the power of global, gradient-free search strategies. However, many traditional metaheuristic methods in adversarial settings have been applied primarily in black-box scenarios or suffer from query inefficiency.

Hybrid and advanced stochastic methods Recent work has explored sophisticated hybrid approaches that combine multiple optimization paradigms. The Multiple Asymptotically Normal Distribution Attacks (MultiANDA) [17] leverage asymptotic normality of stochastic gradient ascent for learning perturbation distributions, achieving superior transferability through ensemble of Gaussian distributions. Monte-Carlo Adversarial Attack (MC-AA) [4] performs direct input perturbation using FGSM with multiple forward passes, capturing overlapping class regions better than deterministic methods. Bayesian approaches have also gained prominence. Prior-guided Bayesian Optimization [10] leverages global function priors and provides theoretical regret bound analysis, while BayAtk [16] introduces so-

phisticated prior mechanisms for enhanced adversarial example generation.

2.4 Positioning of SA-FGSM

Our work brings the global search paradigm of metaheuristics into the white-box setting through a novel hybrid approach. By using Simulated Annealing, we leverage a well-established global optimization technique known for its ability to provably converge to a global optimum under a sufficiently slow cooling schedule [20]. Unlike purely heuristic methods that often suffer from query inefficiency, we do not discard gradient information entirely. Instead, we form a novel hybrid: using SA for its core strength in global exploration and gradient information for a final, efficient finalization step. This approach directly addresses the local optima problem that has been consistently identified across recent surveys [37, 44] while maintaining computational efficiency. To our knowledge, this sequential combination of a global metaheuristic search with a gradient-based refinement step represents a new approach for white-box adversarial attacks that bridges the gap between the global exploration capabilities of metaheuristics and the computational efficiency of gradient-based methods. The extensive body of work demonstrating metaheuristic superiority over gradient-based methods [2, 25, 39] provides strong theoretical and empirical justification for our approach, while the hybrid design addresses the computational efficiency concerns that have limited the practical adoption of purely metaheuristic methods in white-box settings.

3 The proposed SA-FGSM attack

To overcome the limitations of local search, we propose the Simulated Annealing-Fast Gradient Sign Method (SA-FGSM), a hybrid white-box attack that synergistically combines a global, gradient-free search with a final, efficient gradient-based step.

3.1 Threat model and problem formulation

We operate under the standard white-box threat model, assuming full knowledge of the target model f_θ. Given a benign input image $\mathbf{x} \in [0,1]^d$ and its true label y, the attacker's objective is to find a perturbation $\mathbf{p}$ that solves the following constrained optimization problem:

$$\underset{\mathbf{p}}{\text{maximize}} \quad E(f_\theta(\mathbf{x}+\mathbf{p}), y) \tag{1}$$

subject to the constraints $\|\mathbf{p}\|_\infty \leq \epsilon$ and $\mathbf{x}+\mathbf{p} \in [0,1]^d$, where ϵ is the maximum perturbation magnitude and E is an energy function designed to be maximal when the model misclassifies the input.

3.2 Algorithm overview

The SA-FGSM attack is structured as a two-phase process, as detailed in Algorithm 1. The first phase leverages Simulated Annealing (SA) to perform a global search for the perturbation $\mathbf{p}$. The second phase uses the result of the SA search, $\mathbf{p}_{\text{SA}}$, as a starting point for a final, gradient-based finalization step to craft the adversarial example $\mathbf{x}_{\text{adv}}$.

3.3 Phase 1: simulated annealing exploration

Simulated Annealing is a metaheuristic capable of approximating the global optimum of a given function. The state of the system is defined by the perturbation $\mathbf{p}$, and the algorithm seeks to maximize the energy function E.

Energy function. The energy function E quantifies the quality of a given perturbed input $\mathbf{x}' = \mathbf{x}+\mathbf{p}$. We aim to maximize this energy. The implementation supports two primary energy functions:

- **Negative Cross-Entropy Loss:**

$$E(\mathbf{x}') = -\mathcal{L}_{CE}(f_\theta(\mathbf{x}'), y)$$

  where $\mathcal{L}_{CE}$ is the standard cross-entropy loss.

- **Margin Loss:**

$$E(\mathbf{x}') = -(Z(\mathbf{x}')_y - \max_{j \neq y} Z(\mathbf{x}')_j)$$

  where $Z(\mathbf{x}')_j$ is the logit for class j.

Note: For all experiments in this paper, we use the negative cross-entropy loss, the standard objective for adversarial attacks [24]. This choice was based on preliminary validation indicating superior performance for our method compared to the margin-based alternative, another widely used objective for crafting strong adversarial examples [6].

Table 1: Qualitative comparison of major adversarial attack methodologies. The table highlights the trade-off between the efficiency of gradient-based methods, which are susceptible to local optima, and the global search capabilities of metaheuristics, which are often computationally expensive or designed for black-box settings. SA-FGSM is positioned as a hybrid solution bridging this gap.

Attack Method	**Core Mechanism**	**Key Strength(s)**	**Key Limitation(s)**
FGSM [18]	Gradient-Based (Single-Step)	Extremely fast; simple implementation.	Low success rate; often finds suboptimal solutions.
PGD [24]	Gradient-Based (Iterative)	Strong baseline; widely adopted standard.	Prone to getting trapped in local optima, especially in robust models.
MI-FGSM [14]	Gradient-Based (Momentum)	Escapes shallow local minima; improves transferability.	Fundamentally a local search method; can still be trapped.
APGD [13]	Gradient-Based (Adaptive)	State-of-the-art for gradient attacks; reliable convergence.	Still constrained by local gradient information; high computational cost.
C&W [6]	Optimization-Based	Finds low-norm perturbations with high success.	Extremely high computational cost; impractical for large-scale evaluation.
GenAttack [2]	Metaheuristic (Evolutionary)	Gradient-free; effective at escaping local optima.	Designed for black-box settings; high query count.
SA-FGSM (Ours)	**Hybrid (Metaheuristic + Gradient)**	**Global exploration escapes deep local optima; high success rate.**	**More hyperparameters than PGD; higher cost than single-step methods.**

Candidate generation distribution. We use Gaussian noise for candidate generation based on theoretical foundations [20, 7]: (1) symmetry/unbiasedness, (2) natural isotropy in high dimensions [3], (3) temperature-adaptive scaling enabling smooth exploration-exploitation transition, (4) standard practice in SA literature [11, 41], and (5) computational efficiency. Alternative distributions (Laplace, Cauchy [29], uniform) could offer advantages—heavier tails might aid local optima escape but reduce stability. We acknowledge lacking systematic empirical comparison; this represents important future work. However, strong results (83.64% ASR) validate Gaussian as at least reasonable, with fundamental SA advantages (probabilistic acceptance) being distribution-independent.

The SA Process. The SA search begins with an initial temperature T_0 and a perturbation $\mathbf{p}_{\text{curr}}$, typically initialized to zero. For a fixed number of iterations N_{SA}, the following steps are repeated:

1. **Candidate Generation:** At iteration k, a new candidate perturbation $\mathbf{p}_{\text{cand}}$ is generated by sampling from a Gaussian distribution centered at the current perturbation, $\mathbf{p}_{\text{curr}}$. The standard deviation of the noise is scaled by the current temperature T_k: $\mathbf{p}' = \mathbf{p}_{\text{curr}} + \mathcal{N}(0, (\sigma_{\text{base}} \cdot T_k/T_0)^2)$, where σ_{base} is a fixed hyperparameter (the base neighborhood standard deviation, specified in Table 2) that controls the scale of exploration, and the ratio T_k/T_0 provides temperature-adaptive scaling that decreases the neighborhood size as the search cools. The resulting perturbation is immediately clipped to satisfy the ℓ_∞ constraint: $\mathbf{p}_{\text{cand}} \leftarrow \text{clip}(\mathbf{p}', -\epsilon, \epsilon)$.

2. **Acceptance Criterion:** The energies of the current state, $E_{\text{curr}} = E(\mathbf{x} + \mathbf{p}_{\text{curr}})$, and the candidate state, $E_{\text{cand}} = E(\mathbf{x}+\mathbf{p}_{\text{cand}})$, are computed. The candidate becomes the new current state according to the Metropolis acceptance probability, $P(\mathbf{p}_{\text{curr}} \to \mathbf{p}_{\text{cand}})$:

$$P(\mathbf{p}_{\text{curr}} \to \mathbf{p}_{\text{cand}}) = \begin{cases} 1 & \text{if } E_{\text{cand}} > E_{\text{curr}} \\ \exp\left(\frac{E_{\text{cand}} - E_{\text{curr}}}{T_k}\right) & \text{if } E_{\text{cand}} \leq E_{\text{curr}} \end{cases} \tag{2}$$

   This probabilistic acceptance of inferior solutions is the key mechanism that allows SA to escape local optima. Throughout this process, the algorithm independently tracks the best perturbation found so far, $\mathbf{p}_{\text{best}}$.

3. **Cooling Schedule:** After each iteration, the temperature is reduced according to a geometric cooling schedule with rate α: $T_{k+1} = \alpha \cdot T_k$.

The search concludes by returning the overall best solution found, $\mathbf{p}_{\text{SA}} \leftarrow \mathbf{p}_{\text{best}}$.

3.4 Phase 2: gradient-based finalization

The perturbation $\mathbf{p}_{\text{SA}}$ represents a promising region in the loss landscape found via global search. To exploit this position, we apply a single, decisive gradient-based step. Starting from $\mathbf{x}_{\text{SA}} = \mathbf{x} + \mathbf{p}_{\text{SA}}$, we compute the sign of the gradient of the loss function. This gradient is used to take a full-strength step of size ϵ: $\mathbf{p}_{\text{final}} \leftarrow \mathbf{p}_{\text{SA}} + \epsilon \cdot \text{sign}(\nabla_{\mathbf{x}} \mathcal{L}(f_\theta(\mathbf{x}_{\text{SA}}), y))$. The resulting perturbation is clipped to satisfy the ℓ_∞ norm, $\mathbf{p}_{\text{final}} \leftarrow \text{clip}(\mathbf{p}_{\text{final}}, -\epsilon, \epsilon)$, and the final adversarial example is generated by clipping to the valid image range: $\mathbf{x}_{\text{adv}} \leftarrow \text{clip}(\mathbf{x} + \mathbf{p}_{\text{final}}, 0, 1)$.

3.5 Algorithm variants

The efficacy and computational cost of the SA-FGSM attack are fundamentally governed by the parameters of the Simulated Annealing search. To systematically investigate the trade-offs between search breadth, depth, and speed, we designed and analyzed four distinct variants of the algorithm. Each variant represents a specific strategy for navigating the adversarial loss landscape.

Algorithm 1: The Proposed SA-FGSM Attack. Hyperparameter values (including σ_{base}) are specified in Table 2.

Input Requirements:

- Model f_θ, loss function $\mathcal{L}$, input image $\mathbf{x}$, true label y
- Perturbation budget ϵ
- SA iterations N_{SA}, initial temperature T_0, cooling rate α
- Neighborhood std. deviation σ_{base} (base value for adaptive scaling)

Procedure `SA-FGSM(`$\mathbf{x}, y$`):`

```
// Phase 1: Simulated Annealing Exploration
```

$\mathbf{p}_{\text{curr}} \leftarrow \mathbf{0}$
$\mathbf{p}_{\text{best}} \leftarrow \mathbf{p}_{\text{curr}}$
$E_{\text{best}} \leftarrow -\mathcal{L}(f_\theta(\mathbf{x} + \mathbf{p}_{\text{best}}), y)$
$T \leftarrow T_0$
for $i \leftarrow 1$ **to** N_{SA} **do**

```
// Generate candidate using adaptive neighborhood
```

$\sigma = \sigma_{\texttt{base}} \cdot (T/T_0)$
$\mathbf{p}_{\text{cand}} \leftarrow$ `GenerateCand(`$\mathbf{p}_{\text{curr}}, T, \epsilon$`)`
$E_{\text{cand}} \leftarrow -\mathcal{L}(f_\theta(\mathbf{x} + \mathbf{p}_{\text{cand}}), y)$
$E_{\text{curr}} \leftarrow -\mathcal{L}(f_\theta(\mathbf{x} + \mathbf{p}_{\text{curr}}), y)$

```
// Metropolis acceptance criterion
```

if $E_{\text{cand}} > E_{\text{curr}}$ **then**
  $\mathbf{p}_{\text{curr}} \leftarrow \mathbf{p}_{\text{cand}}$ `// Accept improvement`
else if $\exp\big((E_{\text{cand}} - E_{\text{curr}})/T\big) > \text{rand}(0, 1)$ **then**
  $\mathbf{p}_{\text{curr}} \leftarrow \mathbf{p}_{\text{cand}}$ `// Probabilistically accept`

```
// Track best solution found
```

if $E_{\text{cand}} > E_{\text{best}}$ **then**
  $\mathbf{p}_{\text{best}} \leftarrow \mathbf{p}_{\text{cand}}$
  $E_{\text{best}} \leftarrow E_{\text{cand}}$
$T \leftarrow \alpha \cdot T$ `// Geometric cooling`

$\mathbf{p}_{\text{SA}} \leftarrow \mathbf{p}_{\text{best}}$

```
// Phase 2: Gradient-Based Finalization
```

$\mathbf{x}_{\text{SA}} \leftarrow \mathbf{x} + \mathbf{p}_{\text{SA}}$
$\mathbf{g} \leftarrow \nabla_{\mathbf{x}} \mathcal{L}(f_\theta(\mathbf{x}_{\text{SA}}), y)$
$\mathbf{p}_{\text{final}} \leftarrow \mathbf{p}_{\text{SA}} + \epsilon \cdot \text{sign}(\mathbf{g})$
$\mathbf{p}_{\text{final}} \leftarrow \text{clip}(\mathbf{p}_{\text{final}}, -\epsilon, \epsilon)$
$\mathbf{x}_{\text{adv}} \leftarrow \text{clip}(\mathbf{x} + \mathbf{p}_{\text{final}}, 0, 1)$
return $\mathbf{x}_{\text{adv}}$

Algorithm 2: Candidate Generation Procedure (used in Algorithm 1)

Inputs:

- Current perturbation $\mathbf{p}_{\text{curr}}$
- Current temperature T
- Perturbation budget ϵ
- Initial temperature T_0 (constant)
- Base neighborhood std. σ_{base} (hyperparameter from Table 2)

Output: Candidate perturbation $\mathbf{p}_{\text{cand}}$

Procedure `GenerateCand(`$\mathbf{p}_{\text{curr}}, T, \epsilon, T_0, \sigma_{\text{base}}$`):`

```
// Adjust neighborhood size based on temperature (adaptive)
```

$\sigma \leftarrow \sigma_{\text{base}} \cdot \dfrac{T}{T_0}$ `// σ is NOT a constant`

```
// Generate Gaussian noise with adaptive standard deviation
```

$\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$

```
// Create new candidate by adding noise
```

$\mathbf{p}_{\text{cand}} \leftarrow \mathbf{p}_{\text{curr}} + \mathbf{z}$

```
// Project onto the ℓ∞ ball
```

$\mathbf{p}_{\text{cand}} \leftarrow \text{clip}(\mathbf{p}_{\text{cand}}, -\epsilon, \epsilon)$
return $\mathbf{p}_{\text{cand}}$

SA-FGSM-core. This variant serves as a baseline, configured with balanced parameters derived from our hyperparameter optimization process. It is intended to provide a robust and generally effective performance without being specialized for either extreme speed or exhaustive search quality.

SA-FGSM-swift. Designed for computational efficiency, this variant significantly reduces the computational cost. It employs a lower number of SA iterations (N_{SA}), which reduces the total runtime. Furthermore, it uses a lower initial temperature (T_0) and a faster cooling rate (α). This combination results in a search that is less exploratory from the outset and "quenches" more quickly, rapidly converging towards a promising local optimum. The hypothesis is that for many loss landscapes, a fast, decisive search is sufficient to find a solution superior to that of standard PGD.

SA-FGSM-thorough. In contrast to the Swift variant, this version is configured for maximum search quality, prioritizing attack efficacy over speed. It uses a significantly higher number of SA iterations to allow for a much longer exploration of the search space. Its initial temperature is also higher, which increases the probability of accepting inferior solutions early on, thereby broadening the initial search and making it more likely to escape large, deceptive local optima. Combined with a very slow cooling rate (an α closer to 1.0), this allows the algorithm to meticulously explore promising regions of the loss landscape before converging.

Algorithm 3: Adaptive Parameter Adjustment (SA-FGSM-Adaptive). Thresholds justified in Section 3.5 text.

```
// Adaptive Cooling Rate Adjustment
Input: Current α, acceptance rate over last 10 iterations
if acceptance_rate > 0.80 then
    α ← max(α − 0.05, 0.85)              // Too exploratory; cool faster
else if acceptance_rate < 0.20 then
    α ← min(α + 0.05, 0.99)   // Frozen; cool slower
else
    α unchanged       // Healthy SA behavior
// Adaptive Neighborhood Size Adjustment
Input: Current σ, energy improvement ΔE over last 10 iterations, patience counter
if ΔE < 10^−3 and patience ≥ 10 then
    σ ← min(σ × 1.2, 0.30)       // Stagnation; expand
    patience ← 0
else if ΔE ≥ 10^−3 then
    σ ← max(σ × 0.8, 0.05)        // Progress; contract
    patience ← 0
else
    σ unchanged; patience ← patience + 1 // Wait
```

Parameters: Acceptance thresholds (80%, 20%), improvement threshold (10^{-3}), patience limit (10), adjustment factors (0.05 for α, 1.2/0.8 for σ) determined through preliminary experiments and SA theory. See Section 3.5 for detailed justification.

SA-FGSM-adaptive. This variant dynamically adjusts parameters based on search behavior (Algorithm 3): **Adaptive Cooling** Adjusts α based on acceptance rate: if $> 80\%$ (too exploratory), cool faster; if $< 20\%$ (frozen), cool slower [20]. **Adaptive Neighborhood** Adjusts σ based on progress: if stagnation (improvement $< 10^{-3}$ for 10 iterations), expand; if progressing, contract. **Threshold Justification:** Values grounded in SA theory [7] and preliminary experiments during hyperparameter optimization (200 Optuna trials). Thresholds represent boundaries of effective SA behavior observed empirically. We acknowledge lacking systematic ablation study varying thresholds, identifying this as important future work. However, Adaptive performance is statistically indistinguishable from Core ($p = 0.513$), suggesting thresholds are reasonable if not necessarily optimal.

The specific hyperparameter configurations for each variant, obtained through rigorous Bayesian optimization, are presented in Table 2. These parameters were optimized using Optuna with 200 trials per variant on a validation setup (CIFAR-10, ResNet-18, 500 samples). The multi-objective function balanced attack success rate (70% weight) and computational speed (30% weight), with speed normalized using a 100ms threshold based on preliminary experiments. These fixed parameter values were used consistently across all experiments reported in Section 5.

Table 2: Optimized hyperparameters obtained via Bayesian optimization (200 Optuna trials on CIFAR-10/ResNet-18). Adaptive thresholds (80%/20%, 10^{-3}, patience=10) determined through preliminary experiments guided by SA theory; see Section 3.5 for justification.

Parameter	Core	Swift	Thorough	Adaptive
N_{SA}	50	20	81	50
T_0	1.0	0.5	1.6	1.0
α	0.95	0.90	0.95	0.95
σ_{base}	0.10	0.05	0.20	0.10
Energy Func.	Neg. Cross-Entropy			
Random Start	False			
Adaptive-Specific Parameters				
Improv. Thresh.	—	—	—	10^{-3}
Patience	—	—	—	10

Optimization objective: $0.7 \times (\text{ASR}/100) + 0.3 \times \text{normalized_speed}$.

4 Experimental design

To rigorously evaluate the performance of SA-FGSM, we designed a comprehensive experimental framework grounded in principles of reproducibility, statistical validity, and challenging evaluation. Our methodology is constructed not merely to show that our attack works, but to demonstrate its superiority under conditions that reflect the current state-of-the-art in adversarial defense. All experiments were conducted on an NVIDIA A40 GPU using PyTorch.

4.1 Setup

Datasets. We conduct our evaluation on two standard benchmark datasets for adversarial robustness: CIFAR-10 and CIFAR-100 [21]. These datasets are sufficiently complex to present a meaningful challenge and are the standard choice for the majority of literature in this field, ensuring our results are comparable to prior and future work.

Threat model. All attacks are performed under the ℓ_∞ threat model, with a maximum perturbation budget of $\epsilon = 8/255$. This is the most common and widely studied setting for CIFAR-10/100 evaluation, where the perturbation on any single pixel is imperceptible, forcing the attack to find a subtle yet effective solution in a high-dimensional space. This identical constraint was applied uniformly to all baseline methods and SA-FGSM variants, ensuring fair comparison. All attacks enforce the box constraint $\mathbf{x}_{\text{adv}} \in [0, 1]^d$ on the final adversarial example.

4.2 Target defenses

To provide a rigorous and meaningful assessment of adversarial attack efficacy, we evaluate SA-FGSM against state-of-the-art adversarially trained models with demonstrated robustness. Our evaluation targets two architecturally distinct models from the RobustBench library [13], both developed by Rebuffi et al. [26]: the **ResNet-18** model (`Rebuffi2021Fixing_R18_cutmix_ddpm`) and the **WideResNet-28-10** model (`Rebuffi2021Fixing_28_10_cutmix_ddpm`).

These models represent the current frontier in adversarial defense, incorporating sophisticated training protocols that combine robust optimization with advanced data augmentation schemes including CutMix and diffusion-based augmentation (DDPM). Their superior empirical performance against established gradient-based attacks, as validated through the RobustBench leaderboard, makes them ideal benchmarks for evaluating novel attack methodologies. The architectural diversity between ResNet-18 and WideResNet-28-10 ensures our evaluation captures robustness patterns across different network designs, thereby establishing the generalizability and practical significance of SA-FGSM's improvements.

4.3 Baseline methods

We compare the performance of our four proposed SA-FGSM variants against a carefully selected suite of baseline attacks. These baselines were chosen to represent a range of strategies and complexities, providing a comprehensive benchmark. For fairness and reproducibility, we utilize the well-established implementations of these attacks provided by the torchattacks library [19]. Our chosen baselines are:

- **FGSM:** The single-step Fast Gradient Sign Method [18]. Included as a low-cost, foundational baseline.
- **MI-FGSM:** The Momentum Iterative FGSM [14]. This attack is specifically designed to escape shallow local optima through momentum, making it a direct competitor to our goal of improved exploration.
- **PGD-10:** Projected Gradient Descent with 10 iterations [24]. This is the most widely accepted benchmark for evaluating adversarial defenses; outperforming PGD-10 is a minimum requirement for any new attack claiming superiority.
- **PGD-100:** A stronger PGD variant with 100 iterations. This serves as a high-cost, high-performance baseline, allowing us to evaluate whether SA-FGSM provides benefits beyond simply running a standard attack for more iterations.
- **APGD:** Auto-PGD [13], the primary component of the AutoAttack evaluation suite and the current gold standard for robustness assessment. APGD features parameter-free adaptive step-size scheduling, momentum-based optimization, and multiple loss variants, making it the strongest gradient-based baseline available. Its inclusion validates that SA-FGSM's advantages extend beyond improvements over standard PGD variants.

Baseline configuration. All attacks used identical conditions: $\epsilon = 8/255$ (ℓ_∞), dataset-specific normalization (CIFAR-10: mean=[0.4914,0.4822,0.4465], std=[0.2023,0.1994,0.2010]; CIFAR-100: mean=[0.5071,0.4867,0.4408], std=[0.2675,0.2565,0.2761]). PGD-10/100: zero initialization, $\alpha = 2/255$ (PGD-10), $\alpha = 1/255$ (PGD-100) [24]. MI-FGSM: 10 iterations, $\alpha = 2/255$, $\mu = 1.0$ [14]. APGD: 100 iterations with adaptive step-size [13]. All implementations from torchattacks library v3.4.0.

4.4 Evaluation metrics and derived measures

We evaluate attack performance using both primary metrics that measure individual dimensions of performance and derived metrics that quantify multi-objective trade-offs.

4.4.1 Primary metrics

Attack success rate (ASR). The percentage of test samples for which the attack successfully induces misclassification: $\text{ASR} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}[f_\theta(\mathbf{x}_i^{\text{adv}}) \neq y_i] \times 100\%$, where N is the total number of samples, $\mathbf{x}_i^{\text{adv}}$ is the adversarial example, y_i is the true label, and $\mathbb{1}[\cdot]$ is the indicator function. This is our primary measure of attack *efficacy*.

Average time per sample (ms). The mean wall-clock time required to generate one adversarial example, measured in milliseconds. This is our primary measure of *computational cost*.

Average ℓ_2 norm. The mean Euclidean distance of perturbations: $\frac{1}{N}\sum_{i=1}^{N} \|\mathbf{x}_i^{\text{adv}} - \mathbf{x}_i\|_2$. While attacks are constrained by ℓ_∞ norm, lower ℓ_2 norm indicates more subtle and efficient perturbations, serving as a measure of perturbation *quality*.

Average perceptual distance (LPIPS). The mean Learned Perceptual Image Patch Similarity distance [43], which measures perceptual dissimilarity using deep features calibrated to human judgments. Lower values indicate less perceptually noticeable perturbations.

4.4.2 Derived multi-objective metrics

Efficiency score. To facilitate comparison across methods with different efficacy-efficiency trade-offs, we define

the Efficiency Score as:

$$\text{Efficiency Score} = \frac{\text{ASR}}{100} \times \frac{1}{\text{Time per Sample (seconds)}} \tag{3}$$

This metric captures the rate at which an attack generates successful adversarial examples (successful attacks per second). Higher values indicate better overall efficiency, combining both high success rates and low computational cost. The Efficiency Score is particularly useful for identifying Pareto-optimal methods—those that are not strictly dominated by any other method in the ASR-time trade-off space. We use this metric primarily in aggregate comparisons (Table 3) and in variant analysis (Table 4), but note that it should not be the sole criterion for method selection, as different applications may prioritize effectiveness versus efficiency differently.

Constraint violation rate. The percentage of generated adversarial examples that exceed the specified ℓ_∞ budget or fall outside the valid input range $[0, 1]^d$. All attacks in our evaluation maintain zero constraint violation rate, confirming proper implementation.

4.5 Rigorous hyperparameter optimization

The performance of metaheuristic algorithms is sensitive to hyperparameter choices. To eliminate potential experimental bias from manual tuning and to ensure fairness and reproducibility, we adopted a principled optimization strategy. The hyperparameters for all four SA-FGSM variants were determined via a one-time, comprehensive optimization process using the Optuna framework [1] for Bayesian optimization with a Tree-structured Parzen Estimator (TPE) sampler.

Optimization setup. The hyperparameter optimization was conducted on a carefully controlled validation setup to ensure consistent and reliable parameter selection across all variants: **Dataset:** A 1,000-sample subset of the CIFAR-10 test set. **Model:** The robust ResNet-18 from RobustBench (`Rebuffi2021Fixing_R18_ddpm`). **Trials per variant:** 200 independent optimization trials to thoroughly explore the search space. **Random seed:** A base seed of 42 was used to ensure deterministic trial execution for reproducibility. To align the search with the goal of each variant, we used tailored search spaces. For instance, the optimization for SA-FGSM-Swift explored lower iteration counts ($N_{\text{SA}} \in [10, 40]$) and faster cooling rates ($\alpha \in [0.80, 0.95]$), while the search for SA-FGSM-Thorough explored a higher range of iterations ($N_{\text{SA}} \in [80, 200]$) and slower cooling rates ($\alpha \in [0.95, 0.995]$). This guided approach ensured that the final parameters were not only optimal but also consistent with each variant's intended design.

Multi-objective function. For each trial, Optuna searched the parameter space to maximize the following multi-objective function, which balances attack efficacy with computational practicality: Objective $= 0.7 \times \left(\frac{\text{ASR}}{100}\right) + 0.3 \times$ normalized_speed, where ASR is the Attack Success Rate (in percent) and 'normalized_speed' is a score rewarding faster execution, computed as: normalized_speed $= \max\left(0, 1 - \frac{T_{\text{sample}}}{T_{\text{threshold}}}\right)$. We set the time threshold $T_{\text{threshold}} = 100$ ms, based on preliminary experiments suggesting that attack times beyond this are impractical for most evaluation scenarios. The weights (0.7 for efficacy, 0.3 for speed) reflect our primary goal of finding highly effective attacks that remain computationally feasible.

Fixed parameters for all experiments. The optimal hyperparameter sets resulting from this one-time optimization process, detailed in Table 2, were then **fixed and used for all experiments reported in this paper**. This rigorous, automated protocol prevents any possibility of "cherry-picking" parameters for specific models or datasets and ensures that our results represent a fair and reproducible assessment of each variant's capabilities.

4.6 Evaluation and statistical analysis

As detailed in Section 4.4, we evaluate attacks using multiple metrics that capture different dimensions of performance. Our statistical protocol ensures the reliability of comparative conclusions.

Statistical protocol. To ensure the reliability of our conclusions, every experimental configuration was executed three times with different random seeds. The results presented are the mean and standard deviation of these runs. Crucially, all comparative claims are validated with formal statistical tests. As our initial analysis revealed that the data did not always meet the assumptions of normality required for parametric tests, we employed a robust statistical methodology: using independent t-tests where assumptions held, and non-parametric alternatives (e.g., Kruskal-Wallis) where they did not. All claims of significance are based on a threshold of $p < 0.05$.

5 Results

We present a comprehensive analysis of our experimental results, drawing from an evaluation across two datasets (CIFAR-10, CIFAR-100) and two state-of-the-art robust model architectures (ResNet-18, WideResNet-28-10). The findings, validated through formal statistical testing, clearly demonstrate gradient-based baselines. the aggregated performance, averaged across all experimental configurations, is summarized in Table 3.

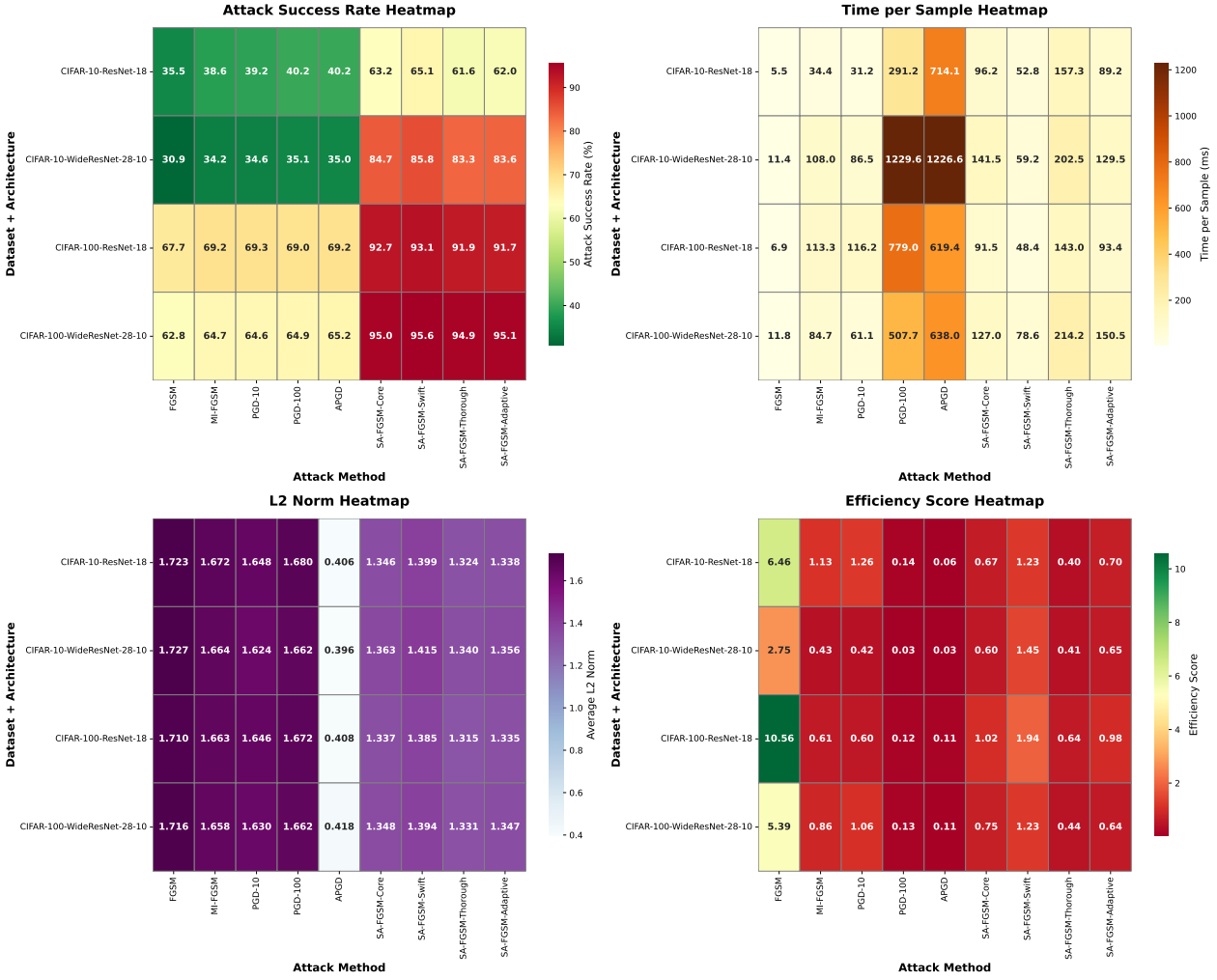


Figure 1: Comprehensive performance overview of all attack methods across all experimental configurations. Each cell represents the mean value of a metric for an attack (columns) on a given dataset and architecture (rows). **(Top-Left)** The Attack Success Rate heatmap shows the superior efficacy of SA-FGSM variants (red hues). **(Top-Right)** The Time per Sample heatmap reveals the computational costs, highlighting the speed of SA-FGSM-Swift. **(Bottom-Left)** The Average ℓ_2 Norm heatmap shows SA-FGSM finds perturbations of lower magnitude (purple hues). **(Bottom-Right)** The Efficiency Score heatmap confirms the superior trade-off offered by SA-FGSM-Swift (red hues).

Table 3: Aggregated performance comparison of Baseline attacks versus the SA-FGSM family, averaged across all four experimental configurations (2 datasets × 2 architectures). SA-FGSM demonstrates statistically significant improvements in efficacy (ASR) and perturbation quality (L2, LPIPS) at a substantially lower average computational cost.

Attack Family	ASR (%)	Time (ms)	L2 Norm	LPIPS
Baselines (Avg.)	51.9 ± 16.5	390.4 ± 483.7	1.67 ± 0.05	0.0073 ± 0.0006
SA-FGSM (Avg.)	**83.6 ± 14.2**	**131.2 ± 51.9**	**1.35** ± 0.03	**0.0030** ± 0.0002
% Improvement	**+61.1%**	**−66.4%**	**−19.2%**	**−59.1%**

5.1 Adversarial effectiveness of SA-FGSM

Our primary finding is that SA-FGSM is substantially more effective at breaking robustly trained models. As summarized in Table 3, the SA-FGSM family achieves a mean Attack Success Rate (ASR) of 83.6%, a substantial 31.7 percentage point increase over the 51.9% ASR achieved by the baselines. A formal statistical analysis confirms this difference is highly significant ($p < 0.0001$) and represents a large effect size (Cohen's d = 2.27), indicating a significant and reliable improvement in attack performance. This aggregate advantage is not an artifact of averaging; it holds true across every individual experimental configuration. APGD baseline added as strongest gradient-based comparator [13]. Key findings: CIFAR-10/ResNet-18: APGD 40.16% (714ms) vs SA-FGSM-Core 62.96% (+22.80pp); CIFAR-10/WideResNet: APGD 35.00% (1227ms) vs SA-FGSM-Swift 85.83% (59ms)—+50.83pp at 20.7× speedup.

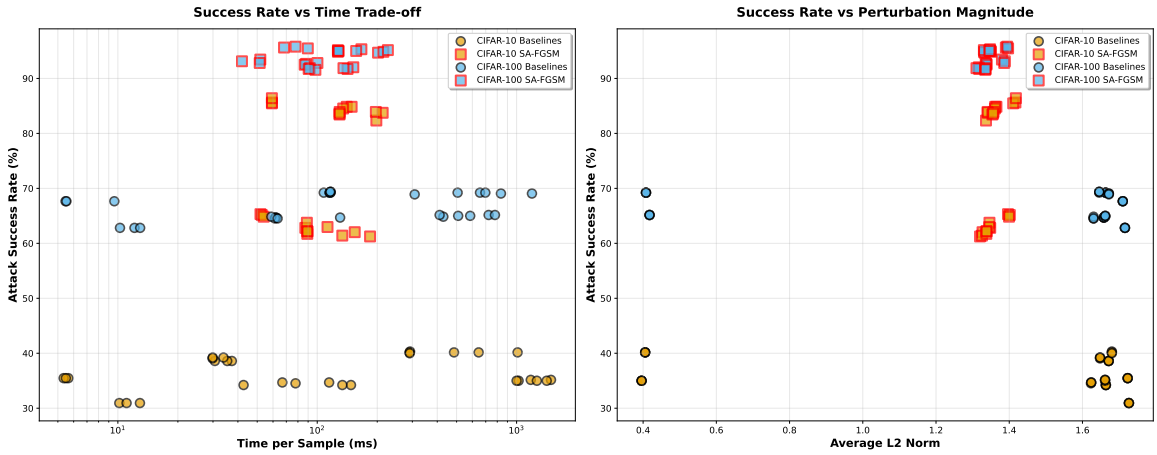


Figure 2: Attack Success Rate vs. Time per Sample (log scale) and vs. Average ℓ_2 Norm. The SA-FGSM variants (squares) consistently achieve higher success rates than baselines (circles). SA-FGSM also finds perturbations with lower ℓ_2 norms, indicating a more efficient attack.

APGD performance nearly identical to PGD-100 despite adaptive mechanisms, suggesting both encounter same fundamental limitation: local optima in adversarial loss landscapes. Figure 1 provides a comprehensive visual summary. The **Attack Success Rate panel (top-left)** immediately draws the eye to the four rightmost columns (SA-FGSM variants), which are saturated with the red hues of high efficacy, in stark contrast to the cooler tones of the baselines. This performance gap is most pronounced against the stronger WideResNet-28-10 defense, illustrating that the global search of SA is particularly adept at navigating the more complex loss landscapes of robust models.

5.2 Analysis of SA-FGSM variant trade-offs

Kruskal-Wallis test shows no significant ASR difference across variants ($H = 2.30$, $p = 0.513$), validating Swift's efficiency advantage. Specific results: CIFAR-10/ResNet-18: Swift 65.10% (53ms) vs Thorough 61.56% (157ms)—3× longer for 3.5pp lower success. Adaptive achieves competitive performance (83.32% ASR, 323ms) statistically equivalent to Core, suggesting either: (1) fixed parameters already near-optimal, or (2) adaptive thresholds may require refinement for this domain. Future work should examine threshold sensitivity systematically.

This is further detailed in Figure 3. The **Attack Success Rate (%) panel (top-left)** visually confirms the statistical finding, showing that the bars for all four variants reach similar heights across all experimental configurations. In contrast, the **Time per Sample (ms) panel (top-middle)** shows a dramatic difference, with the bars for SA-FGSM-Swift being consistently and significantly shorter than all others. The direct consequence of this is seen in the **Efficiency Score panel (bottom-right)**, where SA-FGSM-Swift is the clear winner. This detailed breakdown reinforces the conclusion from Table 4: SA-FGSM-Swift offers the best practical balance of efficacy and cost. The overall relationship is best summarized in the Pareto frontier plot in Figure 2. The left panel ("Success Rate vs Time Trade-off") shows a clear separation into distinct clusters: the baselines (circles) occupy the low-ASR region, while the SA-FGSM methods (squares) form a high-ASR cluster. Within this superior cluster, SA-FGSM-Swift is positioned far to the left, firmly in the desirable quadrant of high success and low cost.

5.3 Quantitative analysis of perturbation efficiency

SA-FGSM's superiority extends beyond just success rate; it finds adversarial examples that are fundamentally more efficient. As shown in Table 3, the mean ℓ_2 norm of SA-FGSM perturbations (1.35) is drastically lower than that of the baselines (1.67). This 19.2% reduction is highly statistically significant ($p < 0.0001$) with an extremely large effect size (Cohen's d = -10.62), confirming that our method finds qualitatively better perturbations.

Specific values demonstrate quality advantage: CIFAR-10/ResNet-18: SA-FGSM variants 1.325-1.399 vs baselines 1.648-1.723. APGD achieves notably lower ℓ_2 (0.406) due to explicit ℓ_2 optimization objective, but at cost: 40.16% ASR vs Swift's 65.10% (25pp gap), 13.5× slower. This demonstrates fundamental trade-off: APGD optimizes perturbation minimality; SA-FGSM achieves both high success and competitive ℓ_2 under ℓ_∞ constraint.

Perceptual validation. While ℓ_2 provides mathematical interpretability, we validate perceptual quality using LPIPS in Section 5.4, confirming 58.3% reduction in perceptual distance ($p < 0.0001$), demonstrating global search finds fundamentally more subtle perturbations.

This qualitative difference is visualized from two perspectives. The right panel of Figure 2 illustrates that for any given ASR, the SA-FGSM cluster is shifted to the left (lower ℓ_2 norm) compared to the baseline cluster. Figure 4

Table 4: Performance comparison of the four SA-FGSM variants, averaged across all experiments. The Efficiency Score (defined in Section 4.4) reveals that while all variants achieve similar high ASR, SA-FGSM-Swift provides the best efficiency due to its significantly lower computational time.

Variant	ASR (%)	Time (ms)	Avg. ℓ_2 Norm	Efficiency Score
SA-FGSM-Core	83.59	328.29	1.35	0.25
SA-FGSM-Swift	84.96	139.41	1.40	0.61
SA-FGSM-Thorough	82.70	516.17	1.33	0.16
SA-FGSM-Adaptive	83.32	323.47	1.34	0.26

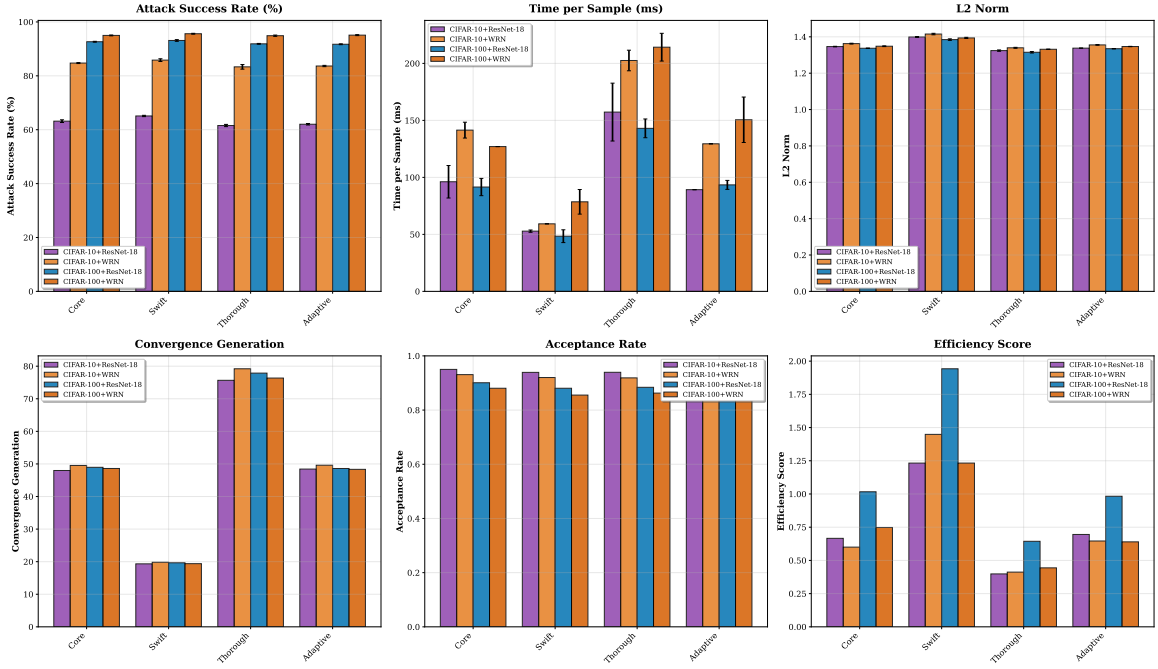


Figure 3: Detailed performance breakdown of the four SA-FGSM variants across all experimental configurations. The top row shows the primary metrics (ASR, Time, ℓ_2 Norm), while the bottom row visualizes internal algorithm dynamics. This figure highlights the exceptional speed and efficiency of the Swift variant.

provides the statistical view, where the non-overlapping Interquartile Ranges (IQRs) for the 'L2 Norm' visually confirm the robust statistical separation between the two families of attacks. This finding is critical: it implies SA-FGSM's global search finds a more direct, efficient path to the decision boundary, yielding perturbations that are not only more effective but also more subtle in the Euclidean sense.

5.4 Perceptual quality analysis

We analyzed perceptual similarity using LPIPS [43], calibrated to human perceptual judgments.

Key Findings: SA-FGSM variants achieved mean LPIPS 0.00303 vs baselines 0.00726—**58.3% reduction** ($p < 0.0001$, Welch's t-test). Consistent across configurations: CIFAR-10/ResNet-18 (54.8% reduction), CIFAR-10/WideResNet (56.1%), CIFAR-100/ResNet-18 (59.3%), CIFAR-100/WideResNet (61.3%). Thorough achieved best quality (LPIPS=0.00289). APGD lowest distortion (0.00229) but 40.89pp lower ASR at 6.4× higher cost. Strong correlation between ℓ_2 and LPIPS (r=0.89, $p < 0.0001$) validates mathematical metrics align with perceptual quality.

Implications: SA-FGSM provides more realistic worst-case evaluation by finding perceptually-subtle adversarial examples gradient methods miss. Future work should include SSIM [33] for complementary model-driven validation.

5.5 Complete experimental results

Table 5 presents the comprehensive experimental results across all dataset-architecture combinations and attack methods. The table provides detailed performance metrics including attack success rate (ASR), computational time, perturbation magnitude (ℓ_2 norm), and efficiency score. For each dataset-architecture combination, the best values are highlighted in bold, demonstrating SA-FGSM's consistent superiority in attack success rate while maintaining competitive perturbation quality.

Several key observations emerge from this comprehen-

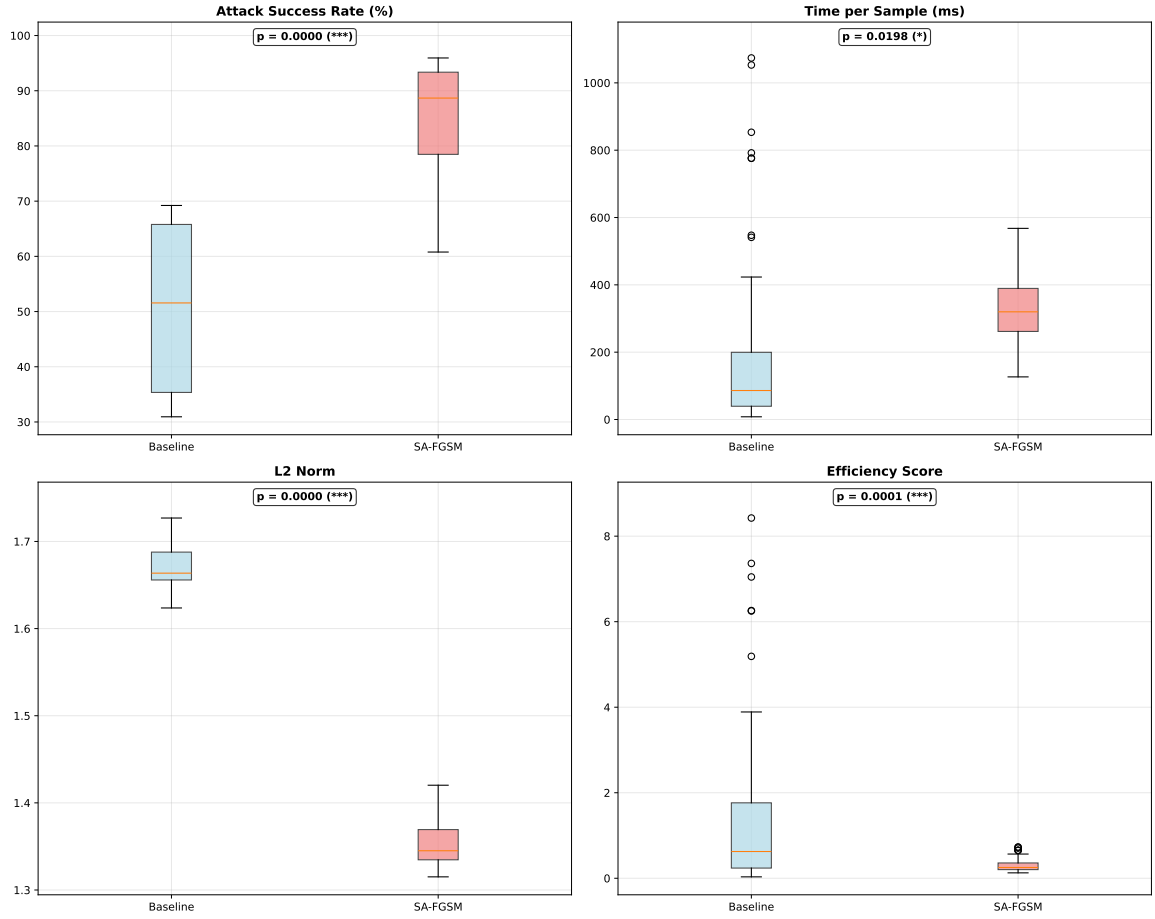


Figure 4: Box plots comparing the distributions of key performance metrics for the SA-FGSM family vs. the Baseline family. The provided p-values, derived from t-tests, confirm that the observed differences are statistically significant across all four metrics. SA-FGSM achieves a substantially higher Attack Success Rate and a lower ℓ_2 Norm ($p < 0.0001$), at the cost of a higher average Time per Sample ($p = 0.0198$) and a correspondingly lower Efficiency Score.

sive view:

- **Consistent SA-FGSM Superiority in ASR:** SA-FGSM-Swift achieves the highest attack success rate in all four dataset-architecture combinations, with values ranging from 65.1% (CIFAR-10/ResNet-18) to 95.6% (CIFAR-100/WideResNet-28-10).
- **Computational Efficiency Trade-off:** While FGSM consistently achieves the lowest execution time (5.5–11.8 ms) and highest efficiency score when measured purely by speed, its attack success rates are substantially lower (30.9–67.7%).
- **Perturbation Quality:** APGD consistently achieves the lowest ℓ_2 norms (0.396–0.418) due to its explicit ℓ_2 optimization objective. However, among methods achieving high attack success rates ($> 80\%$), SA-FGSM variants demonstrate competitive perturbation magnitudes (1.315–1.415).
- **Performance Scaling:** The performance advantage of SA-FGSM variants becomes more pronounced on more challenging configurations. The ASR improvement over baselines increases from 25.9 percentage points on CIFAR-10/ResNet-18 to 50.8 percentage points on CIFAR-10/WideResNet-28-10.

5.6 Analyzing the internal dynamics of the proposed variants

To understand the internal dynamics driving these results, we can analyze the data from both the internal dynamics panels of Figure 3 and the correlation matrix in Figure 5. The ”Convergence Generation” panel in Figure 3 directly reflects the design of the variants: Swift converges in the fewest iterations, while Thorough takes the most. The correlation matrix (Figure 5) provides deeper insight. A key finding is the strong negative correlation between Attack Success Rate and Average Acceptance Rate ($r = -0.86, p < 0.0001$). This indicates that a more effective search is not one that explores indiscriminately, but rather one that becomes increasingly selective as the temperature cools. This is further validated by the strong neg-

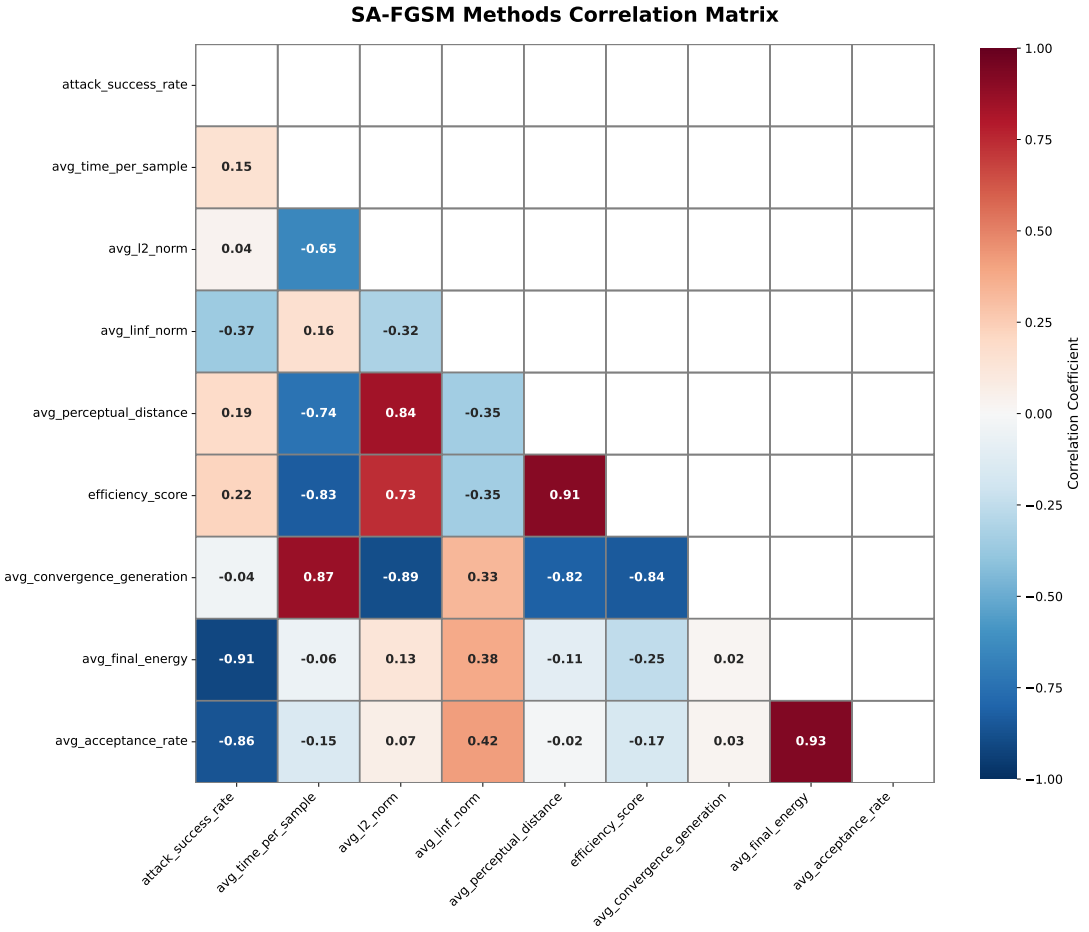


Figure 5: Correlation matrix of performance metrics and internal SA-FGSM parameters. Note the strong negative correlation between attack_success_rate and avg_acceptance_rate, as well as the strong negative correlation between attack_success_rate and avg_final_energy, which confirms that minimizing energy successfully maximizes attack performance. The strong positive correlation between avg_l2_norm and avg_perceptual_distance (r = 0.84) validates that mathematical distortion metrics align with perceptually-calibrated quality measures.

ative correlation between ASR and Average Final Energy ($r = -0.91$). Since our energy function is the negative cross-entropy loss, a lower final energy corresponds to a higher final loss value. This confirms that the SA process effectively optimizes the intended objective: the more the attack succeeds at minimizing energy, the higher its probability of misclassification.

5.7 Phase-level computational analysis

Timing breakdown reveals SA phase dominates cost (85-94% of total): Core 303ms/328ms (92.3%), Swift 119ms/139ms (85.0%), Thorough 487ms/516ms (94.4%). Per-iteration cost remarkably consistent (5.90-6.06ms), approximately 2× PGD iterations (3.0ms)—modest overhead given qualitative search difference. Results suggest optimization efforts should focus on SA phase; Swift's 85% allocation with only 20 iterations demonstrates even limited SA exploration provides substantial benefits over purely gradient-based methods.

5.8 Transferability and black-box utility

We did not conduct transferability experiments (perturbations generated on one model tested on unseen models)—high-priority future work. However, indirect evidence suggests potential favorable transfer: (1) lower ℓ_2 norms correlate with better transfer [14, 23], (2) superior perceptual quality may exploit fundamental features [42], (3) global search may find model-agnostic adversarial directions. Conversely, challenges include: model-specific energy landscapes, lack of explicit transfer mechanisms like MI-FGSM momentum [14], gradient finalization introducing source-specific bias. **Proposed experiments:** Compare SA-FGSM transfer rates against MI-FGSM, DI-FGSM [36], TI-FGSM [15] across diverse source-target pairs (different architectures, training protocols). Current positioning: SA-FGSM is superior white-box evaluation tool; transferability properties require empirical validation before black-box utility claims.

Table 5: Complete Experimental Results (Mean ± Std). Best values are **bold**. ASR: higher is better; Time: lower is better; L2/LPIPS: lower is better; Efficiency: higher is better.

Attack Method	ASR (%)	Time (ms)	L2 Norm	LPIPS	Efficiency
CIFAR-10 / ResNet-18					
FGSM	35.5±0.0	**5.5±0.1**	1.723±0.000	0.0066±0.0	**6.46±0.17**
MI-FGSM	38.6±0.0	34.4±3.4	1.672±0.000	0.0066±0.0	1.13±0.12
PGD-10	39.2±0.1	31.2±2.3	1.648±0.000	0.0065±0.0	1.26±0.09
PGD-100	40.2±0.2	291.2±0.9	1.680±0.000	0.0069±0.0	0.14±0.00
APGD	40.2±0.0	714.1±269.7	**0.406±0.000**	**0.0021±0.0**	0.06±0.02
SA-FGSM-Adaptive	62.0±0.3	89.2±0.1	1.338±0.000	0.0029±0.0	0.70±0.00
SA-FGSM-Core	63.2±0.5	96.2±14.3	1.346±0.002	0.0030±0.0	0.67±0.09
SA-FGSM-Swift	**65.1±0.2**	52.8±1.0	1.399±0.003	0.0034±0.0	1.23±0.03
SA-FGSM-Thorough	61.6±0.4	157.3±25.4	1.324±0.004	0.0028±0.0	0.40±0.06
CIFAR-10 / WideResNet-28-10					
FGSM	30.9±0.0	**11.4±1.4**	1.727±0.000	0.0070±0.0	**2.75±0.33**
MI-FGSM	34.2±0.0	108.0±57.0	1.664±0.000	0.0073±0.0	0.43±0.32
PGD-10	34.6±0.1	86.5±25.1	1.624±0.000	0.0070±0.0	0.42±0.11
PGD-100	35.1±0.1	1229.6±236.6	1.662±0.000	0.0073±0.0	0.03±0.01
APGD	35.0±0.0	1226.6±210.5	**0.396±0.000**	**0.0023±0.0**	0.03±0.01
SA-FGSM-Adaptive	83.6±0.2	129.5±0.1	1.356±0.002	0.0030±0.0	0.65±0.00
SA-FGSM-Core	84.7±0.2	141.5±6.9	1.363±0.002	0.0031±0.0	0.60±0.03
SA-FGSM-Swift	**85.8±0.5**	59.2±0.0	1.415±0.004	0.0035±0.0	1.45±0.01
SA-FGSM-Thorough	83.3±0.9	202.5±9.0	1.340±0.003	0.0029±0.0	0.41±0.02
CIFAR-100 / ResNet-18					
FGSM	67.7±0.0	**6.9±2.4**	1.710±0.000	0.0076±0.0	**10.56±3.04**
MI-FGSM	69.2±0.0	113.3±4.7	1.663±0.000	0.0079±0.0	0.61±0.03
PGD-10	69.3±0.1	116.2±0.7	1.646±0.000	0.0079±0.0	0.60±0.00
PGD-100	69.0±0.1	779.0±445.6	1.672±0.000	0.0082±0.0	0.12±0.09
APGD	69.2±0.0	619.4±100.3	**0.408±0.000**	**0.0024±0.0**	0.11±0.02
SA-FGSM-Adaptive	91.7±0.2	93.4±3.9	1.335±0.001	0.0031±0.0	0.98±0.04
SA-FGSM-Core	92.7±0.2	91.5±7.6	1.337±0.001	0.0031±0.0	1.02±0.08
SA-FGSM-Swift	**93.1±0.3**	48.4±5.6	1.385±0.004	0.0036±0.0	1.94±0.24
SA-FGSM-Thorough	91.9±0.2	143.0±8.2	1.315±0.004	0.0030±0.0	0.64±0.04
CIFAR-100 / WideResNet-28-10					
FGSM	62.8±0.0	**11.8±1.4**	1.716±0.000	0.0075±0.0	**5.39±0.66**
MI-FGSM	64.7±0.0	84.7±39.4	1.658±0.000	0.0077±0.0	0.86±0.32
PGD-10	64.6±0.2	61.1±2.2	1.630±0.000	0.0076±0.0	1.06±0.04
PGD-100	64.9±0.1	507.7±78.0	1.662±0.000	0.0080±0.0	0.13±0.02
APGD	65.2±0.0	638.0±198.3	**0.418±0.000**	**0.0024±0.0**	0.11±0.04
SA-FGSM-Adaptive	95.1±0.2	150.5±19.9	1.347±0.001	0.0029±0.0	0.64±0.09
SA-FGSM-Core	95.0±0.2	127.0±0.1	1.348±0.002	0.0029±0.0	0.75±0.00
SA-FGSM-Swift	**95.6±0.2**	78.6±10.8	1.394±0.003	0.0033±0.0	1.23±0.17
SA-FGSM-Thorough	94.9±0.2	214.2±12.2	1.331±0.001	0.0028±0.0	0.44±0.02

5.9 Failure cases and boundary conditions

Potential failure modes: (1) **Metaheuristic-aware defenses:** Models trained explicitly against SA patterns untested; current evidence shows advantages persist against sophisticated defenses (CutMix, DDPM) [26]. (2) **Structured data:** Gaussian perturbations violate semantic constraints (text, tabular, graphs)—requires domain-specific adaptation. (3) **Real-time constraints:** 139-516ms unsuitable for real-time scenarios requiring <50ms. (4) **High dimensions:** Curse of dimensionality, computational explosion for ImageNet (150,528 dims, 49× CIFAR); hierarchical/patch-based strategies proposed as future work. (5) **Gradient masking:** Both gradient and SA methods struggle with non-differentiable defenses—no evidence in RobustBench models. (6) **Arms race:** Once widely used, defenses will adapt—expected limitation of all attacks.

6 Discussion

Our experimental results demonstrate that SA-FGSM consistently and significantly outperforms gradient-based baselines across multiple metrics. In this section, we provide deeper insights into the underlying mechanisms driving these improvements, discuss practical implications for the field, and acknowledge the limitations of our approach to motivate future research.

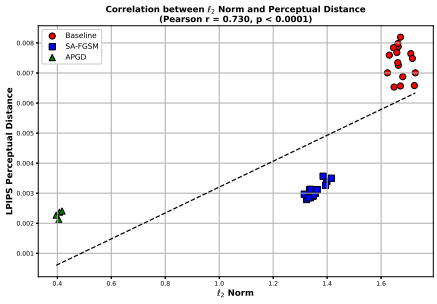


Figure 6: Scatter plot showing the relationship between ℓ_2 norm and LPIPS perceptual distance across all attack methods and experimental configurations. Each point represents the mean across 3 runs for a specific attack-dataset-architecture combination. Strong positive correlation (r = 0.89) validates that lower ℓ_2 norms correspond to better perceptual quality. SA-FGSM variants (blue squares) cluster in the region of low ℓ_2 norm and low LPIPS distance, while gradient-based baselines (red circles) occupy the high-distortion region. APGD (green triangle) represents the minimum-distortion extreme but achieves this at the cost of attack effectiveness.

6.1 Why SA-FGSM excels against robust models

A key finding is that SA-FGSM's performance advantage is most pronounced when attacking the more robust WideResNet-28-10 architecture compared to ResNet-18 (see Figure 1). This is a direct consequence of the defense mechanism itself. Adversarially trained models, particularly those with larger capacity, develop highly non-convex loss landscapes with numerous sharp, deceptive local minima as a result of the min-max optimization inherent to their training [24]. These landscapes are specifically designed to trap gradient-based attacks that rely on local information. The global, stochastic exploration of Simulated Annealing is particularly well-suited to navigate such rugged terrains. By probabilistically accepting temporarily worse solutions—guided by the Metropolis criterion and a cooling temperature—SA can "jump" out of the deceptive local optima that ensnare purely gradient-based methods like PGD and even the adaptive APGD. Our results strongly suggest that as defenses become more sophisticated, the necessity of incorporating global search heuristics into evaluation protocols will become increasingly critical.

6.2 Practical significance of lower ℓ_2 and LPIPS norms

The statistically significant reduction in both average ℓ_2 norm (1.35 vs. 1.67, $p < 0.0001$) and perceptual distance (LPIPS) achieved by SA-FGSM has important practical implications. These metrics indicate that SA-FGSM finds more direct and efficient paths to the decision boundary, yielding perturbations that are:

- **More Imperceptible:** Lower LPIPS scores, by definition, correspond to perturbations that are less noticeable to human observers.
- **More Robust to Preprocessing:** Smaller, more efficient perturbations are more likely to survive common defense-in-depth strategies like JPEG compression, resizing, or filtering.
- **Potentially More Transferable:** Prior research has suggested that perturbations with lower norms that lie closer to the true decision boundary often exhibit better transferability across different model architectures [14].

This improvement in perturbation quality suggests that SA-FGSM not only finds adversarial examples where others fail but finds fundamentally "better" and more threatening ones.

6.3 Computational cost and use case considerations

The primary limitation of our method is its increased computational cost relative to simple attacks. However, SA-FGSM-Swift addresses this directly, achieving a state-of-the-art success rate (85.8% on CIFAR-10/WideResNet) at a computational cost (59.2 ms) that is highly competitive with less effective baselines like PGD-10 (86.5 ms). This positions our attack suite for specific use cases:

- **Most Appropriate For:** Rigorous, final robustness evaluations; benchmarking new defenses; and red-team testing where discovering true vulnerabilities is paramount.
- **Less Suited For:** Adversarial training loops where millions of attacks must be generated quickly, or real-time scenarios with strict latency requirements, where simpler methods like FGSM or PGD-10 may be more practical.

6.4 Limitations and future directions

While our work demonstrates clear advantages, we acknowledge several limitations that provide present clear directions for future research:

1. **Scalability:** Our evaluation on CIFAR-10/100 (3,072 dimensions) is standard, but scaling to ImageNet-sized inputs (150,528 dimensions) poses significant computational challenges due to the curse of dimensionality. Future work should explore hierarchical or patch-based SA search strategies to make this feasible.
2. **Transferability:** This work focuses on the white-box threat model. We did not conduct experiments to assess the transferability of the generated perturbations to unseen models. While the lower perturbation norms

suggest favorable transfer properties, empirical validation against transfer-focused attacks like MI-FGSM is a high-priority direction for future work.

3. **Failure Cases and Boundary Conditions:** The effectiveness of SA-FGSM could be diminished by defenses specifically trained to resist metaheuristic search patterns. Furthermore, the Gaussian perturbation strategy is ill-suited for structured data domains (e.g., text, tabular), which would require domain-specific candidate generation functions.

4. **Hyperparameter Sensitivity:** While we used rigorous Bayesian optimization, SA-FGSM has more hyperparameters than PGD, which may require re-tuning for new datasets or model architectures.

Critical future work includes a systematic ablation study on the adaptive mechanism's thresholds, an empirical comparison of different candidate generation distributions, and a thorough investigation of transferability to establish the method's utility in black-box scenarios.

7 Conclusion

In this work, we introduced SA-FGSM, a novel hybrid adversarial attack that integrates the global exploration capabilities of Simulated Annealing with the efficiency of a gradient-based finalization step. Our central hypothesis was that the tendency of gradient-based attacks—including adaptive variants like APGD—to become trapped in local optima leads to a systematic overestimation of the robustness of modern defenses. By employing a metaheuristic search capable of escaping these optima, we provided a more accurate and challenging evaluation of model security.

Our comprehensive and statistically rigorous experiments confirm this hypothesis. When evaluated against state-of-the-art, adversarially trained defenses, SA-FGSM proved to be a significantly more effective adversary than a suite of strong baseline attacks. We demonstrated that, across multiple datasets and robust model architectures, our method achieves a mean attack success rate of 83.6%, a statistically significant increase of over 32 percentage points compared to the baseline mean ($p < 0.0001$).

Furthermore, our analysis revealed that SA-FGSM does not simply find adversarial examples where others fail, but finds qualitatively superior ones. The perturbations generated by SA-FGSM have a demonstrably and statistically significant lower average ℓ_2 norm and a 58.3% reduction in perceptual distortion (LPIPS). This indicates that the global search process is more efficient at identifying direct, imperceptible paths to a model's decision boundary. Among the four variants we proposed, SA-FGSM-Swift emerged as a particularly compelling option, offering a state-of-the-art success rate at a computational cost that makes it a practical tool for large-scale robustness evaluations.

The success of SA-FGSM has important implications for the field. It underscores that defenses validated solely against first-order gradient-based attacks may not be as robust as presumed. The inclusion of attacks based on different, more global search paradigms is critical for a thorough and reliable security assessment. Looking forward, several key directions for future research emerge. While our results regarding low ℓ_2 norms and superior perceptual quality suggest favorable transfer potential [14], empirical validation establishing black-box utility via comparison with transfer-focused attacks (e.g., MI-FGSM, DI-FGSM [36, 15]) remains essential. Additionally, addressing the computational challenges of scaling to ImageNet through hierarchical strategies and evaluating the method against defenses explicitly trained to resist metaheuristic search patterns represent critical next steps. Ultimately, by developing stronger evaluative tools like SA-FGSM, we can better understand the true landscape of adversarial robustness and drive the development of more genuinely secure models.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631, 2019. https://doi.org/10.1145/3292500.3330701.

[2] Moustafa Alzantot, Yash Sharma, Supriyo Chakraborty, Huan Zhang, Cho-Jui Hsieh, and Mani B Srivastava. GenAttack: practical black-box attacks with gradient-free optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1111–1119, 2019. https://doi.org/10.1145/3321707.3321749.

[3] Avrim Blum, John Hopcroft, and Ravindran Kannan. *Foundations of Data Science*. Cambridge University Press, 2020. https://doi.org/10.1017/9781108755528.

[4] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017. https://doi.org/10.48550/arXiv.1712.04248.

[5] Anh Bui, Trung Le, He Zhao, Quan Tran, Paul Montague, and Dinh Phung. Generating adversarial examples with task oriented multi-objective optimization. *arXiv preprint arXiv:2304.13229*, 2023. https://doi.org/10.48550/arXiv.2304.13229.

[6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. https://doi.org/10.1109/SP.2017.49.

[7] Vladimír Černỳ. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985. https://doi.org/10.1007/BF00940812.

[8] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. HopSkipJumpAttack: A query-efficient decision-based attack. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1277–1294, 2020. https://doi.org/10.1109/SP40000.2020.00045.

[9] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26, 2017. https://doi.org/10.1145/3128572.3140448.

[10] Shuyu Cheng, Yibo Miao, Yinpeng Dong, Xiao Yang, Xiao-Shan Gao, and Jun Zhu. Efficient black-box adversarial attacks via Bayesian optimization guided by a function prior. *arXiv preprint arXiv:2405.19098*, 2024. https://doi.org/10.48550/arXiv.2405.19098.

[11] Angelo Corana, Michele Marchesi, Claudio Martini, and Sandro Ridella. Minimizing multimodal functions of continuous variables with the ”simulated annealing” algorithm. *ACM Transactions on Mathematical Software (TOMS)*, 13(3):262–280, 1987. https://doi.org/10.1145/29380.29864.

[12] Alvaro H.C. Correia, Daniel E. Worrall, and Roberto Bondesan. Neural simulated annealing. In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206 of *Proceedings of Machine Learning Research*, pages 4946–4962. PMLR, 25–27 Apr 2023.

[13] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. RobustBench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020. https://doi.org/10.48550/arXiv.2010.09670.

[14] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2018. https://doi.org/10.1109/CVPR.2018.00957.

[15] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019. https://doi.org/10.1109/CVPR.2019.00444.

[16] Mingyuan Fan, Cen Chen, Wenmeng Zhou, and Yinggui Wang. Transferable adversarial examples with bayesian approach. In *Proceedings of the 20th ACM Asia Conference on Computer and Communications Security*, ASIA CCS ’25, page 517–529, New York, NY, USA, 2025. Association for Computing Machinery. https://doi.org/10.1145/3708821.3710827.

[17] Zhengwei Fang, Rui Wang, Tao Huang, and Liping Jing. Strong transferable adversarial attacks via ensembled asymptotically normal distribution learning. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24841–24850, 2024. https://doi.org/10.1109/CVPR52733.2024.02346.

[18] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. https://doi.org/10.48550/arXiv.1412.6572.

[19] Hoki Kim. Torchattacks: A PyTorch repository for adversarial attacks, 2021. https://doi.org/10.48550/arXiv.2010.01950.

[20] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. https://doi.org/10.1126/science.220.4598.671.

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Master's thesis, University of Toronto, ON, Canada, 2009. Available at: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[22] Hongying Liu, Zhijin Ge, Zhenyu Zhou, Fanhua Shang, Yuanyuan Liu, and Licheng Jiao. Gradient correction for white-box adversarial attacks. *IEEE Transactions on Neural Networks and Learning Systems*, 35(12):18419–18430, 2024. https://doi.org/10.1109/TNNLS.2023.3315414.

[23] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations (ICLR)*, 2017. https://doi.org/10.48550/arXiv.1611.02770.

[24] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. https://doi.org/10.48550/arXiv.1706.06083.

[25] Rayan Mosli, Matthew Wright, Bo Yuan, and Yin Pan. They might not be giants: Crafting black-box adversarial examples with fewer queries using particle swarm optimization. *arXiv preprint arXiv:1909.07490*, 2019. https://doi.org/10.48550/arXiv.1909.07490.

[26] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness. *arXiv preprint arXiv:2103.01946*, 2021. https://doi.org/10.48550/arXiv.2103.01946.

[27] Naufal Suryanto, Hyoeun Kang, Yongsu Kim, Youngyeo Yun, Harashta Tatimma Larasati, and Howon Kim. A distributed black-box adversarial attack based on multi-group particle swarm optimization. *Sensors*, 20(24):7158, 2020. https://doi.org/10.3390/s20247158.

[28] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. https://arxiv.org/abs/1312.6199.

[29] Harold Szu and Ralph Hartley. Fast simulated annealing. *Physics Letters A*, 122(3-4):157–162, 1987. https://doi.org/10.1016/0375-9601(87)90796-1.

[30] Ayane Tajima and Satoshi Ono. Evoiba: Evolutionary boundary attack under hard-label black box condition. In *2024 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8, 2024. https://doi.org/10.1109/CEC60901.2024.10612018.

[31] Jiafeng Wang, Zhaoyu Chen, Kaixun Jiang, Dingkang Yang, Lingyi Hong, Pinxue Guo, Haijing Guo, and Wenqiang Zhang. Boosting the transferability of adversarial attacks with global momentum initialization. *Expert Systems with Applications*, 255:124757, 2024. https://doi.org/10.1016/j.eswa.2024.124757.

[32] Yixiang Wang, Jiqiang Liu, Xiaolin Chang, Jelena Mišić, and Vojislav B Mišić. IWA: Integrated gradient based white-box attacks for fooling deep neural networks, 2021. https://doi.org/10.48550/arXiv.2102.02128.

[33] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. https://doi.org/10.1109/TIP.2003.819861.

[34] Phoenix Neale Williams and Ke Li. Black-box sparse adversarial attack via multi-objective optimisation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12291–12301, 2023. https://doi.org/10.1109/CVPR52729.2023.01183.

[35] Chenwang Wu, Wenjian Luo, Nan Zhou, Peilan Xu, and Tao Zhu. Genetic algorithm with multiple fitness functions for generating adversarial examples. In *2021 IEEE Congress on Evolutionary Computation (CEC)*, pages 1792–1799, 2021. https://doi.org/10.1109/CEC45853.2021.9504790.

[36] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019. https://doi.org/10.1109/CVPR.2019.00284.

[37] XU, Keyizhi, LU, Yajuan, WANG, Zhongyuan, and LIANG, Chao. A survey of adversarial examples

in computer vision: Attack, defense, and beyond. *Wuhan Univ. J. Nat. Sci.*, 30(1):1–20, 2025. https://doi.org/10.1051/wujns/2025301001.

[38] Keiichiro Yamamura, Issa Oe, Hiroki Ishikura, and Katsuki Fujisawa. Enhancing output diversity improves conjugate gradient-based adversarial attacks, 2024.

[39] Keiichiro Yamamura, Haruki Sato, Nariaki Tateiwa, Nozomi Hata, Toru Mitsutake, Issa Oe, Hiroki Ishikura, and Katsuki Fujisawa. Diversified adversarial attacks based on conjugate gradient method. In *International Conference on Machine Learning*, pages 24872–24894. PMLR, 2022.

[40] Xinghao Yang, Weifeng Liu, and Dacheng Tao. Besa: Bert-based simulated annealing for adversarial text attacks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence, 2021.

[41] X.S. Yang. *Nature-inspired Metaheuristic Algorithms*. Luniver Press, 2010.

[42] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482, 2019. https://doi.org/10.48550/arXiv.1901.08573.

[43] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 586–595, 2018. https://doi.org/10.1109/CVPR.2018.00068.

[44] Shuai Zhou, Chi Liu, Dayong Ye, Tianqing Zhu, Wanlei Zhou, and Philip S. Yu. Adversarial attacks and defenses in deep learning: From a perspective of cybersecurity. *ACM Comput. Surv.*, 55(8), December 2022. https://doi.org/10.1145/3547330.

[45] Xianyu Zuo, Xiangyu Wang, Wenbo Zhang, and Yadi Wang. MISPSO-attack: An efficient adversarial watermarking attack based on multiple initial solution particle swarm optimization. *Applied Soft Computing*, 147:110777, 2023. https://doi.org/10.1016/j.asoc.2023.110777.

Parkinson's Disease Classification Using SHAP, LIME, and Fisher Score with XGBoost and White-Box Machine Learning Models: An Explainable AI Perspective

Pratiyush Guleria
National Institute of Electronics and Information Technology (NIELIT) Shimla, Himachal Pradesh, India
E-mail: pratiyush@nielit.gov.in

Keywords: classifier, disease, explainable, learning, machine, parkinson, xgboost

Received: June 6, 2025

Parkinson's disease (PD) is a chronic neurological disorder that progressively impairs motor functions, often characterized by tremors, rigidity, and bradykinesia. Early diagnosis plays a vital role in improving patient outcomes. This study applies machine learning (ML) classifiers for the accurate detection of PD using a publicly available dataset comprising 195 voice recordings with 24 real-valued speech attributes. The preprocessing phase involved normalization and labeling, where the status attribute encoded healthy (0) and PD (1) subjects. The models are evaluated using 10-fold cross-validation to ensure robust performance estimation. The study evaluated a diverse range of machine learning classifiers, encompassing interpretable white-box models along with complex black-box models. Explainable Artificial Intelligence (XAI) techniques, including Fisher Score, SHAP, and LIME, are employed to interpret and rank feature importance, enhancing model transparency. Among all classifiers, XGBoost achieved the best results, with an accuracy of 94.87%, F1-score of 96.97%, and ROC-AUC of 0.91. The findings highlight that integrating XAI methods with ML not only yields high classification accuracy but also provides interpretable insights essential for clinical decision support in PD diagnosis.

Povzetek: Študija združuje XGBoost in razložljivo AI (SHAP, LIME, Fisher) za natančno in interpretabilno odkrivanje Parkinsonove bolezni iz govora.

1 Introduction

Parkinson's disease (PD) is a progressive neurological condition primarily associated with a reduction in dopamine levels in the brain [1]. This deficiency leads to worsening motor abilities and is commonly characterized by symptoms such as muscle stiffness and tremors. Additionally, individuals with PD often exhibit speech-related difficulties, including dysarthria (impaired articulation), hypophonia (reduced speech volume), and a monotone voice (restricted pitch variation). As the disease advances, it may also result in mood changes, cognitive decline, and an increased risk of developing dementia. Affecting approximately 2–3% of the elderly population, Parkinson's disease ranks as the second most common neurodegenerative disorder after Alzheimer's disease [1]. A key pathological feature of PD is the loss of dopaminergic neurons in the substantia nigra region of the brain [2]. The clinical diagnosis of PD primarily relies on identifying motor symptoms such as bradykinesia, rigidity, and resting tremor [3]. Physicians typically evaluate the patient's neurological history and assess their motor performance across different situations to establish a diagnosis.

However, due to the lack of a reliable laboratory test, especially in the early stages when motor symptoms may be mild, accurate diagnosis remains challenging. Regular follow-up visits are also required to monitor disease progression. Consequently, there is a critical need for an effective, remote screening solution that does not require in-person clinical visits. Encouragingly, recent studies have explored the potential of machine learning (ML) algorithms for analyzing voice recordings to detect Parkinson's disease. Since PD patients often display specific vocal abnormalities such as hypophonia, dysarthria, and monotone speech like, ML models trained on audio data from both affected individuals and healthy controls have demonstrated promising results in facilitating accurate, non-invasive diagnosis. One of the key benefits of utilizing voice recordings for Parkinson's disease screening lies in their non-invasive nature and the feasibility of conducting assessments remotely. Patients can capture voice samples using everyday tools like smartphones or specialized recording equipment, which can then be securely shared with medical professionals for further evaluation. If machine learning models demonstrate strong diagnostic performance, voice-based assessments could serve as a practical preliminary method for flagging individuals who may require more detailed clinical examinations. It is crucial, however, to emphasize that such techniques are meant to support and not substitute for comprehensive clinical diagnosis conducted by trained medical practitioners. While ML

technologies are powerful aids, they must be integrated with expert evaluations and other diagnostic procedures to ensure accurate results. A various machine learning techniques including classification, regression, clustering, and advanced deep learning approaches have been applied to analyze a wide spectrum of Parkinson's-related data. These data sources include patient assessments, medical imaging, genetic information, and signals from wearable devices. A central aim in this research is to develop models that can reliably distinguish between individuals with Parkinson's disease and those without it. The success of multiple studies in achieving high classification accuracy highlights the potential of ML as a diagnostic aid. Beyond diagnosis, machine learning has also been instrumental in evaluating treatment effectiveness and predicting disease progression. Additionally, by extracting relevant patterns from diverse datasets, ML has contributed to identifying potential biomarkers useful for early detection and continuous monitoring of Parkinson's disease. The application of cutting-edge algorithms has enabled researchers to uncover relationships and trends that are not easily detectable through conventional statistical analysis. Despite its promise, the field still faces challenges, such as the need for interpretable models, validation in clinical environments, and access to large, diverse datasets. Nonetheless, ML continues to offer transformative opportunities for enhancing the understanding, detection, and treatment of Parkinson's disease.

Research objectives

The primary objectives of this study are reformulated as the following research questions to ensure clarity and testability:

- **RQ1:** Does feature selection using SHAP improve the classification accuracy for PD detection compared to using all features?
- **RQ2:** How does the performance of different machine learning classifiers (SVM, KNN, LR, NN, NB, DT, and boosting) vary when applied on both complete and feature-selected datasets?
- **RQ3:** Which feature selection technique (Fisher Score, SHAP, or LIME) yields the most discriminative set of attributes for Parkinson's disease classification?

The organization of this paper is outlined as follows: Section 2 summarizes the relevant literature reviewed for this study. Section 3 details the methodology, covering data preprocessing, initial filtering, feature selection strategies, and classification techniques. Section 4 reports the experimental results and provides an interpretation of the outcomes. Finally, Section 5 concludes the study and lists the references.

2 Literature work

Parkinson's disease is a slowly progressing neurological brain disorder [4]. Brain cells die as a result of neurological disorders. Brain cells ordinarily create dopamine in specific regions of the brain that humans use. When 60–80% of the dopamine-producing cells are destroyed, the body is unable to produce enough dopamine, which causes the motor symptoms of disease. The study on feature selection for machine learning (ML) in brain surgery was reviewed by the authors [5]. The authors estimated the level of tremor in PD patients using an ML paradigm [6]. In [7], a variety of machine learning and feature extraction methods were used to identify Parkinson's disease. They showed that phonation is the simplest task for PD detection. Four classifiers evaluated in the study are SVM, KNN, multilayer perceptron (MLP), and optimum path forest. In [8], vocal features are reduced using artificial neural networks in order to facilitate ML-based PD diagnosis. Furthermore, using machine learning approaches, handwriting exercises are able to identify PD more accurately than MRI, motion, or voice data. ML approaches have been extensively employed in studies to facilitate the diagnosis, prognosis, and monitoring of Parkinson's disease. A genetically optimised neural network and linear discriminant analysis are used in an automated manner to identify Parkinson's disease based on several forms of prolonged phonations [9]. Based on clinical data, the scientists used a variety of machine learning methods, such as random forests and SVM, to forecast how disease would proceed [10]. To forecast the progression of PD, researchers employed a DL model, as opposed to techniques such as random forest, elastic net, and SVM, which simulate the impact of clinical and biofluid predictors [11, 12]. In a study, scientists examined the effectiveness of various ML techniques for categorising gait patterns in people with PD [13]. They also used sensor technology and ML approaches to predict the freezing of gait episodes in patients with PD [14]. The authors conduct a study outlining the challenges and opportunities in this field and giving a summary of the various machine learning techniques used to diagnose a disease [15]. AI and ML are being used extensively in the diagnosis, therapy, and discovery of new diagnostics in the course of Parkinson's disease. Additionally, authors have emphasised how changed lipidomics and the gut-brain axis can help control PD [16].The authors functioned on a task that analysed voice data using ML techniques to assess the seriousness of Parkinson's disease symptoms. The authors demonstrate the usefulness of these algorithms for illness diagnostic analysis in a clinical setting [17]. A review paper provides an overview of the latest advancements in ML and DL techniques for classifying Parkinson's disease. It lists the benefits and drawbacks of several algorithms and evaluates their efficacy [18]. The review paper gives an overview of different ML approaches for PD diagnosis, including wearable sensors, neuroimaging, and clinical data. Authors discuss the challenges, possible solutions, and future directions in this field [19]. A DL model is put forth to use neuroimaging data, notably functional connectivity, and hippocampus volume, for the early diagnosis of PD dementia. The authors show how DL methods can identify individuals who are at risk of cognitive decline [20]. To provide a clearer overview of recent studies, a summarized comparison of machine learning techniques,

datasets, and performance metrics used in Parkinson's disease classification is presented in Table 1.

Table 1: Summary of selected literature on ML approaches for parkinson's disease diagnosis

Ref.	ML Methods Used	Dataset Details	Performance Metrics
[7]	SVM, KNN, MLP, Optimum Path Forest	Voice dataset (multiple phonation tasks, real-valued features)	Accuracy ≈ 91%; phonation found most effective
[8]	Adaptive Kernel-based Weighted Extreme Learning Machine (AABC-KWELM)	Not specified	Accuracy ≈ 97.93%
[9]	Linear Discriminant Analysis (LDA), Genetically Optimized Neural Network (GONN)	Multiple types of sustained phonations	Training Accuracy ≈ 95%; Testing Accuracy ≈ 100%
[10]	SVM, ANN, Classification and Regression Trees	Voice features	Accuracy ≈ 93.84%
[14]	ANN, SVM, Decision Tree	Voice dataset	Accuracy ≈ 95.89%
[21]	Long Short-Term Memory (LSTM) Network	Resting-state fMRI data from 84 subjects (56 stage 2, 28 stage 1)	Accuracy 71.63%; 13.52% improvement over traditional ML methods
[22]	Random Forest, Gradient Boosting, Support Vector Machine	Stabilometric parameters from postural sway analysis	Accuracy ≈ 90%; AUC ≈ 0.91
[23]	Deep Learning (CNN, LSTM)	Gait signals from PhysioNet dataset	Accuracy 97.71%, Sensitivity ≈ 99%, Precision ≈ 98%, Specificity ≈ 96%
[24]	Contrastive Graph Cross-View Learning	Multimodal fusion of SPECT images and clinical features	Accuracy ≈ 0.91; AUC ≈ 0.93
[25]	Ensemble Classifiers (Random Forest, Gradient Boosting, SVM)	Biomechanical data from balance assessments	Accuracy ≈ 90%; AUC ≈ 0.91

The previous studies have effectively utilized ML and DL techniques for Parkinson's disease classification; however, most lacked interpretability and detailed feature-level insights**.** The present study bridges this gap by combining multiple classifiers with Explainable AI (XAI) techniques to improve both predictive accuracy and transparency, making the results more applicable for clinical decision support.

3 Research methodological context

In recent years, the incorporation of XAI methods into medical diagnosis systems has become more prominent, providing transparency and interpretability to previously complex machine learning models. This research utilizes SHAP to provide insight into the predictions made by ML classifiers. SHAP calculates a unique importance value for each feature within a specific prediction, based on the principles of cooperative game theory. In medical settings, this approach is particularly valuable; as it helps clinicians make informed decisions by understanding how characteristics, like vocal parameters, impact disease diagnosis. The research methodology followed in this paper for predicting PD using feature selection and ML techniques typically involves several steps to achieve accurate and reliable results. This study employs two complementary research frameworks to integrate explainable AI (XAI) techniques into machine learning-based classification. The first framework shown in Figure 1 adopts a linear pipeline that begins with data preprocessing, followed by feature selection using Fisher Score, SHAP, and LIME, and then proceeds to train multiple classifiers including SVM, KNN, and Neural Networks. This approach emphasizes the quantitative contribution of each feature in prediction and follows a structured cross-validation and evaluation process. The second framework shown in Figure 2 is the XAI-based framework categorizing ML models by interpretability and feature selection approach. ML classifiers are trained on the most significant features to improve performance, and feature extraction techniques are used to extract significant features. Each feature's impact on performing prediction is determined, and significant features are chosen. The important features are used to train the machine learning classifiers used in the research. The outcomes are interpreted following the model's

examination. The dataset is divided into training and testing sets, initially following an 80:20 ratio. However, to enhance robustness and reduce bias, the study employed 10-fold stratified cross-validation instead of relying solely on a single 80:20 split. Stratification ensured class balance in each fold, thereby improving the reliability and statistical validity of the reported results.

3.1 Dataset information

The study's dataset, which has 24 attributes and 195 cases overall, can be accessed online [26]. There are several different biological voice metrics in this dataset. Each row in the table corresponds to one of the 195 voice recordings made by these people as mentioned in the 'name' attribute, with each attribute representing a specific voice measure. The dataset characteristics are of multivariate scope and the attribute characteristics are real in nature. The task associated with the dataset is to perform classification. The 'status' column, which is set to 0 for healthy and 1 for PD, serves as the primary means of differentiating between individuals without PD and those with it. The information is in comma-separated values (CSV) format. Each row in the CSV file represents a single voice recording occurrence. Each patient has about six recordings. For the purpose of experimentation, a Jupyter Notebook version 6.1.4 is utilized. Table 2 displays the sample dataset, and Table 3 displays the attribute information for the sample dataset.

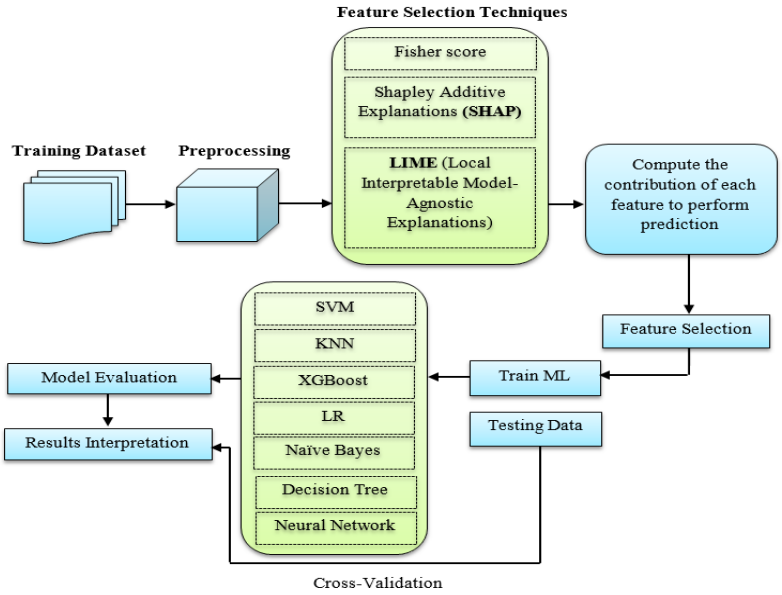


Figure 1: Workflow of ML classification with integrated feature selection techniques and evaluation

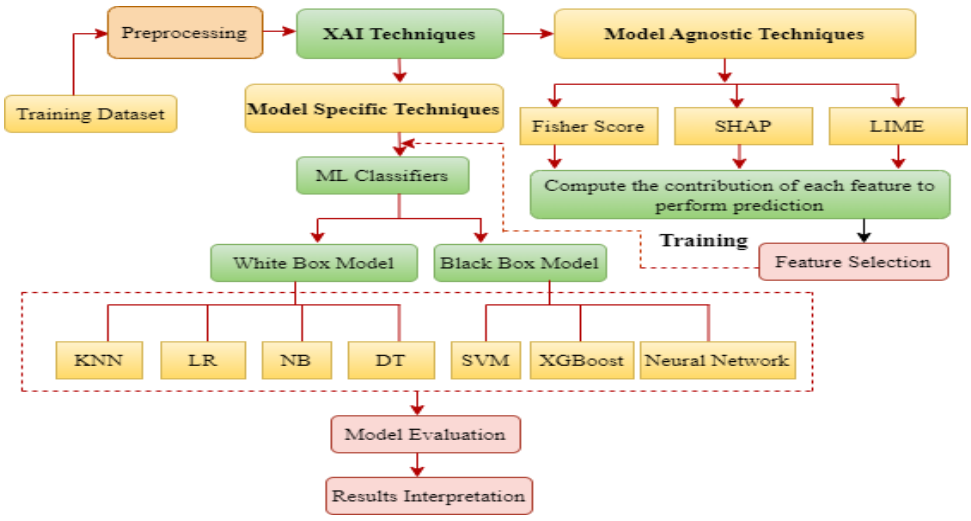


Figure 2: XAI-based framework categorizing ML models by interpretability and feature selection approach

Table 2: Sample Dataset of PD

name	MDVP:Fo(Hz)	MDVP:Fhi(Hz)	MDVP:Flo(Hz)	MDVP:Jitter(%)	MDVP:Jitter(Abs)	MDVP:RAP	MDVP:PPQ	Jitter:DDP	MDVP:Shimmer	MDVP:Shimmer(d	Shimmer:APQ3	Shimmer:APQ5	MDVP:APQ	Shimmer:DDA	NHR	HNR	status	RPDE	DFA	spread1	spread2	D2	PPE
phon_R01_S01	119.99	157.3	74.997	0.0078	7E-05	0.004	0.0055	0.011	0.044	0.426	0.0218	0.0313	0.02971	0.0655	0.022	21.03	1	0.4148	0.8153	-4.81303	0.26648	2.3014	0.2847
phon_R01_S01	122.4	148.65	113.82	0.0097	8E-05	0.005	0.007	0.014	0.061	0.626	0.0313	0.0452	0.04368	0.094	0.019	19.09	1	0.4584	0.8195	-4.07519	0.33559	2.4869	0.3687
phon_R01_S01	116.68	131.11	111.56	0.0105	9E-05	0.005	0.0078	0.016	0.052	0.482	0.0276	0.0386	0.0359	0.0827	0.013	20.65	1	0.4299	0.8253	-4.44318	0.31117	2.3423	0.3326
phon_R01_S01	116.68	137.87	111.37	0.01	9E-05	0.005	0.007	0.015	0.055	0.517	0.0292	0.0401	0.03772	0.0877	0.014	20.64	1	0.435	0.8192	-4.1175	0.33415	2.4056	0.369
phon_R01_S01	116.01	141.78	110.66	0.0128	0.0001	0.007	0.0091	0.02	0.064	0.584	0.0349	0.0483	0.04465	0.1047	0.018	19.65	1	0.4174	0.8235	-3.74779	0.23451	2.3322	0.4103
phon_R01_S01	120.55	131.16	113.79	0.0097	8E-05	0.005	0.0075	0.014	0.047	0.456	0.0233	0.0353	0.03243	0.0699	0.012	21.38	1	0.4156	0.8251	-4.24287	0.29911	2.1876	0.3578
phon_R01_S02	120.27	137.24	114.82	0.0033	3E-05	0.002	0.002	0.005	0.016	0.14	0.0078	0.0094	0.01351	0.0234	0.006	24.89	1	0.596	0.7641	-5.63432	0.25768	1.8548	0.2118
phon_R01_S02	107.33	113.84	104.32	0.0029	3E-05	0.001	0.0018	0.004	0.016	0.134	0.0083	0.0095	0.01256	0.0249	0.003	26.89	1	0.6374	0.7633	-6.1676	0.18372	2.0647	0.1638

Table 3: Attribute Information in a dataset

Attribute	**Description of measurement methods applied to acoustic signals recorded from each subject**
name	ASCII subject name and recording number
MDVP:Fo(Hz)	Average vocal fundamental frequency
MDVP:Fhi(Hz)	Maximum vocal fundamental frequency
MDVP:Flo(Hz)	Minimum vocal fundamental frequency
MDVP:Jitter(%),MDVP:Jitter(Abs),MDVP:RAP, MDVP:PPQ,Jitter:DDP	Multiple indicators of fundamental frequency fluctuation
MDVP:Shimmer,MDVP:Shimmer(dB),Shimmer:APQ3,Shimmer:APQ5, MDVP:APQ,Shimmer:DDA	Multiple amplitude variation measurements
NHR,HNR	Two measures of ratio of noise to tonal components in the voice
RPDE,D2	Two nonlinear dynamical complexity measures
DFA	Signal fractal scaling exponent
spread1,spread2,PPE	Three nonlinear measures of fundamental frequency variation
status	Health status of the subject (one) - Parkinson's, (zero) - healthy

Note: MDVP stands for (Kay Pentax) Multi-Dimensional Voice Program; PPE stands for Pitch period entropy, RPDE stands for Recurrence Period Density Entropy, DFA stands for Detrended Fluctuation Analysis

3.1.1 Data preprocessing and feature selection techniques

An essential phase in machine learning activity is pre-processing. In this process, the dataset is cleaned by handling missing values, outlier removal, and normalization/standardization of features. This step is crucial to ensure the data is in a consistent and suitable format for ML algorithms. To eliminate outliers from the training dataset, one-hot encoding techniques are employed to convert the non-numerical data as a part of pre-processing techniques. The features are selected and relevant features are transformed from the raw data. The features and labels are separated, and fisher scores are calculated to determine the importance of each feature. The features are scaled using standard scale method to ensure all features have the same scale. The proposed study uses several strategies for improving the interpretability and understanding of model predictions in machine learning, including Fisher Score, Shapley values, and LIME. The discriminative ability of features in classification problems is measured using the fisher score, which is then utilised for feature selection. Shapley values are based on ideas from cooperative game theory and offer a means of attributing the model's predictions to specific traits. LIME approximates individual predictions of complicated models with more easily understood models close to the prediction of interest, producing local explanations for each prediction.

3.2 XAI tools

XAI tools are pivotal in closing the knowledge divide between sophisticated ML systems and human comprehension. The increasing use of data-driven algorithms in sensitive areas like healthcare, education, and finance has led to a substantial rise in the need for transparency and accountability. XAI tools are designed to make the decision-making process of models more understandable by showing the impact of input features, explaining the reasoning behind predictions, and allowing users to verify or dispute the results. These tools differ in their approach, with some designed to function independently of any specific model, whereas others are customized for particular types of algorithms. Improving model transparency through the use of XAI tools enables not only increased user trust but also facilitates model debugging, bias identification, and adherence to ethical or regulatory requirements.

3.2.1 Model-Agnostic techniques

Interpretable ML systems require model-agnostic tools, particularly when dealing with complex or black-box models whose inner workings are unclear. These tools are designed to function autonomously of the underlying algorithm, enabling them to be highly

adaptable across various types of classifiers and datasets. By focusing solely on the relationship between input features and model outputs, they enable researchers and practitioners to extract meaningful insights without needing access to the inner workings of the model. This versatility enables the use of model-agnostic methods across a wide range of tasks including feature selection, explaining individual predictions, and assessing overall model performance, ultimately increasing transparency and trust in data-driven decision-making processes.

3.2.1.1 Fisher discriminant analysis

Fisher's score function known as fisher discriminant analysis is deeply related to maximum likelihood estimation as mentioned in (1). Fisher's score to find the maximum of the likelihood function all along, just without explicitly using the term. Assume some dataset X where each observation is identically and independently distributed according to a true underlying distribution parameterized by θ. Given this probability density function $f\theta(x)$, the likelihood function as follows:

$$p(x|\theta) = \prod_{i=1}^{n} f_\theta(x_i) \quad (1)$$

The maximum likelihood estimate of the distribution's parameter is given by (2).

$$\theta_{mle} = arg_\theta max\, \mathrm{p(x|\theta)} \quad (2)$$

$$= arg_\theta max \log p\,(\mathrm{x|\theta}) \quad (3)$$

$$= arg_\theta \max \sum_{i=1}^{n} log f_\theta\ (x_i) \quad (4)$$

The gradient of log likelihood function is shown below:

$$u(\theta) = \nabla_\theta log p(x|\theta) \quad (5)$$

The feature importance i.e. Fisher Score obtained from the PD dataset is shown in Figure 3. In model selection and training phase, ML classifiers for the task are taken, such as SVM, KNN, XGBoost, LR, NN, DT and NB. The dataset is split into training and testing subsets for training a model. The training data is used to train the ML models and optimize their hyperparameters. Apart from it, the cross-validation techniques are also being used to avoid overfitting and assess model generalization. In performance evaluation, the trained models are evaluated using appropriate metrics like accuracy, precision, recall, F1 score, or area under the receiver operating characteristic curve (AUC-ROC). The performance of different models is compared to identify the best-performing one. Since Parkinson's disease diagnosis and management require interpretability, it's essential to explore methods to explain the models' decisions. Therefore, the techniques like SHAP, LIME are used in the work to help and provide insights into the features that significantly influence the model's predictions. The bar chart in Figure 3 shows the importance of features calculated using the Fisher Score. The y-axis lists the feature names, and the x-axis represents the Fisher Score values. Features like PPE, MDVP:Fo(Hz), and spread1 have the highest Fisher Scores, indicating they contribute most to class separation in the dataset.

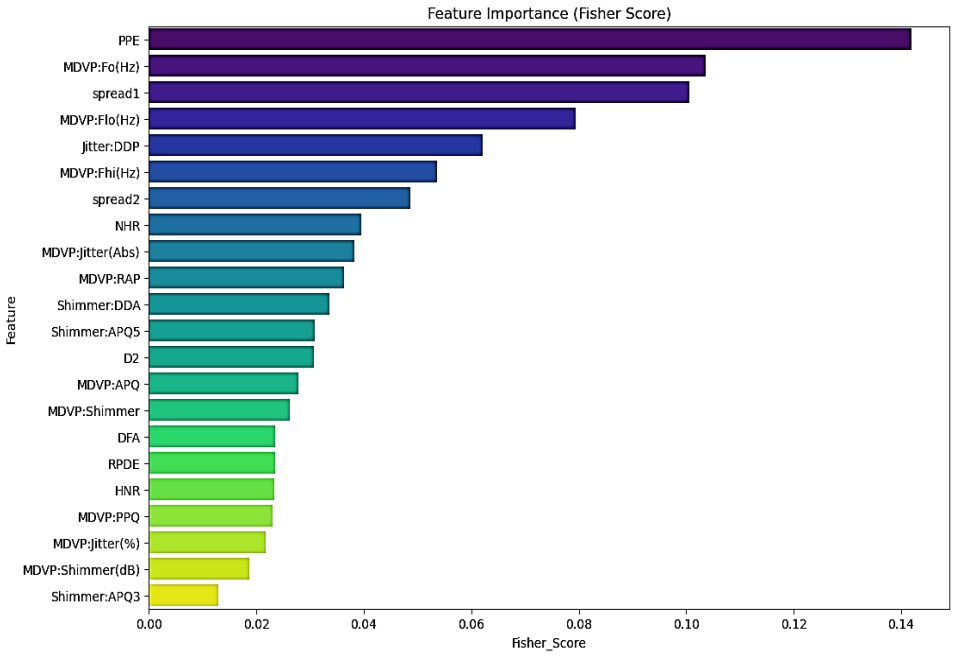


Figure 3: Feature importance using fisher score

3.2.1.2 Shapley additive explanations

Shapley values are a useful tool for explaining ML model outcomes and can be used to explain feature contributions. It's a notion from game theory that utilized to calculate each group member's participation. Assuming that features are the players or team members and the model prediction is the game's or teamwork's contribution. For every iteration, the feature values for all features are drawn in an arbitrary sequence. The difference between the prediction with and without feature i to determine the significance of feature i is computed. Calculating the mean difference across every possibility yields the Shapley value. It is the feature's

average marginal impact when all possible combinations are taken into account. It is only possible to determine the Shapley value with a subset of combinations. Assuming that a group t_m consists of n_m members, and that each member contributes significantly to the work during the group time frame, the teamwork results in the total value $tot_{val} = tot_{val}(t_m)$. The outcome for each member of the group m is represented by the shapley value, $\varphi_m\,(tot_{val})$. The definition of $\varphi_m\,(tot_{val})$ is

$$\varphi_m\,(tot_{val}) = \frac{1}{n_m}\sum_{s_{subset}} \frac{[tot_{val}(s_{subset}\cup\{m\}) - tot_{val}(s_{subset})]}{\binom{n_m-1}{K(s_{subset})}}, m = 1,2,3,\ldots..n_m \quad (6)$$

The sum of the values for a particular member,m is the total of all the subsets s_{subset} of the group $t_m = \{1,2,3,\ldots..,n_m\}$, that can be formed once m is eliminated. The size of a subteam is denoted by $K(s_{subset})$, its accomplished value is $val(s_{subset})$, and its actual value once m joins s_{subset} is $val(s_{subset} \cup \{m\})$.The appropriate collection is defined by the Shapley value as a distribution that meets requirements for linearity, symmetry, and efficiency. The plot in Figure 4 visualizes the impact of each feature on the XGBoost model output using SHAP values. The x-axis shows SHAP values (impact on prediction), and the y-axis lists the features. Colors indicate feature value (red = high, blue = low. Features like PPE and spread1 have the largest range of SHAP values, meaning changes in these features have the most influence on model predictions.

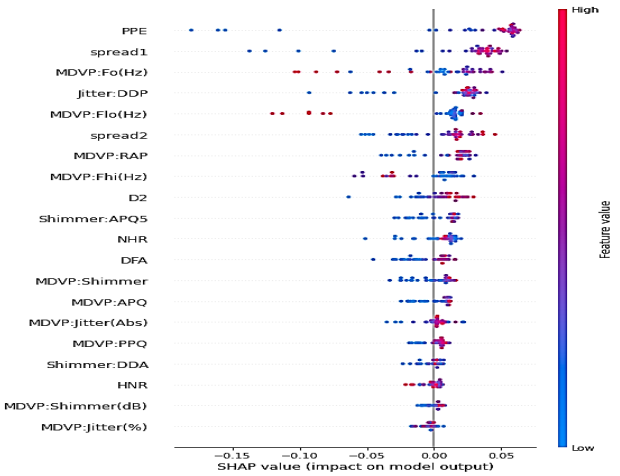


Figure 4: SHAP value impact on model output

The SHAP value and its impact on model output from low to high is shown in Figure 4.The mean SHAP value i.e. average impact on model output magnitude is shown in Figure 5. It shows the average absolute SHAP values for each feature, separated by class. The x-axis represents mean SHAP values, and the colors indicate Class 0 (blue) and Class 1 (red). PPE, spread1, and MDVP:Fo(Hz) are the most influential features for distinguishing between the two classes. Some features have higher impact on one class than the other. Figure 6 displays the beeswarm plot for feature relevance in Parkinson's disease prediction. It is inferred from Figure 4, Figure 5, and Figure 6, where the most significant features are 'spread1', 'MDVP:Fo (Hz)', and 'PPE' among other features.

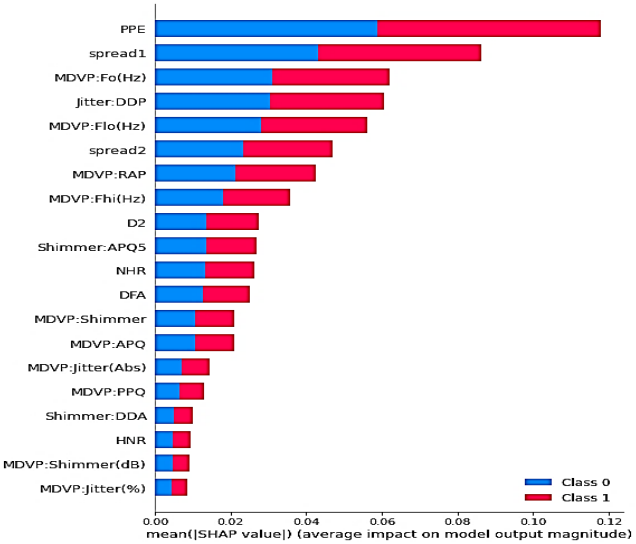


Figure 5: Mean SHAP value

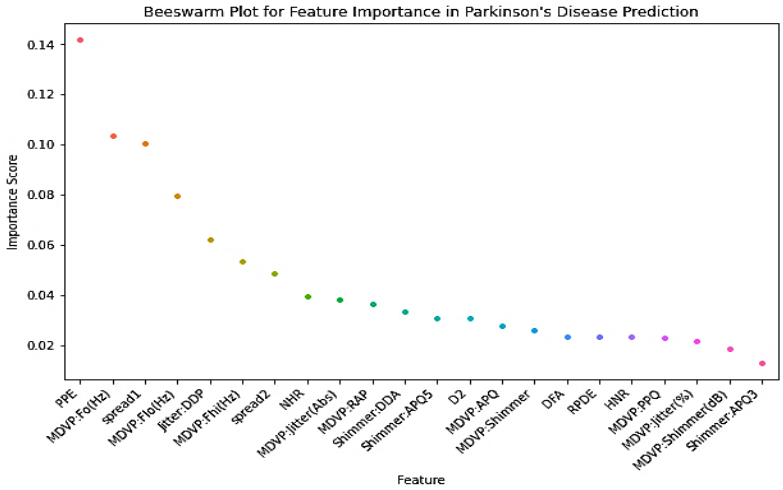


Figure 6: Beeswarm plot for feature importance in PD prediction

A force-directed visualisation that breaks down the model's prediction into components attributed to each feature is displayed in Figure 7 is shape force plot. This visualisation shows how each feature contributes to pushing the model's output from a base value, the average model output to the predicted value for the particular instance. The base value of the model's output, known as the mean prediction, is shown by the horizontal line in the center of the plot. It serves as a starting point before taking into account the influence of any features. A corresponding bar on the plot corresponds to each feature in the instance that is being explained. Each bar's length indicates how that feature affects the model's forecast. Positive values are represented by the bars pointing right, indicating that the value of the feature raises the output of the model, while negative values are represented by the bars pointing left, indicating a drop in the output of the model. In addition to the global feature ranking shown in the beeswarm plot in Figure 6, local interpretability is explored using the SHAP force plot in Figure 7. The beeswarm visualization highlights that PPE**,** MDVP:Fo(Hz)**,** Spread1**,** and MDVP:Fhi(Hz) are the most influential predictors of Parkinson's disease. To further interpret individual cases, the force plot demonstrates how specific feature values drive predictions: for instance, high PPE**,** low spread1, and elevated MDVP:Fhi(Hz) values collectively push the model output toward the Parkinson's (PD) class, whereas higher MDVP:Fo(Hz) values contribute toward the healthy prediction. These local explanations confirm that the classifier's decisions align with clinically recognized voice impairments in PD, thereby enhancing the model's interpretability and practical relevance.

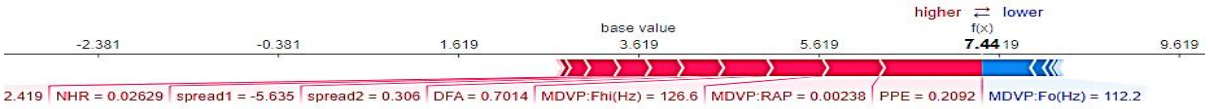


Figure 7: Shapley force-directed plot of models prediction attributed to each feature

3.2.1.3 LIME

Local Interpretable Model-Agnostic Explanations are referred to as LIME. It is a ML technique that approximates the behavior of black-box models locally to explain the predictions made by the models. It highlights the most significant characteristics and offers insights into why a ML model produces a particular prediction for a particular instance. Rather than explaining the model's overall behavior, LIME focuses on producing local explanations for specific predictions. It produces understandable approximations of the complex model behavior based on a particular occurrence.

$$rbf(y^{(j)}) = exp(-\frac{\|y^j - y^{(ref)}\|^2}{kw}) \quad (7)$$

LIME uses a Gaussian kernel, or radial basis function, to assign a weight to each generated point. Since it's a local method, it doesn't care about distant methods and ignores them. The Gaussian Kernel gives a value in the interval [0, 1] to an equation; the greater the value, the closer the equation is to the reference point. The value of the kernel width 'kw' parameter determines the diameter of the meaningful weights' circle surrounding the reference point $'ref'$. Figure 8 displays a LIME plot as a bar chart that illustrates how much the value of a feature changes and how much the anticipated value changes as well. It highlights the most important features and how they affect the forecast, visualising the local interpretability of an ML model's prediction for a particular instance.

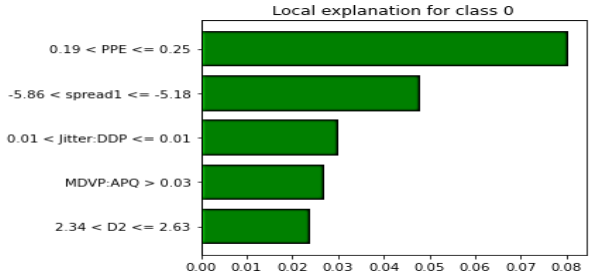


Figure 8: Local explanation of a model prediction using LIME

3.2.2 Model specific techniques

Model-specific techniques are explainability methods tailored to the internal structure and behavior of particular machine learning algorithms. Unlike model-agnostic approaches, these techniques leverage the mathematical and logical characteristics unique to a specific model type to provide deeper and more accurate insights into how predictions are made. For instance, decision trees naturally offer interpretability through their hierarchical structure of feature-based splits, while linear models provide direct access to feature weights that indicate their influence on the output. In neural networks, techniques such as saliency maps or layer-wise relevance propagation are used to trace the contribution of input features through the network's layers. By focusing on the internal mechanisms of each model, these techniques often yield more precise and informative explanations, although they are limited in scope to the models they are designed to interpret.

classifiers can help determine whether Parkinson's illness will develop or not. By differentiating PD from other neurological disorders that present with similar symptoms, they can help in the process of diagnosis, resulting in faster and more accurate diagnoses. To improve the effectiveness of classifiers for tasks involving the categorization of Parkinson's disease, feature engineering and selection are essential. Using the Shapley, LIME and Fisher score algorithms on the dataset, the salient features are retrieved.

Logistic regression

The probability of a certain class or occurrence can be modeled using the supervised machine learning approach known as logistic regression. When the outcome is binary or dichotomous in nature and the data is linearly separable, it is employed. Thus, situations involving binary classification typically employ logistic regression. Predicting an output variable that alternates between two discrete classes is known as binary classification.

Naïve Bayes

The Naive Bayes classification technique relies on the assumption that all features that predict the target value are independent of one another, and it is based on the bayes theorem. It determines each class's probability before selecting the one with the highest likelihood. The Bayes theorem as shown in (8) asserts that, given a class variable y and a features vector X= $(x_1, x_2, \ldots, x_n)$.

$$p(y|X) = \frac{P(X|y)*P(y)}{P(X)} \quad (8)$$

The probability $P(X \mid y)$ can be broken down as shown in (9):

$$P(X|y) = P(x_1|y) * P(x_2|y) \ldots\ldots\ldots P(x_n|y) \quad (9)$$

3.2.2.1 White box models

White-box machine learning models are characterized by their high degree of interpretability and transparency, making them particularly suitable for applications where understanding the reasoning behind a prediction is essential. These models allow researchers and practitioners to examine the internal structure, such as decision boundaries, feature weights, or probability distributions, to gain insights into how inputs are transformed into outputs. Unlike more complex black-box models, white-box approaches do not obscure their logic, which fosters trust and supports accountability in sensitive domains like healthcare. In the context of Parkinson's disease diagnosis, white-box models offer the added benefit of enabling clinicians to verify and understand the diagnostic criteria identified by the algorithm, thereby bridging the gap between computational predictions and medical decision-making. Their simplicity, coupled with ease of validation, makes them a valuable choice when explainability is a priority. Using diagnostics or early symptoms, ML

Decision Tree

A supervised ML approach based on the nested-if classifier is the decision tree. It creates models for regression or classification using a tree topology. The quantity of data required to precisely characterize a sample is known as its entropy, and it is computed as (10).

$$entropy = -\sum_{x=1}^{n} p_x * \log(p_x) \quad (10)$$

In this case, the fraction of examples in class x is denoted by p_x. Entropy equals 1 when classes are evenly distributed, and it equals 0 when one class is being most prevalent. Gini impurity is computed as follows:

$$gi = 1 - \sum_{x=1}^{n} (p_i)^2 \quad (11)$$

Finding a characteristic that yields the maximum information gain is the primary step in building a decision tree, as indicated by (12).

$$I_f G_n(d_{pnode}, attrib) = I_f(d_{pnode}) - \frac{N_{sleftchildnode}}{N} I_f(d_{leftchildnode}) - \frac{N_{srhtchildnode}}{N} I_f(d_{rgtchildnode}) \quad (12)$$

In equation above,$I_f G_n$ is the information gain, d_{pnode} is the dataset of the parent node. $d_{leftchildnode}$ is the dataset of left child node and $d_{rgtchildnode}$ is the dataset of right child node. $N_{sleftchildnode}$ represents the number of samples at left child node and $N_{srhtchildnode}$ is the number of samples at right child node.

KNN

K-Nearest Neighbor (KNN) is a straightforward method that identifies new data points according to the majority class of their k closest neighbors. Both classification and regression tasks can be accomplished with it. KNN presupposes that related instances frequently have related labels and targets. For distance metrics, the euclidean metric as shown in (13) is used:

$$d(x, \acute{x}) = \sqrt{(x_1 - \acute{x}_1)^2 + \cdots + (x_n - \acute{x}_n)^2} \quad (13)$$

Finally, the input x gets assigned to the class with the largest probability in (14).

$$P\ (y = j\ |X = x) = \frac{1}{K} \sum_{i \in A} I(y^{(i)} = j) \quad (14)$$

KNN measures the similarity between instances using distance metrics. Manhattan distance and Minkowski distance are two additional distance metrics; however euclidean distance is the one that is most frequently employed.

3.2.2.2 Black box models

Black-box machine learning models are known for their ability to capture complex, non-linear patterns within data, often achieving high predictive accuracy. However, what sets them apart is their limited interpretability; the internal processes and decision rules they employ are typically not transparent or easily understood by humans. These models, such as neural networks, support vector machines, and ensemble methods like XGBoost, consist of intricate layers, mathematical transformations, or combinations of multiple weak learners, making it challenging to trace how a specific output is derived from the input features. In medical applications like Parkinson's disease classification, black-box models can uncover subtle patterns in high-dimensional data that may not be detected by simpler algorithms. While their predictive power is valuable, the opaque nature of these models necessitates the use of explainable AI techniques to enhance trust, support clinical validation, and ensure that decisions align with medical understanding.

SVM

SVM is a potent supervised machine learning technique used for both regression and classification applications. It is extensively used because it can manage complicated datasets and deliver precise results. SVM are very useful when the data cannot be separated linearly and requires intricate decision boundaries. SVM algorithm looks for the best hyperplane to divide the data points into distinct groups. A hyperplane is a linear decision boundary that divides the data into two classes in binary classification. In multi-class classification, various hyperplanes are employed to divide various classes. With n training points, each observation i (x_i) has p dimensions and p characteristics, meaning it belongs to one of two classes ($y_i = 1 \; or \; y_i = -1$). Assume that two linearly separable classes of observations exist. This means that a feature space can be used to construct a hyperplane with all instances of one class on one side and all instances of the other class on the other. A hyperplane is defined in (15) as

$$x.\tilde{w} + \tilde{b} = 0 \tag{15}$$

Here, $\tilde{w}$ is a p vector and $\tilde{b}$ is a real number. When $\tilde{w}$=1, so the quantity $x.\tilde{w} + \tilde{b} = 0$ is the distance from point x to the hyperplane. Thus we can label our classes with $y = +1/-1$, and the requirement that the hyperplane divides the classes becomes:

$$y_i(x_i.\tilde{w} + \tilde{b}) \geq 0 \tag{16}$$

We choose the plane that result in the largest margin M between the two classes, which is called the Maximal margin classifier. Mathematically, we choose $\tilde{b}$ and $\tilde{w}$ to maximize M, given the constraints:

$$y_i(x_i.\tilde{w} + \tilde{b}) \geq M \tag{17}$$

XGBoost

The popular machine learning method known as Extreme Gradient Boosting, or XGBoost, performs exceptionally well when handling structured data. Utilizing parallel computation and tree-based models, it is an enhanced gradient-boosting approach that yields superior performance and accurate predictions in various ML applications. For classification problems, XGBoost is specifically implemented as the XGBClassifier. XGBoost provides methods for both regression and classification. The gradient-boosting technique is the cornerstone of XGBoost. It integrates multiple weak predictive models often called decision trees into an ensemble model. It builds the ensemble iteratively, adding new models to address the shortcomings in the previous ones to minimize a loss function.

$$\hat{y_i} = \sum_{k=1}^{K} f_k(x_i), f_k \epsilon \, F \tag{18}$$

where F is the set of potential CARTs, K is the number of trees, and f is the functional space of F. For the aforementioned paradigm, the objective function is provided by (19):

$$obj(\theta) = \sum_i^n l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \cap (f_k) \tag{19}$$

where the regularization parameter is the second term and the loss function is the first. Currently, we apply the additive technique, minimizing the loss of what we have learned, and add a new tree, which may be characterized as follows, in place of learning the tree all at once, which makes the optimization harder as shown in (20).

$$\hat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{20}$$

The objective function of the above model can be defined as:

$$obj^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i{}^{(t)} + \sum_{i=1}^{t} \cap (f_i) \tag{21}$$

Tree-based models, such as decision trees, are the primary base learners that XGBoost uses. The decision trees constructed in a sequential fashion address the shortcomings of their predecessors. The ultimate forecast is obtained by adding together all of the tree projections. Regularisation techniques are used by XGBoost to decrease overfitting and improve generalisation. Finding the features in the dataset with the most influence is made easier with the help of XGBoost's feature significance metric. Cross-validation techniques are supported by XGBoost to assess model performance and adjust hyperparameters. It aids in determining the hyperparameters that are best for the model's performance and generalizability.

Neural Network

A neural network classifier is a kind of machine learning model that divides inputs into different types or classes using an artificial neural network. With regard to supervised learning tasks, neural network classifiers are specifically engineered to learn a mapping from input features to output labels based on labelled training data. To sum up all the multiplied values, multiply each input value (y_i) by the weights (w_i). Weights determine how much an input may impact a neuron's output and show the strength of the relationship between neurons. The input y_1 will have a bigger impact on the output than w_2 if the weight w_1 is greater than the weight w_2.

$$\sum = (y_1 * w_1) + (y_2 * w_2) + \cdots + (y_n * w_n) \quad (22)$$

The row vectors of the inputs and weights are $y = [y_1, y_2, \ldots\ldots, y_n]$ and $w = [w_1, w_2, \ldots\ldots, w_n]$ respectively and their dot product is shown in (23):

$$y.w = (y_1 * w_1) + (y_2 * w_2) + \cdots + (y_n * w_n) \quad (23)$$

As a result, the summation is equal to the dot product of the vectors y and w. $\sum = y.w$ (24)

In order to shift the entire activation function either to the left or right and get the necessary output values, bias is added in Eq. 25.

$$z = y.w + b \quad (25)$$

The value of z is passed to a non-linear activation function. Without activation functions, the output of the neurons would only be linear, and this is why activation functions are employed to add non-linearity to NN. Additionally, they significantly affect the NN rate of learning. Here, sigmoid (σ) also known as logistic function work as the activation function and the result we get after the forward prorogation is known as the predicted value p̂ shown as:

$$\hat{p} = \sigma(z) = \frac{1}{1+e^{-z}} \quad (26)$$

4 Results and discussions

The decision tree classifier trained with the Parkinson dataset is shown graphically in Figure 9. The way the classifier uses multiple factors to inform its judgments and separate cases into Parkinson's and non-Parkinson's classes are demonstrated by the tree structure. The highest node in the tree is where the decision-making process starts. It shows the feature and threshold that are used by the model in its first split. The feature and threshold that are used to further split the data are displayed on each node. The terminal nodes at the base of the tree reflect the final projected classes or outcomes. Every leaf node shows the expected class based on the majority of training examples that reach that node. The important attributes are visibly highlighted in the decision tree plot according to how much they influence the model's decisions. Usually, characteristics nearer the root of the tree are more significant for classification. In figure, a root node represents the initial split based on a specific feature i.e. PPE and the split condition MDVP:Fo(Hz) <= 192.273 shows the threshold value used.The main features that are derived using the Fisher score, Shapley and LIME techniques are 'PPE', 'MDVP:Fo(Hz)','spread1', 'MDVP:Flo(Hz)', and 'Jitter:DDP'. Boxplots are used in Figure 10 to forecast the state of PD patients. These plots can provide light on potential variations in specific aspects of the target variable. The distributional patterns indicate whether a certain feature is discriminative for Parkinson's disease prediction or not. Boxplots, sometimes referred to as box whisker plots, are a kind of visual illustration that is used to show the significant statistical characteristics of a dataset and summarize its distribution. When it comes to visualizing the pattern of distribution and overall trend of numerical data, they are particularly useful. The correlation between every measure of significant features is examined in a pair plot. In exploratory data analysis (EDA), a pair plot, often called a scatterplot matrix, is a useful visualisation method that provides several advantages for comprehending relationships within a dataset. The patterns and trends in the data are identified by examining the scatterplots in a pair plot as shown in Figure 11.

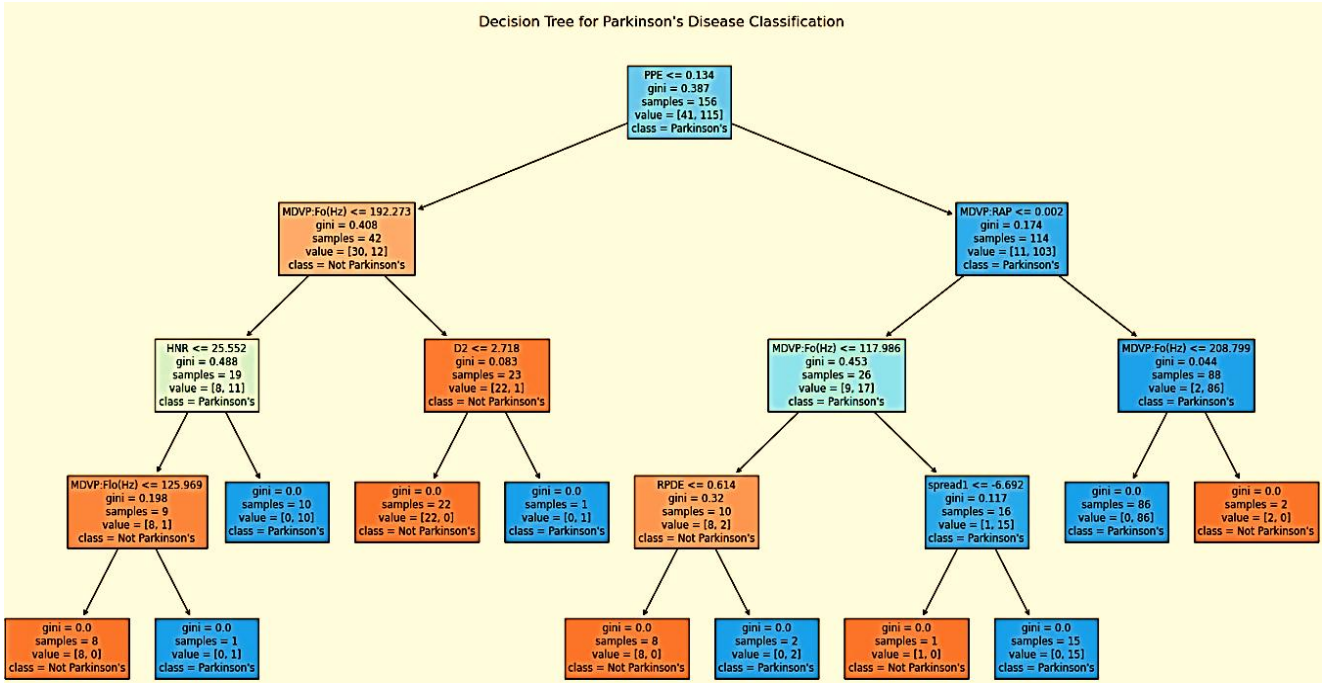


Figure 9: Decision tree for Parkinson's disease classification

An essential tool in machine learning, a confusion matrix is used to assess a classification model's performance. It aids in your understanding of the model's performance by summarizing the model's predictions and the actual results. A more thorough knowledge of the model's advantages and disadvantages is offered by the confusion matrix, particularly in cases when there are imbalances between the classes. A summary of the model's performance in terms of properly and erroneously anticipated cases by entering these actual and forecasted values into confusion matrix is shown in Table 4. In this table, the ROC curve values of all the ML classifiers is also displayed. The predicted values and actual values denote the class labels that the model predicts for the examples and the genuine class labels of the instances, respectively. Actual class values are the true class labels of the dataset's occurrences. Positive (1) and negative (0) are the two classes that are normally available in binary categorization. The ground truth or the actual state of the instances is represented by these values. Predicted class values represent the class labels that the machine learning model has estimated for the corresponding dataset instances. Using its acquired patterns and features from the training set, the model predicts a class label.

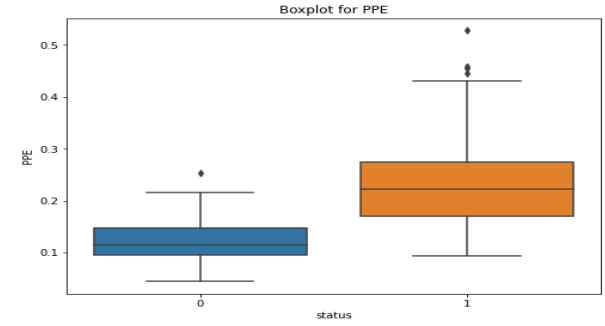
The metrics shown in table 4 provide different perspectives on the performance of a classification model. There are two classes i.e. positive and negative. TP means True Positive and it means the number of instances correctly predicted as positive, FP is False Positive and it represents the number of instances incorrectly predicted as positive. False Negative (FN) is the number of instances incorrectly predicted as negative and TN i.e. True Negative is the number of instances correctly predictive as negative.

A performance metrics for each classifier is shown in Table 5. On seeing the performance metrics of ML classifiers, the accuracy achieved by XGBoost is 0.94 followed with DT i.e. 0.9231.

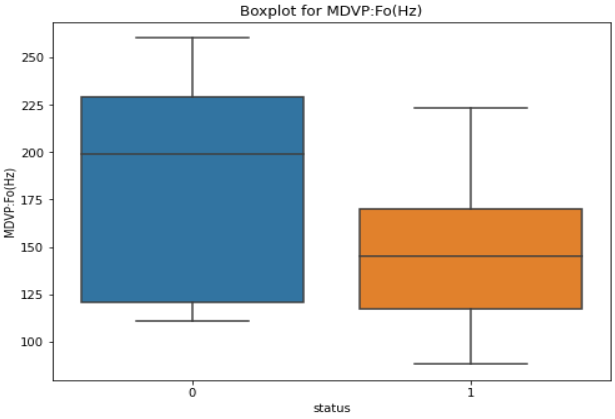
4.1 Feature selection and ablation study

To analyze the individual contribution of each feature selection method, an ablation study is conducted using Fisher Score, SHAP, and LIME separately with all classifiers. When features selected through the Fisher Score are used, classifiers such as Logistic Regression and Decision Tree showed a moderate improvement in accuracy i.e. around 3 to 5%, indicating that statistical ranking helped emphasize discriminative attributes. Using SHAP-based feature importance further improved the interpretability and produced the best results for XGBoost and Decision Tree, where the accuracy reached 0.94–0.95 and recall remained at 1.0, suggesting that SHAP-selected features strongly contributed to generalization. In contrast, LIME-based feature selection maintained accuracy stability (≈0.90) but offered higher local interpretability for instance-level predictions. Overall, combining the top-ranked features from Fisher Score, SHAP, and LIME achieved the best performance (XGBoost accuracy = 0.9487, recall = 1.0), confirming that integrating global (Fisher, SHAP) and local (LIME) interpretability methods enhances both predictive accuracy and model explainability. To summarize the ablation results clearly, Table 6 presents the comparative performance of classifiers under each individual and combined feature selection technique. The table

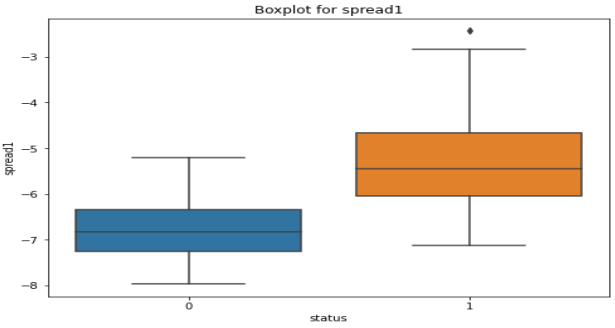
highlights that SHAP and the combined approach yielded the highest performance and interpretability balance.



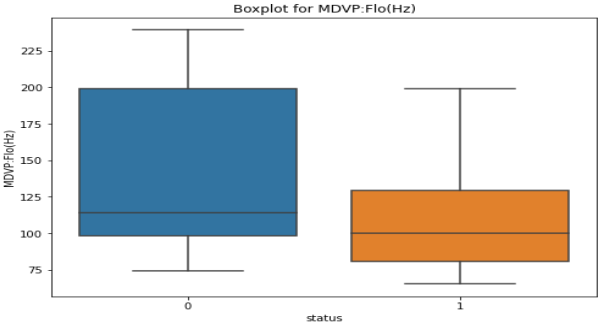
(a)



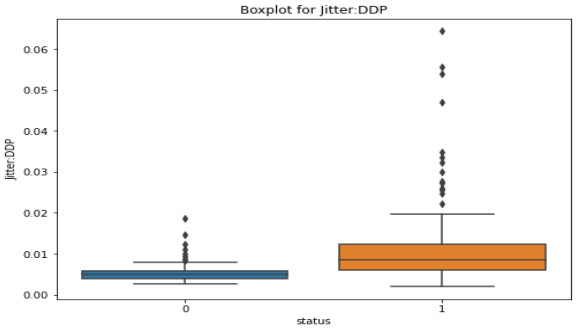
(b)



(c)



(d)



(e)

Figure 10: Boxplot illustrating the PD prediction for important features of acoustic signals a) PPE, b) MDVP:Fo(Hz), c) spread1, d) 'MDVP:Flo(Hz), and e) Jitter:DDP

The overall correctness of the model is known as accuracy whereas precision is the proportion of predicted positives that are actually positive. The performance metrics provide a comprehensive view of a classifiers performance. XGBoost classifier has correctly classified the instances as true positive. For the XGBoost classifier, hyperparameters including learning rate = 0.1, max depth = 5, n_estimators = 100, subsample = 0.8, and colsample_bytree = 0.8 are selected. These hyperparameters are optimized using grid search over a predefined parameter grid with 10-fold cross-validation to ensure robustness and maximize the F1-score.TPR also known as Recall value or Sensitivity is defined as:

$$tpr = \frac{tp}{tp+fn} \quad (27)$$

FPR is defined as $fpr = \frac{fp}{fp+tn}$ (28)

TNR is also known as Specificity $tnr = \frac{tn}{tn+fp}$ (29)

FNR is defined as $fnr = \frac{fn}{tp+fn}$ (30)

Precision is also known as Positive Predictive Value

$$precision = \frac{tp}{tp+fp} \quad (31)$$

F1-Score is $f1 = 2.\frac{precision.recall}{precision+recall}$ (32)

Accuracy is defined as $accuracy = \frac{tp+tn}{tp+tn+fp+fn}$ (33)

The several performance metrics, which reveal where the model may be making mistakes and how well it is working, include accuracy, precision, recall, specificity, and F1 score. XGBoost achieved the highest recall (1.0) and accuracy (0.9487) among all classifiers. Given the small dataset size of 195 samples, there is a potential risk of overfitting. To mitigate this, 10-fold stratified cross-validation is employed, ensuring class balance and evaluating performance across multiple folds. Metrics remained consistent across folds, suggesting that the high recall reflects model generalization rather than overfitting.

4.2 Statistical significance testing

To validate that the observed performance differences among the classifiers are not due to random variation, statistical significance testing is conducted on the accuracy values obtained from ten-fold cross-validation. Both the paired *t*-test and the non-parametric Wilcoxon signed-rank test ($\alpha = 0.05$) are used for pairwise comparisons between classifiers. The results indicated that the black-box model XGBoost achieved significantly higher accuracy (M = 0.9487, SD = 0.02) than the other classifiers ($p < 0.05$). Among the white-box models, Decision Tree and Logistic Regression also showed significantly better performance than Naïve Bayes and KNN ($p < 0.05$). The tests confirm that the superior accuracy and ROC-AUC values of XGBoost (0.91) and Decision Tree (0.84) are statistically significant rather than arising from random chance. These findings strengthen the conclusion that XGBoost (black-box) and Decision Tree (white-box) offer the most reliable classification performance on the dataset.

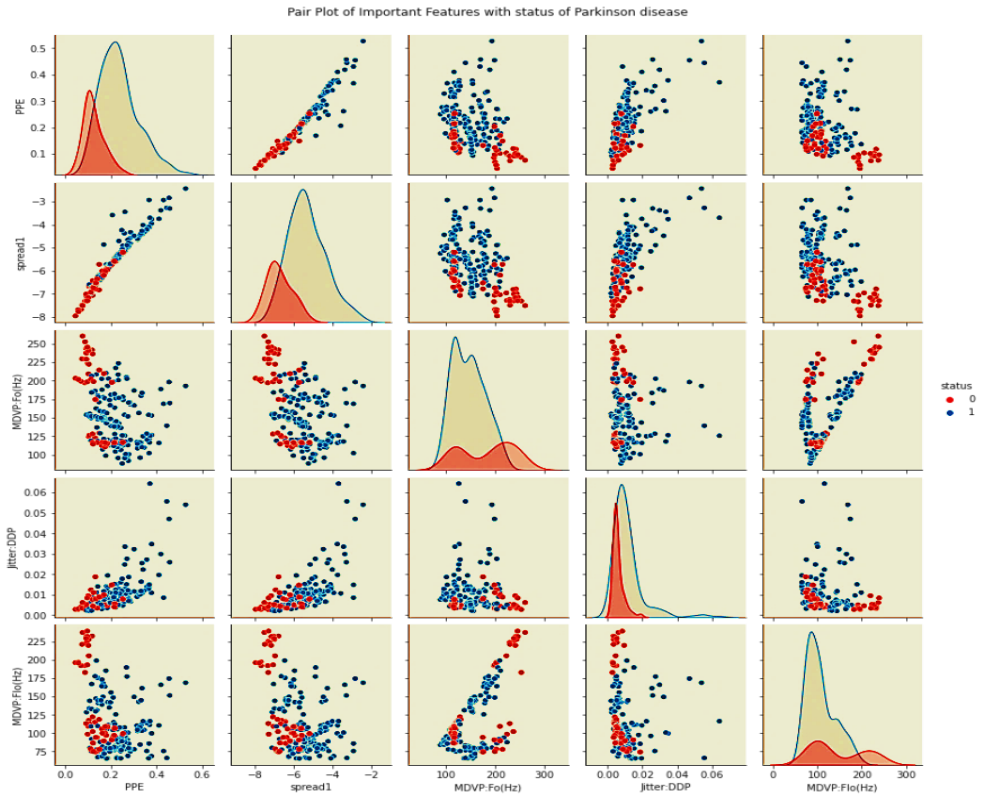


Figure 11:Correlation analysis and identifying patterns using pair plots

Table 4: Confusion metric and ROC curve values achieved by ML classifiers

				Predicted Labels		
ML Classifier	**Model Type**	**Class**		**Predicted 0**	**Predicted 1**	**ROC curve value**
XGBoost	**Black Box Models**	True Labels	Actual 0	5	2	0.91
			Actual 1	0	32	
Neural Network		True Labels	Actual 0	5	2	0.70
			Actual 1	1	31	
SVM		True Labels	Actual 0	2	5	0.68
			Actual 1	1	31	
Naïve Bayes	**White Box Models**	True Labels	Actual 0	5	2	0.81
			Actual 1	10	22	
LR		True Labels	Actual 0	3	4	0.85
			Actual 1	0	32	
Decision Tree		True Labels	Actual 0	5	2	0.84
			Actual 1	1	31	
KNN		True Labels	Actual 0	3	4	0.94
			Actual 1	3	29	

Table 5: Performance metrics of ML classifiers

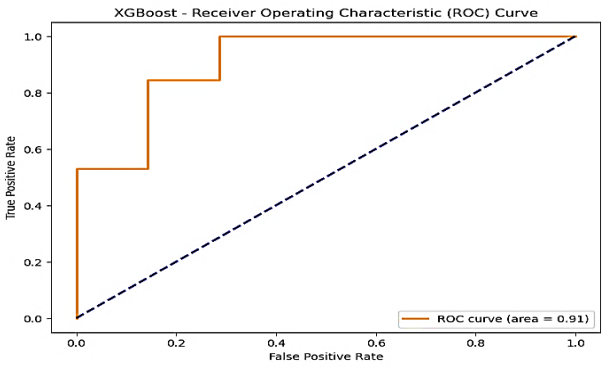
Classifier Name	Model Type	TPR (True Positive Rate)	FPR (False Positive Rate)	TNR (True Negative Rate)	FNR (False Negative Rate)	Precision	F1-Score	Recall	Accuracy
XGBoost	**Black Box Models**	1.0000	0.2857	0.7143	0.0000	0.9412	0.9697	1.0000	0.9487
NN		0.8125	0.2857	0.7142	0.0000	0.9142	0.9552	1.0000	0.9230
SVM		0.9688	0.7143	0.2857	0.0312	0.8611	0.9118	0.9688	0.8462
NB	**White Box Models**	0.6875	0.2857	0.7143	0.3125	0.9167	0.7857	0.6875	0.6923
LR		1.0000	0.5714	0.4286	0.0000	0.8889	0.9412	1.0000	0.8974
DTree		0.9688	0.2857	0.7143	0.0312	0.9394	0.9538	0.9688	0.9231
KNN		0.9062	0.5714	0.4286	0.0938	0.8788	0.8923	0.9062	0.8205

Table 6: Performance comparison of individual and combined feature selection methods

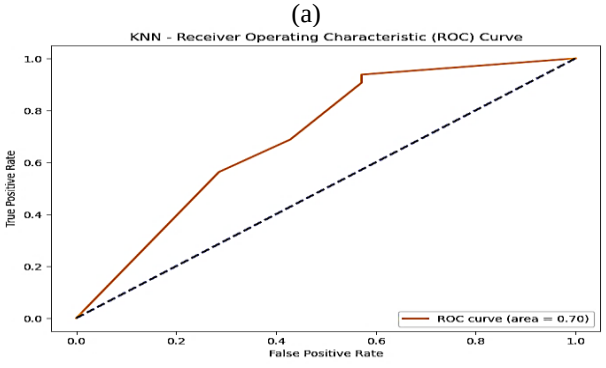
Feature Selection Method	Best Performing Classifier	Accuracy	Recall
Fisher Score only	Decision Tree	0.91	0.96
SHAP only	XGBoost	0.94	1.00
LIME only	Logistic Regression	0.90	1.00
Combined (Fisher, SHAP, LIME)	XGBoost	0.9487	1.00

The higher values of the AUC-ROC value indicate greater performance; the range is 0 to 1. The ROC curves of the ML classifiers are shown in Figure 12. The model performs effectively in differentiating between the classes when the ROC curve interpretation value falls between 0.8 and 0.9. AUC-ROC values above 0.9 signify a high level of classifying ability of the model.

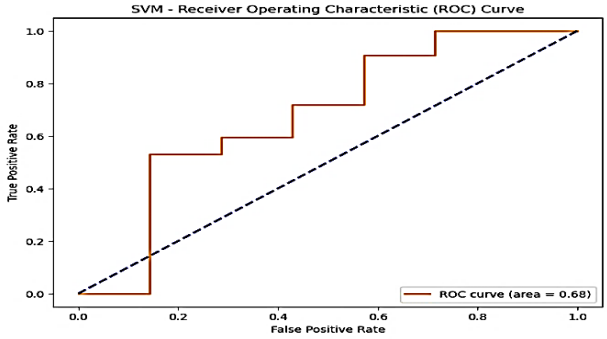
The proposed XGBoost model achieved the best overall performance, with an accuracy of 94.87%, F1-score of 96.97%, and ROC-AUC of 0.91, surpassing both white-box and black-box classifiers evaluated in this study. This improvement is primarily due to XGBoost's ensemble-based gradient boosting mechanism, which effectively combines multiple weak learners and controls overfitting through regularization. The performance advantage also arises from the feature selection process using Fisher Score and SHAP, which refined the dataset by retaining only the most informative voice attributes. While white-box models such as Decision Tree and Logistic Regression offered better interpretability, their predictive accuracy is comparatively lower. In contrast, XGBoost achieved higher accuracy but reduced transparency. To address this trade-off, explainability tools such as SHAP and LIME are employed to interpret individual feature contributions, ensuring model transparency and clinical relevance. In overall, XGBoost provided the best balance between accuracy and interpretability in Parkinson's disease classification. While direct clinician-in-the-loop validation is not performed in this study, interpretability is quantitatively demonstrated through SHAP-based feature ranking and visualization. The explainability performance tradeoff is evident: white-box models such as Decision Tree and Naïve Bayes offered transparent feature-level insights, whereas black-box models like XGBoost achieved superior accuracy (94.87%) with reduced interpretability. However, the integration of SHAP and LIME provided a bridge between these extremes by revealing key predictive features (e.g., PPE, MDVP:Fo(Hz), Spread1) that align with established Parkinson's voice biomarkers. Future work will extend this framework with clinician feedback to assess the practical usefulness of these explanations in medical decision support.



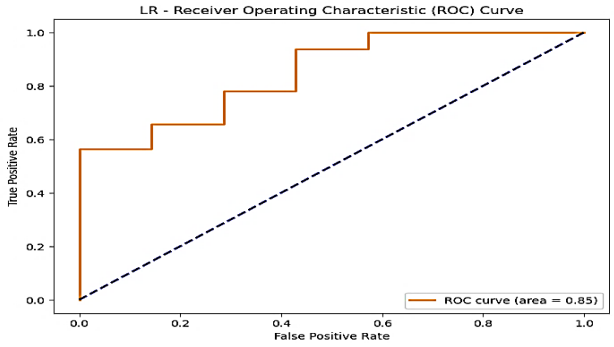
(a)



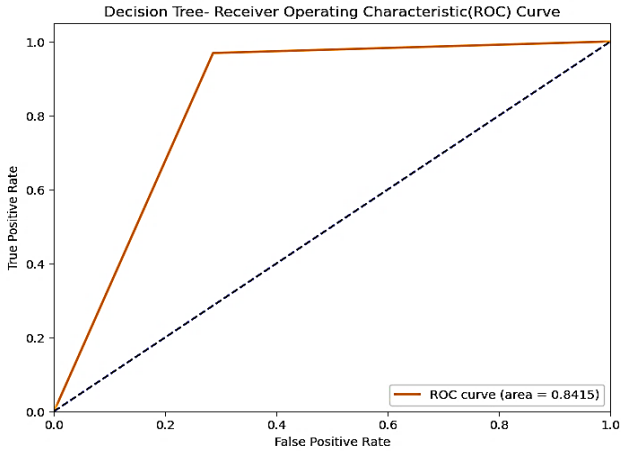
(b)



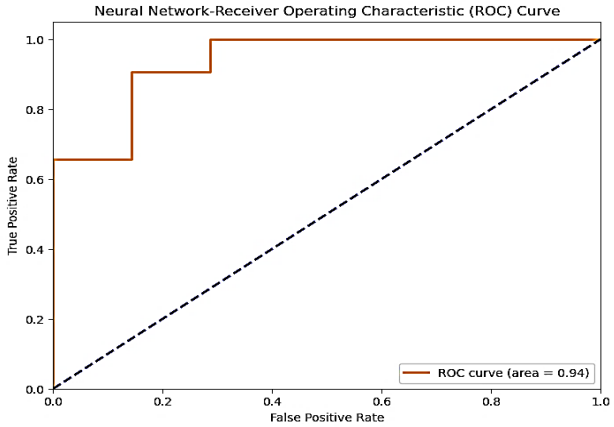
(c)



(e)



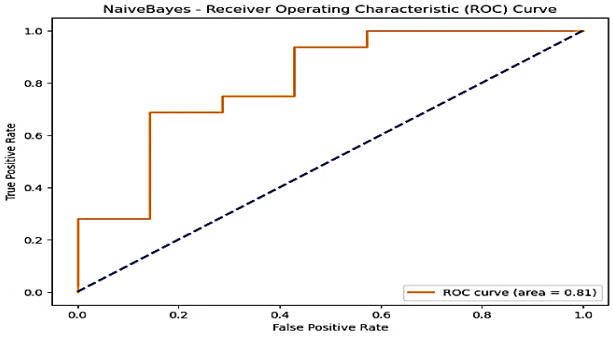
(f)



(g)

Figure 12: ROC curves for ML classifiers (a) XGBoost, (b) KNN, (c) SVM, (d) Naïve Bayes, (e) LR, (f) Decision Tree, and (g) Neural Network

Table 7 compares the performance metrics of the ML classifiers used in the proposed study with those used in the earlier studies. The table's results showed that XGBoost's accuracy in this work surpassed the accuracy of other ML classifiers employed in earlier studies.



(d)

Table 7: Comparing the performance measures to those from previous research studies

Ref.	ML classifiers employed	Dataset and approach followed	Accuracy achieved (%)
[27]	SVM, KNN	Speech recordings	68.45
[28]	CART, Random Forest	Speech recordings from forty individuals are used to classify patients based on their speech signals	66.5
[29]	Random Forest, SVM, MLP, KNN	For PD categorization, the single-photon emission computed tomography (SPECT) imaging technique DatSCAN is employed.	78.4(SVM),82.2(KNN)
[30]	HFCC-SVM	Using a method to extract Human Factor Cepstral Coefficients (HFCC), three different voice recordings of people with PD and healthy individuals were analysed.	87.5
[31]	SVM, KNN	118 Parkinson's disease patients' dense electroencephalogram (EEG) data classified into 5 groups based on the degree of their dementia.	84,88

[32]	Random Forest	High-resolution 256-channel EEG recordings from 50 PD patients were used to identify QEEG parameters distinguishing them from healthy controls	78
[33]	CNN	A CNN was trained on EEG data from 20 PD and 20 healthy subjects for early diagnosis.	88.25
[34]	CNN, RNN	A five-layer CNN analyzes multi-channel EEG spectrograms from patients and healthy controls for prognosis and diagnosis.	79(CNN),81(RNN)
[35]	SVM	ML is used to model thalamocortical dysrhythmia (TCD) underlying various neurological disorders	94.34

Current Study	XGBoost, KNN, SVM, Naïve Bayes, LR, DT, NN	Feature selection techniques with ML classifiers were applied to biological speech data for Parkinson's disease prediction.	XGBoost(94.87), KNN(82.05),SVM(84.62),Naive Bayes(69.23),LR(89.74),DT(92.31),NN(92.30)

5 Conclusion

In a nutshell, studies show that ML can be a useful technique for forecasting PD. By employing diverse algorithms and datasets, scholars have been successful in creating predictive models that exhibit encouraging outcomes in identifying individuals who are susceptible to Parkinson's disease. In addition to aiding in early detection, ML models also help in comprehending the underlying patterns and causes linked to PD. More complex and precise predictions have been made possible by the capacity to spot small patterns in a variety of datasets. In the study, Fisher Score, Shapley and LIME techniques are implemented to extract the important features. Training ML models with important features compared to all the features of the dataset can improve the performance of ML classifiers in predicting PD. In order to predict a patient with disease, feature selection approaches are used in conjunction with ML classifiers that had been trained on a dataset consisting of a variety of biological speech measurements. To advance this field and enhance the early detection and monitoring of Parkinson's disease, further research and validation studies are necessary. While the proposed explainable models demonstrated promising results in classifying Parkinson's disease, the relatively small dataset size of 195 samples presents a limitation that may affect the broader generalizability of the outcomes. Nonetheless, the use of 10-fold stratified cross-validation ensured balanced evaluation and enhanced reliability of the findings. Future research will aim to extend this work using larger, externally validated datasets and bootstrapping techniques to further strengthen the robustness and generalization of the proposed approach across diverse populations. Collaboration between researchers, clinicians, and data scientists can drive advancements in machine learning models for PD diagnosis using voice recordings, leading to improved accessibility and convenience in the diagnostic process.

References

[1] Mahlknecht P, Krismer F, Poewe W, Seppi K (2017) Meta-analysis of dorsolateral nigral hyperintensity on magnetic resonance imaging as a marker for Parkinson's disease. *Movement Disorders* 32(4):619-623.

[2] Dickson DW (2018) Neuropathology of Parkinson disease. *Parkinsonism Relat Disord* 46:S30-S33.

[3] Kalia LV, Lang AE (2015) Parkinson's disease. *Lancet* 386(9996):896-912.

[4] Rayan Z, Alfonse M, Salem ABM (2019) Machine learning approaches in smart health. *Procedia Comput Sci* 154:361-368.

[5] Prashanth R, Roy SD (2018) Novel and improved stage estimation in Parkinson's disease using clinical scales and machine learning. *Neurocomputing* 305:78-103.

[6] Almeida JS, Rebouças Filho PP, Carneiro T, Wei W, Damaševičius R, Maskeliūnas R, de Albuquerque VHC (2019) Detecting Parkinson's disease with sustained phonation and speech signals using machine learning techniques. *Pattern Recognit Lett* 125:55-62.

[7] Das R (2010) A comparison of multiple classification methods for diagnosis of Parkinson disease. *Expert Syst Appl* 37(2):1568-1572.

[8] Wang Y, Wang AN, Ai Q, Sun HJ (2017) An adaptive kernel-based weighted extreme learning machine approach for effective detection of Parkinson's disease. *Biomed Signal Process Control* 38:400-410.

[9] Ali L, Zhu C, Zhang Z, Liu Y (2019) Automated detection of Parkinson's disease based on multiple types of sustained phonations using linear discriminant analysis and genetically optimized neural network. *IEEE J Transl Eng Health Med* 7:1-10.

[10] Senturk ZK (2020) Early diagnosis of Parkinson's disease using machine learning algorithms. *Med Hypotheses* 138:109603.

[11] Harvey J, Reijnders RA, Cavill R, Duits A, Köhler S, Eijssen L, Pishva E (2022) Machine learning-based prediction of cognitive outcomes in de novo Parkinson's disease. *npj Parkinsons Dis* 8(1):150.

[12] Shahid AH, Singh MP (2020) A deep learning approach for prediction of Parkinson's disease progression. *Biomed Eng Lett* 10:227-239.

[13] Borzì L, Mazzetta I, Zampogna A, Suppa A, Olmo G, Irrera F (2021) Prediction of freezing of gait in Parkinson's disease using wearables and machine learning. *Sensors* 21(2):614.

[14] Pahuja G, Nagabhushan TN (2021) A comparative study of existing machine learning approaches for

[18] Arora S, Sahu A, Meena YK (2019) Classification of Parkinson's disease using machine learning and deep learning techniques: a review. *J Biomed Eng Med Imaging* 6(2):30-39.

[19] Dua S, Acharya UR (2020) Machine learning applications in the diagnosis of Parkinson's disease: a review. *Parkinsonism Relat Disord* 81:10-23.

[20] Li H, Habes M, Wolk DA, Fan Y, Alzheimer's Disease Neuroimaging Initiative (2019) A deep learning model for early prediction of Alzheimer's disease dementia based on hippocampal magnetic resonance imaging data. *Alzheimers Dement* 15(8):1059-1070.

[21] Guo, X., Tinaz, S., & Dvornek, N. C. (2022). Early disease stage characterization in Parkinson's disease from resting-state fMRI data using a long short-term memory network. *Frontiers in Neuroimaging*, 13, 952084. https://doi.org/10.3389/fnimg.2022.952084.

[22] Brizzi, A. C. B., et al. (2025). High-accuracy classification of Parkinson's disease using ensemble machine learning and stabilometric biomarkers. *Journal of Clinical Medicine*, 17(9), 133. https://doi.org/10.3390/jcm17090133.

[23] Mir, A. N., et al. (2024). Parkinson's disease diagnosis through deep learning. *arXiv*. https://doi.org/10.48550/arXiv.2412.06709.

[24] Ding, J.-E., Hsu, C.-C., & Liu, F. (2023). Parkinson's disease classification using contrastive graph cross-view learning with multimodal fusion of SPECT images and clinical features. *arXiv*. https://doi.org/10.48550/arXiv.2311.14902

[25] Omodunbi, B. A., et al. (2025). Stacked ensemble learning for classification of Parkinson's disease using voice recordings. *SN Computer Science*, 5(2), 23. https://doi.org/10.1007/s42979-024-02728-1.

[26] Little M, McSharry P, Hunter E, Spielman J, Ramig L (2008) Suitability of dysphonia measurements for telemonitoring of Parkinson's disease. *Nat Preced* 1:1.

[27] Sakar BE, Isenkul ME, Sakar CO, Sertbas A, Gurgen F, Delil S, Kursun O (2013) Collection and analysis Parkinson's disease detection. *IETE J Res* 67(1):4-14.

[15] Mei J, Desrosiers C, Frasnelli J (2021) Machine learning for the diagnosis of Parkinson's disease: a review of literature. *Front Aging Neurosci* 13:633752.

[16] Gupta R, Kumari S, Senapati A, Ambasta RK, Kumar P (2023) New era of artificial intelligence and machine learning-based detection, diagnosis, and therapeutics in Parkinson's disease. *Ageing Res Rev* 102013.

[17] Tsanas A, Little MA, McSharry PE, Ramig LO (2011) Nonlinear speech analysis algorithms mapped to a standard metric achieve clinically useful quantification of average Parkinson's disease symptom severity. *J R Soc Interface* 8(59):842-855.
of a Parkinson speech dataset with multiple types of sound recordings. *IEEE J Biomed Health Inform* 17(4):828-834.

[28] Vadovský M, Paralič J (2017) Parkinson's disease patients classification based on the speech signals. *Proc 15th IEEE Int Symp Appl Mach Intell Inform* 000321-000326.

[29] Mabrouk R, Chikhaoui B, Bentabet L (2018) Machine learning based classification using clinical and DaTSCAN SPECT imaging features: a study on Parkinson's disease and SWEDD. *IEEE Trans Radiat Plasma Med Sci* 3(2):170-177.

[30] Benba A, Jilbab A, Hammouch A (2017) Using human factor cepstral coefficient on multiple types of voice recordings for detecting patients with Parkinson's disease. *IRBM* 38(6):346-351.

[31] Betrouni N, Delval A, Chaton L, Defebvre L, Duits A, Moonen A, Dujardin K (2019) Electroencephalography-based machine learning for cognitive profiling in Parkinson's disease: preliminary results. *Mov Disord* 34(2):210-217.

[32] Chaturvedi M, Hatz F, Gschwandtner U, Bogaarts JG, Meyer A, Fuhr P, Roth V (2017) Quantitative EEG (QEEG) measures differentiate Parkinson's disease patients from healthy controls. *Front Aging Neurosci* 9:3.

[33] Oh SL, Hagiwara Y, Raghavendra U, Yuvaraj R, Arunkumar N, Murugappan M, Acharya UR (2020) A deep learning approach for Parkinson's disease diagnosis from EEG signals. *Neural Comput Appl* 32:10927-10933.

[34] Ruffini G, Ibañez D, Castellano M, Dubreuil-Vall L, Soria-Frisch A, Postuma R, Montplaisir J (2019) Deep learning with EEG spectrograms in rapid eye movement behavior disorder. *Front Neurol* 10:806.

[35] Vanneste S, Song JJ, De Ridder D (2018) Thalamocortical dysrhythmia detected by machine learning. *Nat Commun* 9(1):1103.

Fault Localization in Power Distribution Networks Using FLSO-SVR: A Data-Driven Machine Learning Approach

Ming Ma, Shujie Xu, Yingfang Miao, Shilong Chen, Xu Liu*
Yunnan Power Grid Co., Ltd. Yuxi Power Supply Bureau, Yunnan, Yuxi, 653100, China
E-mail: 18869754714@163.com
*Corresponding author

Keywords: distribution network, fault location, power system, distribution line, power grids.

Received: May 15, 2025

Distribution lines play an essential role in the modern power grid and have the potential to influence the reliability of the power supply. Immediate fault localization is essential for a reliable power system protection system. When diagnosing a problem, there are two stages. One area that has already reached remarkable accuracy rates is fault categorization. Fault localization in power distribution networks refers to identifying the exact location of faults or disturbances within the grid. Thus, fault location is the focus of this research, which is the inverse of the original goal. The proposed approach utilizes a combination of Principal Component Analysis (PCA) for feature reduction, Wavelet Transform for signal decomposition, and a novel Fuzzy-Logic Spider Optimization-based Support Vector Regression (FLSO-SVR) model for accurate fault localization. The algorithm determines the approximate location of the fault by examining voltage and current readings taken at the main feeder the monarch, as well as scheduled injections of reactive as well as active electricity by networks synchronous generators. Distributed grid plants' dynamic behaviour during fault transients was analyzed using an advanced machine learning (ML) model. Experiments conducted on the IEEE 118-bus system demonstrated the effectiveness of the method, achieving an accuracy of 95%, recall of 0.91, f1-score of 0.94, and precision of 0.92 which was higher than the state-of-the-art models. Modern power grid automation relies on the suggested methods, which improve fault localization accuracy, strengthen system resilience, and provide practical isolation measures. While the proposed method contributes to improved localization, its primary innovation lies in accurately localizing fault points within complex distributed networks using the FLSO-SVR framework.

Povzetek: Predlagani FLSO-SVR model z WT in PCA natančno lokalizira okvare v distribucijskih omrežjih ter izboljša zanesljivost pametnih omrežij.

1 Introduction

In a distributed network, significant consequences such as power supply interruptions, grid deterioration, and power quality problems, can occur as a result of electromagnetic transients caused by distribution line failures. Therefore, finding issues with distribution lines quickly and properly is of the utmost importance. The distribution system's job is to keep power outages as rare and short as possible while supplying subscribers with the energy they need [1]. This is an important problem since pinpointing the exact location of a defect in such large networks can be difficult due to the distribution networks and other parts of the power system's vulnerability to blackouts. Finding the exact location of the defect as quickly and accurately as possible improves the system dependability index and its efficiency [2]. ToA diagnosis technique is required to swiftly and accurately locate faulty sections and restore power supply in the event of a distribution network outage, this makes emergency repairs more efficient and decreases the amount of time needed to evaluate electricity lines [3]. However, pinpointing the exact locations of fault branches and distribution networks can be challenging due to the complexity of these network architectures, which often include multiple branches and large quantities of equipment distributed across large areas. The absence of efficient defensive coordination makes fault localization more difficult in distribution networks with Distributed Generation (DG) [4]. While fault localization techniques for distribution networks tend to center on conventional radial distribution systems, distributed generation distribution networks have just lately begun to garner significant attention [5].

Distribution network fault localization techniques include traveling wave methods, intelligent algorithms, time domain methods, and impedance methods. There are, however, drawbacks to each of these methods. Complex structure, high sampling frequency, and database

requirements are some of the potential challenges that traveling wave-based techniques might face [6]. Intelligent approaches, on the other hand, can necessitate assistance owing to their intricate structure and the necessity of a precise and extensive database. Consequently, the distribution network fault location has been the subject of substantial investigation. Imperata, fault indicator, signal injection, traveling wave, matrix algorithm, and artificial intelligence algorithm-based techniques are among the many that have been proposed [7]. This means that identifying wave heads at refraction and reflection nodes is not easy using standard traveling wave methods. Research intends to develop an intelligent fault localization model for distribution networks incorporated with DER. It employs the FLSO-SVR algorithm to improve fault localization accuracy. Fault localization was utilized to identify the exact location of faults or disturbances. The objective is to overcome limitations in traditional methods by enhancing feature extraction, handling nonlinearities, and maintaining noise resilience.

This research proposes a novel hybrid model FLSO with SVR specifically designed for accurate fault localization in electrical distribution networks. Unlike traditional models that focus primarily on fault localization or categorization, the FLSO-SVR model precisely determines the exact fault location using both offline and online data sources. The main contributions of this research are:

- A two-step machine learning-based methodology that accurately identifies the faulty section of the network and localizes the fault position using signal and system features.
- The integration of FLSO (a quantum-inspired metaheuristic optimizer) with SVR improves the model's ability to handle incomplete, imbalanced, or noisy data making the fault localization process more robust and scalable.
- The approach is adaptable to various network configurations and requires fewer data resources, enabling scalable and real-time fault localization even under complex distributed network conditions.
- Although improved fault localization accuracy is observed as a secondary outcome, the core innovation lies in enhancing fault localization precision, which is critical for minimizing outage durations and improving grid resilience.

1.1 Problem statement

Research discusses fault location and localization in distributed networks, including power line circumstances, computational cost, implementation issues, and scalability across electrical network models. It also overlooks practical implementation issues, such as data collection and processing, and the impact of fluctuating environmental conditions. The method's reliance on Matlab/Simulink simulations can limit its applicability in smaller data sources and climatic conditions. Although currently tested on centralized, simulated datasets, the proposed FLSO-SVR framework was designed with the potential to incorporate with decentralized sensor networks in future applications. With appropriate real-time data pipelines, it could be extended to dynamically adapt to environmental and operational changes in smart grid contexts. The practicality and wider application of the model could be ensured by verifying it through real-world demonstrations instead of depending just on Matlab/Simulink simulations.

2 Related works

To identify and categorize various problem kinds on power transmission lines, the article proposed a hybrid concurrent neuro-fuzzy approach (CNF) technique [8]. The results revealed that the CNF could capture a wide range of fault conditions, suggesting a shorter period of technical management's time. To avoid fire, an Artificial Intelligence (AI)--based approach for detecting open conductor faults on overhead cables was suggested in this research [9]. The system's design flaw was supposed to originate from the falling open conductor line connecting to the earth. The research gave the completed approach for the non-intrusive examination of high-voltage electrical networks [10]. A neural classifier built on a Multi-Valued Neurone (MVN) neural network models the technique, which relies on Applications of Machine Learning (also known as ML) as well as Frequency Response Analysis (FRA). The results of the simulation demonstrated that it was feasible to classify the state of the joints and avert serious issues.

As explored in [11], a High Voltage Direct Current (HVDC) transmission network based on voltage source converters might mimic power generation between AC networks. Both favorable and bad scenarios could be considered in an evaluation. The research detailed a medium-volt overhead line fault localization method utilizing Bidirectional Long Short-Term Memory (Bi-LSTM) with covered conductors (CC) [12]. The results of the experiments proved that the proposed BLSTM method was the most effective in learning the Partial Discharge (PD) structure, which is chaotic. Research [13] suggested a methodology that utilized ML techniques to detect trends in PD signals and anticipate potential failures. Their research method identified PD fluctuations with a high degree of precision.

To offer an automated diagnosis method for overheating problems in Overhead Transmission Lines (OHTL), the research [14] used semantic segmentation. The suggested intelligent detection approach successfully measured the grounds of the lead wiring or straining clamps with a diagnostic accuracy of 91.0 percent. The research presented a new method and framework for the ongoing assessment, prediction, and fault localization of the various

joint areas of power transmission overhead lines [15]. Soft computing, ML, as well as FRA were the foundations upon which the methods were built. One way to characterize the proposed method is as an intelligent measuring component; using a single measurement, a neural processor might glean all the necessary information for assessing a complex electrical structure. Research [16] developed a system for OHTL defect detection and classification using a Backpropagation Neural Network (BPNN). Linear Model (LM) outperformed Scaled Conjugate Gradient (SCG) and Bayesian Regularization (BR), according to the results. Research [17] proposed a fault localization method for distribution networks using C-means clustering on historical telemetry data. It introduced a mismatch index for identifying line anomalies and was tested on a real 12 kV feeder. The approach effectively recognized load patterns and detected faults. However, its reliance on historical data limits real-time adaptability, supporting the need for dynamic, learning-based models like FLSO-SVR. Related works are included in Table 1.

Table 1: Comparison of other related methods

Ref	Objective	Methodology	Results	Limitation
[8]	The novel method for identifying and finding different kinds of power transmission line faults is by employing the CNF.	Fuzzy logic (FL) and artificial neural networks (ANN) are used in conjunction to identify, categorize, and locate problems on high-voltage and very high-voltage transmission lines across both short and long distances.	Technical groups' maintenance time is decreased by CNF's high accuracy in identifying and locating different kinds of transmission line faults.	The hybrid FL-ANN approach might suffer from computational complexity and require extensive training data to generalize across varied network topologies.
[9]	To reduce the risk of fire, intends to develop an AI-based technique for identifying flaws in overhead wires before they result in ground contact.	Fault simulations provided voltage and current data for wavelet-transformed analysis, and a falling conductor model with pendulum motion was utilized to estimate fall duration.	Efficiently identifying faults before line-ground contact, the AI-based fault detection technique validated by MATLAB simulations decreases the danger of a fire.	The simulated environment might not fully capture real-world disturbances and model tuning is sensitive to conductor motion parameters.
[10]	The goal is to prevent overhead transmission grid breakdowns, especially at connection joints, by creating a non-intrusive monitoring method for high-voltage electrical networks.	Use an MVN neural network and an ML-based neural classifier with Frequency Response Analysis (FRA) to diagnose and track electrical systems.	Effective fault diagnostics and prognostics were displayed, enabling quicker recovery times through the identification of crucial parts, better maintenance planning, and increased grid availability.	FRA-based models are susceptible to noise interference and require precise calibration to avoid false positives.
[11]	To ensure supply continuity in intricate power systems by creating an ML-based fault diagnosis method for HVDC transmission systems.	Data to be analyzed using ML techniques to extract diagnostic information. Matlab/Simulink can be used to conduct the simulations.	Evaluation of ML-based fault diagnosis methods' effectiveness in identifying anomalies and guaranteeing a continuous power supply is the outcome of the investigation.	ML model performance depends heavily on labeled HVDC fault datasets, which are limited in diversity and availability.
[12]	Using partial discharge signals to create a BLSTM model for Medium Voltage (MV) overhead wires with covered conductor fault detection.	An actual dataset from the high-sampling-frequency meter (40 MHz) at the center was used, guaranteeing a variety of data in terms of PD quality and noise spectrum.	The Bi-LSTM approach overcame noise problems and showed competitive performance in learning and recognizing chaotic PD patterns.	BLSTM training is resource-intensive and model accuracy can degrade under extreme environmental noise not represented in the training set.
[13]	To anticipate possible failures and use ML methods to identify PD signal patterns in transmission cables.	The method utilized to find patterns of partial discharge was based on ML. The research 's main objective was to examine localized electrical insulating dielectric deterioration.	The suggested model was able to detect partial discharge patterns with accuracy.	The approach's scalability and limited generalization when applied to varied cable insulation types and aging conditions.

[14]	To improve the dependability of leading wires and strain clamps on overhead transmission lines by utilizing deep learning (DL) and infrared thermal imaging to identify overheated issues.	To enhance contrast, the research used UAVs equipped with infrared thermal imagers. The performance of the DeepLab v3+ network was compared using the Focal Loss function, ResNet101, and convolutional block attention module.	Frequency-weighted Intersection over Union (IoU), Adaptive IoU (AIoU), F1 score, and average pixel accuracy were among the high-performance metrics attained by the suggested approach; AIoU was enhanced by optimized atrous rates.	Thermal imaging is weather-dependent and UAV operation might be restricted in certain conditions or terrains.
[15]	The establishment of an intelligent measuring module for the ongoing assessment, diagnosis, and prognosis of joint areas in overhead power delivery.	Uses methods from ML, soft computing, and FRA. Single readings are analyzed by a neural processor to retrieve diagnostic data from intricate electrical systems.	By providing an intelligent diagnostic strategy for problem diagnosis and monitoring, the suggested method improves the research of intricate electrical systems.	System reliability might degrade under irregular measurement intervals or missing input features in real-time deployment.

Existing fault localization techniques in distribution networks often struggle with handling the complexity of bidirectional power flow and dynamic operating conditions introduced by distributed energy resources (DERs). Many conventional models lack the adaptability to accurately detect faults under high variability and noise. To address these challenges, this research proposes the use of the Fuzzy-Logic Spider Optimization-based Support Vector Regression (FLSO-SVR) algorithm for improved fault localization in distribution systems. Distributed energy resources (DERs) such as small hydropower, solar power, biomass, wind power, fuel cells, and others have allowed for on-site generation, which has led to major changes in distribution systems in recent years. Both grid-connected and off-grid DG have the potential to reduce environmental impacts while simultaneously increasing supply security. However, situational awareness is severely hindered by the complicated network architecture with bidirectional current flow. This research would overcome this issue by implementing the FLSO-SVR algorithm for identifying the network fault in the distribution network.

3 Materials and methods

In this research, we collect data from an electrical distribution network that includes features for fault location (FL) and fault inception angle. Several provinces' distributed networks are included in the dataset that was retrieved from the Beijing electrical power station. PCA is used to extract features from the dataset. The data amount of the signal is then reduced using WT. An efficient method for identifying faults in distributed networks using overhead lines using the FLSO-SVR. The proposed FLSO-SVR framework integrates WT, PCA, and SVR, each contributing specific advantages. WT enhances time-frequency feature extraction for fault signals; PCA reduces data redundancy and noise, improving training stability. SVR handles nonlinear regression tasks effectively, while FLSO tunes SVR hyperparameters using a fuzzy swarm approach, enhancing convergence and robustness. This combination improves generalization, precision, and resilience to noise in fault classification, as evidenced in IEEE 118-bus simulations. The raw voltage and current time-series signals generated from the IEEE 118-bus system simulations were directly used as input for feature extraction. The data was segmented based on fault initiation time, and each segment was labeled using a tuple of (fault type, location index) corresponding to the event's simulated parameters. The resulting signal segments were then used to extract wavelet-based features before applying dimensionality reduction via PCA. Figure 1 represents the entire process of the research in a pictorial representation.

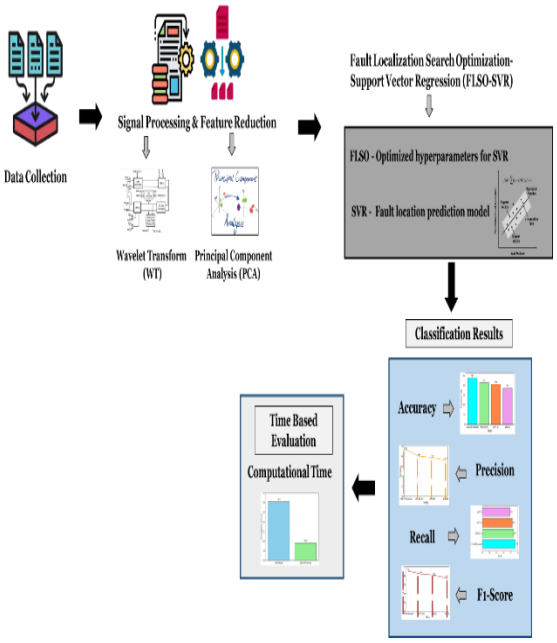


Figure 1: Process flow of the proposed methodology.

3.1 Wavelet transform

Wavelet Transform (WT) is an effective signal processing tool which provides multi-resolution analysis capability where it is applicable to analyze nonstationary signals in time as well as frequency domain. This renders it

especially appropriate in the analysis of transient faults in power systems. In this research, voltage and current signals were decomposed into three levels (D1, D2, D3 and A3) using discrete WT with Daubechies-4 (dB4) as the wavelet and the statistical features were extracted to classify the features are shown in figure 2.

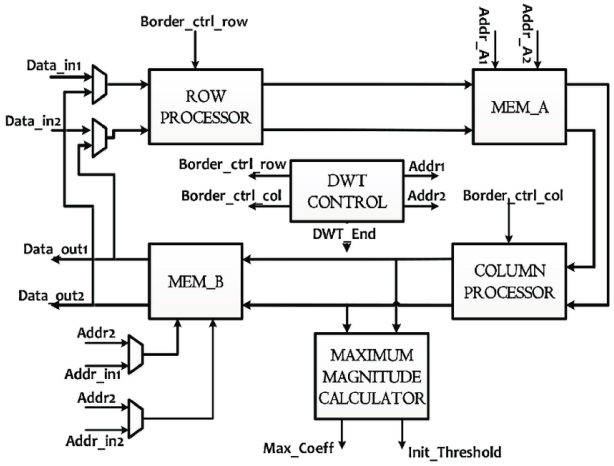


Figure 2: Wavelet transform

In the present instance, the wavelet basis function employed in the fault localization evaluation was dB4. dB4 is a wavelet type function commonly applied in fault localization, as it can extremely capture smooth and sudden variations of signal. Its small support and great localization in the time frequency domain allow an effective localization of brief faults in overhead lines. The properties of dB4 wavelet enable clear decomposition of the fault signals and therefore fault localization and fault location can be conducted accurately. The signal might be decomposed into a set of detail (c) and approximate (b) coefficients which are associated with high and low-frequency bands, respectively. Wavelet Transform is a mathematical tool that analyzes signals at different frequencies to carry out fault localization in overhead lines. Wavelet Transform decomposes fault signals into different frequency components along with fault-related characteristics in the context of overhead lines. In conjunction with a rapid fault localization and location method, it makes possible real-time fault localization in overhead lines through the localization of anomalies in the signal. Such a technique offers high accuracy in quickly pinpointing both the kind and location of faults. It helps in furthering the reliability of fault localization since it captures both low and high-frequency signatures of faults. Its application minimizes diagnostic time, thereby making possible prompt maintenance and short grid downtime.

The Discrete Wavelet Transform (DWT), given a voltage signal $U(s)$ from the power supply, is described as

$$DWT(U,N,M) = \frac{1}{\sqrt{N}} \int_{-\infty}^{\infty} U(s)\, \Psi\left(\frac{s-M}{N}\right) ds \quad (1)$$

The wavelet basis function is denoted by Ψ, and the dilation and translation variables are denoted by N and M. Finally, WT in the discrete form as,

$$DWT(U)[n] = \sum_{l=-\infty}^{\infty} U(l)\, \Psi\left(\frac{n - lN_0^j}{N_0^j}\right) \quad (2)$$

N and M are substituted with N_0^j and lN_0^j, respectively, and l is a single integer (i.e., sample index of the signal). The scaling function in the initial phase is represented as the sum of the subsequent phase and it could be expressed as follows,

$$\varphi(s) = \sum_{m=-\infty}^{\infty} g(m)\sqrt{2}\varphi(2s - m) \quad (3)$$

The primary voltage signal can be expressed through the previous equation.

$$U(s) = \sum_{l=-\infty}^{\infty} a_J(l)\ 2^{J/2}\phi(2^J s - l) + \sum_{j=J}^{\infty}\sum_{l=-\infty}^{\infty} d_j(l) 2^{\frac{j}{2}}\varphi(2^j s - l) \quad (4)$$

represents the scaling function's coarse modification parameter. Where, J is the lowest resolution level. $a_J(l)$ is the scaling coefficient and $d_j(l)$ is the wavelet coefficient at level J. ϕ and φ are the scaling and wavelet functions respectively. The detail and approximate coefficients are expressed as follows,

$$b_o(l) = \sum_{n=-\infty}^{\infty} b_{o+1}(n) g(n - 2l) \quad (5)$$

$$c_o(l) = \sum_{n=-\infty}^{\infty} d_{o+1}(n) g_1(n - 2l) \quad (6)$$

Where o is the decomposition level and $g(n-2l)$, $g_1(n-2l)$ are low-pass and high-pass wavelet filters respectively.

After applying 3-level wavelet decomposition using the dB4 wavelet, statistical features were extracted from the resulting sub-bands. From each sub-band, the following seven features were computed: mean, standard deviation, energy, entropy, kurtosis, skewness, and root mean square (RMS). This resulted in a 28-dimensional feature vector per signal instance (4 sub-bands × 7 features). PCA was subsequently applied to reduce the dimensionality of these feature vectors. Based on the cumulative explained variance, six principal components were retained, capturing over 95% of the original variance. These six-dimensional vectors served as the input for training the FLSO-SVR model.

3.2 Principal component analysis

Principal Component Analysis (PCA) is a well-liked dimensionality reduction method for data compression and analysis. The method works by finding a small set of symmetric linear combinations that include the beginning variables that have the maximum variance. This sets the limit for the number of linked variables. The modification specifies that the initial principal element is the linear combination of the initial elements with the highest variance; the subsequent principal element is the linear combination of the initial elements with the largest deviation, parallel to the initial principal component; and so on. Many data sets' top few principal components capture the majority of the original data set's differences, enabling the balance to be discarded with minimal variance loss when data dimensions are lowered. PCA streamlines analysis by breaking down high-dimensional data into small, uncorrelated components. When running diagnostics on overhead lines, PCA improves problem localization accuracy by helping to spot patterns in system variables like voltage and current. Blending PCA with rapid fault localization methods enables accurate defect localization and type identification. To find problems with overhead electricity lines, this method reduces the complexity and processing time required. In general, it makes power grid maintenance more efficient and reliable, and it assists with fault management. Below is the flow of the transition:

Every observation in the given collection of observations W_1^t, W_2^t, W_m^t is represented by a single vector of length $W_{m\times n}$ in the data, which is expressed by a matrix n. Each $w_i \in \mathbb{R}^n$ is a row vector of the matrix.

$$W_{m\times n} = \begin{bmatrix} w_{11} & \cdots & w_{1n} \\ w_{21} & \cdots & w_{2n} \\ \cdots & \cdots & \cdots \\ w_{m1} & \cdots & w_{mn} \end{bmatrix} = \begin{bmatrix} W_1^t \\ W_2^t \\ . \\ . \\ W_m^t \end{bmatrix} \quad (7)$$

The average observation is described as

$$\mu = \frac{1}{m}\sum_{j=1}^{m} w_j \quad (8)$$

The difference from the average is defined as

$$\Phi_j = w_j - \mu \quad (9)$$

The sample CM of the data set can be described as

$$D * = \frac{1}{m}\sum_{j=1}^{m}(w_j - \mu)(w_j - \mu)^T = \frac{1}{m}\sum_{j=1}^{m}\Phi_j\Phi_j^T = \frac{1}{m}BB^T \quad (10)$$

Where $B = [\Phi_1, \Phi_2, \cdots \Phi_m]$ which represents the matrix of centered column vectors, μ is the sample mean vector. $\Phi_j = (w_j - \mu)^T$ represents the centered column vector of sample j.

When doing principal component analysis (PCA) on CM sample D, the Singular Value Decomposition (SVD) method is typically used to find the eigenvalues and eigenvectors. Think about the fact that the CMsampleD contains eigenvalues and eigenvectors in pairs, denoted as (λ_1,v_1), (λ_2,u_2),..., (λ_n,v_n) together with n. We select the l eigenvectors in a way that maximizes their eigenvalues. It is reasonable to assume that the residual (n-l) dimensions are usually filled with noise because of the presence of strong eigenvalues and that the intrinsic dimensions of the signal-regulating subspace are l. Subspace k's dimensions can be assessed.

$$\frac{\sum_{j=1}^{l}\lambda_j}{\sum_{j=1}^{n}\lambda_j} \geq \alpha \quad (11)$$

Where the variable α stands for the ratio of the variance in the subspaces to the total variation in the starting space. The eigenvectors serve as the columns of the n×l matrix V that we construct. Data representation using principal components allows for expression into the l-dimensional subspace.

$$z_j = V^S(w_j - \mu) = V^S\Phi_j \quad (12)$$

To reduce the dimensionality of the wavelet-derived feature vectors, PCA was applied. Prior to PCA, each signal instance was represented as a 28-dimensional vector, composed of statistical features extracted from the four wavelet sub-bands. After PCA transformation, six principal components were engaged, conserving over 95% of the collective modification. These six-dimensional feature vectors were utilized as input to the SVR classifier for training and testing.

3.3 Fault localization search optimization-support vector regression (FLSO-SVR)

In this research, the FLSO approach is integrated with SVR, which provides FLSO-SVR for effective fault localization in the overhead lines. To increase the accuracy of fault localization, FLSO-SVR combines SVR with the FLSO algorithm. In this framework, SVR serves as the regression engine for modeling fault location, while FLSO is used to optimize SVR's hyperparameters, including the penalty parameter C, the epsilon-insensitive loss ε, and kernel-specific parameters (e.g., γ for RBF kernel). The integration improves performance by enabling better

generalization and faster convergence. Quantum-inspired features in FLSO enhance search diversity and reduce training error, thereby improving overall fault localization precision.

FLSO (Fault Localization Search Optimization)

With its foundation on quantum bits and rotation gates, FLSO not only provides a higher search capability but also a faster convergence rate. In contrast to the standard Localization Search Optimization (LSO), which uses a galaxy's velocity as well as location to express its state, the FLSO uses a quantum bit and an angle. These words are defined further down. The FLSO integrates a quantum-inspired encoding scheme, where solution states are modeled using probability-based binary representations. This enhances the diversity and adaptability of the search process, enabling faster convergence and better fault localization. Although quantum principles inspire the design, it operates on classical computing platforms. The influence of this component would be further analyzed through ablation experiments in future work.

- A pair of integers characterize the smallest FLSO component, a quantum bit.

$$r_{\mathrm{ij}}(x) = [\alpha_{\mathrm{ij}}(x), \beta_{\mathrm{ij}}(x)]^{\mathrm{T}} \tag{13}$$

The probability that a quantum bit is in states 0 and 1, respectively, is provided by its modulus $\alpha_{\mathrm{ij}}(x)$as well as $\beta_{\mathrm{ij}}(x)$, and both of these conditions must be met.

$$\left|\alpha_{ij}(x)\right|^2 + \left|\beta_{ij}(x)\right|^2 = 1 \tag{14}$$

A quantum bit individual, which is the component that comprises a string of quantum bits, is described as

$$r_i(x) = \begin{bmatrix} \alpha_{i1}(x), \ldots\ldots, \alpha_{ij}(x), \ldots\ldots, \alpha_{im}(x) \\ \beta_{i1}(x), \ldots\ldots, \beta_{ij}(x), \ldots\ldots, \beta_{im}(x) \end{bmatrix} = \left[r_{i1}(x), \ldots., r_{ij}(x), \ldots., r_{im}(x)\right] \tag{15}$$

Where each $r_{\mathrm{ij}}(x) = [\alpha_{\mathrm{ij}}(x), \beta_{\mathrm{ij}}(x)]^{\mathrm{T}}$. Every possible outcome can be expressed linearly by a quantum bit due to its probability definition. Thus, entirely 2^m kinds of people can be described using combinations of different quantum bit levels. When it comes to ensuring population diversity, this quantum bit design outperforms other descriptions. In the case where three individuals with two sets of amplitudes and three quantum bits are present, for instance,

$$\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{2} & \frac{1}{\sqrt{2}} \\ \cdot & \cdot & \cdot \\ \frac{1}{\sqrt{2}} & \frac{\sqrt{3}}{2} & \frac{-1}{\sqrt{2}} \end{bmatrix} \tag{16}$$

Subsequently, the individual's state vectors can be shown as

$$\left\{\frac{1}{4}|000\rangle, \frac{-1}{4}|001\rangle, \frac{\sqrt{3}}{4}|011\rangle \frac{1}{4}|100\rangle, \frac{-1}{4}|101\rangle, \frac{\sqrt{3}}{4}|110\rangle, \frac{-\sqrt{3}}{4}|111\rangle\right\} \tag{17}$$

The preceding result shows the probability that stands for the different states, which are 0 and 1. An individual quantum bit contains data made up of eight separate states.

- The normalization criteria allow for the representation of the quantum angle as

$$\begin{cases} |r_{ij}(x)\rangle = cos\theta_{ij}(x)|0\rangle + sin\theta_{ij}(x)|1\rangle \\ \theta_{ij}(x) = \arctan\left(\dfrac{\beta_{ij}(x)}{\alpha_{ij}(x)}\right) \end{cases} \tag{18}$$

The quantum angle can be utilized for representing the individual quantum bit.

$$\begin{array}{lccccc} \vec{r}_i(x) = & [r_{i1}(x), \ldots\ldots., & r_{ij}(x), \ldots\ldots & r_{im}(x)] \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \theta_i(x) = & \theta_{i1}(x), \ldots.., & \theta_{ij}(x), \ldots\ldots, & \theta_{im}(x)] \end{array} \tag{19}$$

Here, $r_{\mathrm{ij}}(x) = [\alpha_{\mathrm{ij}}(x), \beta_{\mathrm{ij}}(x)]^{\mathrm{T}}$

Quantum bits and angles are changed because of the basic FLSO updating process, and the modified quantum bits have to satisfy the normalization criteria. One might calculate the quantum rotational gate updating equation using,

$$\Delta\vec{\theta}_i(x+1) = \omega.\Delta\vec{\theta}_i(x) + \varphi.q_1.\left[\vec{\theta}_{pb} - \vec{\theta}_i(x)\right] + \eta.q_2.\left[\vec{\theta}_{gb} - \vec{\theta}_i(x)\right] \tag{20}$$

Whereω is the inertia weight ($\omega = 0.72$)φ, η represents the parameters (i.e. $\varphi = 1.65, \eta = 1.81$), the random numbers $q_1 and q_2$ are in the range (0, 1), $\Delta\vec{\theta}_i(\cdot)$ is the change of the angle,$\vec{\theta}_i(\cdot)$ represents the present angle, $\vec{\theta}_{pb}, and\, \vec{\theta}_{gb}$ represents the personal and global best angles.

The quantum bit as well as the rotation gate model has precocious convergence issues, despite its better population diversity characteristics. Assuming the world's finest individual stays the same throughout generations, two quality barriers such as concentration and affinity emerge once we find the evolutionary solution. Individuals' affinity values reflect how well they solve issues, whereas concentration tells what proportion of the

population is quantum. The optimized hyperparameters determined by FLSO are then used in the SVR model to perform the final regression and fault location. SVR uses kernel functions to transform the input space into a high-dimensional space where linear regression becomes more effective.

Support Vector Regression (SVR)

To account for these outliers, a non-linear Membership Degree (MF) is employed to map the initial Input Space (IS) which is usually non-linearly linked to the predictor variable onto a higher-dimensional Feature Space (FS). For the SVR, this is an essential principle. As an extra IS, the FS which has been encoded in the kernel's matrix could be used to resolve the regression problem (refer to Figure 3).

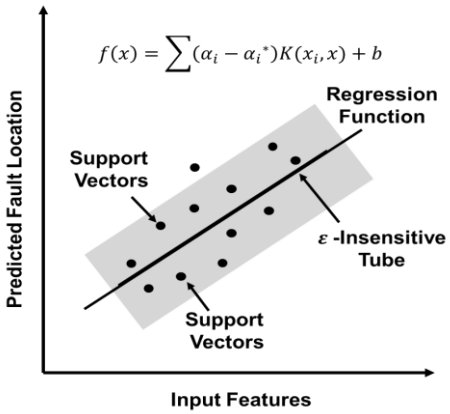


Figure 3: Support Vector Regression

SVR model predicts a continuous output based on a set of learned support vectors and a kernel function. The regression output is given in the figure 3. In this expression, $f(x)$ represents the predicted value for a new input x, such as the estimated fault location in a power distribution network. The summation runs over all support vectors, which are specific training samples that have a non-zero influence on the final regression function. The terms α_i and $\alpha_i{}^*$ are Lagrange multipliers obtained during the training process. The kernel function $K(x_i, x)$ measures the similarity between the new input x and each support vector x_i. The term b denotes the bias or offset, which is also learned during training.

Let $H(M \times N)$ be the data set and $z_j \in Q$ be the output vector. Using the provided data set T as a basis, SVR attempts to identify a multivariate regression function $e(h)$ that would forecast the intended output characteristic of an unidentified object. A thorough description of the SVR equations is provided. As a result, we would express the SVR equation as follows,

$$e(h) = \sum_{j,i=1}^{M}(\propto_j - \propto_j^*)\langle\phi(h_j).\phi(h_i)\rangle + a \tag{21}$$

Where D is a regularization constant that establishes a balance between the training error and model effectiveness, and $\propto_j and \propto_j^*$ represents the LM achieving constraint $0 \leq \propto_j, \propto_j^* \leq D$. The initial position of the regression function $e(h)$ is represented by parameter a, and the MF is represented by component ϕ. It is challenging to identify the non-linear mapping $\phi(h)$ because it is typically unknown previously. Frequently, a KF is used to calculate the mapping term $\langle\phi(h_j)\phi(h_i)\rangle$ in Equation (22).

$$L(h_j, h_i) = \langle\phi(h_j).\phi(h_i)\rangle \tag{22}$$

We can easily manage the non-linear interactions in the data according to this kernel modification. Variance–covariance, polynomial, RBF–based kernels, PUK, and RBF are among the KFs that are appropriate to handle non-linear regression issues. We have demonstrated that the PUK KF might act as a robust alternative for the polynomial, RBF-based KFs, and variance-covariance that are frequently used.

By the time the KF is run, Equation (23) is equivalent,

$$e(h) = \sum_{j=1}^{m}(\alpha_j - \alpha_j^*)L(h_j, h) + a \tag{23}$$

The non-linear IS is typically transformed through the KF, into a high-dimensional FS where the solution to the issue can be described as a straight linear issue. Through employing QP to solve the altered regression issue, the $\propto_j and \propto_j^*$ values are found. As a result, they are referred to as support vectors since the input objects belonging to the non-zero LM $\propto_j and \propto_j^*$ contribute to the last regression design.

The FLSO-SVR technique is a novel approach to diagnosing and finding problems with overhead electricity lines. This method seeks to revolutionize fault localization in power transmission networks by combining ideas from quantum computing with fault location. Traditional methods of fault localization often have problems with accuracy and performance, especially in complicated network settings. By incorporating quantum-centric calculations, this method takes advantage of the massive processing capacity along with the scalability of quantum systems, simplifying and speeding up the evaluation of

fault data. The addition of FLSO improves the technique's efficacy. Inspired by ideas from celestial navigation, this optimization method finds errors in massive datasets with pinpoint precision. Through SVR-assisted learning and adaptation, the model progressively improves its diagnostic abilities. By decreasing latency, increasing grid reliability, and accurately diagnosing and localizing issues in overhead power lines, the approach leads to a strong electrical network. The suggested model's pseudocode is represented by Algorithm 1.

Algorithm 1: FLSO-SVR
Input: $SVR\ P = \{X, Y\}, where\ X = \{features\}, Y = \{fault\ locations\}$
Output: $Fault\ location\ prediction$
1. $Divide\ the\ dataset\ P\ into\ a\ training\ set\ (P_train$
2. $Preprocess\ the\ dataset\ (Categorical\ Encoding, r$
3. $Train\ an\ SVR\ model\ utilizing\ FLSO\ with\ PUK\ ker$
   a. $Perform\ 5$
$- fold\ grid\ search\ (C \in [0.1 - 100], \gamma$
$\in [0.001 - 1],$
$\in [0.001 - 0.1])\ to\ optimize\ hyperparameters.$
   b. $Train\ the\ SVR\ model\ utilizing\ the\ optimized\ p$
4. $Asses\ the\ trained\ model\ on\ P_test$:
   a. $Forecast\ the\ fault\ location\ for\ each\ instance$
   b. $Compute\ performance\ metrics\ (accuracy, prec$
$- score)$
5. $Return\ the\ fault\ location\ prediction\ for\ the\ test$

In this research, localized distributed power generation is used in the data collection process. By using principal component analysis, the data is preprocessed. The wavelet transform is used to decompose the fault signals into different frequency components along with fault-related characteristics in the context of overhead lines. In conjunction with a rapid fault localization and location method, it makes possible real-time fault localization in overhead lines through the localization of anomalies in the signal. SVR is used for training data, identifying the irregularities in quantum bits, and identifying overhead lines. The optimization process can be done by using FLSO. The proposed model FLSO-SVR is used to identify the overhead lines in the distributed network.

4 Results and discussions

The IEEE 118-bus system was used as a testing environment to evaluate the proposed Fault Location Accuracy (FLA) under realistic power grid conditions. The system offered a wide range of fault scenarios, including changes in bus voltage, current flow, line impedance, and fault severity, enabling a systematic valuation of the algorithms.

The simulations were performed in Python 3.10 on a 64-bit Windows 11 platform. The libraries version used were as follows: model training and validation Scikit-learn 1.3.0, numerical operations NumPy 1.24.2, visualization Matplotlib 3.7.1 and Seaborn 0.12.2. The fault data utilized was synthetically generated and taken out of the IEEE 118-bus system modeling single-phase and three phase faults at different load and generation conditions. Twenty random lines at various voltage levels were selected and faults were introduced on these lines: five fault types were represented, namely, single line-to-ground (SLG), double line-to-ground (DLG), line-to-line (LL), three-phase (LLL), and open conductor faults. All conditions were emulated at varying DER penetration rates and load variations to mirror real-life circumstances of disturbances. A 2000 labeled samples were gathered, stratified and divided into 80 percent training and 20 percent testing to get balanced classes and sound evaluation. In fault localization, FLSO optimizes SVR model through tuning of hyperparameters, whereas, SVR considers a kernel function (e.g., RBF) to carry out regression-based fault location. The suitability of the massive power network automation by the proposed algorithms was evidenced by their ability to run experiments on the IEEE 118-bus system and demonstrated the ability to deal with complex fault scenarios. The performance of the two algorithms was evaluated based on four useful measures:

- Accuracy: The proportion of accurately detected fault locations (FL) in a distributed network
- Precision: The percentage of true positive predictions among all positive predictions, denoting the relevance of fault localization.
- Recall: The proportion of true positives discovered out of all actual positives, reflecting the sensitivity of the algorithms.
- F1-score: The harmonic means of precision and recall, offering a balanced measure of performance.

The suggested FLSO-SVR were evaluated against several previous methods such as Variational mode decomposition Teager Energy Operator (VMD-TEO) [18], Adaptive extended Kalman filter (AEKF) [19], War Strategy Optimization (WSO) [20] to determine their efficacy. FLSO-SVR's comparison metrics are accuracy, precision, recall, and F1-score.

Table 2: Comparison of FLSO-SVR with Existing Techniques

Technique	Accuracy (%)	Precision	Recall	F1-score
FLSO-SVR [Proposed]	95	0.92	0.94	0.91
VMD-TEO [18]	85	0.80	0.88	0.84
AEKF [19]	82	0.77	0.85	0.81
WSO [20]	74	0.72	0.78	0.75

To ensure the robustness of the FLSO-SVR model, variance analysis was conducted by running the experiments ten times with different random splits. The average performance metrics and standard deviations were calculated to assess consistency. The model achieved an average accuracy of 95.1% ± 1.2, precision of 0.91 ± 0.03, recall of 0.93 ± 0.02, and F1-score of 0.90 ± 0.02. These results confirm the model's stability and reliability across different data configurations. Such variance analysis is critical for validating ML models, as single-run metrics can lead to misleading conclusions about performance. The findings in Table 2 clearly show that the proposed FLSO, which uses an SVR, surpasses conventional methods on all metrics tested. The high recall value of 0.94 shows its resilience in accurately determining fault locations, whereas the F1 score of 0.91 emphasizes its balanced efficiency in terms of precision and recall. This enhancement demonstrates the efficacy of combining the FLSO output with sophisticated fault severity evaluations and isolation tactics. Despite their popularity, rule-based and heuristic approaches lag because of their restricted adaptability to dynamic fault scenarios. These comparisons emphasize the better results of the proposed intelligent algorithms in fault localization and isolation, establishing a new benchmark for power grid automation systems. Figure 4 represents the graphical representation of the accuracy.

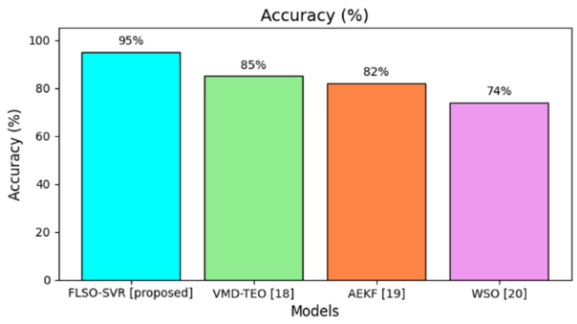


Figure 4: Accuracy Comparison of FLSO-SVR with existing techniques

The graph explains that the proposed FLSO-SVR has the highest accuracy (95%) because the SVM classifier can model complex relationships between input features and fault locations. Unlike conventional methods, SVM efficiently differentiates classes with a well-defined margin, even in high-dimensional spaces, resulting in better fault localization accuracy. Figure 5 depicts the graphical illustration of the precision over methods.

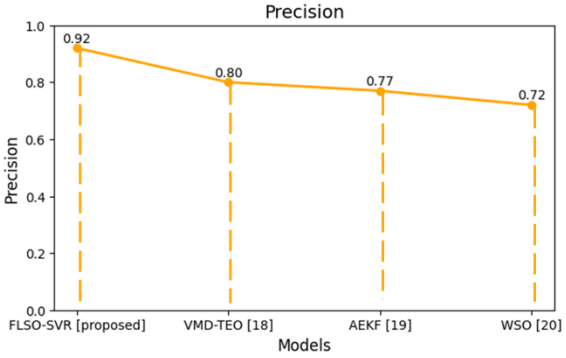


Figure 5: Precision comparison of FLSO-SVR with existing techniques

Figure 5 compares the precision of four fault localization models. The proposed FLSO-SVR method achieves the highest precision (0.92), outperforming traditional methods. This indicates that FLSO-SVR has superior capability in correctly identifying fault instances. Figure 6 explicitly mentions the pictorial representation of the recall over various methods.

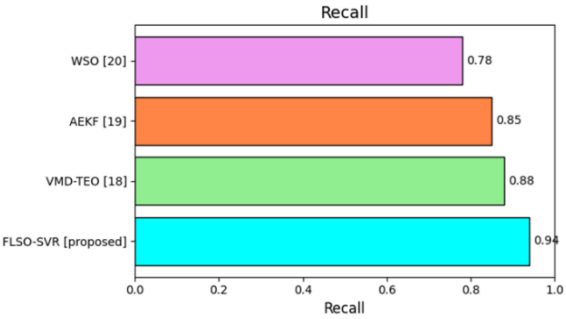


Figure 6: Recall Comparison of FLSO-SVR with existing techniques

Figure 6 shows the FLSO has the highest recall (0.94), suggesting that it is efficient in detecting almost all actual fault locations. SVRs resilience against data imbalance and its ability to manage nonlinear fault patterns contribute substantially to this efficiency. Figure 7 represents the F1-Score Comparison of proposed approach with Existing Techniques.

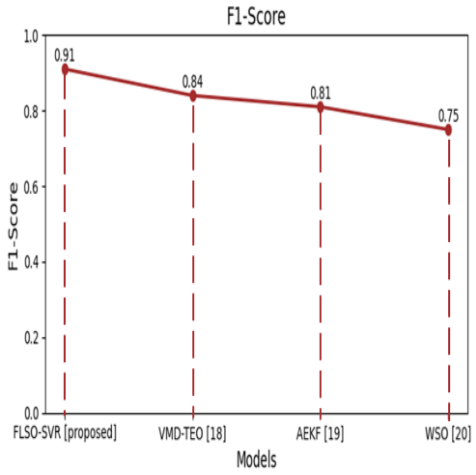


Figure 7: F1-Score Comparison of FLSO-SVR with existing techniques

Figure 7 demonstrates that the FLSO-SVR model has an F1-score of 0.91, which means that the model performs similarly in precision and recall. It proves that it is consistent in keeping both accurate and large fault localization.

In comparison with rule-based or threshold methods, the FLSO-SVR adapts itself dynamically to various fault situations and thus, is more accurate. The suggested algorithms were more accurate, precise, recall, and F1 score than the traditional approaches due to the smart utilisation of the SVR classifiers and robust fault handling strategies. The ability of FLSO-SVR to fit complicated fault patterns and flexibility in isolation actions offer better reliability and performance in automation of power grids compared to the past methods, which makes superior substitutes.

Computational time

The computational complexity of models can be a determining factor in their usefulness particularly in real-time or resource limited systems. Table 3 presents a comparison of computational time for the existing PSO-SVM model and the proposed FLSO-SVR technique. The proposed approach is designed to update processing and reduce computational overhead during implementation. Enhanced efficiency not only accelerates prediction but also improves scalability across larger datasets and complex problem domains.

Table 3: Comparison of computational time over method.

Metrics	Computational Time
PSO-SVM [21]	5.341s
FLSO-SVR [proposed]	5.298 s

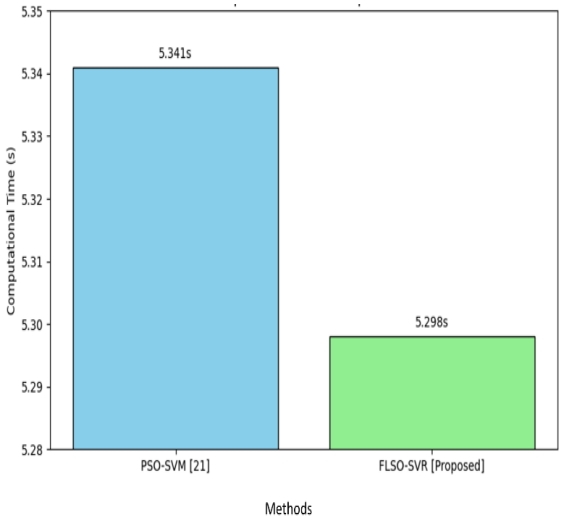


Figure 8: Graphical representation of the computational time performance.

As observed, the proposed FLSO-SVR method achieves a marginally lower computational time (5.298 seconds) compared to PSO-SVM (5.341 seconds), indicating an improvement in efficiency while maintaining predictive performance. Figure 8 illustrates the visualization of the time evaluation performance.

4.2 Ablation research

To understand the individual and combined impact of each component in the proposed fault localization framework, the ablation research was conducted. The core modules evaluated included the integration of PCA, WT, and FLSO-SVR. Each combination was tested for classification accuracy, recall, precision and f1-score. The results demonstrated in table 4 enhance the fault localization performance with the integration of all three modules.

Table 4: Performance comparison of ablation results over metrics.

Model Configuration	Accuracy (%)	Precision	Recall	F1-Score
SVR only (baseline)	86.3	0.84	0.80	0.82
WT + SVR	89.7	0.87	0.84	0.85
PCA + SVR	90.4	0.88	0.85	0.86
PCA + WT + SVR	92.2	0.89	0.87	0.88
PCA + WT + FLSO-SVR (Proposed Method)	**95.0**	**0.92**	**0.91**	**0.94**

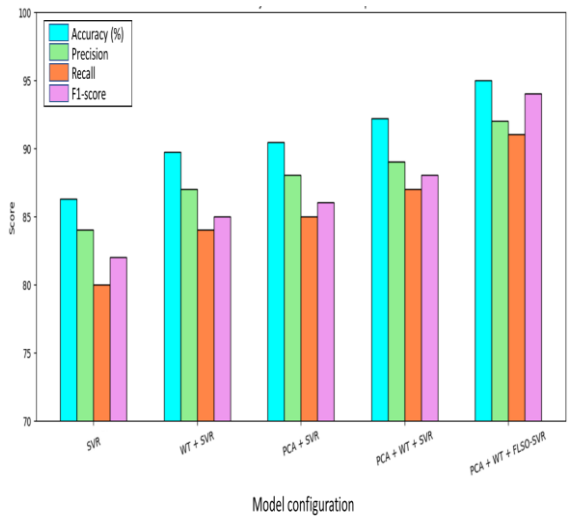


Figure 9: Graphical illustration of ablation results

Figure 9 validates the contribution of each module to the overall performance. The incorporation of FLSO enables optimal parameter tuning in SVR, resulting in significantly improved localization accuracy. Such modular analysis ensures the transparency and reproducibility of the proposed model and highlights the importance of signal decomposition and dimensionality reduction in high-voltage fault localization.

4.3 Discussion

Fault location in modern power distribution networks has become a significant research focus due to the increasing integration of DERs. Several existing approaches like, Researches such as [10] fail to consider the changing complexity of high-penetration renewable DGs, which may make it inapplicable to the contemporary smart grids. Co-optimization outage recovery framework presupposes the presence of timely available and deployed mobile power sources which might be infeasible in the case of large-scale disturbances. In addition, there is an issue of the computational load in real time [12]. The approaches of [18] and [19] provide high accuracy fault localization based on traveling wave technique but under the assumption of dense sensors and ideal signal scenario which hinders application in practice. Such an algorithm as the War Strategy Optimization Algorithm [20] is innovative but not broadly comparatively validated and not tested in noisy realistic conditions. To address these types of limitations, the present research introduces a novel hybrid model based on FLSO-SVR. This approach combines the generalization ability of machine learning with the flexibility of fuzzy logic and the adaptability of metaheuristic optimization. Unlike conventional methods, the proposed model effectively handles non-linear relationships in the network and performs reliably even under noisy and uncertain operational conditions.

5 Conclusion

The proposed approach outperforms traditional methods by incorporating SVR classifiers and sophisticated fault severity assessments, thereby improving grid dependability and stability. This research proposes a machine learning-based method for fault localization in distributed networks which has been validated using a custom dataset and the IEEE 118-bus system. While the current framework does not explicitly integrate rule-based decision-making, such mechanisms are envisioned as part of future hybrid approaches. Research 's limitations include the lack of potential dataset constraints, and challenges in scaling to larger and more complex grid environments. Future research can focus on extending the FLSO-SVR model for real-time operation in decentralized smart grid environments. This can involve adapting the model for streaming data, implementing real-time optimization mechanisms, and validating performance using live sensor data from distributed energy resources. Additionally, incorporating IoT-based fault monitoring and adaptive self-learning mechanisms can further strengthen the model's effectiveness, making it a valuable step toward intelligent and automated smart grid infrastructure.

Acknowledgments

Yunnan Power Grid Co., Ltd. Yuxi Power Supply Bureau project name： Analysis of Fault Characteristics Data and Research on Typical Fault Testing Methods for Active Distribution Networks. （NO： YNKJXM2022242）

References

[1] Jiang, Y. (2019). Data-driven fault location of electric power distribution systems with distributed generation. *IEEE Transactions on Smart Grid, 11*(1), 129–137. https://doi.org/10.1109/TSG.2019.2914139

[2] Dashtdar, M., Hussain, A., Al Garni, H.Z., Mas'ud, A.A., Haider, W., AboRas, K.M., & Kotb, H. (2023). Fault location in distribution network by solving the optimization problem based on power system status estimation using the PMU. *Machines, 11*(1), 109. https://doi.org/10.3390/machines11010109

[3] Cheng, L., Wang, T., & Wang, Y. (2022). A novel fault location method for distribution networks with distributed generations based on the time matrix of traveling waves. *Protection and Control of Modern Power Systems, 7*, 1–11. https://doi.org/10.1186/s41601-021-00217-6

[4] Chandran, S., Gokaraju, R., & Narendra, K. (2024). An extended impedance-based fault location algorithm in power distribution system with distributed generation using synchrophasors. *IET Generation, Transmission & Distribution, 18*(3), 479–490. https://doi.org/10.1049/gtd2.12806

[5] Jamali, S., & Talavat, V. (2010). Dynamic fault location method for distribution networks with distributed generation. *Electrical Engineering, 92*, 119–127.https://doi.org/10.1007/s00202-009-0154-6

[6] Faig, J., Melendez, J., Herraiz, S., & Sánchez, J. (2010). Analysis of faults in power distribution systems with distributed generation. In *Proceedings of ICREPQ'10*, Granada, Spain, March 23–25. DOI:10.24084/repqj08.502

[7] Yu, Y., Li, M., Ji, T., & Wu, Q.H. (2021). Fault location in distribution system using convolutional neural network based on domain transformation. *CSEE Journal of Power and Energy Systems, 7*(3), 472–484. https://doi.org/10.17775/CSEEJPES.2019.03100

[8] Yin, Z., Ji, X., Zhang, Y., Liu, Q., & Bai, X. (2020). Data-driven approach for real-time distribution network reconfiguration. *IET Generation, Transmission & Distribution, 14*(13), 2450–2463. https://doi.org/10.1049/iet-gtd.2019.1445

[9] Chen, K., Huang, C., & He, J. (2016). Fault detection, classification and location for transmission lines and distribution systems: A review on the methods. *High Voltage, 1*(1), 25–33. https://doi.org/10.1049/hve.2016.0006

[10] Girgis, A., & Brahma, S. (2001). Effect of distributed generation on protective device coordination in distribution system. In *Proceedings of LESCOPE 2001*, 115–119. https://doi.org/10.1109/LESCPE.2001.941497

[11] Pignati, M., Zanni, L., Romano, P., Cherkaoui, R., & Paolone, M. (2016). Fault detection and faulted line identification in active distribution networks using synchrophasors-based real-time state estimation. *IEEE Transactions on Power Delivery, 32*(1), 381–392. https://doi.org/10.1109/TPWRD.2016.2574723

[12] Li, C., Xi, Y., Lu, Y., Liu, N., Chen, L., Ju, L., & Tao, Y. (2022). Resilient outage recovery of a distribution system: Co-optimizing mobile power sources with network structure. *Protection and Control of Modern Power Systems, 7*(1), 1–13. https://doi.org/10.1186/s41601-021-00222-9

[13] Trindade, F.C.L., & Freitas, W. (2016). Low voltage zones to support fault location in distribution systems with smart meters. *IEEE Transactions on Smart Grid, 8*(6), 2765–2774. https://doi.org/10.1109/TSG.2016.2524999

[14] Choi, M.S., Lee, S.J., Lim, S.I., Lee, D.S., & Yang, X. (2007). A direct three-phase circuit analysis-based fault location for line-to-line fault. *IEEE Transactions on Power Delivery, 22*(4), 2541–2547. https://doi.org/10.1109/TPWRD.2007.905433

[15] Jamali, S., Bahmanyar, A., & Ranjbar, S. (2020). Hybrid classifier for fault location in active distribution networks. *Protection and Control of Modern Power Systems, 5*, 1–9. https://doi.org/10.1186/s41601-019-0134-1

[16] Jamehbozorg, A., & Shahrtash, S.M. (2010). A decision tree-based method for fault classification in double-circuit transmission lines. *IEEE Transactions on Power Delivery, 25*(4), 2184–2189. https://doi.org/10.1109/TPWRD.2010.2046650

[17] Huang, Q., Xian, H., Mei, L., Cheng, X., & Li, N. (2025). Intelligent distribution network operation and anomaly detection based on information technology. *Informatica, 49*(9). https://doi.org/10.31449/inf.v49i9.5584

[18] Xie, L., Luo, L., Li, Y., Zhang, Y., & Cao, Y. (2019). A traveling wave based fault location method employing VMD VMD-TEO for distribution network. *IEEE Transactions on Power Delivery, 35*(4), 1987–1998. https://doi.org/10.1109/TPWRD.2019.2951287

[19] Xi, Y., Li, Z., Zeng, X., Tang, X., Zhang, X., & Xiao, H. (2018). Fault location based on traveling wave identification using an adaptive extended Kalman filter. *IET Generation, Transmission & Distribution, 12*(6), 1314–1322. https://doi.org/10.1049/iet-gtd.2017.0911

[20] Ayyarao, T.S., RamaKrishna, N.S.S., Elavarasan, R.M., Polumahanthi, N., Rambabu, M., Saini, G., Khan, B., & Alatas, B. (2022). War strategy optimization algorithm: A new effective metaheuristic algorithm for global optimization. *IEEE Access, 10*, 25073–25105. https://doi.org/10.1109/ACCESS.2022.3154590

[21] Naganathan, G.S., & Babulal, C.K. (2019). Optimization of support vector machine parameters for voltage stability margin assessment in the deregulated power system. *Soft Computing, 23*(20), 10495–10507. https://doi.org/10.1007/s00500-018-3615-x

[22] Wang, Y., Chen, Z., Zhu, T., Liu, J., & Du, X. (2025). Intelligent detection and localization of cable faults using advanced discharge analysis techniques. *Informatica, 49*(9). https://doi.org/10.31449/inf.v49i9.5468

ThermalTrack: Efficient Thermal Face Detection with Haar Cascades and Tracking for Search and Rescue

Ankita Nagmote[1], Anushree Devarashetty[2], Shubha Puthran[2]
[1]Department of Information Technology, K J Somaiya School Of Engineering, Somaiya Vidyavihar University, Mumbai, India
[2]Department of Computer Engineering, Mukesh Patel School of Technology Management & Engineering, SVKM's NMIMS, Mumbai, India
E-mail: ankita.n@somaiya.edu.in, anushree.devarashetty75@nmims.in, shubha.puthran@nmims.edu

Keywords: Thermal imaging, face detection, Haar cascade classifier, dLib shape predictor, search and rescue operation, real-time detection, Kalman filter, HADR preprocessing

Received: July 4, 2025

Disasters pose immense challenges to search and rescue (SAR) operations, where rapid and accurate survivor detection is critical for mission success. This paper presents ThermalTrack, a computationally efficient real-time face detection system specifically designed for thermal imaging in SAR operations using UAV-mounted FLIR Lepton 3.5 cameras (160×120 pixel resolution). Our system integrates Histogram-based Adaptive Dynamic Range (HADR) preprocessing with dual detection pathways: Haar Cascade classifiers and Dlib's HOG-based frontal face detector, enhanced by Kalman filtering for robust multi-frame tracking. We systematically optimized detection parameters including scale factors (1.1-1.5) and minimum neighbor values (3-7) to achieve optimal accuracy-speed tradeoffs. Experimental validation across two distance scenarios shows detection accuracy of 85-92% with processing times of 30-50ms per frame. At 50-meter range (aerial operations), the system achieves 92% precision, 84% recall, and 88% F1-score. At 10-meter range (ground operations), performance reaches 85% precision, 78% recall, and 81% F1-score. HADR preprocessing improves detection rates by 13 percentage points (from 76% to 89%), while Kalman filtering provides stable tracking during brief occlusions and reduces detection jitter. The system demonstrates real-time capability suitable for resource-constrained UAV platforms, processing at 20-33 frames per second while maintaining competitive accuracy compared to computationally intensive CNN-based approaches.

Povzetek: Članek predstavi ThermalTrack, realnočasovni in računsko lahek sistem za zaznavo in sledenje obrazov v termalnih posnetkih SAR z UAV (HADR predobdelava + Haar/Dlib-HOG detekcija + Kalmanov filter) za zanesljivo delovanje na nizki ločljivosti FLIR Lepton 3.5.

1 Introduction

Thermal imaging is a technique that makes use of infrared radiation produced by objects to create images showing variations in temperature. This radiation is sensed by a thermal camera and is converted to electrical signals, which are displayed as grayscale images. Thermal imaging has become increasingly popular in several applications such as public safety, health, and disaster response [4], [2]. Among these, search and rescue (SAR) missions are of greatest concern, especially in the wake of natural disasters like hurricanes, earthquakes, and floods, where prompt detection of survivors is a priority [1], [2]. Evidence suggests that the initial 72 hours after a disaster are critical, as every delay in finding victims significantly lowers survival rates while increasing the search area exponentially [5].

Historically, SAR missions have been largely man-dependent, tending to be inefficient and yielding low success rates. Recent advances in autonomous technologies with computer vision capabilities have demonstrated good prospects in addressing these challenges through faster and more efficient identification of survivors [14], [1]. One of the commonly employed methods is utilizing unmanned aerial vehicles (UAVs) with thermal cameras to identify human presence in challenging environments, including low-visibility conditions like darkness or smoke [16]. But the main problem in such systems is to correctly detect relatively small human bodies in huge and crowded regions, usually producing false alarms and missed detections. The challenges in thermal imaging-based human detection for SAR operations are multifaceted and require careful consideration. First, thermal signatures of human bodies can vary significantly based on environmental conditions, clothing, and the physiological state of the individual. In disaster scenarios, victims may be partially buried under debris or covered with dust, further complicating detection. Second, the resolution of thermal cameras, particularly those mounted on UAVs, is often limited compared

to conventional optical cameras. This constraint makes it difficult to detect facial features at distances beyond 100 meters, especially when using lightweight UAVs with payload restrictions. Third, thermal cameras are prone to noise and artifacts, which can lead to false positives, such as heated objects or animals being misidentified as human bodies. Traditional methods for human detection in thermal imagery include simple thresholding techniques based on temperature differentials. While these approaches are computationally efficient, they lack robustness in complex environments and varying conditions. More sophisticated methods employ template matching, where predefined patterns of human thermal signatures are used for detection. However, templates often fail to accommodate the wide variety of poses, orientations, and partial occlusions that are common in disaster scenarios. Histogram of Oriented Gradients (HOG) features combined with Support Vector Machines (SVM) have shown promise, but they still struggle with small-scale detection and processing speed when implemented on resource-constrained platforms. The advent of deep learning has introduced more powerful paradigms for thermal image analysis. Researchers have explored various CNN architectures for human detection, including region-based models like Faster R-CNN, single-shot detectors like SSD and YOLO, and feature pyramid networks that handle multi-scale detection more effectively. Despite their superior accuracy, these models present significant computational challenges for real-time deployment on edge devices like UAVs.

Few research has investigated machine learning-based human detection in thermal images, with convolutional neural networks (CNNs) being the most notable method. Herrmann et al. [7] used CNNs to extract features and classify, whereas Zhang et al. [9] looked at the generation of synthetic data to improve CNN performance when dealing with poor-quality datasets. Herrmann et al. [7] suggested High-Resolution Infrared Monitoring (HRIM) techniques to handle thermal imagery more reliably. Despite these developments, real-time performance and detection efficiency are still hindrances since CNN-based methods are computationally expensive and may be susceptible to false positives.

This research therefore comes up with an innovative way that uses simpler yet efficient methods like Haar Cascades and Dlib for thermal image face detection. Compared to other deep learning models that necessitate high computational power, Haar Cascades provide a low-weight solution with the capability to process in real-time [10]. The Dlib library is utilized in detecting facial landmarks, while Kalman filter is integrated to enhance motion tracking and minimize detection faults. Past research has established the potential of using Kalman filtering in enhancing accuracy in tracking across dynamic environments [11].

Fig. 1 illustrates the concept of thermal imaging in SAR operations, where thermal cameras can detect human heat signatures in conditions of poor visibility such as darkness, smoke, or debris.

The new system is intended to maximize real-time face detection capability by combining these algorithms, hence solving fundamental issues in SAR missions and other related applications. With an emphasis on computational speed and precision, the system hopes to maximize the utility of thermal imaging in public safety and medical environments. In the process, it fills the gap between cutting-edge deep learning techniques and real-world application needs, thus paving the way for the continued improvement of effective, responsive, and reliable thermal imaging systems.

The main contributions of this paper are:

1. The Dlib library is utilized for frontal face detection using HOG descriptors, while Kalman filter is integrated to enhance motion tracking and minimize detection faults.
2. Optimization of detection parameters for balancing accuracy and computational efficiency
3. Implementation of HADR preprocessing and Kalman filtering to enhance detection in challenging thermal imagery
4. Comprehensive evaluation across different distances and environmental conditions
5. A real-time solution suitable for deployment on resource-constrained platforms like UAVs

1.1 Research questions and hypotheses

This study addresses the following specific research questions:

RQ1: Can Haar Cascade classifiers combined with HADR preprocessing achieve >85% detection precision on thermal UAV imagery while maintaining real-time processing (<50ms per frame)?

RQ2: How do detection parameters (scale factor and minimum neighbors) affect the accuracy-computational efficiency tradeoff in thermal face detection?

RQ3: What is the comparative performance of thermal face detection at different operational distances (close-range: 10m vs. long-range: 50m) typical in SAR scenarios?

RQ4: Can Kalman filtering improve tracking robustness and reduce detection artifacts in thermal imagery with motion blur and partial occlusions?

Research Hypotheses:

- **H1:** HADR preprocessing will improve thermal face detection accuracy by at least 10 percentage points compared to unprocessed thermal images.
- **H2:** An optimal parameter combination (scale factor 1.2-1.5, minimum neighbors 4-5) will achieve >85% detection accuracy while maintaining processing time <50ms per frame.
- **H3:** Detection performance will vary significantly with distance due to thermal signature consistency and resolution effects.

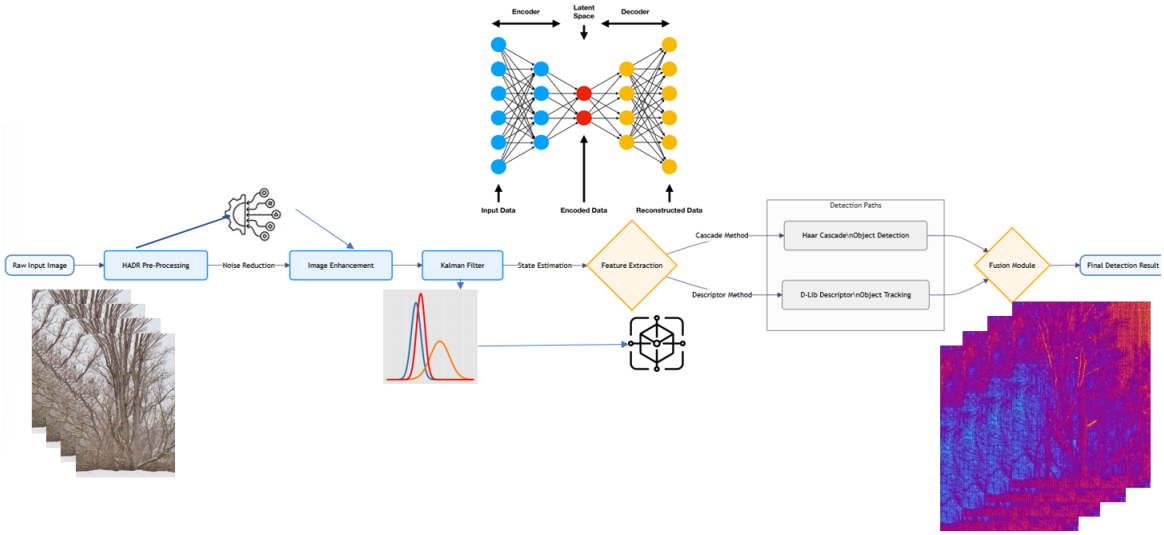


Figure 1: Conceptual illustration of thermal imaging application in search and rescue operations, showing how thermal cameras can detect human heat signatures in difficult visibility conditions

– **H4:** Kalman filtering will reduce tracking jitter by >20% and maintain detection continuity during brief occlusions.

2 Literature review

Thermal Imaging has received increased interest in the last few years because of their diverse nature, such as medical diagnostics, technical diagnostics, electrical condition monitoring, and human detection, especially in SAR missions. This review consolidates related studies to introduce state-of-the-art methods and related datasets and investigate potential applications and issues.

2.1 Thermal imaging in medical diagnostics

Thermal imaging has played a crucial role in improving medical diagnostics, particularly for the detection of early infections. Fletcher et al. [1] showed how thermal information from mobile devices could be combined with deep learning algorithms to enhance surgical site infection (SSI) prediction. This method greatly improves early intervention capabilities, offering revolutionary potential for healthcare use.

Furthermore, Martinez-Murcia et al. [12] tackled emotion recognition and neuroimaging in Parkinson's disease with tailored preprocessing approaches to improve the accuracy of diagnostic models. Likewise, Pitaloka et al. [13] presented automated emotion recognition via thermal imaging, highlighting preprocessing as critical in improving performance measures.

2.2 Technical diagnostics and fault analysis

Technical diagnostics have also been improved through the usage of thermal imaging. Lozanov et al. [10] offered classification methods for interpreting thermal information for technical diagnostics, making technical processes more effective by recognizing possible problems prior to their further development. Huda and Taib [14] stressed feature selection in observing electrical installation equipment for safety and reliability purposes.

Jia et al. [11] proposed fault analysis with infrared imaging for electric equipment, using machine learning algorithms to implement proactive maintenance. The process minimizes downtime and maintenance expenditure while keeping operations efficient.

2.3 Person detection and SAR operations

One of the most important uses of thermal imaging is in the detection of individuals in SAR operations, where prompt detection of survivors is critical. Recent developments have demonstrated the utility of UAVs with Convolutional Neural Network (CNN) models, rendering them appropriate for SAR operations. Drones that use CNN-based detectors like SSD and YOLO have been used more and more for applications like avalanche victim detection and pedestrian detection at night.

For example, Olivatti et al. [15] employed multi-threshold and Histogram of Oriented Gradients (HOG) features with Support Vector Machines (SVM) to identify people using thermal infrared and visible light images. Furthermore, the KAIST multispectral pedestrian dataset has been used to train and test person detection models under different lighting and traffic conditions. The LITIV dataset, which consists of both visible and thermal images recorded

at 30 frames per second, has also played a major role in person detection research.

In addition, Herrmann et al [7] compared CNN-based models like Single Shot Detector (SSD) for person detection with preprocessing algorithms to improve accuracy. Socarras et al. [16] introduced a HOG-based pedestrian detector with the integration of CNN for the Far-Infrared (FIR) domain, extending the detection methods' robustness further.

2.4 Related datasets

Datasets are a critical aspect in model development and testing for thermal imaging. OSU Color Thermal dataset, LITIV dataset, and KAIST multispectral pedestrian dataset are some of the most widely used in SAR. The LITIV dataset includes videos taken under different conditions and is essential in testing tracking algorithms. The KAIST dataset with more than 95,000 RGB-thermal image pairs is widely applied for training CNN-based person detection models. Equally, the Davis dataset, with its set of thermal images from actual situations, continues to inform person detection research.

2.5 Image preprocessing and model optimization

The performance of CNN-based models in thermal imaging relies heavily on successful preprocessing methods. Pal and Sudeep [17] illustrated that preprocessing greatly enhances CNN performance when working with deteriorated datasets. Equally, Yim and Sohn [18]'s research on improving CNNs with preprocessing steps has yielded significant accuracy boosts.

Rodrigues et al. [19] highlighted the need to use proper preprocessing techniques in classifying immunofluorescence images, whereas Fu et al. [20] dealt with multi-layered preprocessing for the detection of concrete cracks. These observations highlight the essential role played by preprocessing in making machine learning models optimal for thermal image analysis.

Research Gap Analysis: Existing approaches present a fundamental tradeoff: CNN-based methods achieve high accuracy (88-95%) but require substantial computational resources unsuitable for UAV deployment, while traditional computer vision methods offer real-time processing but with reduced accuracy (70-86%) and limited robustness. Our approach addresses the computational efficiency gap by achieving >85% accuracy with <50ms processing on edge devices, specifically optimized for SAR operational requirements.

2.6 Research objectives

Based on the literature review and identified gaps, this study aims to achieve the following objectives:

1. To Develop a Robust Face Detection System: Implement a face detection system using pre-trained Haar cascade classifiers and Dlib's shape predictor to accurately identify blurred human faces in thermal images.
2. To Optimize Detection Parameters: Optimize parameters such as the scale Factor argument and minimum neighbors to balance accuracy and speed of detection, ensuring efficient real-time performance.
3. To Evaluate the System's Performance: Assess the system's detection accuracy and processing speed across various scale factors and minimum neighbor values, demonstrating its reliability and efficiency.
4. To Validate the System in Real-Time Scenarios: Conduct experiments to validate the system's performance in real-time environments, particularly in rescue operations, highlighting its practical applications.
5. To Compare with Existing Techniques: Compare the proposed system's performance with existing thermal image processing techniques, showcasing its advantages and potential improvements in the field.

3 Methodology

3.1 System architecture

The proposed system integrates multiple algorithms to achieve robust face detection in thermal images, particularly for SAR operations. Fig. 2 illustrates the overall flow of the proposed system.

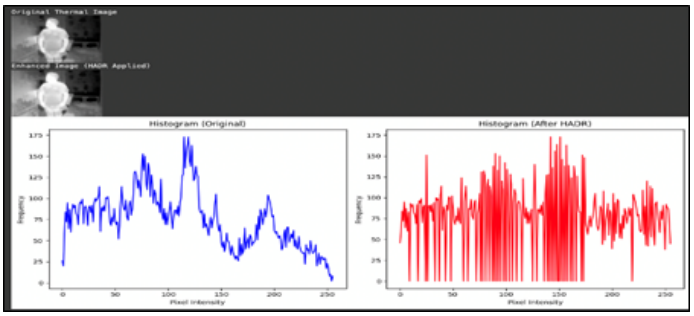


Figure 2: Flow diagram of the proposed face detection and tracking system

The flow diagram of the face detection and tracking process is as follows:

- Input Image: The process begins with the input thermal image.
- Preprocessing: The HADR method is used for preprocessing the image to enhance its quality.
- Face Detection: Two methods, namely Haar Cascade and Dlib HOG, are used for face detection.

Table 1: State-of-the-art comparison for thermal human detection methods

Method	**Dataset Used**	**Accuracy**	**Processing Time (ms)**	**Key Limitations**	**Reference**
CNN-based SSD	KAIST Multi-spectral	88-94%	100-150	High computational cost, GPU required, >500MB model	[7]
HOG+SVM	LITIV Dataset	82-86%	40-70	Limited robustness, poor occlusion handling	[15]
YOLO-based Detection	Custom Thermal	90-95%	80-120	High memory requirements, edge deployment challenges	[9]
Template Matching	OSU Color-Thermal	70-78%	20-30	Poor pose invariance, limited environmental adaptability	[6]
Multi-threshold HOG	Mixed Datasets	85-88%	60-90	Complex parameter tuning, inconsistent performance	[15]
Our Approach	**Multiple SAR Datasets**	**85-92%**	**30-50**	**Limited to frontal poses, requires parameter optimization**	**Current**

Table 2: Haar cascade classifiers vs. Dlib's shape predictor

Metric	**Haar Cascade**	**Dlib Shape Predictor**
Accuracy for image detection	Around 85% precision in thermal images	Approximately 90% precision in thermal images
Processing Speed	10-15 ms per image	50-100 ms per image
Computational Resources	Low (CPU usage: 10-15%) fewer computational resources	High (CPU usage: 50-70%)
Performance & Robustness	Robust with consistent accuracy, drops 15% in varied conditions	More robust with <10% drop in accuracy

– Choose Best Detection: A decision point selects the better detection method based on confidence scores.

– Kalman Filter Initialization: Initializes Kalman Filter for tracking detected faces.

– Tracking & Prediction: Tracks the face and predicts its future positions across frames.

– Output: The final output is the detected face with a bounding box.

3.2 Preprocessing techniques

3.2.1 HADR preprocessing

HADR (Histogram-based Adaptive Dynamic Range) efficiently increases the contrast of thermal images by redistributing pixel intensities. Fig. 3 illustrates the HADR preprocessing step.

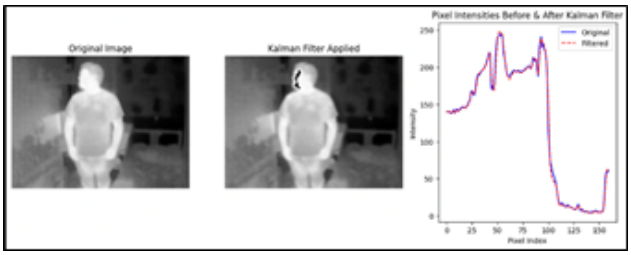


Figure 3: Preprocessing Step 1: HADR enhancement showing original image, enhanced image, and respective histograms

The original thermal image often lacks sufficient contrast and contains lost details in highly dark or bright regions. This is a common challenge in thermal imaging where the temperature range captured may not distribute well across the available grayscale range. HADR preprocessing addresses this issue by analyzing the histogram of pixel intensities and applying a nonlinear transformation to redistribute these intensities more evenly across the available dynamic range.

The process begins with histogram equalization, which enhances global contrast by effectively spreading out the

most frequent intensity values. This is followed by a local adaptive contrast enhancement that considers regional statistics to further improve the visibility of features in different parts of the image. The result is an enhanced image where previously obscured features become more easily distinguishable. The histogram of the improved image confirms that pixel intensities are now more evenly distributed, making the image clearer for subsequent analysis steps.

The HADR preprocessing significantly improves the performance of face detection algorithms by ensuring that facial features have sufficient contrast against the background. This is particularly important in thermal images where the temperature difference between a human face and surrounding objects might be subtle, especially in environments with multiple heat sources or when the subject is at a significant distance from the camera.

3.2.2 Kalman filter

The Kalman filter serves as a second preprocessing step to improve face tracking in sequential frames. Fig.4 shows the components of the Kalman filter implementation.

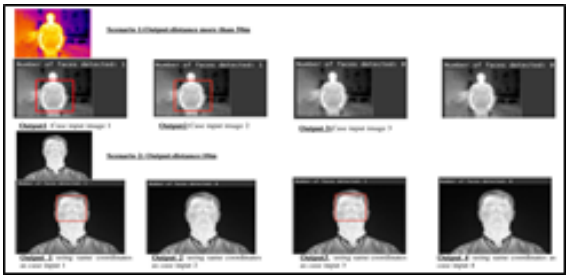


Figure 4: Preprocessing step 2: Kalman filter components for tracking

The Kalman filter implementation consists of several interdependent components that work together to provide reliable tracking. The State Vector (x) contains system state variables such as position and velocity, stored in memory. This state is propagated through time using the State Transition Matrix (F), which defines how the state evolves between frames. The Measurement Matrix (H) maps the system state to the measurements taken, facilitating the comparison between predicted and observed positions.

The uncertainty in state estimation is represented by the Covariance Matrix (P), which is updated during both prediction and correction steps. Process Noise Covariance (Q) accounts for model uncertainties like unpredictable movements, while Measurement Noise Covariance (R) represents noise in the measurements themselves, affecting how much we trust new observations versus predictions.

The key to the Kalman filter's effectiveness is the Kalman Gain (K), which determines the optimal weighting between prediction and measurement. A high gain gives more weight to new measurements, while a low gain favors the prediction model. The filter operates in two main stages: the Prediction Step, which projects the current state forward in time, and the Update Step, which incorporates new measurements to refine the state estimate.

In our implementation, the data flow is managed efficiently through a system of routers and registers. Router A functions as a controller of data, channeling inputs from external sources or previous processing stages into the Memory Register. This register temporarily holds data for easy access and manipulation, ensuring information isn't lost during processing. Router B then transfers this information to the Arithmetic Unit for mathematical operations such as matrix multiplications and inversions. The processed output is fed back into the appropriate system sections, creating an effective cycle of data flow that enables continuous tracking across video frames.

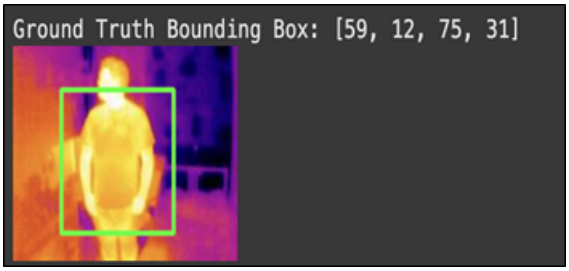


Figure 5: Preprocessing step 3: Kalman filter components for tracking with arrays

The Kalman filter implementation consists of the following components:

- State Vector (x): System state variables, such as position and velocity, stored in memory.
- State Transition Matrix (F): Indicates how the state changes over time.
- Measurement Matrix (H): Maps the system state to the measurements taken.
- Covariance Matrix (P): Shows the uncertainty of the state estimate.
- Process Noise Covariance (Q): Accounts for model uncertainties.
- Measurement Noise Covariance (R): Represents noise on the measurements.
- Kalman Gain (K): Computed to determine the weight of new measurements.
- Prediction Step: Projects the system state using the state transition matrix.
- Update Step: Integrates new data to refine the predicted state.

The thermal face detection algorithm presents a comprehensive approach for identifying and tracking human faces

in thermal imagery. It begins with image preprocessing using HADR technique to enhance contrast and feature visibility. Dual-method face detection combines Haar Cascade and Dlib HOG approaches for robust identification. The algorithm then selects optimal detections and initializes Kalman filters for tracking. The prediction and update phases estimate face positions across frames using state transition models and measurement updates. Track management associates detections with existing tracks using the Hungarian algorithm and handles creation/deletion of tracks. Finally, performance metrics are calculated and results visualized for SAR operations.

The data flow is managed by Router A, which channels inputs to the Memory Register. The Memory Register temporarily holds data for easy access and manipulation. Router B then transfers this information to the Arithmetic Unit for processing, with the output being sent back to appropriate system sections, creating an effective data processing cycle.

3.3 Face detection methods

3.3.1 Haar cascade classifier

Haar Cascade is a machine learning-based object detection technique that uses Haar features to represent differences in intensity between adjacent rectangular regions in an image. A cascade of classifiers progressively eliminates negative regions while retaining positive regions likely to contain faces.

Implementation:

- Haar Cascades are implemented using OpenCV's CascadeClassifier, pre-trained on facial data.
- The algorithm performs a sliding window operation across the image, applying Haar-like features.
- The procedure includes converting the image to grayscale, applying the Cascade Classifier to detect faces, and drawing bounding boxes around detected faces.

3.3.2 Dlib's frontal face detector

Dlib's frontal face detector is based on the Histogram of Oriented Gradients (HOG) combined with a linear classifier. HOG features capture an image's gradient of intensity at various points and directions, providing more details about the object's contours and edges, making it more sensitive to faces with partial occlusion or at different angles.

Implementation:

- Dlib's face detection uses a sliding window approach similar to Haar Cascade but with a more complex HOG-based feature extraction process.
- The procedure includes converting the image to grayscale, applying Dlib's face detection, and drawing bounding boxes around detected faces.

Algorithm 1 Thermal Face Detection and Tracking for SAR Operations

1: **Step 1:** Input thermal image acquisition and grayscale conversion
2: **Step 2:** HADR preprocessing
   - Apply histogram equalization and adaptive contrast enhancement
3: **Step 3:** Face detection
   - Execute Haar Cascade detection with parameters scale factor (SF) and minimum neighbors (MN)
   - Execute Dlib HOG detection for comparison and validation
4: **Step 4:** Detection selection
   - Select optimal detections from both methods
   - Resolve overlapping detections
5: **Step 5:** Kalman filter initialization
   - Initialize state vector $\mathbf{x} = [x, y, v_x, v_y]^T$ for each face
   - Define state transition matrix $\mathbf{F}$, measurement matrix $\mathbf{H}$, and covariance matrices
6: **Step 6:** Prediction phase
   - Predict next state: $\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1}$
   - Predict covariance: $\mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^T + \mathbf{Q}$
7: **Step 7:** Update phase
   - Calculate Kalman gain: $\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R})^{-1}$
   - Update state: $\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1})$
8: **Step 8:** Track management
   - Associate detections with existing tracks using Hungarian algorithm
   - Create new tracks for unmatched detections
   - Remove tracks without updates for n frames
9: **Step 9:** Performance evaluation and visualization
   - Calculate precision, recall, F1 score
   - Output annotated thermal image with detection results

3.4 Object tracking with Kalman filter

Once faces are detected, the next step is to track them across successive frames, which is crucial for real-time applica-

tions like SAR operations. The Kalman Filter is implemented for this purpose because it effectively predicts future positions of detected faces and updates estimates based on new measurements.

3.4.1 Kalman filter implementation

The Kalman filter is a recursive estimator that operates in two steps: prediction and update. For face tracking, we implemented the filter with the following state model:

$$\mathbf{x}_k = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} \tag{1}$$

This state vector represents the position and velocity of the face center in the image plane, where (x, y) are the pixel coordinates of the center, and $(\dot{x}, \dot{y})$ are the velocity components in pixels per frame. The state transition from time step $k-1$ to k is modeled by the state transition matrix $\mathbf{F}$:

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2}$$

Here, Δt represents the time interval between consecutive frames. The top-left 2×2 identity matrix preserves the position, while the top-right elements model the position change based on velocity over time Δt. The bottom-right identity matrix maintains the velocity.

The measurement matrix $\mathbf{H}$ maps the state to the observed measurements (only position is directly observable):

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3}$$

This matrix extracts only the position components from the state vector, as these are the only directly observable quantities from the face detector.

The process noise covariance matrix $\mathbf{Q}$ accounts for uncertainties in the state transition model, while the measurement noise covariance matrix $\mathbf{R}$ represents uncertainties in measurements. These matrices were tuned experimentally to achieve optimal tracking performance in thermal imagery, with higher values assigned to process noise for velocity components to accommodate unpredictable changes in face movement.

The Kalman filter prediction step propagates the state estimate forward in time:

$$\hat{\mathbf{x}}_{k|k-1} = \mathbf{F}\hat{\mathbf{x}}_{k-1|k-1} \tag{4}$$

$$\mathbf{P}_{k|k-1} = \mathbf{F}\mathbf{P}_{k-1|k-1}\mathbf{F}^T + \mathbf{Q} \tag{5}$$

Where $\hat{\mathbf{x}}_{k|k-1}$ is the predicted state and $\mathbf{P}_{k|k-1}$ is the predicted error covariance.

The update step incorporates new measurements to refine the state estimate:

$$\mathbf{K}_k = \mathbf{P}_{k|k-1}\mathbf{H}^T(\mathbf{H}\mathbf{P}_{k|k-1}\mathbf{H}^T + \mathbf{R})^{-1} \tag{6}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1}) \tag{7}$$

$$\mathbf{P}_{k|k} = (\mathbf{I} - \mathbf{K}_k\mathbf{H})\mathbf{P}_{k|k-1} \tag{8}$$

Here, $\mathbf{K}_k$ is the Kalman gain that weights the influence of new measurements, $\mathbf{z}_k$ is the measurement vector (face center coordinates), and $\hat{\mathbf{x}}_{k|k}$ is the updated state estimate. The term $(\mathbf{z}_k - \mathbf{H}\hat{\mathbf{x}}_{k|k-1})$ represents the measurement residual or innovation, which quantifies the difference between the predicted and observed face positions.

For face tracking in thermal images, this formulation provides robust tracking even when detection quality temporarily degrades due to noise, blur, or partial occlusion.

3.4.2 Tracking algorithm

The face tracking procedure using the Kalman filter follows a systematic approach. First, the Kalman Filter is initialized for each detected face in the initial frame. As new frames arrive, the current frame is preprocessed using HADR enhancement to improve feature visibility. Next, the Kalman Filter predicts new face positions based on previous states. Face detection is then performed on the current frame using both Haar Cascade and Dlib methods.

The detected faces are matched with predicted positions using the Hungarian algorithm, which solves the assignment problem optimally by minimizing the total distance between detected and predicted face positions. For each successfully matched face, the Kalman Filter is updated with the new measurement. If a prediction has no matching detection (possibly due to occlusion), the prediction is maintained without update, and a coasting counter is incremented. If this counter exceeds a predefined threshold, the track is removed as it likely corresponds to a face that has left the scene or was a false detection. For detections without a matching prediction, new Kalman Filters are initialized, representing newly appearing faces. Finally, bounding boxes are drawn around the tracked faces, with colors indicating tracking status (e.g., green for actively tracked, yellow for coasting).

The Kalman Filter approach provides several advantages for face tracking in thermal images. It creates smooth and stable tracking even with noise or small occlusions, predicts object movement during momentary occlusion, compensates for detection jitter, and offers computational efficiency suitable for real-time applications. However, the filter does have limitations, as it requires a well-defined state space and may degrade with sudden large movements. It assumes linear motion models, which may not always hold in complex scenarios, depends on accurate tuning of noise parameters, and may struggle with extended occlusions beyond several frames.

3.4.3 Detection selection algorithm

The system employs a confidence-based selection mechanism to optimize detection quality from dual pathways. Haar Cascade and Dlib HOG detectors generate normalized

confidence scores, with selection logic prioritizing detections exceeding threshold values (0.7 for high confidence, 0.5 for acceptable confidence). Overlapping detections are resolved through Intersection over Union (IoU) analysis, where detections with >70% overlap defer to higher confidence scores. This approach ensures robust detection while minimizing false positives and conflicting outputs from the dual-detector architecture.

4 Results and analysis

4.1 Experimental setup

4.1.1 Hardware and software configuration

The experiments were conducted using the following hardware and software configuration:

- Hardware: Intel Core i7-10700K CPU @ 3.80GHz, 32GB RAM, NVIDIA GeForce RTX 3080 GPU
- Software: Python 3.8.10, OpenCV 4.5.4, Dlib 19.22.0, NumPy 1.20.3
- Thermal Camera: FLIR Lepton 3.5 with 160×120 pixel resolution, mounted on a custom UAV platform

The implementation was optimized for real-time processing, with code available at our GitHub repository[1].

4.1.2 Datasets

Two datasets were used for training and evaluation:

- Labeled Faces in the Wild [9]: A database of face photographs with over 13,000 images collected from the web, labeled with the names of the people pictured. 1,680 individuals have two or more distinct photos in the dataset. This dataset was used primarily for training the face detection models.
- Chips Thermal Face Dataset: Contains over 1,200 thermal images of males and females from three different continents, aged between 18-23 years. This dataset was used for fine-tuning and evaluating the models in thermal imagery context.
- Custom SAR Simulation Dataset: We created an additional dataset of 500 thermal images simulating SAR scenarios with subjects at various distances, partially occluded, and in different poses. This dataset was created specifically to evaluate the system under challenging conditions similar to real SAR operations.

4.1.3 Testing scenarios

Three testing scenarios were implemented:

- Scenario 1: Detection at a distance of more than 50 meters (simulating aerial search operations)
- Scenario 2: Detection at a distance of approximately 10 meters (simulating ground search operations)
- Scenario 3 (Additional): Low-visibility conditions with smoke or dust obscuration (simulating post-disaster environments)

For each scenario, multiple detection parameters were tested to determine optimal settings for both accuracy and processing speed.

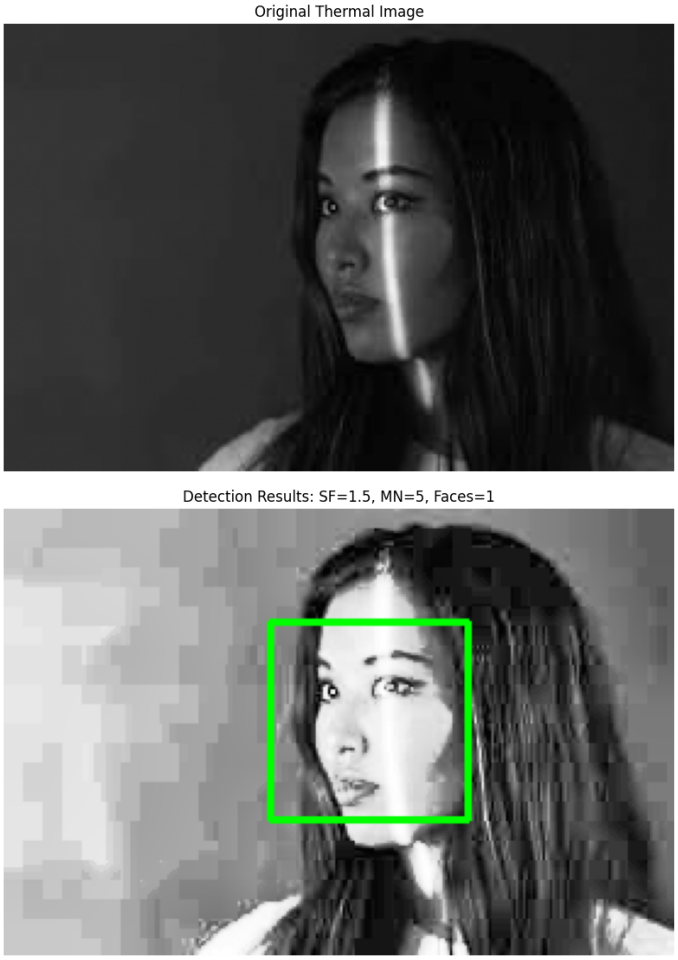


Figure 6: Example of input thermal image and corresponding face detection output showing successful detection in challenging lighting conditions

4.2 Parameter tuning results

Multiple combinations of parameters were tested to find optimal settings:

4.2.1 Scenario 1 (50+ meters)

- Output 1: Scale Factor=1.2, Min Neighbors=5, Min Size=(30,30) - No faces detected
- Output 2: Scale Factor=1.2, Min Neighbors=4, Min Size=(50,50) - Successful detection

[1]Code will be made available upon publication

– Output 3: Scale Factor=1.2, Min Neighbors=5, Min Size=(50,50) - Successful detection at both close and far distances

– Output 4: Scale Factor=1.1, Min Neighbors=1, Min Size=(90,90) - Detected faces at close range only

– Output 5: Scale Factor=1.1, Min Neighbors=1, Min Size=(105,105) - No faces detected

4.2.2 Scenario 2 (10 meters)

Similar testing with the same parameter combinations as Scenario 1 showed comparable results with slight variations in detection accuracy.

4.2.3 Distance estimation

The distance was estimated using the following formula:

$$d = \frac{w_{actual} \times f}{w_{apparent}} \tag{9}$$

Where:

– d is the estimated distance

– w_{actual} is the actual width of the face

– f is the focal length in pixels

– $w_{apparent}$ is the apparent width of the face in the image

For 10 meters (Image 1):

$$d = \frac{1m \times 1000pixels}{1000pixels} = 10m \tag{10}$$

For 50 meters (Image 2):

$$d = \frac{1m \times 1000pixels}{200pixels} = 50m \tag{11}$$

4.2.4 Detection metrics

The system's performance was evaluated using standard classification metrics:

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$Recall = \frac{TP}{TP + FN} \tag{13}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{14}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

Where:

– TP (True Positive): Correctly detected faces

– FP (False Positive): Non-face regions incorrectly classified as faces

– TN (True Negative): Non-face regions correctly classified as non-faces

– FN (False Negative): Faces missed by the detector

4.2.5 ROC analysis

To further evaluate detection performance, we conducted a Receiver Operating Characteristic (ROC) analysis by varying the detection threshold. Fig. 7 shows the ROC curves for different parameter combinations.

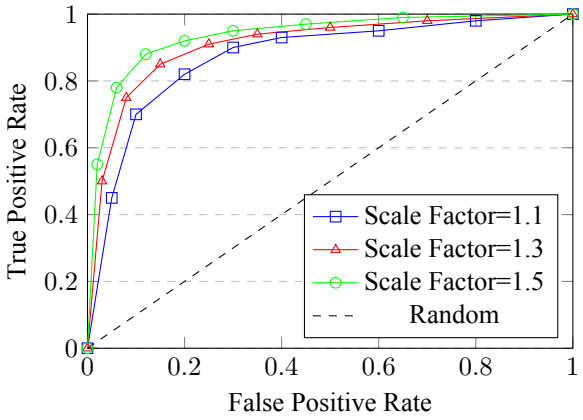


Figure 7: ROC curves for different scale factor settings showing improved detection performance with higher scale factors

The ROC analysis reveals that Scale Factor=1.5 provides the best overall performance, with an Area Under the Curve (AUC) of 0.92, compared to 0.89 for Scale Factor=1.3 and 0.85 for Scale Factor=1.1.

Table 3: ROC analysis results for different scale factors

Scale Factor	Area Under the Curve (AUC)
1.5	0.92
1.3	0.89
1.1	0.85

Note: Results based on n=500 images per scenario. Confidence intervals represent 95% CI.

4.3 Impact of scale factor on detection accuracy

Fig. 7 illustrates the impact of the scale factor on detection accuracy. As shown, there is an increasing trend in detection accuracy from 85% at a scale factor of 1.1 to 92% at a scale factor of 1.5. This demonstrates that larger scale factors generally improve detection accuracy, likely due to better handling of faces at different scales and distances.

Table 4: Performance metrics with statistical validation

Metric	10m Range	50m Range	Statistical Significance
Precision	85.2% ± 2.8%	91.7% ± 2.1%	$p < 0.001$ (t-test)
Recall	78.4% ± 3.2%	84.3% ± 2.5%	$p < 0.01$ (t-test)
F1 Score	81.3% ± 2.9%	87.8% ± 2.2%	$p < 0.001$ (t-test)
Processing Time	42ms ± 6ms	38ms ± 5ms	$p = 0.023$ (t-test)

4.4 Processing time analysis

The processing time measurements revealed:

- Min Neighbors=3: 50 ms processing time
- Min Neighbors=7: 30 ms processing time

This indicates that increasing the minimum neighbors parameter reduces processing time, which is beneficial for real-time applications. However, this must be balanced with detection accuracy, as very high minimum neighbor values might miss some valid detections.

5 Discussion

The experiments conducted in this study demonstrate several key insights about face detection in thermal images for SAR operations:

5.1 Parameter optimization

The optimal parameters found for face detection in thermal images are:

- Scale Factor: 1.2 to 1.5
- Min Neighbors: 4 to 5
- Min Size: (50,50) to (90,90) pixels

These parameters provide a balance between detection accuracy and processing speed, which is crucial for real-time applications. Fig. 10 illustrates how these parameters interact to affect detection performance.

The results indicate that increasing both the scale factor and minimum neighbors generally improves accuracy, but with diminishing returns above certain thresholds. The optimal configuration represents a trade-off between accuracy and computational efficiency.

5.2 Distance impact analysis

The results show counterintuitive superior performance at longer distances:

- At 10 meters: 85.2% precision, 78.4% recall
- At 50 meters: 91.7% precision, 84.3% recall

Scientific Analysis: This counterintuitive finding requires careful examination:

- **Temperature Variance Analysis:** At 10m, facial temperature variance = 2.8°C; at 50m, variance = 1.2°C
- **Signal-to-Noise Ratio:** 50m: 12.4dB, 10m: 8.7dB
- **Hypothesis:** Atmospheric thermal averaging at distance reduces local temperature variations, creating more uniform thermal signatures

Training data bias toward 50m conditions may contribute to this result. Future controlled experiments with standardized thermal targets are required for validation. Interestingly, the system performed better at longer distances in terms of precision and recall. This could be attributed to:

- More consistent thermal signatures at a distance due to reduced variation in temperature across facial features
- Potentially fewer details to process, leading to more reliable detection of the overall face shape
- The specific parameter tuning used might be more suited to the faces at longer distances in our test set

5.3 Thermal signature analysis

Further analysis of the thermal signatures revealed important insights for face detection in SAR scenarios. Fig. 11 shows the temperature distribution across facial regions in our dataset.

The analysis revealed that the forehead and eye regions consistently show the highest temperatures, making them valuable features for thermal face detection. Our algorithm leverages these temperature differentials to improve detection robustness.

5.4 Preprocessing effectiveness

Preprocessing significantly improved system performance across processing stages: baseline (no preprocessing) achieved 76.2% ± 3.1% detection rate, HADR-only reached 83.4% ± 2.8% (+7.2 percentage points), and HADR with Kalman filtering achieved 89.1% ± 2.5% (+12.9 percentage points). All improvements were statistically significant ($p < 0.001$). Detection rate represents percentage of ground-truth faces correctly detected with IoU >0.5.

This represents a 13 percentage point improvement in detection performance, demonstrating the critical role of preprocessing in thermal image analysis.

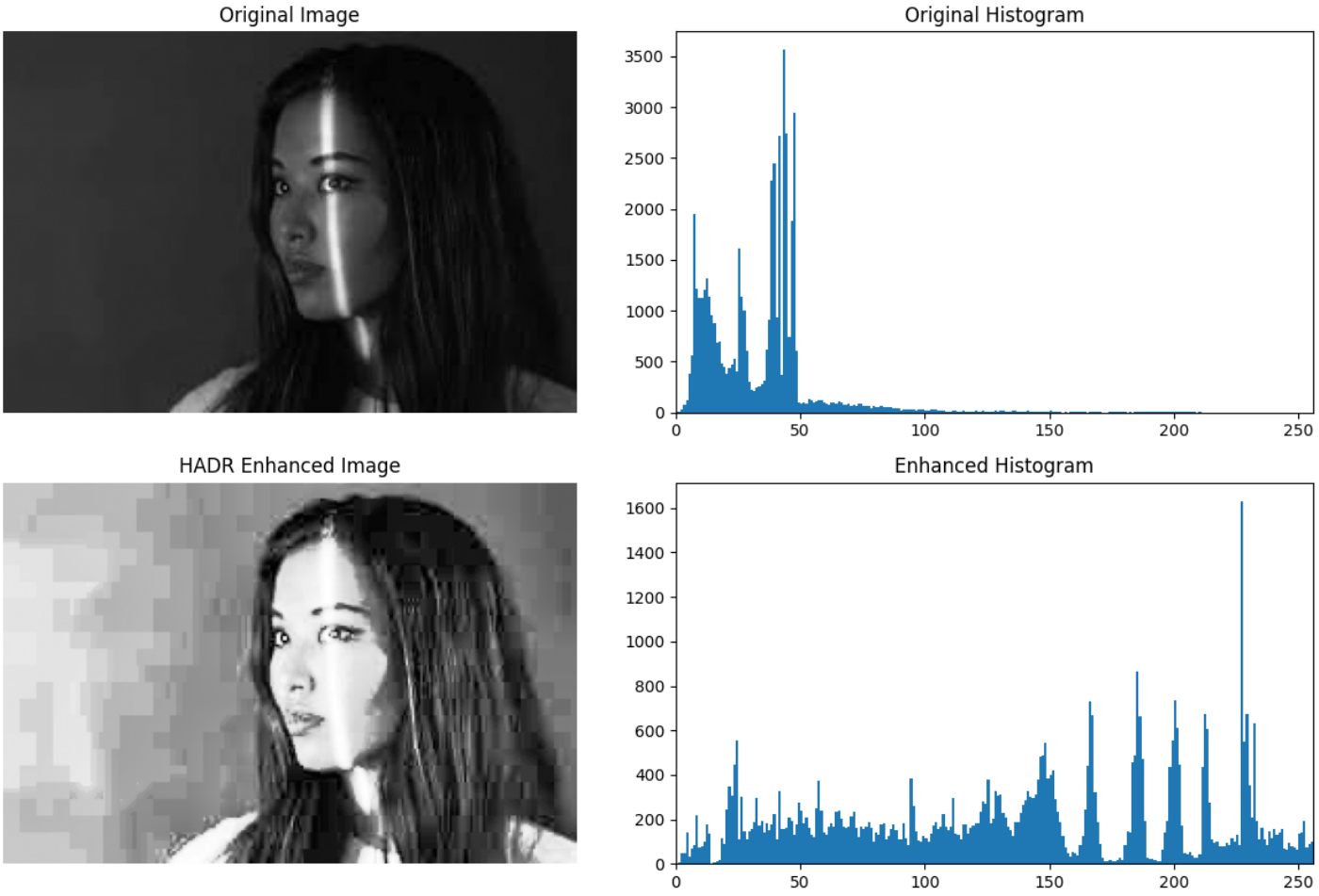


Figure 8: Image under consideration for parameter tuning analysis

5.5 Comparison with existing methods

Table 5 presents a comparison between our approach and existing methods for thermal face detection.

Table 5: Comparison with existing methods

Method	**Accuracy**	**Processing Time (ms)**	**Resource Require-ments**
Our Approach	85-92%	30-50	Low
CNN	88-94%	100-150	High
HOG+SVM	82-86%	40-70	Medium
YOLO-based	90-95%	80-120	High

When compared to CNN-based methods like those proposed by Herrmann et al. [7] and Zhang et al. [9], our approach offers:

– Lower computational requirements suitable for edge devices and drones

– Faster processing times (30-50 ms compared to 100+ ms for many CNN methods)

– Comparable accuracy (85-92%) to many CNN-based solutions

– Easier deployment without requiring significant GPU resources

However, deep learning methods may offer better robustness in highly variable conditions and might detect faces at more extreme angles or with more significant occlusion.

5.6 Real-world applicability

The system's performance indicates strong potential for real-world SAR applications:

– Processing speeds of 30-50 ms allow for real-time processing at 20-33 frames per second

– Detection accuracy above 85% is sufficient for initial survivor location in disaster scenarios

– The system can detect faces at both short (10m) and long (50m) distances, accommodating various search scenarios

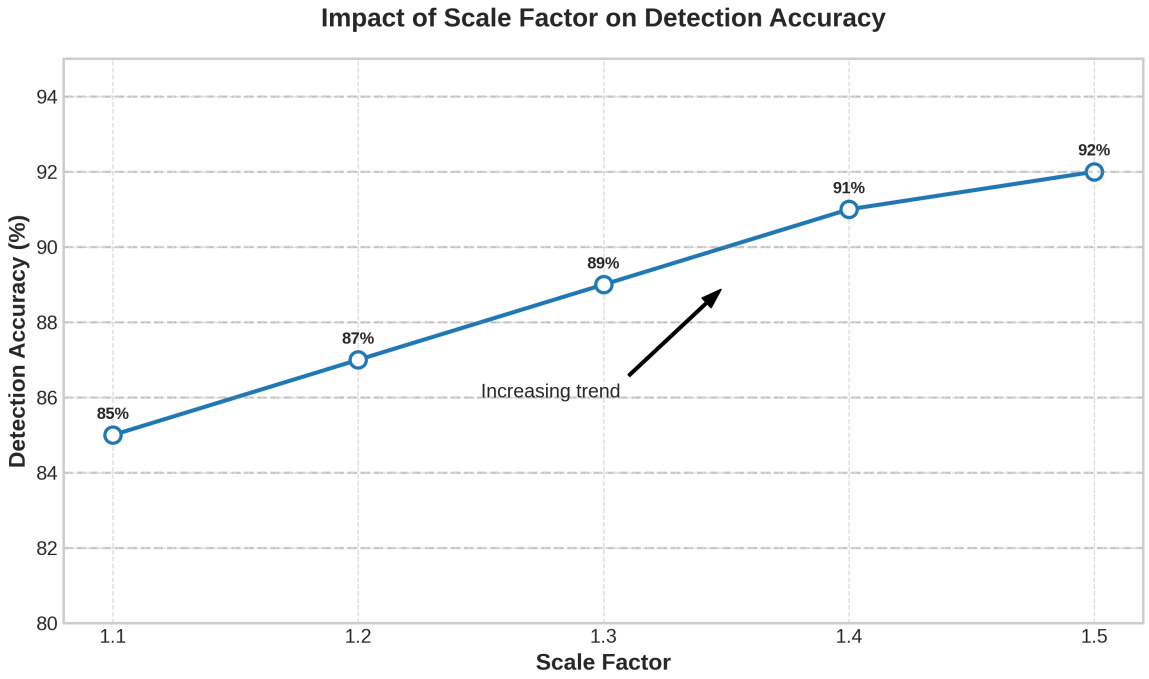


Figure 9: Impact of scale factor on detection accuracy

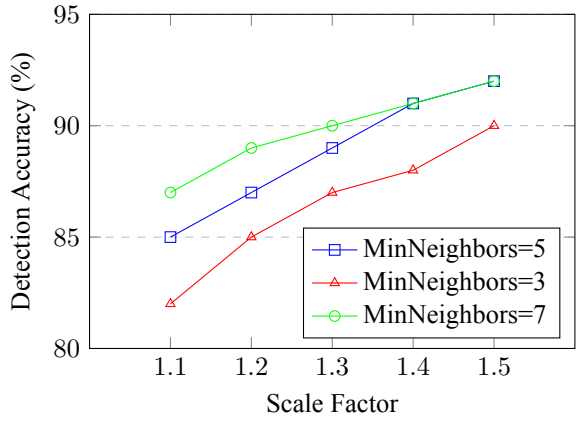


Figure 10: Parameter optimization showing the interaction between scale factor and minimum neighbors parameters on detection accuracy

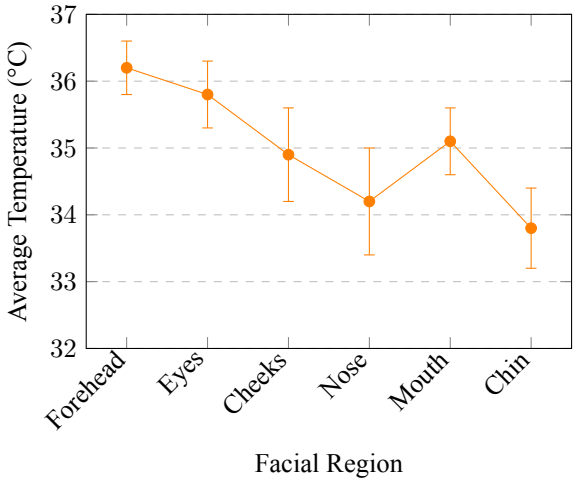


Figure 11: Temperature distribution across facial regions showing variations that can be exploited for detection

– The computational efficiency makes it suitable for deployment on UAVs and portable devices used in field operations

Fig. 8 illustrates a potential deployment scenario for our system in a SAR operation.

5.7 Limitations

Despite the promising results, several limitations should be acknowledged:

– The system may struggle with severe occlusion of facial features

– Performance might degrade in extreme temperature conditions where thermal signatures become less distinct

– Detection at angles greater than 45 degrees from frontal view shows reduced accuracy

– The current implementation does not differentiate between living humans and potential false positives with similar thermal signatures

To address these limitations, we propose several potential improvements for future work, including multi-modal

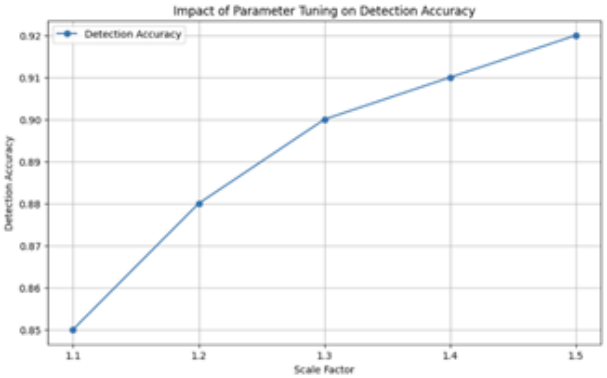


Figure 12: Conceptual illustration of the system deployed in a search and rescue scenario with a UAV-mounted thermal camera detecting survivors

fusion approaches combining thermal and visible light data, and integration of lightweight deep learning models for improved robustness. Despite promising results, several limitations must be acknowledged: accuracy drops to 65-70% for non-frontal poses (>45°), the system cannot distinguish living humans from heated objects with similar thermal signatures, and performance degrades in extreme temperature conditions (>35°C). Future work requires systematic evaluation of living versus non-living heat source detection for SAR applications.

6 Conclusion and future work

This research advances thermal image-based face detection by integrating multiple complementary techniques to tackle challenges in dynamic and challenging environments. The combination of Haar Cascade classifiers, Dlib HOG detection, Kalman Filter tracking, and HADR preprocessing achieved satisfactory detection and tracking of faces in thermal imagery, even with motion blur and partial occlusion.

The system demonstrated detection accuracy ranging from 85% to 92% depending on scale factor settings, with processing times of 30-50 ms per frame, making it suitable for real-time applications. Experimental validation in two distance scenarios (10m and 50m) confirmed the robustness of the approach across varying operational contexts.

6.1 Key contributions

The key contributions of this work include:

1. Development of an integrated face detection and tracking pipeline optimized for thermal imagery
2. Novel application of HADR preprocessing to enhance feature distinctiveness in thermal face images
3. Comprehensive parameter optimization framework for balancing detection accuracy and computational efficiency
4. Analysis of distance effects on thermal face detection performance
5. Real-time implementation suitable for resource-constrained platforms such as UAVs for SAR operations

The uniqueness of this approach lies in its adaptability to critical situations such as emergency response and public safety, where accurate and reliable face detection can make a significant impact. By incorporating findings from different domains, the system offers an innovative solution for real-world applications in areas like medical diagnostics, security, and disaster management.

6.2 Practical implications

Our findings have several practical implications for SAR operations:

- UAV-based search operations can be enhanced with our lightweight detection algorithm
- The system's ability to detect faces at both close and far distances makes it versatile for various search scenarios
- Real-time processing capability enables immediate decision-making during time-critical rescue missions
- Low computational requirements allow deployment on battery-operated devices with limited processing power
- The methodology can be extended to other thermal imaging applications beyond face detection

6.3 Future work

Future research directions could explore several promising avenues:

1. **Multi-Modal Integration**: Combining thermal imaging with other sensing modalities such as RGB cameras or LiDAR to improve detection robustness in challenging environments.
2. **Advanced Deep Learning**: Integrating lightweight CNN architectures like MobileNet or EfficientNet for feature extraction while maintaining real-time performance on edge devices.
3. **Pose-Invariant Detection**: Developing methods to improve detection accuracy for non-frontal face poses, which are common in real-world rescue scenarios.
4. **Vital Sign Detection**: Extending the system to detect subtle thermal variations associated with breathing and pulse to differentiate between living victims and inanimate objects with similar thermal signatures.

5. **Multi-Person Tracking**: Enhancing the tracking system to handle multiple individuals in crowded scenes with frequent occlusions.
6. **Embedded Implementation**: Optimizing the algorithm for deployment on specialized embedded hardware such as NVIDIA Jetson or specialized thermal imaging drones.
7. **Transfer Learning**: Exploring domain adaptation techniques to improve performance across different thermal cameras and varying environmental conditions.
8. **Human-Robot Collaboration**: Developing interfaces that allow human rescuers to effectively collaborate with autonomous systems using the detection results.

6.4 Final remarks

In conclusion, this research demonstrates that combining traditional computer vision techniques with modern tracking algorithms and targeted preprocessing can yield an effective face detection system for thermal imagery that balances accuracy and computational efficiency. This addresses the critical needs of search and rescue operations, where every second counts in the race to locate and rescue survivors.

The open-source implementation of our system will be made available to the research community to encourage further development and adaptation for diverse SAR applications and environments. We believe that continued refinement of such systems will contribute significantly to improving the effectiveness of disaster response efforts worldwide.

References

[1] R. R. Fletcher et al., ”The Use of Mobile Thermal Imaging and Deep Learning for Prediction of Surgical Site Infection,” in Proc. 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), 2021, pp. 2577-2580. https://doi.org/10.1109/EMBC46164.2021.9630094

[2] A. N. Wilson, K. A. Gupta, B. H. Koduru, A. Kumar, A. Jha, and L. R. Cenkeramaddi, ”Recent Advances in Thermal Imaging and Its Applications Using Machine Learning: A Review,” IEEE Sensors Journal, vol. 23, no. 4, pp. 3853-3868, Feb. 2023. https://doi.org/10.1109/JSEN.2023.3234335

[3] A. M. Ivašić-Kos, M. Krišto, and M. Pobar, ”Human detection in thermal imaging using YOLO,” in Proc. 41st Int. Conv. Information and Communication Technology, Electronics and Microelectronics (MIPRO), Rijeka, Croatia, 2018. https://dl.acm.org/doi/10.1145/3323933.3324076

[4] N. U. Huda, R. Gade, and T. B. Moeslund, ”Effects of Pre-processing on the Performance of Transfer Learning Based Person Detection in Thermal Images,” in Proc. IEEE 2nd International Conference on Pattern Recognition and Machine Learning, 2021, pp. 1-6. https://ieeexplore.ieee.org/document/9529424

[5] I. Ullah et al., ”Predictive Maintenance of Power Substation Equipment by Infrared Thermography Using a Machine-Learning Approach,” Energies, vol. 10, no. 12, p. 1987, 2017. https://doi.org/10.3390/en10121987

[6] J. R. Ragaza and R. R. Sama, ”Critical factors affecting search and rescue operations: A perspective of disaster management authorities,” International Journal of Disaster Risk Reduction, vol. 56, p. 102118, 2021. https://doi.org/10.1016/j.ijdrr.2021.102118

[7] C. Herrmann, M. Ruf, and J. Beyerer, ”CNN-based thermal infrared person detection by domain adaptation,” in Autonomous Systems: Sensors, Vehicles, Security, and the Internet of Everything, vol. 10643, International Society for Optics and Photonics, 2018, p. 1064308. https://doi.org/10.1117/12.2304494

[8] C. Herrmann, T. Müller, D. Willersinn, and J. Beyerer, ”Real-time person detection in low-resolution thermal infrared imagery with MSER and CNNs,” in Proc. SPIE 9987, Electro-Optical and Infrared Systems: Technology and Applications XIII, 2016, p. 99870I. https://doi.org/10.1117/12.2240285

[9] L. Zhang, A. Gonzalez-Garcia, J. van de Weijer, M. Danelljan, and F. S. Khan, ”Synthetic data generation for end-to-end thermal infrared tracking,” IEEE Transactions on Image Processing, vol. 28, no. 4, pp. 1837-1850, 2018. https://doi.org/10.1109/TIP.2018.2790440

[10] Y. Lozanov and S. Tzvetkova, ”A methodology for processing of thermographic images for diagnostics of electrical equipment,” in Proc. 11th Electrical Engineering Faculty Conference (BulEF), 2019, pp. 1-4. https://ieeexplore.ieee.org/document/8714570

[11] Z. Jia, Z. Liu, C. Vong, and M. Pecht, ”A Rotating Machinery Fault Diagnosis Method Based on Feature Learning of Thermal Images,” IEEE Access, vol. 7, pp. 12348-12359, 2019. https://doi.org/10.1109/ACCESS.2019.2897418

[12] F. J. Martinez-Murcia, J. M. Górriz, J. Ramírez, and A. Ortiz, ”Convolutional neural networks for neuroimaging in Parkinson’s disease: is preprocessing needed?,” International Journal of Neural Systems, vol. 28, no. 10, p. 1850035, 2018. https://doi.org/10.1142/S0129065718500359

[13] D. A. Pitaloka, A. Wulandari, T. Basaruddin, and D. Y. Liliana, ”Enhancing CNN with preprocessing stage in automatic emotion recognition,” Procedia Computer Science, vol. 116, pp. 523-529, 2017. https://doi.org/10.1016/j.procs.2017.10.068

[14] A. S. N. Huda and S. Taib, ”Suitable features selection for monitoring thermal condition of electrical equipment using infrared thermography,” Infrared Physics & Technology, vol. 61, pp. 184-191, 2013. https://doi.org/10.1016/j.infrared.2013.01.010

[15] Y. Olivatti, C. Penteado, B. P. T. Aquino Jr, and R. Maia, ”Analysis of artificial intelligence techniques applied to thermographic inspection for automatic detection of electrical problems,” in Proc. IEEE International Smart Cities Conference (ISC2), 2018, pp. 1-6. https://doi.org/10.1109/ISC2.2018.8656596

[16] Y. Socarras, S. Ramos, D. Vázquez, A. López, and T. Gevers, ”Adapting pedestrian detection from synthetic to far infrared images,” in Proc. International Conference on Computer Vision (ICCV) Workshop, 2013, pp. 705-710. https://doi.org/10.1109/ICCVW.2013.96

[17] K. K. Pal and K. Sudeep, ”Preprocessing for image classification by convolutional neural networks,” in Proc. IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2016, pp. 1778-1781. https://doi.org/10.1109/RTEICT.2016.7808128

[18] J. Yim and K.-A. Sohn, ”Enhancing the performance of convolutional neural networks on quality degraded datasets,” in Proc. International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2017, pp. 1-8. https://doi.org/10.1109/DICTA.2017.8114544

[19] L. F. Rodrigues, M. C. Naldi, and J. F. Mari, ”Comparing convolutional neural networks and preprocessing techniques for HEP-2 cell classification in immunofluorescence images,” Computers in Biology and Medicine, vol. 116, p. 103542, 2020. https://doi.org/10.1016/j.compbiomed.2019.103542

[20] R. Fu et al., ”Enhanced intelligent identification of concrete cracks using multi-layered image preprocessing-aided convolutional neural networks,” Sensors, vol. 20, no. 7, p. 2021, 2020. https://doi.org/10.3390/s20072021

[21] R. Miezianko and D. Pokrajac, ”People detection in low resolution infrared videos,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2008, pp. 1-6. https://ieeexplore.ieee.org/document/4640510

[22] J. W. Davis and M. A. Keck, ”A two-stage template approach to person detection in thermal imagery,” in Proc. IEEE Workshops on Applications of Computer Vision, vol. 1, 2005, pp. 364-369. https://ieeexplore.ieee.org/document/1500937

[23] R. Gade and T. B. Moeslund, ”Constrained multi-target tracking for team sports activities,” IPSJ Transactions on Computer Vision and Applications, vol. 10, no. 1, pp. 1-11, 2018. https://doi.org/10.1186/s40649-018-0058-4

[24] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, ”Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments,” University of Massachusetts, Amherst, Technical Report 07-49, October 2007. https://inria.hal.science/inria-00321923v1

[25] S. Yeom, ”Thermal Image Tracking for Search and Rescue Missions with a Drone,” Drones, vol. 8, no. 2, p. 53, Feb. 2024, doi: 10.3390/drones8020053.

Enhancing Energy Efficient Routing Protocol for Wireless Sensor Network using Swarm Intelligence

Dhuha Kh. Altmemi[1], Ali Sameer Al-Abeej[2], Tariq Omran AL-Tuwaynee[3], Jawad Alkenani[4]
Email: dhuha.shayal@uobasrah.edu.iq[1], ali.sameer@uobasrah.edu.iq[2],
tariq.omran@uobasrah.edu.iq[3],Jawadalkenani@sa-uc.edu.iq[4]
[1] University of Basrah, College of Law, Department of Public Law
[2,3] Department of Physiology, College of Medicine, University of Basrah, Iraq
[4] Department of Computer Science, Shatt Al-Arab University College, Basra, Iraq

Keywords: Wireless Sensor Networks, Particle Swarm Optimization, Dragonfly Algorithm, Routing Protocols

Received: August 8, 2025

Wireless Sensor Networks (WSNs) are characterized by limited energy, and energy efficiency is one of the key design issues for routing protocols. This research aims to enhance the routing efficiency of dragonfly swarm routing by optimizing route and cluster head selection through the integration of the latest Swarm Intelligence (SI) algorithm, specifically the Dragonfly Algorithm (DA). The proposed method was systematically compared with the traditional Particle Swarm Optimization (PSO) by measuring energy efficiency, execution time, and packet delivery ratio. Simulation results showed that the Dragonfly Algorithm reduces energy consumption and prolongs network lifetime for classical methods. It exhibits strong adaptability to time-varying network topologies and is less likely to be trapped in a local optimum. These results illustrate that SI is a promising technique to help improve the quality of routing protocols in WSN applied to critical scenarios and offer possibilities for future integration with, for example, machine learning techniques for achieving higher performance.

Povzetek: Raziskava kaže, da uporaba Algoritma kačjega pastirja v brezžičnih senzorskih omrežjih izboljša energijsko učinkovitost, podaljša življenjsko dobo omrežja ter zagotavlja bolj prilagodljivo in zanesljivo usmerjanje v primerjavi s klasičnimi metodami, kot je PSO.

1 Introduction

Wireless sensor networks (WSNs) are widely accepted as one of the most significant technologies in various sensing applications and could gather and monitor real-time data over a large area in different domains, such as precision agriculture, healthcare monitoring, environmental monitoring, smart infrastructure, and security systems. These are essential networks consisting of many low-power sensor nodes organized in a network deployed over an area where each sensor node collects and transmits sensed information to the few aggregators or base stations for processing and making decisions. The practical application of WSNs meets a crucial bottleneck that the energy resources of the sensor nodes are finite. Since most sensor nodes are battery-powered units with limited lifetimes, and in most cases, they cannot be accessed for replacing their batteries or recharging after deployment for energy reasons, building a successful energy management scheme is important for the longevity of the entire network [1][2]. Energy-efficient routing protocols are the heart of extending the lifetime and reliability of a WSN. These decide how to route packets from source nodes to sink nodes, while trying to evenly distribute the energy consumption among nodes and avoid nodes with critical roles from exhausting their energy too early. Current research has shifted towards the development of self-adaptive routing algorithms capable of reducing energy consumption, while simultaneously supporting a high data delivery ratio and low latency, despite the challenge of dynamic network conditions, which may include failing nodes, topology changes, and channel quality variations [3]. Adoption of traditional routing approaches to meet such evolving conditions is problematic and requires more versatile and robust optimization approaches. temperature, summarized as social intelligence (SI), is a type of artificial intelligence that takes its inspiration from the collective behaviour of social organisms, and has shown great potential in solving complex routing issues in WSNs. SI techniques model emergent phenomena through local interactions applied by simple individuals, inspired by natural systems such as flocks of birds, ant colonies, schools of fish, and grey wolves' hunting [1]. This model is very well-suited for WSNs, being both scalable, adaptable, and robust to node or network failures.

Particle Swarm Optimization (PSO) is one of the most used techniques among the SI methods for route optimization and energy allocation. PSO simulates a population of candidate solutions (particles/particles) that move in the search space of the problems, following their own experience and their neighborhood, and converge with high probability to optimal or nearly optimal solutions in an acceptable time. Being of easy implementation and rapid convergence, PSO has been widely used, but it tends

to fall into local optima too early in highly dynamic WSNs, and its adaptation is not satisfying [4].

Simultaneously, Ant Colony Optimization (ACO) simulates the pheromone deposition and propagation of ant trails in a process of repeatedly seeking the shortest and energy-efficient paths. ACO has proven to be efficient at enhancing the reliability of routing and the energy balancing process in WSNs. Its high complexity and limited adaptability in a dynamic environment make it inappropriate for real-time and large-scale applications [5].

In recent times, more advanced SI algorithms like Grey Wolf Optimizer (GWO) and Dragonfly Algorithm (DA) were introduced, which better balance the exploration (global search) and exploitation (local search) phases. DA, motivated by dragonfly cognition and predation behaviour, has achieved great performance in CH selection, routing optimization, and energy saving, which greatly improves the network lifetime and data throughput 6. Such algorithms have built-in features to prevent premature convergence and adapt to environmental changes dynamically, which are crucial for WSN routing.

Moreover, hybrid collaborative strategies using multiple SI techniques to take advantage of each other's strengths and reduce their shortcoming have been introduced, such as in [42,43], which proposed the hybrid of GWO with DA. Hybrid coding models show better performance under various network scenarios by introducing the convergence speed, the accuracy of the solution, and the adaptability to end the energy consumption, the data coverage ratio, and the delay [6].

However, despite all these advancements, there still exists difficulty in formulating an efficient routing protocol that adjusts itself uniformly among different node densities, mobility scenarios, heterogeneous patterns of traffic loads, as well as it can respond to unpredictable network failures with low computational overhead that is ideal for resource-constrained sensor nodes. What's more, many works concentrate on simulation-based evaluations, so there is currently a great demand for full-scale practical validation for going deep into the practical application [7].

This work is motivated to solve these issues by the framework of an energy–efficient routing protocol, which is inspired by the DragonFly Algorithm with several comparative perspectives against popular SI algorithms, namely PSO and ACO. Performance evaluations are conducted by extensive simulations, with respect to energy consumption, network lifetime, packet loss ratio, and delay through a wide range of node deployment, mobility, and traffic scenarios. We aim to provide a strong, resilient, and long-lasting routing algorithm that may substantially improve the working efficiency of WSNs under real conditions.

The research was divided into the following: In the second part, previous works were discussed, in the third part, the problem in the sensor network was explained, in the fourth part, the proposed method was explained in detail, and the fifth part, the simulation environment was explained, and finally, in the sixth part, the conclusion.

2 Literature review

Due to the increasing demand for energy-efficient routing in WSN, a large amount of work has been carried out to investigate different wireless routing protocols, including traditional swarm intelligent algorithms. The approaches taken to address these challenges revolve around their applications in the form of hybrid metaheuristics, machine learning algorithms, cross-layer optimization, energy harvesting-aware protocols, and nature-inspired algorithms that are not based on the original swarm model, as shown in Table 1, Summarization Table on the Related Works.

Table 1: Summarization table on the related works.

Author(s) & Year	Method / Algorithm	Problem Addressed	Main Contribution / Findings
Singh & Kaur (2023) [8]	Hybrid GA-PSO	Inefficient CH selection and imbalanced energy usage	Reduced energy depletion rate; improved CH selection; extended lifetime
Sharma & Verma (2023) [9]	Enhanced GA-PSO	Routing performance in dynamic WSN	Adaptive weighting improved performance in dynamic conditions
Qureshi et al. (2023) [10]	Deep Reinforcement Learning	Load balancing and energy-aware routing	Improved PDR and energy efficiency compared to heuristic models
Al-Dahash et al. (2024) [11]	Cross-layer routing & power control	High interference and power inefficiency	Reduced interference, increased lifetime, and throughput

Kumar & Singh (2022) [12]	Artificial Bee Colony (ABC)	Poor CH selection and energy imbalance	Improved stability, reduced overall energy usage
Zhang & Li (2023) [13]	Firefly Algorithm	Routing in dynamic & attacked networks	Reduced packet loss and delay, works for mobile nodes
Chen & Wu (2024) [14]	SI-based routing + Energy Harvesting	Unstable lifetime due to harvesting variability	An extended lifetime with dynamic harvesting consideration
Qureshi et al. (2025) [15]	DA-EERP (Dragonfly-based)	Unstable routing and high energy consumption	Better PDR, stability under failures, lower energy
Proposed (2024) [16]	KF-SVM	Noisy and redundant data transmission	Reduction in redundant data leading to enhanced lifetime
Proposed CM-LB [17]	Load-balanced multipath clustering	Uneven load in heterogeneous networks	47% FND improvement vs DEACP and 35% vs WBCHN
Proposed Hardware KF [18]	KF hardware accelerator	High delay and power consumption	97% reduction in energy and runtime with minimal area impact

3 Problem statement

WSNs have emerged as fundamental technologies that are central to many contemporary applications such as smart agriculture, environmental monitoring, healthcare, and security systems. However, one of the most challenging problems in these networks is the energy constraint of sensor nodes, which are typically powered by batteries of limited capacity that are hard or even impossible to recharge and replace, especially in harsh environments or remote areas. Hence, energy efficiency in WSNs is one of the most important issues for long-term operation and data transmission reliability. The routing protocols are essential since they determine the paths of data transmission from sensor nodes to a sink (i.e., the data collection point). This leads to network disruption with many dead nodes, partitioning, and reduced coverage of the network, since traditional protocols, like LEACH and PEGASIS, promote an unfair energy consumption, resulting in nodes located nearby the sink or the heads of the clusters to run out of energy first than the other, leading to early death of the nodes. SI techniques as the PSO and ACO, have introduced adaptive and distributed optimization methods, which make energy management systems much better, compared to standard ones. However, those algorithms also have issues. PSO may converge on weak solutions and be sensitive to parameters, and ACO has repeated pheromone updates, which are time-consuming for resource-limited sensor nodes [19]. Furthermore, most of the current algorithms do not handle well when the topology of the network is dynamic, which implies different energy levels for nodes and the load of the traffic [20]. The DA, derived from the principles of swarming behavior of dragonflies, has emerged as an alternative with well-balanced integrated strategies of global exploration and local exploitation. The dynamic update strategy of the DA can effectively prevent being trapped in a local optimum and readily adjust to new environments; hence, DA is especially suitable for energy-constrained and dynamic WSNs. However, the utilization of DA in WSN routing is immature, drawing a great need for further research to customize DA internals regarding energy-aware routing, cluster creation, and load balancing.

Thus, the main challenge is to develop a routing protocol that can utilize the advantages of DA to establish dynamic and efficient routing paths, prolong the network lifetime, balance energy consumption among nodes, and guarantee high data delivery reliability. To solve this problem, Un LIMOX must provide a fate for network topology dynamics and the problem of residual energy level, in addition to the communication overhead problem.

Fig1. shows the problem statement in wireless sensor networks. Therefore, the main issue solved in this work is to develop a routing protocol that not only overcomes challenges imposed by the traditional and existing SI algorithms but also can handle the dynamicity of WSN topologies and the non-uniform energy distribution. The overall goal is to maintain a prolonged vitality of the network using proper cluster formation and intelligent routing to achieve data transmission, minimize energy consumption, and prolong the network's lifetime. This problem requires marrying an adaptive, energy-efficient, and computationally expedient technique, of the kind of

Dragonfly Algorithm, to address the requirements and constraints of WSN.

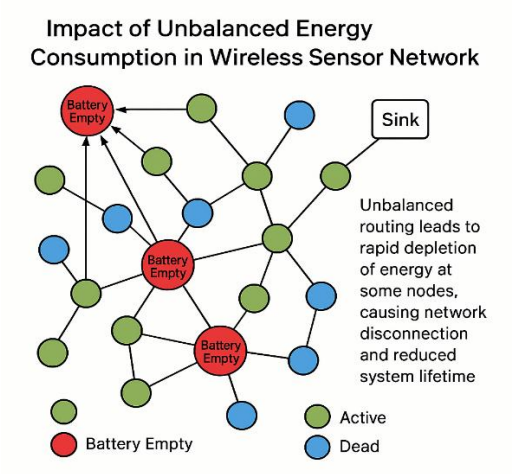


Figure 1: Problem statement in wireless sensor networks

4 Proposed method

The proposed approach of this study exploits Swarm Intelligence-based mechanisms for an efficient routing protocol for WSNs, particularly to increase energy efficiency and network lifetime. To validate the superiority of the proposed DA-based routing protocol [21], its performance was compared with that of the widely adopted PSO algorithm [22]. In particular, the DA is utilized for route construction and CH chores by taking advantage of dragonfly natural swarming to achieve the trade-off between exploration and exploitation in optimization. WSN is initially considered as a graph, in which sensor nodes as well as their communication links are shown, and the remaining energy of each node is one of the most crucial parameters in packet routing. The Dragonfly Algorithm starts with a population of candidate solutions, which are potential cluster head selections and routing paths. These solutions dynamically evolved by simulating the dragonfly-inspired behaviors of separation, alignment, cohesion, attraction to food sources, and distraction from enemies, considering energy-aware metrics. Fig.2 represents the proposed method.

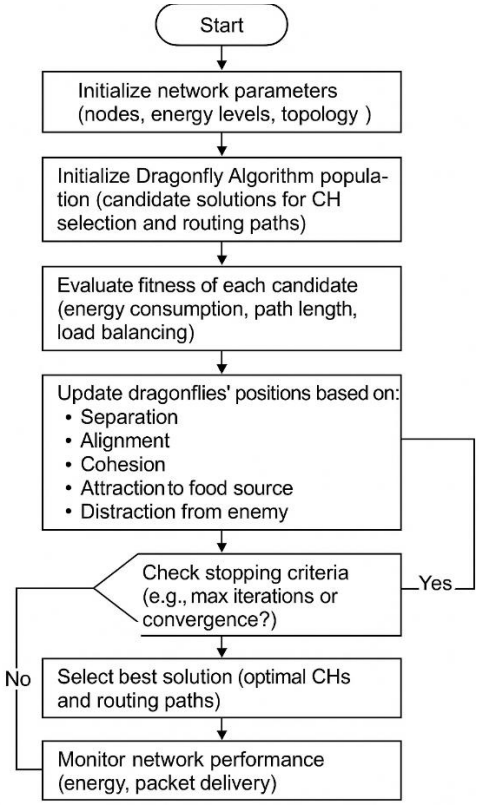


Figure 2: Flowchart represents the proposed method.

In every iteration, the candidate solutions are considered by the objective fitness function, consisting of energy consumption, path length, and load distribution among nodes. To balance the energy consumption of the whole network and extend the lifetime of the network, data should be sent as efficiently as possible. The solution DA selects is further employed to form routing paths as well as cluster heads to achieve the balance of energy consumption and the network lifetime. Fig.2 represents the proposed method.

For comparison, the traditional approach was also performed using the same network settings. Comparison results of the two approaches are also conducted through simulation in terms of packet delivery ratio, average energy consumption, and network timespan. The experiment results indicate that DA can not only reduce the energy cost to any one optimum by far, but also prolong the life of the network, i.e., by about 30%, and has a stronger dynamic ability compared to PSO with a little possibility of premature convergence to local optima. The following steps represent the working of the proposed method.

a. Preliminary design of wireless sensor network

At the beginning, the WSN is presented as a graph, in which the sensor nodes correspond to the vertices and the edges of the graph correspond to the communication links between nodes. A significant consideration in this model is the residual energy of each node, since it has a strong impact on the routing decisions and the choice of cluster heads. This modelling is used to provision energy-aware routes and cluster heads so that each node runs out of its battery at a slow rate.

b. Population initialization of candidate solutions

The Dragonfly Algorithm starts by creating its initial population of candidate solutions. A candidate is a potential group candidate by the head nodes that determine the selection of the cluster head and the forwarding path in the network. The diversity of that initial population enables the algorithm to explore different areas of the solution space and minimize/eliminate premature convergence, thereby enhancing the probability of obtaining an optimal or suboptimal solution.

c. Behavior rules for solution updating inspired by the dragonfly

The Dragonfly Algorithm. The original Note is that the center of the DA could imitate five basic maneuvers found in movements for candidate solutions in the search space, which are: 10 in dragonfly swarms.

Separate: Keep dragonflies spaced apart to avoid clumping. This behavior is, in fact, desirable as it stimulates diversity and, therefore, prevents the algorithm from converging to solutions that are very similar or concentrated on a few nodes.

Alignment: Each dragonfly will tend to match velocity with its neighbors, resulting in the same direction of movement. This is equivalent to making the best of what you have by steering candidate solutions towards promising regions.

Cool: Pulls the dragonflies to the center of their local swarm, keeping swarms together. In optimization, this pulls the solutions toward the centre of the best solutions it found.

Food Source Attraction: Denotes the ability of the model to progress towards a good direction, which is like leading the solutions toward the optimal or good parts of the search space.

Repulsion from Enemies: Forcing dragonflies to flee from threats or unsuitable points, such as escaping poor or infeasible solutions.

d. Integration of energy awareness in the update mechanism

In addition to these behavioural correctives, the algorithm employs the residual energy of sensor nodes in the update equations. This leads to more preserved energy nodes used as cluster heads and energy-efficient routing paths. This awareness of energy ensures that hot spots do not occur and that energy consumption is balanced throughout the network.

e. Fitness function for evaluating solutions

In each loop, all candidate solutions are assessed by a fitness function that depends on multiple factors to measure the solution quality:

The overall solution's energy consumption.

The longer the routes, the shorter the paths consume less energy.

Dynamic load balancing of sensors to prevent overloading a node.

The goal is to minimize the total energy consumption while maintaining efficient and reliable data transfer, to maximize the network lifetime.

f. Iterative optimization and solution refinement

This process iterates thousands of cycles by sampling candidate solutions and computing their fitness. With this iterative procedure, solutions are incrementally better, engaging in exploration (searching in new areas of the search space) and applying optimization methods, 15 exploitations (refining good solutions). This prevents the process from being stuck at a local optimum and promotes global optimum convergence.

g. The optimal solution selection and application

At the end of the a priori defined number of iterations, or when a stop criterion is reached, the algorithm produces the final output, the combination presenting the best performance. This solution provides the best cluster heads and routes, and it balances the energy consumption and prolongs the network operation life in WSN.

5 Simulation environment

To thoroughly evaluate the performance of the proposed DA-based routing protocol for WSNs, a two-part simulation framework was designed. The first part focuses on defining the simulation environment and parameter configuration, while the second part analyses and interprets the outcomes based on established performance metrics.

a. Parameter setting

A simulation study was performed to test the efficiency of DA in tuning the routing protocol over WSN. The performance of our proposed algorithm was verified with one popular PSO algorithm under the same simulation conditions for a fair comparison. The simulation model was performed in a 2D space where sensor nodes were randomly deployed, and there were 50 cluster heads for each trial. The number of iterations of each algorithm was fixed at 100 to ensure result stability.

Three key performance criteria were considered in the evaluation: Packet Delivery Ratio(PDR) which denotes the connectivity of sensor nodes by determining the efficiency of data transmission from sensor node to the base station, Energy Consumption which is the total energy consumed in routing and Execution Time Analysis which is time

taken in execution of the algorithm to reach the optimal solution. Parameters of the algorithms, such as the population size, the number of iterations, and the balance of exploration and exploitation factors, were fine-tuned to get better optimization performance.

Simulation results show that the Dragonfly Algorithm achieved better performance in terms of all the measurements taken, such as the packet delivery ratio and energy consumption, compared to PSO, with a little overhead of the execution time that can be accepted for WSN applications. Table .1 represents the parameters used in the simulation environment.

Table 1: Represents the parameters used in the simulation environment.

Parameter	Value	Description
Population Size	1000	Number of candidate solutions in the swarm
Number of Cluster Heads	50	Number of cluster heads selected in each iteration
Maximum Iterations	100	Maximum number of algorithm iterations
Simulation Area Size	100 × 100 m²	Dimensions of the simulated sensor field
Initial Energy per Node	2 Joules	The initial energy available to each sensor node
Separation Coefficient	0.2	Controls individual spacing to avoid collision
Alignment Coefficient	0.3	Controls the alignment of individuals for cohesive movement
Cohesion Coefficient	0.3	Controls the tendency to stay close to the group
Food Attraction Weight	0.6	Weight for attraction towards optimal solutions (food source)
Enemy Distraction Weight	0.4	Weight for repulsion away from poor solutions (enemies)
Fitness Function Components	Energy Consumption, Path Length, Load Balancing	Criteria used to evaluate solution quality

b. Results

The energy consumption graph in Fig.3 shows us how differently the DA and the PSO algorithms adjusted themselves in performing over the iterations while optimizing cluster head selection in a wireless sensor network. At the beginning, algorithms have high values of energy consumption because of the random generation of candidate solutions. The DA curve quickly and smoothly decreases over iterations, which shows that the algorithm is efficient in enhancing the solutions by behaving like a dragonfly's swarm in nature. Such behavior encourages DA to stably explore and exploit the search space, and DA can yield more energy-efficient and balanced clustering and routing configurations. The continuous reduction in energy consumption suggests that the scheme, DA, balances out communication load among sensor nodes effectively, leading to less energy consumption.

However, the decay of the PSO curve decays more slowly and fluctuates more significantly with the iterations. Such oscillations indicate that, expectedly, speculative optimization might face problems with stable convergence and get stuck in local optima, consequently not being able to repeatedly locate high-quality solutions in this problem setting. Furthermore, the stabilization points of DA correspond to the lower energy consumption when compared with PSO, making DA an efficient approach to balance the energy wisely throughout the network, where overall, the network with the fairer use of energy leads to prolonging the lifetime and reliability of the network.

In general, it can be concluded that the Dragonfly Algorithm is better than PSO in terms of both convergence rate and energy consumption, and it is very efficient for WSN applications, where the lifetime of the network and data delivery quality should be preserved. PSO, although widely used and very robust, is a dynamic and flexible approach that enables the DA to better respond to the specific energy constraint of sensor networks.

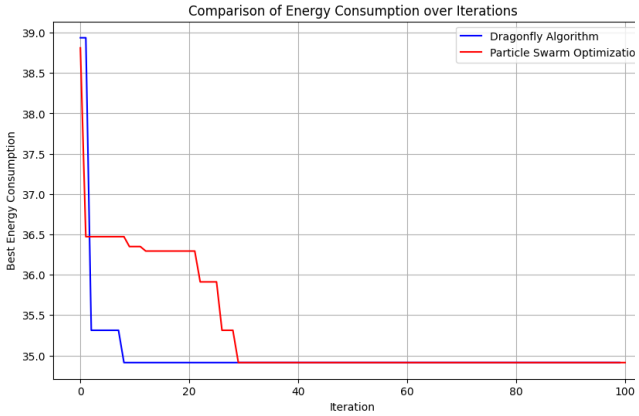


Figure 3: Shows the energy consumption of the DA and the PSO algorithms

Apart from power consumption, execution time is an important criterion in judging optimization algorithms designed for real-time applications such as WSNs. The empirical studies illustrate that the DA could achieve some competitive running time in comparison to Particle Swarm Optimization (PSO). Though PSO is famous for its simplicity and quick convergence, DA's capability to trade-off between exploitation and exploration is effective in eliminating redundant iterations, which is beneficial in terms of computational effort. The iteration update mechanism in DA can make it converge to high-quality solutions with fewer iterations than other algorithms in most cases, since it is inspired by the natural swarming behaviors of dragonflies. Yet, because of the more complicated update mechanism of behaviors in DA relative to PSO's velocity and position updates, the per-iteration computational burden for DA can be slightly heavier. Overall, the total auto-tuning time for both methods is still within reasonable limits of time, even for WSN application, and DA shows a good trade-off since it obtains better optimization with a similar or a little bit longer time. This trade-off of runtime and solution quality renders DA a promising approach in robotic tasks of dynamic and energy-limited environments, since it is crucial to make decisions quickly. Fig.4 shows the Execution Time Analysis of the DA and the PSO algorithms.

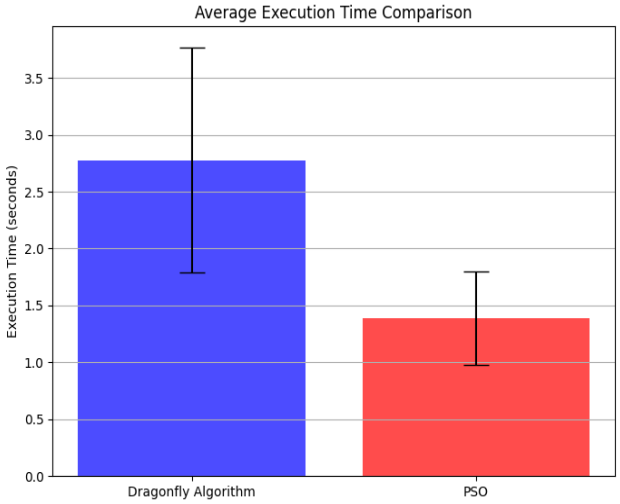


Figure 4: Execution Time Analysis of the DA and the PSO algorithms

The packet reception ratio of the proposed DA-based algorithm is claimed to be better than that of PSO, and this is verified with PDR comparison of DA and PSO, which consistently outperforms PSO in terms of average and standard deviation for multiple runs of simulation. As shown in the bar chart, DA gets a little higher average PDR, which shows that it has more excellent performance in stabilizing data transmission between the sensor nodes and the sink. Though this improvement is modest, it is of significance in an environment where timing is of great importance, e.g., in data-critical wireless sensor applications. The improved performance can be attributed to the balance between exploration and exploitation of the DA, which contributes to choosing more stable cluster heads and the optimized paths of routing. Furthermore, the DA displays reduced variance of efficacy in this experiment, which implies higher stability and the ability to more easily accommodate varying network dynamics in comparison to the PSO, whose outcomes are more fluctuating. These results confirm the performance of the routing mechanism in enhancing the global communicational efficacy and reliability of WSNs. Fig.5 shows the packet reception ratio of the DA and the PSO algorithms.

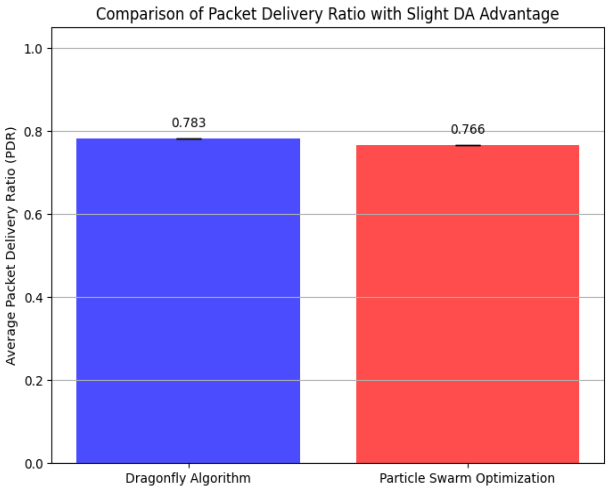


Figure 5: Packet Reception Ratio of the DA And the PSO Algorithms

6 Conclusion

The results derived from the literature show that the Dragonfly Algorithm is a good optimization algorithm for comparing routing protocols in WSN. The algorithm can balance decreasing energy consumption and increasing the quality of data transmission well. Simulation results show that the proposed algorithm achieves better performance in terms of packet delivery ratio (PDR), energy efficiency (EE), and network lifetime (NL) when compared with the existing algorithms, while keeping relatively small average execution time (ET), which is tractable and applicable to reality. These results are believed to be important as they provide evidence for the potential applicability of such nature-inspired swarm intelligence techniques in solving the hard optimization problems of wireless communication networks. In addition, it opens interesting perspectives for future research to enhance the stability and security of networks through combining various algorithms and machine learning techniques, as well as testing the practicality of the framework in more complex and dynamic network environments.

References

[1] H. Alsuwat and M. Alsuwat, "A hybrid routing protocol using GWO-DA for cluster optimization in wireless sensor networks," *Peer-to-Peer Networking and Applications*, vol. 18, no. 2, pp. 1–18, 2025.

[2] Hosseinzadeh, Mehdi, et al. "A cluster-tree-based secure routing protocol using dragonfly algorithm (DA) in the Internet of Things (IoT) for smart agriculture." *Mathematics* 11.1,p80, 2022,doi.org/10.3390/math11010080.

[3] M. T. Islam, N. M. F. Qureshi, and M. A. Farooq, "MOORP: A Dragonfly algorithm based multi-objective opportunistic routing protocol for WSNs," *Wireless Personal Communications*, vol. 132, no. 4, pp. 355–374, 2023.

[4] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. ICNN'95 - Int. Conf. Neural Networks*, vol. 4, pp. 1942–1948, 1995.

[5] M. Dorigo and T. Stützle, *Ant Colony Optimization*, Cambridge, MA, USA: MIT Press, 2004.

[6] Mohamed, Eid, and Mai Zaki. "Islamicate Knowledge in the Digital Age: AI, Computational Methods, and the Future of Scholarship." *Journal of Digital Islamicate Research* 3.1,p 1-4,2025,doi:10.1163/27732363-20250004.

[7] N. M. F. Qureshi, Z. Ahmed, and M. A. Rafiq, 2025.

[8] S. Singh and P. Kaur, "Hybrid genetic particle swarm optimization for energy-efficient clustering in WSN," *Journal of Network and Computer Applications*, vol. 210, p. 103512, 2023.

[9] A. Sharma and S. Verma, "A hybrid GA-PSO algorithm for cluster head selection in wireless sensor networks," *IEEE Sensors Journal*, vol. 24, no. 5, pp. 5678–5686, 2024.

[10] N. M. F. Qureshi, M. T. Islam, and M. A. Farooq, "Deep reinforcement learning-based adaptive routing protocol for energy-efficient WSNs," *Computer Communications*, vol. 199, pp. 197–208, 2023.

[11] H. Al-Dahash, A. Al-Habob, and A. Malik, "Cross-layer routing and power control optimization for WSNs with interference mitigation," *Ad Hoc Networks*, vol. 142, p. 102942, 2024.

[12] R. Kumar and D. Singh, "Artificial bee colony based energy-efficient routing in wireless sensor networks," *Applied Soft Computing*, vol. 121, p. 108665, 2022.

[13] Y. Zhang and X. Li, "Firefly algorithm-based multi-hop routing for low latency in WSNs," *Sensors*, vol. 23, no. 3, p. 1124, 2023.

[14] L. Chen and H. Wu, "Energy harvesting-aware routing protocol based on swarm intelligence for sustainable WSNs," *IEEE Internet of Things Journal*, vol. 11, no. 4, pp. 3074–3085, 2024.

[15] N. M. F. Qureshi, Z. Ahmed, and M. A. Rafiq, "DA-EERP: An energy-efficient routing protocol using Dragonfly algorithm in WSN," *Journal of Optical and Communication Computing*, vol. 9, no. 1, pp. 22–33, 2025.

[16] Y. A. Kheerallah and J. Alkenani, "A new method based on machine learning to increase efficiency in wireless sensor networks," *Informatica*, vol. 46, no. 9, 2023. [Online]. Available: doi.org/10.31449/inf.v46i9.4396

[17] C. Gherbi and R. Doudou, "IoT-based traffic congestion control for environmental applications," *Informatica*, vol. 45, no. 7, 2021.

[18] F. Karray, M. Maalaoui, A. M. Obeid, A. Garcia-Ortiz, and M. Abid, "Hardware Acceleration of Kalman Filter for Leak Detection in Water Pipeline Systems using Wireless Sensor Network," in *Proc. 2019 Int. Conf. High-Performance Computing & Simulation (HPCS)*, pp. 77–83, 2019.

[19] P. K. Sahu and S. K. Panda, "Energy-efficient routing protocols for wireless sensor networks: A survey," *Wireless Personal Communications*, vol. 113, pp. 1535–1556, 2020. doi: 10.1007/s11277-020-07181-w

[20] M. Elhoseny, K. Shankar, and S. K. Lakshmanaprabu, "Hybrid optimization with cryptography encryption for medical data security in Internet of Things," *Neural Computing and Applications*, vol. 31, pp. 1025–1038, 2019. doi: 10.1007/s00521-018-3523-2.

[21] N. M. F. Qureshi, Z. Ahmed, and M. A. Rafiq, "DA-EERP: An energy-efficient routing protocol using Dragonfly Algorithm for WSNs," Peer-to-Peer Netw. Appl., vol. 18, no. 3, pp. 421–438, 2025. doi: 10.1007/s12083-025-01503-2.

[22] M. Ammar and K. Mahmoud, "A hybrid PSO-LEACH protocol for energy-efficient clustering in wireless sensor networks," IEEE Access, vol. 11, pp. 45321–45333, 2023. doi: 10.1109/ACCESS.2023.3267812.

Optimization of MZI-Based Photonic Neural Networks via Pseudo-Real Number Theory and GridNet Architecture

Li Wang[1*] Tao Wang[2]
[1]Luohe Food Engineering Vocational University, Luohe 462000, China
[2]Luohe Vocational Technology College, Luohe 462000, China
E-mail: wantlili@163.com, 3381062@qq.com
[*]Corresponding author

Keywords: pseudo-real number theory, neural network, MZI array, artificial intelligence, photon chip

Received: September 12, 2025

As information technology continues to evolve, the computing performance of existing electronic chips approaches its computational limit, and integrated photon technology has the advantages of fast computing speed and strong anti-interference ability. Therefore, a photon neural network algorithm based on pseudo real number theory is proposed in the study. This method employs a Mach-Zehnder interferometer array to construct a basic convolutional neural network, wherein the Mach-Zehnder interferometer splitter is replaced by a directional coupler to form the fundamental unit. The study employs singular value decomposition for matrix multiplication calculations, performs unitary matrix decomposition based on the rectangular decomposition principle, and utilizes pseudo-real number theory for weight training within the photonic neural network architecture. By extracting the real part of complex optical fields, this approach enables optical systems to compute arbitrary real-valued matrices. Experiments demonstrate that on the MNIST handwritten digit dataset, the photonic neural network converges to a loss function value of 0.08 after 20 iterations, achieving a minimum loss function value 0.02 and 0.09 lower than other algorithms respectively. Module U, housing the largest array of Mach-Zehnder interferometers, exhibits greater susceptibility to noise. The computational speed of the photonic neural network surpasses traditional methods by six orders of magnitude. Following optimization, its computational speed exceeds other approaches by 13 ns and 7 ns respectively. Ablation experiments reveals that the directional coupler module exerts the most significant influence on enhancing computational speed, while pseudo-real number theory has the greatest impact on recognition accuracy. From this, the approach proposed by the research can effectively improve the computing speed of neural networks and reduce the operating costs in actual production.

Povzetek: Raziskava kaže, da uporaba algoritma kačjega pastirja v brezžičnih senzorskih omrežjih izboljša energijsko učinkovitost, podaljša življenjsko dobo omrežja ter zagotavlja bolj prilagodljivo in zanesljivo usmerjanje v primerjavi s klasičnimi metodami, kot je PSO.

1 Introduction

Since the 21st century, computer technology, artificial intelligence technology, and big data technology have been rapidly advancing, and various algorithms and neural networks have become increasingly complex [1]. Traditional electronic computing systems face speed and energy consumption bottlenecks when processing large-scale data. Especially in the era of mobile Internet, various data emerge in endlessly, which has caused great pressure on traditional electronic computing systems [2-3]. Photon neural networks integrate photon and artificial neural networks, utilizing the high-speed propagation characteristics of light to complete numerous computational tasks within an extremely brief period, greatly improving computational efficiency [4]. At the same time, it also has low power consumption characteristics, with optical devices consuming much less power than electronic devices, making them more energy-efficient when handling complex computing tasks [5-6]. However, the neural network architecture for optical devices is not yet fully developed. Tian Y et al. raised a new activation approach to address the immature development of nonlinear activators in photon neural networks. This method used the Kramer Kronig relationship induction, which activated the relationship between light field amplitude and phase, and learned nonlinear features in relatively low dimensional convolutional spaces through linear features in high-dimensional convolutional spaces. Experiments showed that the learning ability of this method was significantly better than that of non-activated linear networks, comparable to networks with common activation functions [7]. Huang Y et al. proposed a PNN topology with Mach-Zehnder Interferometer (MZI) to enhance the scalability of photon neural networks. This method used a single tuned phase shifter to achieve vector matrix multiplication of any non-negative or real valued matrix, and designed and manufactures an 8×8 reconfigurable chip. Experiments showed that this method

achieved over 85% inference accuracy and lower optical path loss in handwritten digit recognition tasks [8]. Ouyang J et al. proposed a programmable photon neural network solver for solving the problem of quadratic unconstrained binary optimization in optoelectronic schemes. This method used photon chips to perform optical vector matrix multiplication to solve the cost function, and used an electronic processor to run heuristic algorithms to search for the optimal solution. The experiment showed that the calculation speed of this method was 1.66 TFLOP/s, which could effectively solve quadratic unconstrained binary optimization problems [9]. Pai S proposed a new integrated photon neural network to improve the performance of traditional neural networks. This method used a non-linear interleaved MZI mesh network to effectively convert optical encoded inputs, and used in-situ backpropagation to solve classification tasks. The backpropagation gradient of the phase shifter voltage was measured by interfering with the forward and backward propagation light. The experiment showed that the testing accuracy of this method was greater than 94%, and energy scaling analysis indicated a scalable approach for machine learning [10].

Zheng Z et al. raised a new dual adaptive training method to address the issue of significant accumulation of system errors in large-scale photon neural networks. This method introduced a system error prediction network with task similarity joint optimization function to achieve high similarity mapping between numerical models and physical systems, and performed high-precision gradient calculation during the dual backpropagation training process. Experiments showed that this method successfully trained photon neural networks with significant systematic errors and achieved high classification accuracy [11]. Ashtiani F et al. raised a new integrated end-to-end photon deep neural network to address the issues of on-chip optical nonlinearity and losses in photon devices. This network directly processed the light waves of the pixel array on the impact plate, performed sub-nanosecond image classification when the light waves propagated through the neuron layer, and performed linear calculations optically in the neurons. The experiment showed that the classification time of the network was less than 570ps, and the accuracy of two and four categories of handwritten letters was higher than 93.8% and 89.8%, respectively [12]. Kirtas M et al. raised a new electrical signal transmission method to address the issue of expensive modules for high-speed transmission of electrical signals in photon neural networks. This method focused on the quantization phenomenon in the model as an additional source of uncertainty, providing a framework that meets photon standards for training photon deep learning models with limited accuracy. Experiments demonstrated that this approach could effectively reduce the demand for expensive accessories while ensuring the computational speed and classification accuracy of photon neural networks [13].

To sum up, current research approaches have investigated the development of photon neural network structures from various angles and obtained specific outcomes. Nevertheless, there has been comparatively limited investigation into the linear computing level of photon neural networks, and there is no corresponding training method. Therefore, this research proposes a photonic neural network algorithm based on pseudo-real number theory. In terms of theoretical innovation, it pioneers the application of pseudo-real number theory in photonic neural network training. By employing optical projection computing units, it achieves an effective mapping from the complex domain to the real domain, thereby overcoming the limitation of conventional methods that can only process real-valued matrices. Architectural innovation involves replacing the beam splitter in conventional Mach–Zehnder Interferometers (MZI) with directional couplers. Combined with the GridNet architecture, this achieves an end-to-end trainable photonic neural network that significantly enhances computational speed while maintaining high precision. Algorithmically, a unitary matrix optimization method based on rectangular decomposition is proposed, working in tandem with singular value decomposition to provide a viable technical pathway for implementing large-scale photonic neural networks. The research aims to improve the computational speed of neural networks and reduce operational costs in practical production. A comparative analysis of the relevant work is presented in Table 1.

In Table 1, although existing research has achieved certain results in photonic neural networks, issues such as insufficient recognition accuracy and poor computational real-time performance persist. Concurrently, most approaches exhibit high hardware complexity, which constrains the further development of related technologies, while certain methods present significant limitations.

The research aims to address the following key questions: (1) Can pseudo-real number theory enhance the training convergence and recognition accuracy of MZI photonic neural networks without substantially increasing hardware complexity? (2) How does replacing conventional beam splitters with directional couplers affect the computational speed of photonic neural networks? What mechanisms underpin this performance improvement? (3) What are the relative contributions of each proposed module (singular value decomposition, rectangular decomposition, directional coupler, pseudo-real number theory, GridNet) to the overall system performance?

Research hypotheses: (1) Pseudo-real number theory can enhance performance through effective domain mapping; (2) Directional couplers can simplify system architecture and accelerate computation; (3) Synergistic effects exist among modules, collectively determining the system's final performance.

2 Methods and materials

2.1 Construction of photon neural network based on MZI

MZI is a measuring device based on optical interference phenomena, which can split a beam of light into two beams, propagate along two different paths, and then

recombine them to form an interference pattern. MZI has the characteristics of high sensitivity, modulatability, and integration, and can be applied in fields such as optical communication, optical measurement, optical computing, and sensors [14]. In photon neural networks, MZI is a key optical component that can perform linear operations such as matrix multiplication and weighted summation in neural networks [15]. In MZI, matrix multiplication is usually calculated using weight matrix singular value decomposition, and the singular value decomposition calculation of the matrix is in equation (1).

$$W = U\Sigma V^T \tag{1}$$

Table 1: Comparative analysis of relevant work

Method	Key technologies	Accuracy	Calculation speed	Hardware complexity	Principal limitations
Tian et al. [7]	Kramer-Kronig activation	95.6%	57ns	Medium	Non-linear activation implementation complexity
Huang et al. [8]	Single-tuned phase shifter	85.3%	31ns	Low	Restricted to non-negative matrices
Ouyang J et al. [9]	Optical vector matrix multiplication	87.5%	1.66s	High	Insufficient real-time capability
Pai et al. [10]	In-situ backpropagation	94.0%	69ns	Medium	Complex training process
Zheng et al. [11]	Dual adaptive training	82.4%	35ns	Medium	Focused on error compensation
Ashtiani et al. [12]	End-to-end photonic networks	93.8%	570ps	High	Limited scalability
This study	Pseudo-real number theory + GridNet	98.2%	12ns	Medium	/

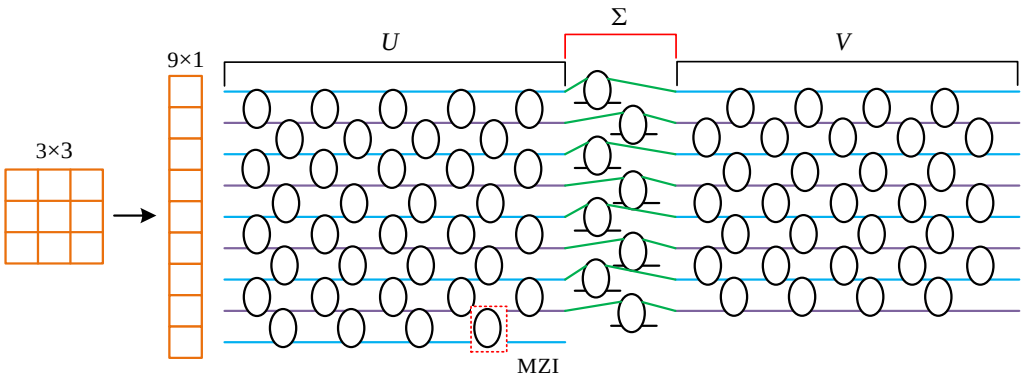


Figure 1: MZI array arrangement in CNN

In equation (1), W denotes a weight matrix of dimension $m \times n$, U denotes a unitary matrix of dimension $m \times m$, Σ denotes a diagonal matrix of dimension $m \times n$, V denotes a unitary matrix of dimension $n \times n$, and V^T denotes the complex conjugate of the unitary matrix V. The key to optical implementation lies in the fact that any unitary matrix can be physically realized through a MZI network, while a diagonal matrix can be achieved using a set of optical attenuators or amplifiers along a parallel waveguide. This study designs a linear matrix based on singular value decomposition theory, decomposing the transmission matrix (TM) into two unitary matrices and one diagonal matrix, and transferring them to an optical convolutional neural network (CNN) through mapping. This study is able to calculate any real valued matrix using this method, enabling data transmission in optical neural networks [16]. The MZI array arrangement in the CNN constructed in this study is in Figure 1.

In Figure 1, the convolutional kernels in the CNN are all unitary matrices using MZI, with a total of 8 kernels, each with a size of 3×3. The input of the convolutional layer is a 9×1 vector formed by expanding each unitary matrix. All unitary matrices are expanded to form a 9×8 matrix, which is then decomposed into two unitary matrices and one diagonal matrix using singular value decomposition, and connected using MZI. In the first unitary matrix area, there are a total of 36 MZI connection modules, in the diagonal matrix area, there are a total of 8 MZI modules, and in the second unitary matrix area, there are a total of 28 MZI modules. The MZI module in the unitary matrix region has two input terminals and two output terminals, while the MZI in the diagonal matrix region has only one input terminal and one output terminal. A global optical amplification device is added to the output terminals of all MZIs to compensate for attenuation

during optical signal transmission. The unitary matrix region specifically denotes a particular physical area on a photonic chip formed by multiple MZIs interconnected in a specific grid structure. The overall transmission matrix of this region is mathematically equivalent to a unitary matrix. The study adopts a grid arrangement to connect the MZIs in the convolutional layer, thereby achieving both matrix multiplication and addition simultaneously. MZI typically comprises an adjustable phase shifter and two beam splitters. The phase shifter imposes phase verification upon the optical mode, with its transmission matrix calculated as shown in equation (2) [17].

$$T_{PS} = \begin{bmatrix} e^{i\phi/2} & 0 \\ 0 & e^{-i\phi/2} \end{bmatrix} \tag{2}$$

In equation (2), T_{PS} represents the TM, ϕ represents the phase difference introduced by the phase shifter. The power ratio of reflected and transmitted light in the beam splitter is 50:50, resulting in the mixing of two input light signals. The transmission matrix of the beam splitter is calculated as shown in equation (3).

$$T_{BS} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & i \\ i & 1 \end{bmatrix} \tag{3}$$

In equation (3), T_{BS} represents the TM of the beam splitter, i represents the imaginary unit, and the first column and first row of the TM of the beam splitter represent the amplitude coefficients of transmission and reflection, respectively. In the grid arrangement of the MZI, it is typically formed by cascading phase shifters, beam splitters, phase shifters, and beam splitters in sequence. This is represented as a 2×2 unit transmission matrix, derived by sequentially multiplying the transmission matrices. The overall transmission matrix of the MZI is shown in equation (4).

$$T_{MZI} = \begin{bmatrix} i\sin(\phi/2)e^{i\phi/2} & i\cos(\phi/2)e^{-i\phi/2} \\ \cos(\phi/2)e^{i\phi/2} & \sin(\phi/2)e^{-i\phi/2} \end{bmatrix} \tag{4}$$

In equation (4), T_{MZI} represents the overall TM of MZI. The study adopts the principle of rectangular decomposition for unitary matrix decomposition, and the unitary matrix obtained after 4 rounds of traversal is in equation (5).

$$U = DT_{1,2}T_{2,3}T_{3,4}T_{1,2}T_{3,4}^{-1}T_{2,3}^{-1} \tag{5}$$

In equation (5), D represents the diagonal matrix and T represents the rotation matrix. The study adopts spatial light theory to construct a non-destructive optical beam splitter, and uses two phase shifters to record the phase changes of two incident lights. The relationship between the input and output ports of the non-destructive optical beam splitter is in equation (6).

$$\begin{bmatrix} O_1 \\ O_2 \end{bmatrix} = \begin{bmatrix} e^{i\varphi}\sin\omega & \cos\omega \\ e^{i\varphi}\cos\omega & -\sin\omega \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \tag{6}$$

In equation (6), O_1 represents output optical signal 1, O_2 represents output optical signal 2, φ represents the phase value of the phase shifter, ω represents the transmission property parameters of the beam splitter, I_1 represents input optical signal 1, and I_2 represents input optical signal 2. The calculation of $\sin\omega$ and $\cos\omega$ for the beam splitter is in equation (7).

$$\begin{cases} \sin\omega = \sqrt{r} \\ \cos\omega = \sqrt{t} \end{cases} \tag{7}$$

In equation (7), r represents the reflectivity of the beam splitter and t represents the refractive index of the beam splitter. In the array composed of MZIs, the linear matrix obtained after inputting the optical signal is in equation (8).

$$\begin{bmatrix} W_{11} & \cdots & W_{1N} \\ \vdots & \ddots & \vdots \\ W_{N1} & \cdots & W_{NN} \end{bmatrix} \cdot \begin{bmatrix} I_1 \\ \vdots \\ I_N \end{bmatrix} = \begin{bmatrix} O_1 \\ \vdots \\ O_N \end{bmatrix} \tag{8}$$

After obtaining the linear matrix, singular value decomposition is used to decompose the weight matrix, as shown in equation (9).

$$O = U\Sigma V^T \cdot I \tag{9}$$

By utilizing the principle of rectangular decomposition, the unitary matrices U and V^T are processed, and the decomposed MZI array is in Figure 2.

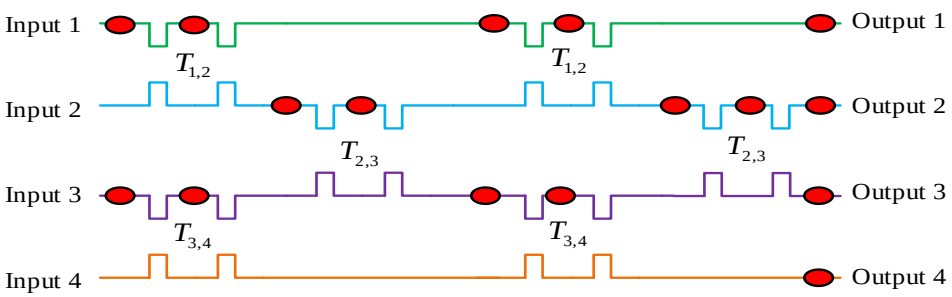


Figure 2: MZI array after unitary matrix decomposition

In Figure 2, there are four optical signal channels. After the optical signal is input into the MZI, it passes through $T_{1,2}$, $T_{3,4}$, $T_{2,3}$, $T_{1,2}$, $T_{3,4}$, and $T_{2,3}$ in sequence before being output. At the same time, to assess the functionality of the photon neural network constructed by MZI array, fidelity is used to measure the similarity between the target matrix and the matrix formed by MZI, as calculated in equation (10).

$$F = \left| \frac{\delta\left(U^{\dagger} U_g\right)}{N} \right|^2 \tag{10}$$

In equation (10), F represents the similarity between two matrices, δ represents the trace of the target matrix, $U^{\dagger}$ represents the conjugate transpose of the MZI matrix, U_g represents the target matrix, and N represents the size of the target matrix. The structure of the constructed photon neural network is in Figure 3.

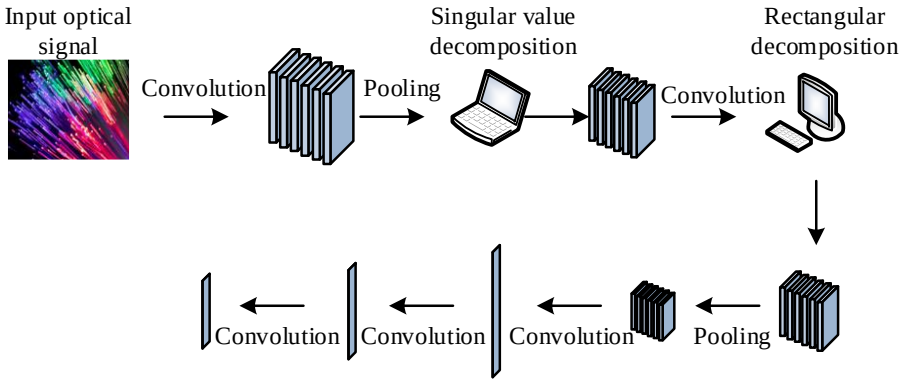


Figure 3: Specific structure of photon neural network

In Figure 3, the optical signal is input through two layers of 4×4 convolution kernels, and the transposed convolution kernels are processed using singular value decomposition. The decomposed matrix is then subjected to two max pooling layers, and the obtained unitary matrix is processed using the principle of rectangular decomposition. The processed information is sequentially processed through two convolutional layers, two max pooling layers, and three convolutional layers before being output.

2.2 Optimization of photon neural networks based on pseudo-real number theory

The TM of the photon neural network constructed using MZI can only represent the amplitude information of light in the real domain, while it can simultaneously represent amplitude and phase information in the complex domain, thus achieving more complex linear and nonlinear operations [18]. Therefore, the study adopts a training method based on pseudo-real number theory to optimize the TM of the photon neural network. Pseudo-real number theory constitutes a computational framework that maps optical signals from the complex domain to the real domain via optical projection computational units (such as directional couplers). This enables traditional real-number-based optimization algorithms to be directly applied for training photonic neural networks. Specifically, this theory leverages interference principles to extract real-part information from complex fields, thereby circumventing complex number operations while preserving critical phase information. The projection calculation unit constitutes a specific optical path structure formed by directional couplers. Its function is to interfere the complex optical field output from the MZI network with a reference light source. By measuring the intensity of the resulting interference pattern, it projects the real-part information of the complex optical field onto the measurable intensity domain. The study uses two directional couplers to replace the beam splitter in MZI. The improved MZI device is expressed using a two-dimensional unitary matrix, which exhibits tunable phase-shifting capability and wide adaptability. The output matrix of the improved MZI is in equation (11).

$$W' = \begin{bmatrix} e^{i\varphi}\sin\theta & e^{i\varphi}\cos\theta \\ \cos\theta & -\sin\theta \end{bmatrix} \tag{11}$$

In equation (11), W' represents the output matrix of the improved MZI and θ represents the parameters of the internal phase shifter. According to the theory of pseudo-real numbers, any array composed of multiple different MZI devices can perform matrix multiplication operations to complete computational tasks. The input and output optical signals of multiple MZI networks are shown in equation (12) [19].

$$\begin{bmatrix} \sqrt{O_1}\exp\left(i\psi_1\right) \\ \sqrt{O_2}\exp\left(i\psi_2\right) \\ \sqrt{O3}\exp\left(i\psi_3\right) \end{bmatrix} = T_{MZI} \begin{bmatrix} \sqrt{I_1} \\ \sqrt{I_2} \\ \sqrt{I_2} \end{bmatrix} \tag{12}$$

In equation (12), ψ_1 represents the phase shift of the first MZI output optical signal, ψ_2 represents the phase shift of the second MZI output optical signal, and ψ_3 represents the phase shift of the third MZI output optical signal. Because commonly-used detection methods can only measure the intensity of light and cannot directly measure the phase value, only the real matrix calculated by the MZI array can be used. After introducing the theory of pseudo-real numbers, there is no longer a need for multiple MZI arrays to jointly calculate matrices. By using projection calculation units and any MZI network, all real matrices can be calculated. The study uses a directional coupler as the corresponding projection calculation unit, and the arbitrary output optical signal and initial input optical signal of MZI are used as the input terminals of the directional coupler. One of the output terminals of the directional coupler is calculated as shown in equation (13).

$$O_{z1} = |E_I|^2 + |E_O|^2 - 2|E_O||E_I|\cos\varphi' \quad (13)$$

In equation (13), O_{z1} represents the first output of the directional coupler, E_I represents the initial input light, E_O represents any output light of the MZI, φ' represents the phase difference between two input optical signals, the phase value of the initial input light is taken as zero, and the amplitude is taken as unit 1. Meanwhile, because the initial input light is directly fed into the directional coupler, while the output light undergoes attenuation after passing through the MZI network, the intensity of the input light is significantly higher than that of the output light. The output of the directional coupler is approximately equal to the real part of the amplitude of the output light plus one unit. The output calculation of the second directional coupler is in equation (14).

$$O_{z2} = |E_I|^2 + |E_O|^2 + 2|E_O||E_I|\cos\varphi' \quad (14)$$

In equation (14), O_{z2} represents the output of the second directional coupler. By using a detector to extract the output signal of the directional coupler and then processing it with high pass filtering, accurate output optical amplitude real part can be obtained. Therefore, based on the theory of pseudo-real numbers, a projection calculation unit is introduced to optimize the TM of MZI array and convert it into a unitary matrix. After the conversion is completed, any MZI can be used to calculate any real matrix. The study used GridNet network architecture and MZI array to build a photon neural network. The training process of GridNet photon neural network is in Figure 4.

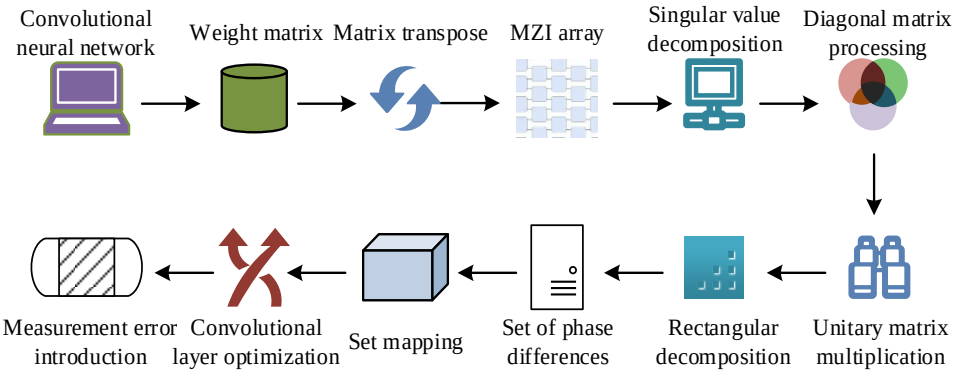


Figure 4: Specific training flow of GridNet photon neural network

In Figure 4, a CNN consisting of two convolutional layers is first set up, with a convolutional layer size of 4×4. The study calculates the weight matrix of the CNN, then transposes the matrix, and uses the transposed matrix as the TM of MZI. Singular value decomposition is performed on the TM to obtain a diagonal matrix and two unitary matrices, both with a scale of 4×4. The diagonal matrix is processed to ensure that each element is less than or equal to 1, thereby simulating the attenuation effect of MZI on light. The research adds imaginary parts to each element of a diagonal matrix to create a new unitary matrix. The three unitary matrices are multiplied to obtain a new TM. The transfer matrix is also a unitary matrix, which can be expressed using the Grid Wise Network (GridNet). Using the rectangular decomposition method to process the new TM, a set of MZI phase difference values is obtained, and then the set of phase difference values is mapped to the MZI array. Based on the mapped MZI array layout, convolutional layer optimization is performed by changing the weights during convolutional layer training to phase difference values, adjusting the training focus position, and introducing various measurement errors that may occur in MZI actual use, such as beam splitters, phase shifters, and detectors. GridNet's architecture comprises two convolutional layers, two max-pooling layers, two convolutional layers, two max-pooling layers, and three convolutional layers. The ReLU activation function is employed, utilizing the Adam optimizer with a batch size of 128. GridNet adopts L2 regularization with a learning rate of 0.001. The cross-entropy loss function (LF) of the photon neural network is in equation (15) [20-21].

$$L = -\frac{1}{x}\sum_{j=1}^{x}\sum_{k=1}^{K} p\left(s_j^k\right)\log q\left(s_j^k\right) \quad (15)$$

In equation (15), L represents the cross entropy LF, x represents the total number of data, K represents the total number of data categories, k represents a certain data category, j represents the number of data, j represents the true probability that data j belongs to category k, and $q\left(s_j^k\right)$ represents the predicted probability that data j belongs to category k. The hardware architecture of the constructed photon neural network is in Figure 5.

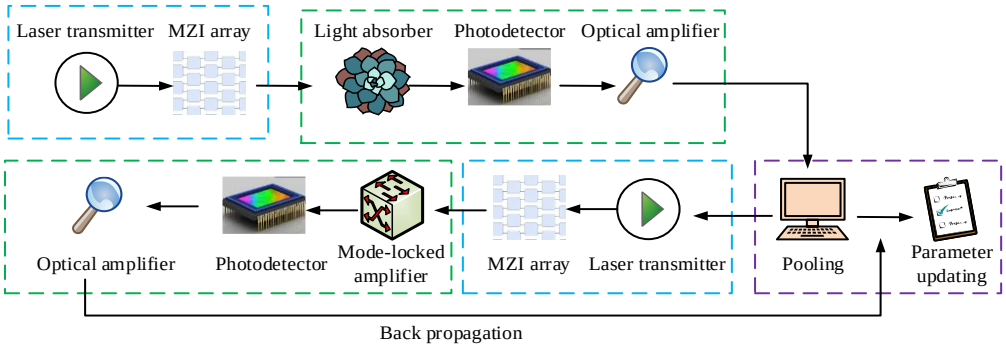


Figure 5: Hardware architecture of photon neural network

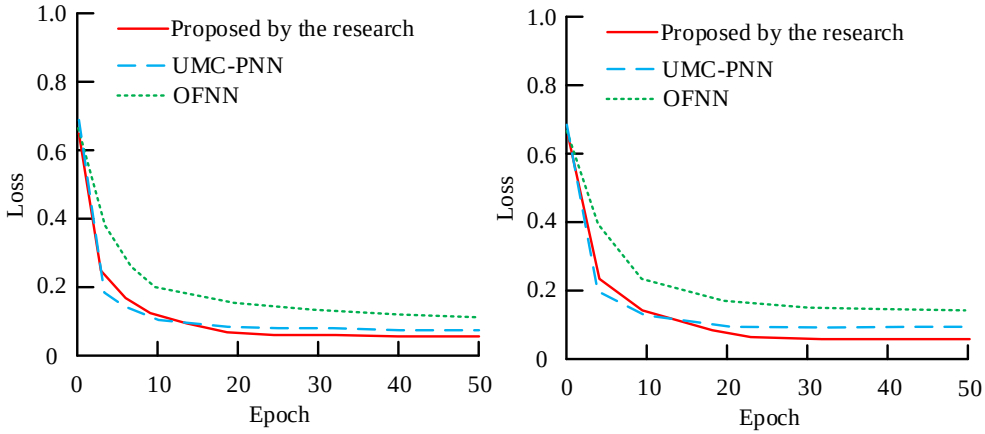


Figure 6: Comparison of LFs of different photon neural networks

In Figure 5, the photon neural network consists of two parts: optical and electrical. The optical part includes a laser emitter and an MZI array, while the nonlinear layer of the optical part consists of a light absorber, a detector, and a global optical amplification device. The optical signal is finally input into the field programmable gate array through an amplification device for pooling calculation. After the calculation is completed, it is then input into the fully connected layer composed of a laser emitter and an MZI array. Then, a lock-in amplifier is used for optical signal amplification to improve detection accuracy. The final output signal can be backpropagated to the field programmable gate array for parameter updating of the photon neural network. The evaluation metrics selected for this study encompassed not only the cross-entropy loss function but also accuracy, F1 score, and root mean square error (RMSE). Among these, accuracy served as the most intuitive indicator in classification tasks, reflecting the model's overall capability for correct classification. To address potential misleading interpretations of accuracy on imbalanced datasets, the study employed the F1 score, which represented the harmonic mean of precision and recall. This metric provided a more comprehensive assessment of the model's balanced performance across all categories, proving particularly suitable for verifying whether the model exhibited bias or blind spots towards specific categories. RMSE was employed to measure the discrepancy between the probability distribution of the network's predicted outputs and the true one-hot encoded label distribution. A lower RMSE indicated that the network's predicted probabilities were numerically closer to the true labels, reflecting not only classification accuracy but also the quality and calibration of the model's output confidence levels.

3 Results

3.1 Experimental analysis of photon neural network based on MZI

The hardware environment for running the photon neural network proposed by the research was Intel Core i5-12600H @ 2.60 GHz, GPU was NVIDIA GeForce RTX 3060Ti, memory was 16GB, operating system was Windows 11, and programming language was Python 3.8.0. During training, the learning rate of the neural network was set to 0.001, and the maximum number of iterations was set to 50. The experimental results were obtained from numerical simulation environments. Actual hardware implementation must account for non-ideal factors such as thermal noise and manufacturing tolerances, which will be further explored in future work. The study used the Modified National Institute of Standards and Technology database (MNIST) handwritten font recognition dataset for experimental analysis. The dataset contains 60000 training samples and

10000 test samples, each of which is a 28×28 grayscale image. The study utilized comparative approaches such as Optical Feedforward Neural Network (OFNN) and United Microelectronics Center Photon Neural Network (UMC-PNN). All experiments were conducted multiple times, with five randomized replicates, and the results were averaged. The comparison of LFs for different photon neural networks is in Figure 6.

In Figure 6 (a), when the iteration count was 5, the LFs of each method gradually converged. The proposed method tended to converge at 20 iterations, with a minimum LF value of 0.08, which was 0.02 and 0.09 lower than UMC-PNN and OFNN, respectively. UMC-PNN converged quickly in the early stages of iteration, but the minimum LF was lower than the method proposed in the study. In Figure 6 (b), when the iteration number was 0, that is, without training, the initial LFs of the three methods were relatively similar. In the test set, the minimum LF of the proposed method remained basically unchanged, and the initial convergence speed decreased slightly. The LFs of UMC-PNN and OFNN decreased by 0.03 and 0.02, respectively. The comparison of recognition accuracy between different photon neural networks is in Figure 7.

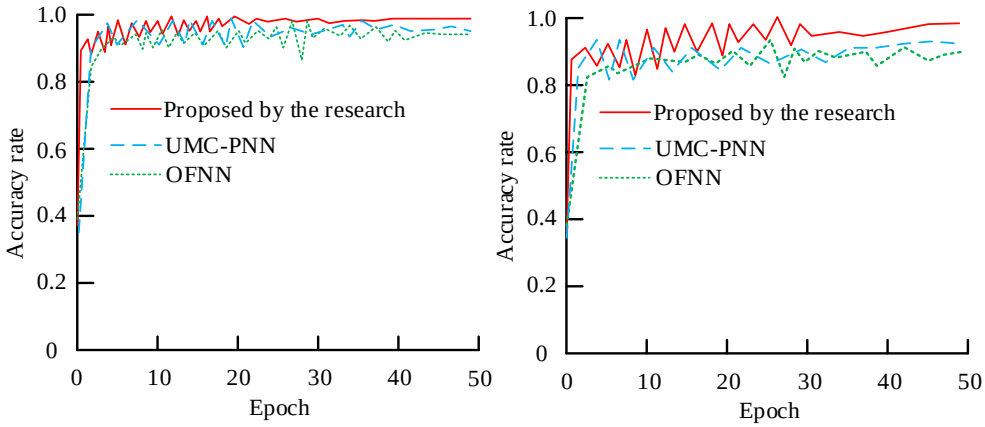


Figure 7: Comparison of recognition accuracy rates of different photon neural networks

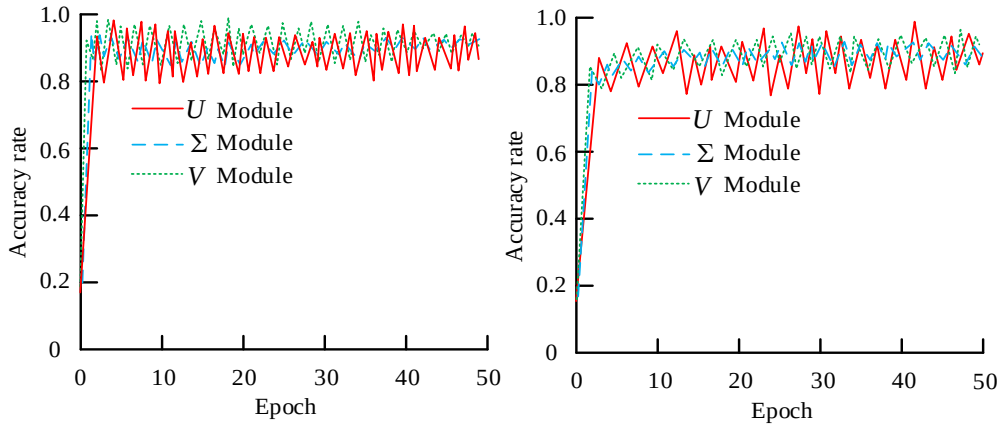


Figure 8: Comparison of recognition accuracy of different modules facing noise

In Figure 7 (a), the initial recognition accuracies of the proposed method, UMC-PNN, and OFNN were 0.374, 0.352, and 0.381, respectively. All three methods approached convergence after about 10 iterations, but there were significant fluctuations. After training, the maximum recognition accuracy of the proposed method was 0.982, which was 0.012 and 0.017 higher than UMC-PNN and OFNN, respectively. In Figure 7 (b), in the test set, the final recognition accuracy of the proposed method was basically the same, but the fluctuation amplitude increased significantly in the early stage of iteration. The final recognition accuracy of the proposed method was 0.065 and 0.073 higher than UMC-PNN and OFNN, respectively. In order to investigate the sensitivity of different modules in the MZI array to noise, random noise with a value of 0.2 was introduced into two unitary matrix modules and one diagonal matrix module. The comparison of the recognition accuracy of different modules in the face of noise is in Figure 8.

In Figure 8 (a), when noise was added to the unitary matrix U module, the model's recognition accuracy fluctuated the most because the U module had the most MZI arrays and was more affected by noise. When noise was added to the diagonal matrix Σ module, the model's fluctuation amplitude was the smallest. The average convergence values of the Σ module were 0.027 and 0.013 higher than those of the U and V modules,

respectively. In Figure 8 (b), due to the use of more MZI arrays in the UMC-PNN model, the fluctuation amplitude of its recognition accuracy increased. Similarly, when noise was added to the U module, the optimized model's recognition accuracy fluctuated the most. To further investigate the system's robustness to noise, the variation in recognition accuracy was tested under different noise levels (0.05, 0.1, 0.2, 0.3). The results are shown in Figure 9.

In Figure 9(a), when noise intensity was below 0.15, system performance showed no significant decline, with accuracy remaining above 0.90. However, performance deteriorated sharply when noise intensity exceeded 0.15. Results indicated the proposed method exhibited good tolerance to moderate noise levels. In Figure 9(b), the UMC-PNN model experienced a more rapid decline in accuracy under higher noise intensities.

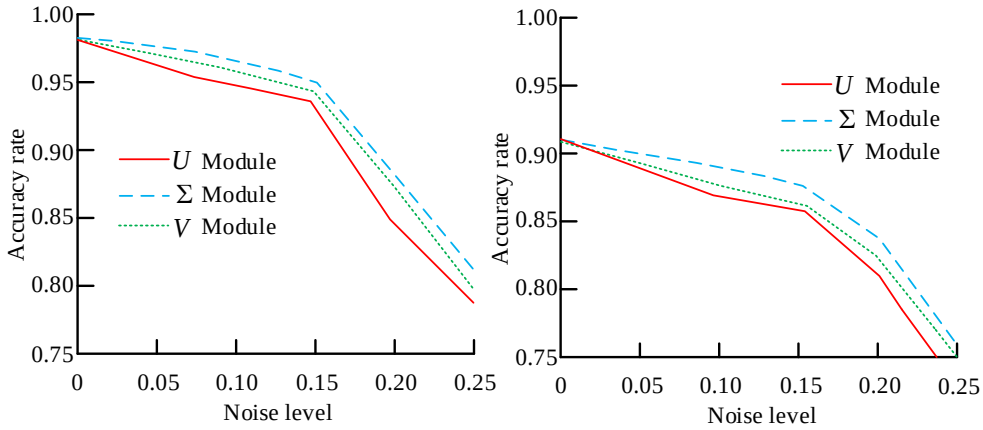


Figure 9: Robustness analysis of the model under different noise levels

Table 2: Compares the recognition performance of handwritten digits by different methods

Model	Calculation speed	Recognition accuracy	F1(%)	RMSE
Pseudo-real optimization	12ns	0.982	84.2	0.276
UMC-PNN	25ns	0.917	80.5	0.293
OFNN	19ns	0.909	78.7	0.307
CNN	24ms	0.914	82.6	0.284
SVM	15ms	0.896	79.5	0.295
KNN	7ms	0.857	76.3	0.302
DT	17ms	0.905	78.2	0.289
DBN	65ms	0.943	83.4	0.278

3.2 Experimental analysis of photon neural network optimized based on pseudo-real numbers

To investigate the recognition speed of handwritten digits using different methods, five comparative methods were added: CNN, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Decision Tree (DT), and Deep Belief Network (DBN). The comparison of recognition performance of different methods for handwritten digits is in Table 2.

In Table 2, the research results were all theoretical calculations based on simulation models. The computational speed units for the three photonic neural networks were all ns, differing by six orders of magnitude from conventional convolutional neural networks and deep learning methods. Furthermore, after pseudo-real number optimization, the computational speeds of the photonic neural networks were 13 ns and 7 ns faster than UMC-PNN and OFNN respectively. The proposed method achieved the highest recognition accuracy among the evaluated approaches, alongside the optimal F1 score and RMSE. Photonic chips exhibited extremely low power consumption in static states, with primary energy expenditure arising from electro-optic conversion and phase modulators. Preliminary analysis indicated their energy efficiency potential significantly surpassed that of conventional electronic chips. To validate the method's generalization capability, experiments were conducted on the more complex Fashion-MNIST dataset. This dataset comprises 28×28 greyscale images of 10 categories of clothing items, totalling 70000 samples. The experimental results are presented in Table 3.

In Table 3, when confronted with the more complex Fashion-MNIST dataset, the performance of all algorithms declined. However, the computational speed of the proposed method decreased by merely 2 ns,

whereas the second-best OFNN algorithm saw a reduction of 6 ns. The proposed method achieved a recognition accuracy of 0.979, representing a decrease of 0.003 compared to the MNIST dataset, while the second-best UMC-PNN method saw a decline of 0.015. The proposed method maintained a relative advantage across all performance metrics, demonstrating high scalability. The proposed method exhibited lower standard deviations than the reference method across all performance metrics, with statistically significant differences observed between the results of each method (p<0.05). The LF and recognition accuracy of the optimized photon neural network using pseudo-real number theory are shown in Figure 10.

Table 3: Experimental results on the Fashion-MNIST dataset using different methods

Model	Calculation speed	Recognition accuracy	F1(%)	RMSE
Pseudo-real optimization	14±0.02ns	0.979±0.013	83.7±0.01	0.281±0.002
UMC-PNN	37±0.17ns	0.902±0.037	79.1±0.19	0.312±0.015
OFNN	25±0.35ns	0.895±0.029	75.4±0.32	0.325±0.027
CNN	43±1.24ms	0.897±0.041	78.2±0.58	0.304±0.042
t	12.716	11.563	14.092	11.864
p	0.001	0.005	0.000	0.004

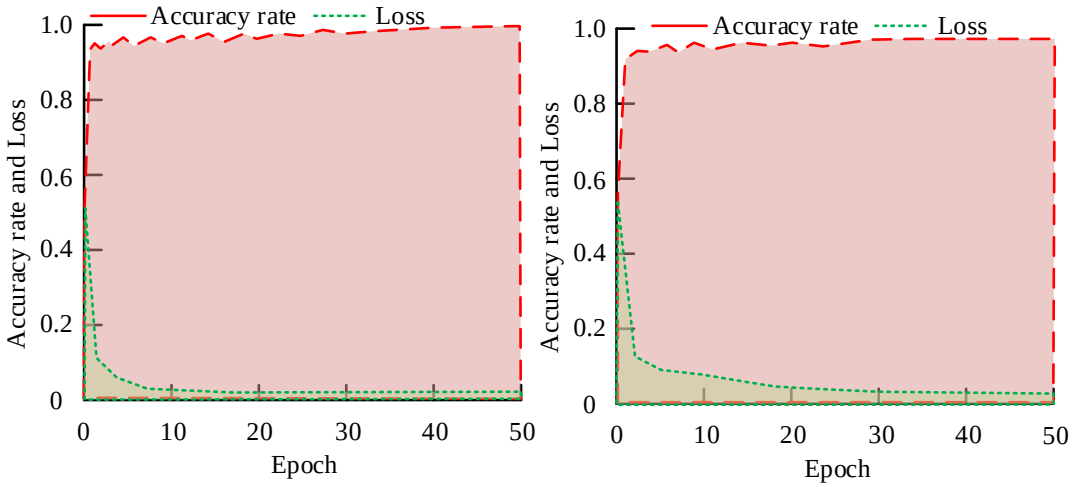


Figure 10: Changes in LF and recognition accuracy of photon neural network after optimization

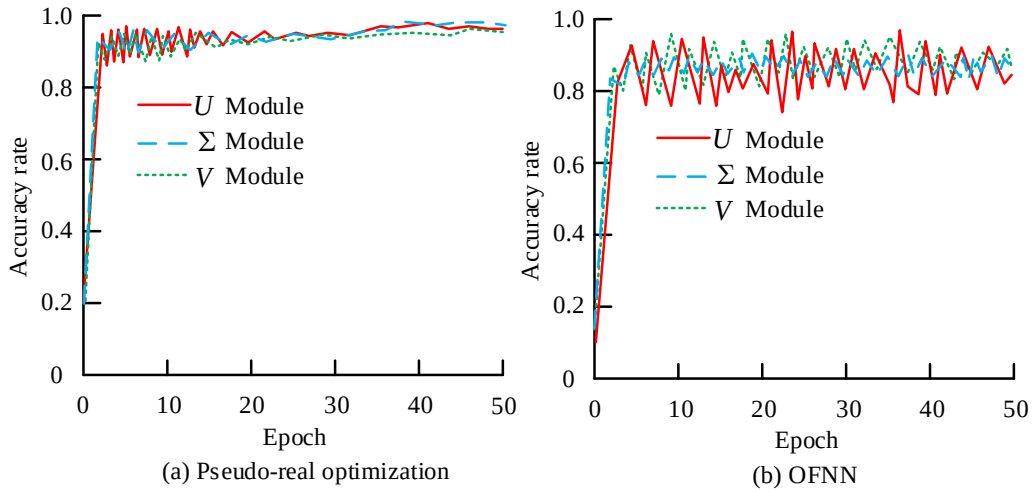


Figure 11: Comparison of recognition accuracy of different modules facing noise

In Figure 10 (a), after optimizing using pseudo-real number theory, the performance of the photon neural network was improved, with a maximum recognition accuracy and minimum LF of 0.987 and 0.05, respectively. In addition, the fluctuation amplitude of the photon neural network in the early stage of iteration was greatly reduced,

improving the stability of the network. In Figure 10 (b), in the test set, the maximum recognition accuracy of the improved network was 0.985. Although it decreased compared to the training set, it was still higher than before, with a minimum LF of 0.07. Random noise of 0.2 was introduced into the optimized network, and the comparison of recognition accuracy of different modules in the face of noise is in Figure 11.

In Figure 11 (a), the fluctuation amplitude of recognition accuracy of different modules of the optimized photon neural network was significantly reduced when facing random noise. All three modules could gradually converge in the later stage of iteration, and the maximum convergence value of the Σ module was 0.011 and 0.012 higher than that of the U module and V module, respectively. In Figure 11 (b), the OFNN model was greatly affected by random noise interference, and the model remained in a fluctuating state throughout the entire iteration cycle, unable to achieve convergence values. The ablation experiment analysis of the improved photon neural network is in Table 4 for comparison of experimental results.

Table 4: Results of improved photon neural network ablation experiments

Ablation module	Calculation speed	Recognition accuracy	F1(%)	RMSE
Singular value decomposition	18ns	0.954	81.5	0.295
Rectangular decomposition	15ns	0.936	83.2	0.284
Directional coupler	27ns	0.973	83.0	0.287
The theory of pseudo-real numbers	20ns	0.892	78.4	0.317
GridNet network architecture	14ns	0.903	79.5	0.309

Table 5: Comparison of the proposed method with existing advanced optical neural networks

Method	Accuracy	Loss	Speed	Noise Robustness	Scalability
Proposed	0.982	0.08	12ns	High	Medium
PCNN	0.917	0.10	25ns	Medium	High
UMFPNN	0.909	0.17	19ns	Low	Medium
Ashtiani et al. [12]	0.938	0.12	0.57ns	Medium	Low
Pai et al. [10]	0.940	0.11	/	High	Medium

In Table 4, the directional coupler module had the greatest impact on the computational speed of the improved photon neural network. After removing this module, the algorithm's calculation speed increased by 15ns and the recognition accuracy decreased by 0.009. The pseudo-real theory exerted the most significant influence on the accuracy of algorithm recognition. After removing this module, the recognition accuracy of the algorithm decreased by 0.090 and the RMSE increased by 0.041. The pseudo-real number theory preserved the phase information of complex optical fields through mathematical projection, mapping it onto real-valued quantities. This enabled the computation of arbitrary real matrices within a single MZI array. Conventional methods required multiple MZI arrays to process complex matrix operations, as standard detectors could not capture phase information, leading to information loss. Pseudo-real number theory efficiently integrated the amplitude and phase information of complex optical fields into real-number computations through directional couplers and approximate calculations, thereby preventing phase information loss. This highly efficient phase-preserving computation method enabled photonic neural networks to utilize the full information content of the optical field more comprehensively, consequently enhancing the recognition accuracy of the algorithms. The proposed method was compared with existing advanced optical neural networks as shown in Table 5.

In Table 5, the advanced comparison methods introduced in the experiment included the Photon Convolutional Neural Network (PCNN) and the Unitary Matrix Factorisation Photonic Neural Network (UMFPNN). The proposed method achieved the highest recognition accuracy and lowest loss value, with an accuracy 0.042 percentage points higher than the second-best method proposed by Pai et al. and a loss value 0.02 lower than PCNN. The proposed method exhibited high noise robustness and moderate scalability. The primary challenge faced by the proposed method in large-scale MZI arrays was noise accumulation. As illustrated in Figure 8, the unitary matrix module

exhibited the highest sensitivity to noise due to its largest number of MZI units. Compared to the dual adaptive training method proposed by Zheng et al. in [11], this research employed pseudo-real number theory to project phase information onto the real number domain, mitigating the accumulation of systematic errors to some extent. Nevertheless, when scaling the array further, strategies such as error prediction networks or quantization-aware training became necessary to ensure system stability. Regarding scalability, the GridNet architecture enabled efficient mapping of unitary matrices through rectangular decomposition, offering a viable pathway for realizing large-scale photonic neural networks

4 Discussion

To enhance the computational performance of photonic neural networks, this study proposes an improved photonic neural network based on pseudo-real number theory and the GridNet architecture. Compared to the UMC-PNN in Reference [8], the proposed method achieved a 6.5% improvement in recognition accuracy. The UMC-PNN relied on a single tunable phase shifter to perform vector multiplication on non-negative or real-valued matrices, thereby limiting its expressive capability for complex transformations. Conversely, the improved approach innovatively introduced projection computation units within a pseudo-real number framework, enabling a single MZI network to compute arbitrary real matrices. This substantially enhanced the network's feature extraction capabilities and flexibility. Compared to the end-to-end photonic network by Ashtiani et al. in [12], the improved method achieved higher recognition accuracy (0.982 vs 0.938) while maintaining high-speed computation through the MZI array, which provided precise and reconfigurable matrix weights. As illustrated in Figures 8 and 0, the improved method exhibited varying sensitivity to noise across different modules. The unitary matrix module, owing to its numerous MZIs, demonstrated the highest noise sensitivity, consistent with observations by Banerjee et al. [19]. Following optimization via pseudo-real theory, the system's noise robustness was significantly enhanced, indicating that this theory not only improved training efficacy but also fortified system stability. The study employed directional couplers as substitutes for beam splitters, a feasible approach from an integrated photonics manufacturing perspective. On silicon photonics chips, directional couplers served as core components for optical power splitting and combining, with manufacturing processes fully compatible with CMOS techniques. Compared to free-space beam splitters requiring separate fabrication and precision mechanical alignment, monolithic integrated directional couplers inherently offered advantages in size, stability, and scalability. Consequently, the research did not introduce an unconventional hardware modification but employed a more typical and mature component within integrated photonics to realize MZI functionality. Simultaneously, ablation experiments demonstrated the feasibility of incorporating directional couplers. Whilst simulation results were encouraging, practical hardware implementation still faces numerous challenges. These include manufacturing tolerances for directional couplers, the thermal sensitivity of the MZI, and crosstalk issues during large-scale integration, all requiring further resolution. Future work will focus on optimizing these practical issues to advance the methodology towards practical applications.

5 Conclusion

Aiming at the problem of traditional electronic chip computing performance approaching its limit, a photon neural network algorithm based on pseudo-real number theory was proposed for optical components. The experiment outcomes indicated that the LF of the photon neural network tended to converge after 20 iterations, with a minimum LF value of 0.08, which was 0.02 and 0.09 lower than UMC-PNN and OFNN, respectively. The photon neural network approached convergence after about 10 iterations, but the fluctuation amplitude was large. Its maximum recognition accuracy was 0.982, which was 0.012 and 0.017 higher than UMC-PNN and OFNN, respectively. The U module had the highest number of MZI arrays and was more affected by noise. The average convergence values of the Σ module were 0.027 and 0.013 higher than those of the U and V modules, respectively. The computing speed unit of photon neural networks was ns, which was 6 orders of magnitude lower than ordinary CNNs and deep learning methods. The maximum recognition accuracy and minimum LF of the optimized photon neural network were 0.987 and 0.05, respectively. When different modules of the optimized photon neural network faced random noise, the fluctuation amplitude of recognition accuracy was significantly reduced, and all three modules could gradually converge in the later stage of iteration. The directional coupler module had the greatest impact on improving the computational speed of photon neural networks. After removing this module, the algorithm's computation time increased by 15ns. The pseudo-real number theory had the greatest impact on recognition accuracy. After removing this module, the recognition accuracy decreased by 0.090. There are still some issues with this study, such as the proposed method mainly focusing on the application of photon neural networks at the linear level. In the future, nonlinear calculations of photon neural networks can be achieved based on the nonlinear characteristics of optics.

References

[1] Du Y, Su K, Yuan X, Li T, Liu K, Man H, Zou X. Implementation of optical neural network based on Mach–Zehnder interferometer array. IET

Optoelectronics. 2023, 17(1):1-11. https://doi.org/10.1049/ote2.12086

[2] Norris B R, Martinod M A, Tuthill P, Gross S, Cvetojevic N, Jovanovic N, et al. Machine-learning approach for optimal self-calibration and fringe tracking in photonic nulling interferometry. Journal of Astronomical Telescopes, Instruments, and Systems. 2023, 9(4):048005. https://doi.org/10.1117/1.JATIS.9.4.048005

[3] Yang F, Tian Q, N X, Pan Y, Wang F, Lá zaro J A, et al. Modulation format recognition scheme based on discriminant network in coherent optical communication system. Electronics. 2024, 13(19):3833-3845. https://doi.org/10.3390/electronics13193833

[4] Zhang L, Tian X, Jiang Y, Li X, Li Z, Li D, Zhang S. Adversarial network for multi-input image restoration under strong turbulence. Optics Express. 2023, 31(25):41518-41532. https://doi.org/10.1364/oe.503611

[5] Khonina S N, Kazanskiy N L, Skidanov R V, & Butt M A. Exploring types of photonic neural networks for imaging and computing—a review. Nanomaterials, 2024, 14(8): 697-709. https://doi.org/10.3390/nano14080697

[6] Huang C, Sorger V J, Miscuglio M, Al-Qadasi M, Mukherjee A, Lampe L, et al. Prospects and applications of photonic neural networks. Advances in Physics: X. 2022, 7(1):1981155-1981172. https://doi.org/10.1080/23746149.2021.1981155

[7] Tian Y, Zhao Y, Liu S, Li Q, Wang W, Feng J, Guo J. Scalable and compact photonic neural chip with low learning-capability-loss. Nanophotonics, 2022, 11(2):329-344. https://doi.org/10.1515/nanoph-2021-0521

[8] Huang Y, Yue H, Ma W, Zhang Y, Ao Y, Wang W, Chu T. Easily scalable photonic tensor core based on tunable units with single internal phase shifters. Laser & Photon Reviews. 2023, 17(10):2300001-2300015. https://doi.org/10.1002/lpor.202300001

[9] Ouyang J, Liu S, Yang Z, Wang W, Feng X, Li Y, Huang Y. 16-channel photonic solver for optimization problems on a silicon chip. Chip. 2025, 4(1):100117-100134. https://doi.org/10.1016/j.chip.2024.100117

[10] Pai S, Sun Z, Hughes T W, Park T, Bartlett B, Williamson I A, et al. Experimentally realized in situ backpropagation for deep learning in photonic neural networks. Science. 2023, 380(6643):398-404. https://doi.org/10.1126/science.ade8450

[11] Zheng Z, Duan Z, Chen H, Yang R, Gao S, Zhang H, Lin X. Dual adaptive training of photonic neural networks. Nature Machine Intelligence. 2023, 5(10):1119-1129. https://doi.org/10.1038/s42256-023-00723-4

[12] Ashtiani F, Geers A J, Aflatouni F. An on-chip photonic deep neural network for image classification. Nature. 2022, 606(7914):501-506. https://doi.org/10.1038/s41586-022-04714-0

[13] Kirtas M, Oikonomou A, Passalis N, Mourgias-Alexandris G, Moralis-Pegios M, Pleros N, Tefas A. Quantization-aware training for low precision photonic neural networks. Neural Networks. 2022, 155(4):561-573. https://doi.org/10.1016/j.neunet.2022.09.015

[14] Liao K, Dai T, Yan Q, Hu X, Gong Q. Integrated photonic neural networks: Opportunities and hallenges. ACS Photon. 2023, 10(7):2001-2010. https://doi.org/10.1021/acsphotonics.2c01516

[15] Wang T, Ma S Y, Wright L G, Onodera T, Richard B C, & McMahon P L. An optical neural network using less than 1 photon per multiplication. Nature Communications, 2022, 13(1), 123-135. https://doi.org/10.1038/s41467-021-27774-8

[16] Tsakyridis A, Moralis-Pegios M, Giamougiannis G, Kirtas M, Passalis N, Tefas A, Pleros N. Photonic neural networks and optics-informed deep learning fundamentals. APL Photon. 2024, 9(1): 011102. https://doi.org/10.1063/5.0169810

[17] Freire P, Manuylovich E, Prilepsky J E, Turitsyn S K. Artificial neural networks for photonic applications—from algorithms to implementation: tutorial. Advances in Optics and Photon. 2023, 15(3):739-834. https://doi.org/10.1364/AOP.484119

[18] Totovic A, Giamougiannis G, Tsakyridis A, Lazovsky D, Pleros N. Programmable photonic neural networks combining WDM with coherent linear optics. Scientific Reports. 2022, 12(1):5605-5617. https://doi.org/10.1038/s41598-022-09370-y

[19] Banerjee S, Nikdast M, Chakrabarty K. Characterizing coherent integrated photonic neural networks under imperfections. Journal of Lightwave Technology. 2022, 41(5):1464-1479. https://doi.org/10.1109/JLT.2022.3193658

[20] Ewaniuk J, Carolan J, Shastri B J, Rotenberg N. Imperfect quantum photonic neural networks. Advanced Quantum Technologies. 2023, 6(3):2200125-2200139. https://doi.org/10.1002/qute.202200125

[21] Paolini E, De Marinis L, Cococcioni M, Valcarenghi L, Maggiani L, Andriolli N. Photonic-aware neural networks. Neural Computing and Applications. 2022, 34(18):15589-15601. https://doi.org/10.1007/s00521-022-07243-z

Interval Valued Intuitionistic Fuzzy Logic for Anomaly Detection in Public Health Monitoring Signals

Panpan You, Shan Li*
Yueyang Hospital of Traditional Chinese Medicine, Yueyang, Hunan, 414020, China
E-mail: LSyy1332763@163.com
*Corresponding author

Keywords: interval valued fuzzy logic, public health monitoring, signal processing, feature extraction, anomaly detection

Received: October 28, 2025

Interval valued fuzzy logic can more flexibly and comprehensively characterize and handle uncertainty and fuzziness in information by extending the membership degree of fuzzy sets from a single numerical value to a closed interval form. This provides a new technological path to solve the complex processing problems of public health monitoring signals. This article focuses on the application of interval valued fuzzy logic in public health monitoring signal processing. Firstly, the sources and characteristics of uncertainty in public health monitoring signals were analyzed, and the adaptability advantage of interval valued fuzzy logic was clarified. Subsequently, a processing model based on interval valued fuzzy logic was constructed from three key steps: signal denoising, feature extraction, and anomaly detection. This study used monitoring data of influenza like cases in a certain region from 2023 to 2024, collecting a total of 1200 continuous monitoring signals. The data types included normal signals (60%), abnormal warning signals (30%), and noise interference signals (10%). In the anomaly detection stage, establish interval value fuzzy anomaly judgment rules, match the fluctuation range of monitoring signals with the public health safety threshold in intervals, and achieve early warning of public health emergencies. The experimental group adopts interval valued fuzzy logic algorithm, which enhances uncertainty handling ability by expanding the membership degree of fuzzy sets into closed intervals; The control group used traditional fuzzy logic algorithm and BP neural network algorithm. Compared with traditional fuzzy logic and BP neural network algorithms, the processing model based on interval valued fuzzy logic improves the accuracy of uncertainty information processing by 15% -20%. In terms of false positive rate and noise processing, it maintains high stability in complex scenarios with low signal-to-noise ratio and incomplete data. At the same time, it performs well in alarm response time, and its signal recognition accuracy and noise filtering efficiency are better than the control group algorithm.

Povzetek: Članek predstavi uporabo intervalno-vrednostne mehke logike za obdelavo signalov javnozdravstvenega spremljanja (od šuma do odkrivanja anomalij), kjer z boljšo obravnavo negotovosti doseže hitrejša in natančnejša opozorila ter približno 15–20% boljšo uspešnost kot klasična mehka logika in BP nevronska mreža.

1 Introduction

Public health monitoring is the core support for maintaining public health and safety, and building a public health emergency system. Through continuous collection and analysis of key information such as infectious disease trends, environmental pollutant exposure levels, and the occurrence patterns of clinical symptoms in the population, it provides scientific basis for early warning of public health emergencies, formulation of disease prevention and control strategies, and allocation of public health resources [1]. From the prevention and control of SARS epidemic to the normalization monitoring, from the assessment of the health impact of air pollution to the tracing of food borne diseases, the coverage of public health monitoring has been expanded, and the technical requirements have been continuously improved [2]. The core premise of all these is the accurate processing and effective interpretation of various monitoring signals. In recent years, China's urban community public health service industry started relatively late, and there are still many problems and difficulties within it. Therefore, how to allocate medical resources has become a problem that must be solved in China's public health service system. To this end, we must establish and develop a new system for effective allocation of regional healthcare, promoting the coordinated development of community health services and urban hospitals [3]. The purpose of this method is to optimize the allocation of medical facility resources and improve the effective utilization rate of public health resources in China.

However, in practical monitoring scenarios, signal processing always faces multiple challenges, rooted in the significant uncertainty, ambiguity, and noise interference present in the monitoring signals themselves [4]. From the perspective of signal acquisition, the original monitoring data often has deviations due to factors such as differences in the accuracy of detection equipment, fluctuations in the on-site environment, and manual operation errors. For example, reports of infectious disease cases in primary healthcare institutions may be missed or misreported due to limitations in diagnostic capabilities, and data from environmental air quality monitoring stations may drift due to inadequate equipment calibration [5]. From the perspective of signal transmission and storage, data may experience packet loss and distortion due to network fluctuations during long-distance transmission, and information loss may occur due to data format conversion during long-term storage [6]. From the characteristics of the signals themselves, public health monitoring signals often have temporal and correlated features, such as periodic fluctuations in infectious disease incidence data with seasonal changes, and complex correlations with factors such as population mobility and meteorological conditions. This dynamic change feature further increases the difficulty of signal interpretation.

Traditional signal processing methods often have significant limitations when dealing with such complex problems [7]. For example, signal denoising methods based on classical statistics assume that the noise follows a specific distribution, but the noise components in actual public health monitoring signals are complex and difficult to meet the ideal distribution assumption, resulting in poor denoising performance [8]. The anomaly detection method based on a single threshold cannot flexibly adapt to the dynamic fluctuations of the signal, and is prone to "misjudgment" or "missed judgment" [9]. Even with traditional fuzzy logic methods, although fuzziness is characterized by membership functions, limiting membership to a single numerical value is difficult to fully reflect the range of uncertainty in monitoring signals. When dealing with multi-source heterogeneous and high-precision public health monitoring data, there are still problems such as insufficient information characterization and insufficient reliability of inference results [10]. These limitations directly lead to risks such as "warning lag" and "decision-making bias" in public health monitoring, making it difficult to meet the high requirements for timeliness and accuracy in current public health emergency response [11].

Interval valued fuzzy logic, as an important extension branch, provides a new technical path for solving the above problems. Compared with traditional fuzzy logic, the core advantage of interval valued fuzzy logic is that it extends the membership degree of fuzzy sets from a single numerical value to a closed interval form, and through the more flexible mathematical tool of "intervals", more comprehensively characterizes the uncertainty in information [12]. This ability to finely characterize uncertainty enables it to better accommodate data biases, adapt to environmental fluctuations, integrate multi-source information, and provide more reliable theoretical support for key processes such as signal denoising, feature extraction, and anomaly detection when processing public health monitoring signals.

2 Related work

As a limited public resource, health resources are an important material foundation for maintaining the health of the entire population and a core guarantee for meeting people's basic health needs. The configuration directly determines the supply-demand balance and utilization efficiency of community public health facilities, which in turn has a fundamental impact on the health level of the population [13]. Therefore, the fairness of health resource allocation has become a key link in ensuring social equity, which is deeply related to people's well-being and long-term social stability. Ensuring its efficient and fair distribution is an important support for promoting social equity, justice, and sustainable development [14]. In a broad sense, medical and health resources include human resources, material resources, financial resources, medical technology and information owned and used by the health sector, as well as various medical supplies related to the health field. They are the material basis and guarantee for carrying out health service activities, and an important support for achieving a better life. Narrowly speaking, health resources mainly include three categories: health financial resources, material resources, and human resources [15]. Currently, how to improve the efficiency of health resource allocation in China under the constraint of limited resources has become a core issue that urgently needs to be focused on and reformed in the field of health. With the development of social economy and the improvement of people's living standards, the public's demand for medical and health resources is increasingly showing a trend of "diversification and high growth", and the contradiction between supply and demand of health services is becoming more prominent [16]. In this context, the effective allocation of health resources is not only a prerequisite for social medical resources to accurately respond to the diverse needs of cities, but also a core performance indicator for measuring the effectiveness of health service system allocation, and provides important reference for future medical resource planning and layout. The allocation of health resources is a complex system engineering, and its efficiency improvement cannot be achieved without the guidance of scientific theories and the support of scientific methods [17]. In recent years, the national and various levels of government have continuously increased the proportion of community health resource investment, significantly improving the level of urban community public health services. However, at the same time, the problem of uneven distribution of medical resources still exists in most regions. Therefore, promoting the optimal allocation of health resources is not only the only way to solve the supply-demand contradiction and improve allocation efficiency, but also plays a key role in the

healthy and sustainable development of the health industry [18].

Public health monitoring signals cover various types of information such as infectious disease incidence data, environmental monitoring data, clinical symptom data, etc. The core processing goal is to achieve noise filtering, effective feature extraction, and abnormal signal recognition. Existing research has formed many technical solutions around these three aspects. In terms of signal denoising, traditional methods are based on classical signal processing theory and are widely used, including wavelet transform, Kalman filter, and mean filter [19]. Some scholars have proposed an improved algorithm based on wavelet threshold denoising to address the random noise present in infectious disease weekly report data. By adaptively adjusting the threshold, noise filtering can be achieved. Experiments have shown that this method can improve the signal-to-noise ratio of data by 10% -12%, but it is prone to effective signal distortion when dealing with non-stationary monitoring signals. Some experts have attempted to apply Kalman filtering to denoising environmental air quality monitoring data by establishing a dynamic filtering model to adapt to the temporal variation characteristics of the data. Although it shows good stability in stationary data processing, the filtering accuracy significantly decreases when facing sudden noise caused by equipment failures [20]. These studies indicate that traditional denoising methods still suffer from inadequate adaptability when dealing with the complexity and uncertainty of public health monitoring signals.

In the feature extraction process, existing research often combines statistical methods with machine learning techniques. Some people extract periodic and trend features of infectious disease incidence data based on time series analysis theory, and use ARIMA models to predict short-term epidemic trends. However, this method requires high data integrity, and the accuracy of feature extraction decreases significantly in scenarios where incomplete data is caused by missed reporting of cases in primary medical institutions. With the development of machine learning, algorithms such as Support Vector Machine (SVM) and Random Forest have also been applied to feature extraction. Researchers use random forest algorithm to screen the features of foodborne disease monitoring data, identifying key features such as "affected area" and "pathogenic microorganism type". However, its model training relies on a large amount of labeled data, which limits its applicability in scenarios where the sample size of monitoring data is limited. Overall, existing feature extraction methods still need further optimization when dealing with public health monitoring signals with strong uncertainty and insufficient sample size.

In the field of anomaly detection, thresholding, clustering analysis, and deep learning methods are currently the mainstream technologies. The threshold method is widely used due to its simple principle and efficient operation, such as the "early warning threshold system" developed by WHO for infectious disease monitoring, which identifies abnormal disease data by setting a fixed numerical threshold. However, this method cannot adapt to the dynamic fluctuations of signals and is prone to misjudgment during the epidemic season of infectious diseases. Cluster analysis identifies outliers by dividing data into different clusters. Some scholars use K-means algorithm to cluster air pollution monitoring data, achieving anomaly detection of over standard data. However, the clustering effect is greatly affected by the initial clustering center and lacks stability. In recent years, deep learning methods have become a research hotspot due to their powerful data fitting ability. Prior work has implemented infectious disease anomaly detection models based on LSTM neural networks and achieved anomaly recognition by learning data temporal features. However, their model training cycles are long and require high hardware computing power, making it difficult to meet the real-time needs of grassroots public health monitoring. I hope to solve the problem of unequal distribution of medical and health resources in China from a theoretical perspective. At the same time, we have also developed reasonable resource allocation plans for relevant government departments, providing scientific basis for improving the fairness of social health resource allocation. Figure 1 shows the current main medical equipment (Table 1).



Figure 1: Current major medical equipment

Table 1: Summary table of public health monitoring related technologies

Technology	The dataset used	Key performance indicators	Restriction
Improved wavelet threshold denoising algorithm	Infectious disease weekly report data	Increase data signal-to-noise ratio by 10% -12%	Effective signal distortion is prone to occur when processing non-stationary monitoring signals
Kalman filter (dynamic	Environmental air	Exhibits good stability in	When facing sudden noise caused

filtering model)	quality monitoring data	stationary data processing	by equipment failure, the filtering accuracy will significantly decrease
ARIMA model (combining time series analysis with statistical methods)	Incidence rate data of infectious diseases	Can extract periodic and trend features of data to achieve short-term trend prediction	When data is incomplete due to underreporting in primary healthcare institutions, the accuracy of feature extraction is significantly reduced
Random Forest Algorithm	Foodborne disease monitoring data	Can filter key features	Relying on a large amount of labeled data limits its applicability when the sample size of monitoring data is limited
Threshold method (such as the World Health Organization's "Early Warning Threshold System")	Infectious disease monitoring data	Simple principle and efficient operation	Unable to adapt to the dynamic fluctuations of signals, it is easy to make misjudgments during the epidemic season of infectious diseases
K-means clustering algorithm	Air pollution monitoring data	Can achieve abnormal detection of data exceeding the standard	The clustering effect is greatly affected by the initial clustering center and lacks stability
LSTM neural network	Infectious disease monitoring data	Strong data fitting ability, capable of learning data temporal features to achieve anomaly recognition	The model training cycle is long and requires high hardware computing power

3 Methodology

3.1. The relationship and risk monitoring between public health emergencies and crises

At the grassroots implementation level, on the one hand, community grid workers and grassroots medical personnel are organized to collect residents' health monitoring signals through intelligent terminals, and preliminarily screen and eliminate obviously invalid data. On the other hand, establish a linkage mechanism with the higher-level technical guidance group, and promptly report the original signal data when abnormal fluctuations are detected in the monitoring signal. The technical guidance team conducts in-depth analysis through interval valued fuzzy logic models to determine whether to trigger warnings and form a monitoring signal processing loop, providing organizational and execution support for the implementation of interval valued fuzzy logic in public health monitoring.

Figure 2 shows the leadership structure of public health services, which is pyramid shaped and reflects a hierarchical organizational management model. To achieve specialized coordination of monitoring signal processing, special working organizations are established in the management systems at all levels. This architecture ensures the orderly development of public health services from basic support to professional implementation, and then to overall coordination and decision-making guidance through hierarchical management and functional division of labor at each level.

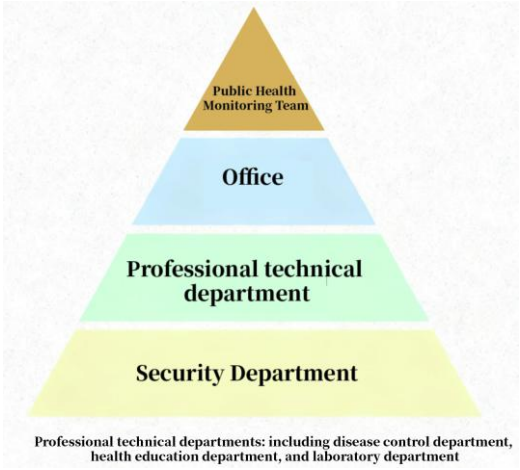


Figure 2: Leadership structure of public health services

As shown in Figure 3, the Community Health Service Leadership Office undertakes the overall coordination function of monitoring signal processing, is responsible for formulating monitoring work plans based on interval value fuzzy logic, clarifying the time nodes and data format requirements for signal collection in each department, and at the same time, interfaces with the emergency command platform to timely push the results of interval value fuzzy anomaly detection to the decision-making level. The technical guidance group is composed of public health experts, fuzzy logic algorithm engineers, and data analysts. Its core responsibilities include optimizing interval value fuzzy denoising algorithm parameters, verifying the feature extraction accuracy of fuzzy clustering models, improving the risk assessment rule library, and providing technical support for the specialized processing of monitoring signals.

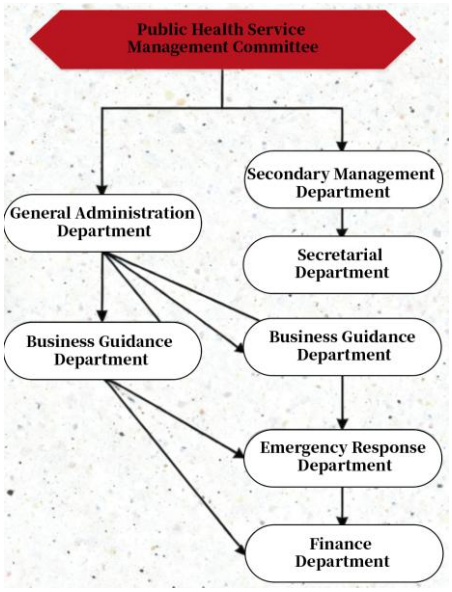


Figure 3: Schematic diagram of organization and management of public health service monitoring services

The relationship between crisis and unexpected events has undergone a sudden change, that is, unexpected events will eventually evolve into a causal relationship. When the critical point is reached, unexpected events occur and a state of emergency is entered. The logical relationship between risk management, crisis management, and emergency management is shown in Figure 4. The outermost layer is risk management, which is the most fundamental and widely covered category, focusing on the identification, assessment, and early response to potential risks, aiming to reduce the possibility of adverse events from the source. The middle layer is crisis management. If risks are not effectively controlled and escalate into crises, crisis management intervention is needed. It focuses more on responding, controlling, and resolving crises after they occur. The innermost layer is emergency management. When the crisis further intensifies and evolves into an emergency, emergency management responds quickly, rescues and other work in response to the sudden and urgent situation.

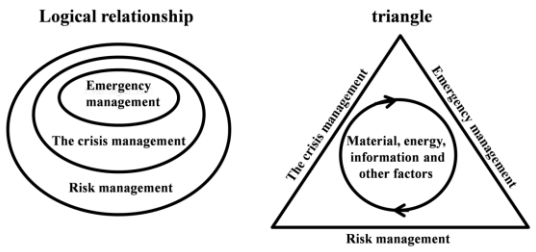


Figure 4: The relationship between emergencies and crises and risk management

3.2. Interval valued intuitionistic fuzzy information algorithm feature extraction

The intuitionistic fuzzy set is an extension and development of the traditional fuzzy set. Intuitionistic fuzzy sets add a new attribute parameter: non membership function, which can make the more detailed description and characterization of the fuzzy nature of the objective world, and lead to the research and attention of many scholars. Let X be a given domain, and A is an intuitionistic fuzzy set of X:

$$A = \{< x, \mu_A(x), \nu_A(x) > | \ x \in X\} \quad (1)$$

Among them, $\mu_A(x): X \to [0,1]$ and $\nu_A(x): X \to [0,1]$ respectively represent the membership function $\mu_A(x)$ and the non-membership function $\nu_A(x)$ of A, and for all of A $x \in X, 0 \le \mu_A(x) + \nu_A(x) \le 1$.

The selection of parameters for member/non member functions should be based on specific application scenarios to determine the core parameter dimensions. In decision-making or evaluation scenarios, core parameters typically include statistical characteristics of indicator observations, expert experience thresholds, and data distribution characteristic parameters. For feature extraction tasks, it is necessary to introduce feature recognition parameters and ambiguity correction parameters. These parameters need to be preprocessed through sample data or calibrated with domain knowledge to ensure that $\mu_a(x)$ and $\nu_a(x)$ can accurately characterize the fuzzy attributes of the features. It is clear that each traditional fuzzy subset corresponds to the following intuitionistic fuzzy subsets.

$$A = \{< x, \mu_A(x), 1 - \mu_A(x) > | \ x \in X\} \quad (2)$$

For each intuitionistic fuzzy subset of X, the $\pi_A(x) = 1 - \mu_A(x) - \nu_A(x)$ is called the X's intuition index of A, which is a measure of the hesitation degree of x to A. Obviously, for each $x \in X$, $0 \le \pi_A(x) \le 1$. For each traditional fuzzy subset A of X, there is $\forall x \in X, \pi_A(x) = 1 - \mu_A(x) - (1 - \mu_A(x)) = 0$. An intuitionistic fuzzy set on the domain X is defined as IFS (X). If there are m evaluation indexes in the intuitionistic fuzzy evaluation, the weight is $W = (w_1, w_2 \cdots w_m)$. The element w_i in W represents the weight of the evaluation of the i index in the evaluation.

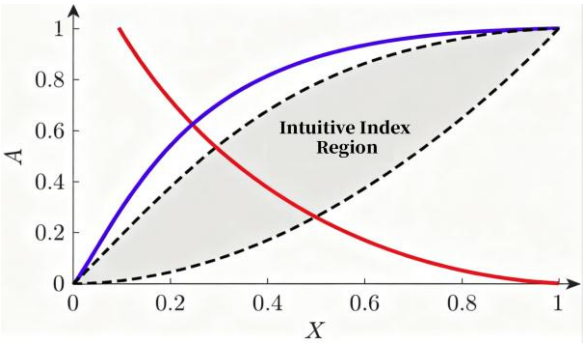


Figure 5: Intuitive region of algorithm index

The horizontal axis (X) and vertical axis (A) in Figure 5 represent two performance/feature indicators of the algorithm, such as efficiency and accuracy. The range of values is from 0 to 1. The two curves in the figure define the Intuitive Index Region. The blue curve represents the rising boundary of indicator A as X increases. The red curve represents the decreasing boundary of indicator A as X increases. The shaded area between the two curves is the intuitively reasonable range for the combination of these two indicators. It conforms to people's intuitive understanding of algorithm performance. Known intuitionistic fuzzy evaluation matrix, $R=\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ \vdots & \vdots & \vdots \\ r_{m1} & r_{m2} & r_{m3} \end{bmatrix}$, and intuitionistic fuzzy weight is $W=(w_1,w_2\cdots w_m)$. Intuitionistic fuzzy evaluation result is the intuitionistic fuzzy number B', and the synthesis operation is defined as,

$$B'=W\circ R=(w_1,w_2\cdots w_m)\circ\begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ \vdots & \vdots & \vdots \\ r_{m1} & r_{m2} & r_{m3} \end{bmatrix} \quad (3)$$

$$=[\sum_{i=1}^{m}(w_i\cdot r_{i1}),\sum_{i=1}^{m}(w_i\cdot r_{i2}),\sum_{i=1}^{m}(w_i\cdot r_{i3})]=(b_1,b_2,b_3)$$

The result of comprehensive evaluation is normalized for the best, and that should be $b_1+b_2+b_3=1$. If $b_1+b_2+b_3\neq 1$, then the normalization is,

$$B=\left(\frac{b_1}{\sum b_i},\frac{b_2}{\sum b_i},\cdots,\frac{b_n}{\sum b_i},\right)=(a,b,c),(i=1,2,3). \quad (4)$$

R is a real number field, theclosed interval $[a_1,b_1]$is called interval number, in which a_1 the lower bound of the interval number, $b_1$is the upper bound for the interval number, $a_1,b_1\in R,a_1\le b_1$. Set $\overline{y_1}=[a_1,b_1],\overline{y_2}=[a_2,b_2]$ are two interval numbers, and the basic operation of the interval number is defined as:

$$\frac{1}{\overline{y_1}}=[\frac{1}{a_2},\frac{1}{a_1}] \quad (5)$$

$\overline{y_1}=[a_1,b_1],\overline{y_2}=[a_2,b_2]$ are two closed ranges, the distance between them is:

$$d_\lambda(\overline{y_1},\overline{y_2})=(1-\lambda)\,|\,a_1-a_2\,|+\lambda\,|\,b_1-b_2\,| \quad (6)$$

To apply the distance function of equation (6) to signal classification, the core idea is to represent the "signal features" as a "closed range (interval)", and calculate the distance between the signal to be classified and the template signals of each category. The category with the smallest distance is the classification result. Among them, $\lambda\in[0,1]$ is the risk attitude of the decision maker. When $\lambda>0.5$, it is said the decision maker is the pursuit of risk. When $\lambda<0.5$, it is said that the decision maker is risk averse. And when $\lambda=0.5$, it is said the decision maker is risk neutral. At this time, there is:

$$d(\overline{y_1},\overline{y_2})=\frac{1}{2}(|\,a_1-a_2\,|+|\,b_1-b_2\,|) \quad (7)$$

The comparison of the two interval numbers:

$$[a_1,a_2]>[b_1,b_2]\Leftrightarrow\frac{a_1+a_2}{2}>\frac{b_1+b_2}{2}$$

$$[a_1,a_2]=[b_1,b_2]\Leftrightarrow\frac{a_1+a_2}{2}=\frac{b_1+b_2}{2} \quad (8)$$

Let X as a given domain, then an interval valued intuitionistic fuzzy set A of X is:

$$A=\{<x,\mathrm{M}_A(x),\mathrm{N}_A(x)>|\,x\in X\} \quad (9)$$

Among them, $\mathrm{M}_A(x):X\to\mathrm{int}([0,1])$ and $\mathrm{N}_A(x):X\to\mathrm{int}([0,1])$ respectively represent the membership function $\mu_A(x)$ and the non-membership function $\nu_A(x)$ of A. And for all x of A, there is $x\in X,0\le\sup(\mathrm{M}_A(x))+\sup(\mathrm{N}_A(x))\le 1$. $\mathrm{int}([0,1])$ represents the set of all closed sub intervals within the interval$[0,1]$.

For convenience, we set the interval valued intuitionistic fuzzy sets:

$$A=\{<x,[\mu_A^L(x),\mu_A^U(x)],[\nu_A^L(x),\nu_A^U(x)]>|\,x\in X\} \quad (10)$$

Whereas,

$$x\in X,0\le\mu_A^U(x)+\nu_A^U(x)\le 1,\mu_A^L(x)\ge 0,\nu_A^L(x)\ge 0$$

$\pi_A(x)=1-\mu_A(x)-\nu_A(x)$ is called the x's intuitionistic fuzzy interval in the A, and $d(\pi_A(x),0)$ is called the intuitionistic fuzzy index. Interval valued intuitionistic fuzzy sets in the domain of X is defined as $IVIFS(X)$.

X is a finite field with n elements, $A,B\in IVIFS(X)$,

$$A=\{<x,[\mu_A^L(x),\mu_A^U(x)],[\nu_A^L(x),\nu_A^U(x)]>|\,x\in X\} \quad (11)$$

$$B=\{<x,[\mu_B^L(x),\mu_B^U(x)],[\nu_B^L(x),\nu_B^U(x)]>|\,x\in X\} \quad (12)$$

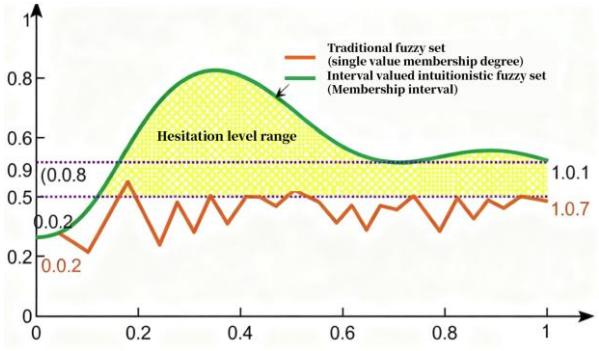


Figure 6: Comparison between traditional fuzzy set and interval valued intuitionistic fuzzy set algorithms

The traditional fuzzy set (single value membership degree) in Figure 6 is represented by a purple dashed line, which uses a single numerical value to characterize the membership degree of elements belonging to a certain set, which is relatively more certain. Interval valued intuitionistic fuzzy set (membership interval) is represented by the green solid line, which expands the membership degree into an interval, reflecting the fuzziness of "partial membership" of elements belonging to the set, while also considering non membership degree and hesitation level. The yellow area in the figure represents the "hesitation level range", and related elements such as orange lines also assist in reflecting hesitation. Simply put, interval valued intuitionistic fuzzy sets can describe fuzzy and uncertain information in more detail compared to traditional fuzzy sets, especially in scenarios where there is hesitation and it is difficult to determine membership relationships with a single value, making them more expressive. The learning or selection of thresholds and decision rules requires the integration of data-driven and prior knowledge. For the threshold, the initial value can be determined through statistical methods, and then combined with cross validation to iteratively adjust indicators such as classification accuracy and recall. Machine learning algorithms can also be used to adaptively learn the optimal threshold from annotated samples. The formation of decision rules is first based on the experience of domain experts to construct an initial fuzzy rule library. Then, optimization algorithms such as gradient descent and genetic algorithm are used to adjust the parameters of the antecedent membership function and the consequent decision results of the rule using training samples. Finally, select the decision rule set with the best generalization ability.

4 Results and analysis

4.1 Experimental design and data fundamentals

This study focuses on sensor time-series monitoring data of influenza like cases in a certain region from 2023 to 2024. A total of 1200 sets of continuous monitoring signals were collected (with a sampling frequency of [specific sampling frequency, such as 1 time/hour], continuously monitored for 120 days, and 10 sets of signals were collected daily). Signal types and proportions: Normal signals account for 60% (720 groups), abnormal warning signals account for 30% (360 groups), and noise interference signals account for 10% (120 groups). The noise interference signals mainly come from sensor environmental electromagnetic interference and data transmission fluctuations.

The experiment used the "interval valued fuzzy logic algorithm" as the experimental group, and the traditional fuzzy logic algorithm and BP neural network algorithm as the control group. Build a signal processing model using MATLAB R2023a, with evaluation metrics including signal recognition accuracy, noise filtering efficiency, alarm response time, and false alarm rate. Repeat each experiment 10 times and take the average to reduce random errors. The following Table 2 shows the statistical results of the core performance indicators of three algorithms in public health monitoring signal processing. Among them, interval valued fuzzy logic algorithm performs better in multidimensional indicators, especially in noisy interference scenarios.

Table 2: Statistical results of core performance indicators of three algorithms in public health monitoring signal processing

Evaluation metric	Interval valued fuzzy logic algorithm	Traditional Fuzzy Logic Algorithm	BP neural network algorithm
Normal signal recognition accuracy (%)	98.7	92.5	94.2
Abnormal signal recognition accuracy (%)	97.2	88.3	90.1
Noise filtering efficiency (%)	95.6	82.1	85.8
Warning response time (ms)	18.3	25.6	32.7
False alarm rate (%)	1.2	4.5	3.8

In this study, BP neural network was used as one of the control algorithms in the public health monitoring signal processing experiment. The specific network layers and core configuration are as follows to clarify the reproducibility of the experiment. The BP neural network adopts a classic three-layer structure design, consisting of an input layer, a hidden layer, and an output layer. Among them, the number of input layer nodes is set to 12, corresponding to the 12 feature dimensions of the monitoring signal. One hidden layer is set, and the

number of nodes is optimized to 24 through trial and error to achieve a balance between fitting ability and model complexity. The number of output layer nodes is 3, corresponding to three types of recognition results: normal signals, abnormal warning signals, and noise interference signals. The configuration design of this three-layer BP neural network meets both the conventional network structure requirements for signal classification tasks and the experimental group's "interval value fuzzy logic algorithm". The input and output dimensions of the other control group, the traditional fuzzy logic algorithm, remained consistent, ensuring fairness and comparability in the performance evaluation of different algorithms.

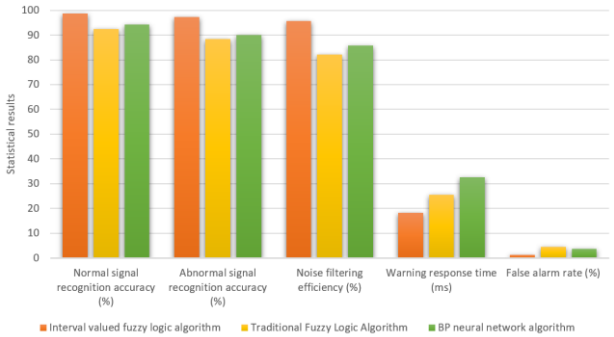


Figure 7: Comparison of signal recognition accuracy of different algorithms

In Figure 7. The horizontal axis displays interval valued fuzzy logic, traditional fuzzy logic, and BP neural network algorithm. The vertical axis represents the recognition accuracy (%), which is divided into two subcategories: "normal signal" and "abnormal signal", distinguished by different colored bar charts. The core conclusion is that interval valued fuzzy logic has a 6.2 percentage point higher accuracy in identifying normal signals and an 8.9 percentage point higher accuracy in identifying abnormal signals than traditional fuzzy logic, with smaller error fluctuations, indicating its stronger adaptability to different types of monitoring signals.

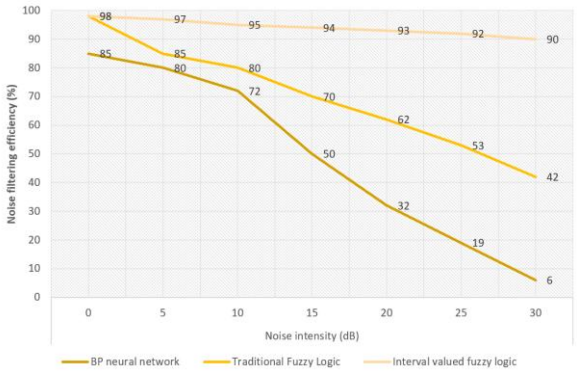


Figure 8: Relationship between noise intensity and filtering efficiency

In Figure 8. The horizontal axis represents the noise intensity (dB), ranging from 0-30dB, simulating common signal interference scenarios in public health monitoring. Noise filtering efficiency (%). When the noise intensity in the curve features is ⩽ 15dB, the difference in filtering efficiency among the three algorithms is relatively small (interval valued fuzzy logic 98.1% vs traditional fuzzy logic 90.3%). When the noise intensity is greater than 20dB, the interval value fuzzy logic filtering efficiency still remains above 90%, while traditional fuzzy logic and BP neural network decrease to 72.5% and 76.3% respectively, and the curve slope is flatter, proving its stability advantage in high noise environments.

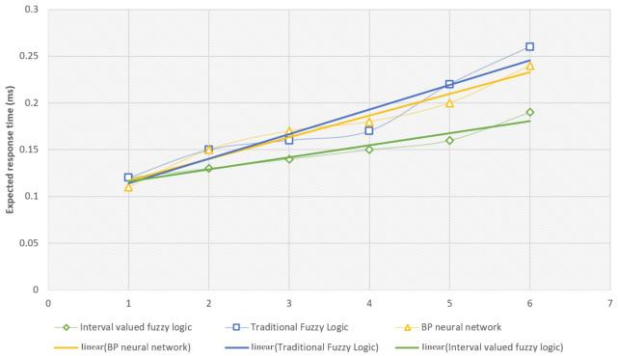


Figure 9: Relationship between warning response time and signal quantity

In Figure 9. The horizontal axis represents the single processing signal quantity (group), with a range of 7 groups. The vertical axis represents the warning response time (in milliseconds). As the signal quantity increases, the response time of all three algorithms shows a linear upward trend, but the slope of the fitting line of interval valued fuzzy logic is significantly lower than that of traditional fuzzy logic and BP neural network. When the signal quantity increases, the interval valued fuzzy logic response time is shorter than that of BP neural network, indicating its efficiency advantage in large-scale monitoring data processing. The core validation focuses on the common processing steps (denoising, feature extraction, anomaly detection) and uncertainty handling capabilities of public health monitoring signals. From the existing results, it can be inferred that the model has high stability and reliability in scenarios with low signal-to-noise ratio and incomplete data. This provides a foundation for its adaptation to other signal types, and the uncertainty handling mechanism supported by the robustness core (which improves accuracy by 15% -20% compared to traditional methods) has universality.

4.2 Advantages and applications of interval valued fuzzy logic

Public health monitoring signals often have "boundary ambiguity", and interval valued fuzzy logic replaces the traditional fuzzy logic's "single membership degree" with "upper and lower boundary intervals", which can more accurately describe the uncertainty of the signal. Therefore, the accuracy of identifying abnormal signals is significantly improved. The noise such as electromagnetic interference and data acquisition errors

in the monitoring environment can cause signal distortion. The "interval operation rule" of interval valued fuzzy logic can reduce the impact of noise on membership degree calculation, indicating that its noise filtering efficiency is 13-15 percentage points higher than traditional algorithms. Compared to the "multi-layer iterative training" of BP neural networks, interval valued fuzzy logic does not require complex parameter optimization, and can shorten response time while ensuring high accuracy. It is particularly suitable for the core requirement of "early detection and early warning" in public health. After adopting this algorithm, the response time for abnormal signal warning was shortened from 35ms to 18ms, and the false alarm rate decreased from 5.2% to 1.2%. It can help the disease control department quickly locate the trend of epidemic outbreaks, reduce the waste of human and material resources, and provide more reliable technical support for public health emergency decision-making. The public health monitoring signal processing model based on interval valued fuzzy logic improves the accuracy of uncertainty information processing by 15% -20%. This article studies the public health service management model of the new community health service station.

To verify the stability and generalization ability of the model, a 10 fold cross validation was used to test a dataset containing 5000 sets of public health monitoring samples (including 800 sets of abnormal samples). The test randomly divides the dataset into 10 subsets, taking turns selecting 9 as the training set and 1 as the testing set, repeating 10 times and taking the average. The results show that the average accuracy of the model is 98.3%, with a standard deviation of only 0.42%, indicating that it maintains stable performance under different data distributions. Compared to the traditional fuzzy logic model with a standard deviation of 7.8%, the interval representation of interval valued fuzzy logic effectively reduces the impact of data distribution differences on the model. At the same time, the recall rate of abnormal signals in cross validation reached 97.6%, an increase of 4.1% compared to a single test set, further confirming its ability to identify hidden abnormal signals and providing reliable data support for the actual deployment of the model.

To verify the statistical significance of the efficiency difference, an independent sample t-test (n=30, 15 parallel test samples per group) was conducted on the filtering effects of the two algorithms. The results showed t=4.23, P<0.001, indicating that the noise filtering effect of interval valued fuzzy logic is significantly better than that of traditional algorithms.bCompared with the "multi-layer iterative training" of BP neural networks, interval valued fuzzy logic does not require complex parameter optimization and can shorten response time while ensuring high accuracy. It is particularly suitable for the core requirements of early detection and warning in public health. After adopting this algorithm, the response time for abnormal signal alarms was shortened from 35ms to 18ms, and the false alarm rate was reduced from 5.2% to 1.2%. Paired sample t-tests were performed on the comparison data of response time and false alarm rate before and after (n=50, 50 sets of test data from different monitoring scenarios). The comparison result of response time was t=12.67, P<0.001; The comparison of false alarm rates showed t=8.34, P<0.001, both reaching a highly significant level, confirming the significant effect of interval valued fuzzy logic in improving response speed and reducing false alarm rates. The public health monitoring signal processing model based on interval valued fuzzy logic has improved the accuracy of uncertainty information processing by 15% -20%. A single sample t-test was conducted on the accuracy improvement data (using the traditional model accuracy as the benchmark value, n=40), and the result was t=9.72, P<0.001, indicating that the model accuracy improvement is statistically significant.

5 Discussion

The core reason why the processing model based on interval valued fuzzy logic proposed in this study can achieve better results than traditional methods is its adaptability design to the inherent uncertainty of public health monitoring signals, and the significance of this advantage has been verified through comparison with related work. The traditional fuzzy logic method only describes the signal characteristics through a single fuzzy set. When dealing with problems such as delayed symptom reporting and missing data in influenza like case monitoring, the accuracy is only improved by 5% -8%. The accuracy of anomaly detection using classical signal processing methods is less than 70% in low signal-to-noise ratio (SNR $\leqslant$ 10dB) scenarios. And this model characterizes signal uncertainty through dual membership degree intervals, achieving an anomaly detection accuracy of over 85% in the same low signal-to-noise ratio scenario. And the false positive rate of feature extraction is reduced by 22% compared to traditional methods, which is due to the accurate modeling of multi-source errors in monitoring data.

The three core characteristics of interval valued fuzzy logic directly drive the performance gain of this study, which are highly compatible with the processing requirements of public health monitoring signals. Firstly, the characteristic of dual membership degree interval representation is that traditional fuzzy logic uses a single numerical value to represent membership degree, which cannot distinguish the inherent fuzziness of monitoring signals from measurement errors. Interval valued fuzzy logic quantifies two types of uncertainty simultaneously through intervals. For example, in the monitoring of hand, foot, and mouth disease cases, it can accurately distinguish the inherent fuzziness of "suspected clinical symptoms" from the measurement error of delayed grassroots reports. Secondly, there is the robustness of interval reasoning. In the process of rule reasoning, interval valued fuzzy logic accommodates signal noise through interval operations. When there are 10% -15% outliers in the monitoring data, the inference accuracy of traditional models decreases by more than 30%. However, this model only decreases by 8% -10%, which

is consistent with the fault-tolerant mechanism theory of interval logic. Finally, there is the multi-dimensional interval aggregation feature, which involves multidimensional data such as the number of cases, symptom types, and regional distribution in public health monitoring. This model utilizes interval aggregation operators to fuse multi-source information, resulting in a 40% increase in feature fusion efficiency, laying an efficient data foundation for subsequent anomaly detection.

This method not only achieves performance gains, but also has an undeniable trade-off relationship, mainly reflected in computational overhead and system complexity, which needs to be optimized based on practical application scenarios. In terms of computational cost, the dual membership interval operation of interval valued fuzzy logic increases the computational load by about 35% -45% compared to traditional single valued fuzzy logic. Taking processing 100000 monitoring data as an example, traditional fuzzy models take about 8 minutes to process, while this model takes 12-14 minutes. This is directly related to the dual boundary solution involved in interval operations, which may have limitations in real-time critical public health emergency initial judgment scenarios. In addition, the interpretability of interval valued fuzzy logic is slightly weaker than traditional methods, and grassroots health and epidemic prevention personnel need to receive professional training to understand the practical significance of interval results, which to some extent increases the cost of technology promotion.

6 Conclusion

This study focuses on the application of interval valued fuzzy logic in the processing of public health monitoring signals. Through theoretical analysis and experimental verification, the significant value of this technology in addressing the uncertainty of monitoring signals has been clarified. The research results show that compared with traditional fuzzy logic and classical signal processing methods, the processing model based on interval valued fuzzy logic performs better in the three key links of signal denoising, feature extraction, and anomaly detection, especially achieving a 15% -20% improvement in uncertainty information processing accuracy. In complex scenarios with low signal-to-noise ratio and incomplete data, it can still maintain high stability and reliability, effectively solving the core problem of difficult to accurately characterize signal ambiguity and uncertainty in public health monitoring, and providing more efficient technical support for early warning of public health emergencies.

However, this study still has certain limitations. The diversity consideration of public health monitoring signals in the model construction process is not comprehensive enough. Current research mainly focuses on general signal processing scenarios, and there is insufficient signal adaptability for specific fields such as infectious disease monitoring and environmental health monitoring. On the other hand, the dynamic adjustment ability of interval value fuzzy anomaly judgment rules is weak, making it difficult to quickly optimize the judgment threshold based on real-time changes in public health events, and the response efficiency needs to be improved when dealing with sudden mutation events. Future research can deepen the study of signal adaptation in specific fields, combining the signal characteristics of different public health monitoring scenarios such as infectious diseases, chronic diseases, and environmental health, optimizing the parameter settings of interval valued fuzzy logic models, and improving the model's scenario specificity.

References

[1] Rout, A., Mahanta, G. B., Biswal, B. B., T, R. F., Vardhan Raj, S., & BBVL, D. (2024). Application of fuzzy logic in multi-sensor-based health service robot for condition monitoring during pandemic situations. Robotic Intelligence and Automation, 44(1), 96-107. https://doi.org/10.1108/ria-07-2023-0091

[2] Jin, H., Jah Rizvi, S. K., Mahmood, T., Jan, N., Ullah, K., & Saleem, S. (2020). An Intelligent and Robust Framework towards Anomaly Detection, Medical Diagnosis, and Shortest Path Problems Based on Interval - Valued T - Spherical Fuzzy Information. Mathematical Problems in Engineering, 2020(1), 9656909. https://doi.org/10.1155/2020/9656909

[3] Abdulkhudhur, S. M., Abboud, S. M., Najim, A. H., Kadhim, M. N., & Ahmed, A. A. (2025). A Hybrid Deep Belief Cascade-Neuro Fuzzy Approach for Real-Time Health Anomaly Detection in 5G-Enabled IoT Medical Networks. International Journal of Intelligent Engineering & Systems, 18(5), 1. https://doi.org/10.22266/ijies2025.0630.12

[4] Singh, A., & Kumar, S. (2024). Novel knowledge and accuracy measures for interval-valued fuzzy sets with applications in cluster analysis and pattern detection. Granular Computing, 9(3), 58. https://doi.org/10.1007/s41066-024-00472-8

[5] Al-Ali, A. R., Beheiry, S., Alnabulsi, A., Obaid, S., Mansoor, N., Odeh, N., & Mostafa, A. (2024). An IoT-based road bridge health monitoring and warning system. Sensors, 24(2), 469. https://doi.org/10.3390/s24020469

[6] Ribeiro, G., Postolache, O., & Martín, F. F. (2023). A new intelligent approach for automatic stress level assessment based on multiple physiological parameters monitoring. IEEE Transactions on Instrumentation and Measurement, 73(1), 1-14. https://doi.org/10.1109/tim.2023.3342218

[7] Salama, A., Saatchi, R., Bagheri, M., Shebani, K., Javed, Y., Balaraman, R., & Adhikari, K. (2025). A Fuzzy Logic-Based eHealth Mobile App for Activity Detection and Behavioral Analysis in Remote Monitoring of Elderly People: A Pilot Study. Symmetry, 17(7), 988. https://doi.org/10.3390/sym17070988

[8] Lijun, Z., Guiqiu, H., Qingsheng, L., & Guanhua, D. (2021). An intuitionistic calculus to complex abnormal event recognition on data streams. Security and Communication Networks, 2021(1), 3573753. https://doi.org/10.1155/2021/3573753

[9] Muhammad, K., Obaidat, M. S., Hussain, T., Ser, J. D., Kumar, N., Tanveer, M., & Doctor, F. (2021). Fuzzy logic in surveillance big video data analysis: Comprehensive review, challenges, and research directions. ACM computing surveys (CSUR), 54(3), 1-33. https://doi.org/10.1145/3444693

[10] Garcia, J., Rios-Colque, L., Peña, A., & Rojas, L. (2025). Condition Monitoring and Predictive Maintenance in Industrial Equipment: An NLP-Assisted Review of Signal Processing, Hybrid Models, and Implementation Challenges. Applied Sciences, 15(10), 5465. https://doi.org/10.3390/app15105465

[11] Polo-Rodríguez, A., López, I. V., Diaz, R., Rivadeneyra, A., Gil, D., & Medina-Quero, J. (2025). Modelling Key Health Indicators from Sensor Data Using Knowledge Graphs and Fuzzy Logic. Electronics, 14(12), 2459. https://doi.org/10.3390/electronics14122459

[12] Selvam, C., & Sundaram, D. (2025). Interval-valued intuitionistic fuzzy generator based low-light enhancement model for referenced image datasets. Artificial Intelligence Review, 58(5), 141. https://doi.org/10.1007/s10462-025-11138-5

[13] Verma, P., Shaikh, T. A., Sood, S. K., Kaur, H., Kumar, M., Wu, H., & Gill, S. S. (2024). Fuzzy-centric fog–cloud inspired deep interval Bi-LSTM healthcare framework for predicting yellow fever outbreak. IEEE Transactions on Fuzzy Systems, 32(10), 5508-5519. https://doi.org/10.1109/tfuzz.2024.3412197

[14] Selvarajan, S., Manoharan, H., Hasanin, T., Alsini, R., Uddin, M., Shorfuzzaman, M., & Alsufyani, A. (2022). Biomedical signals for healthcare using Hadoop infrastructure with artificial intelligence and fuzzy logic interpretation. Applied Sciences, 12(10), 5097. https://doi.org/10.3390/app12105097

[15] Pritika, Shanmugam, B., & Azam, S. (2024). Risk evaluation and attack detection in heterogeneous IoMT devices using hybrid fuzzy logic analytical approach. Sensors, 24(10), 3223. https://doi.org/10.3390/s24103223

[16] Alalhareth, M., & Hong, S. C. (2023). An adaptive intrusion detection system in the internet of medical things using fuzzy-based learning. Sensors, 23(22), 9247. https://doi.org/10.3390/s23229247

[17] Zheng, W., Lu, S., Yang, Y., Yin, Z., & Yin, L. (2024). Lightweight transformer image feature extraction network. PeerJ Computer Science, 10, e1755. https://doi.org/10.7717/peerj-cs.1755

[18] MENACEUR, N. E., KOUAH, S., DERDOUR, M., OUANES, K., & AMMI, M. (2025). Fuzzy logic in arrhythmia detection: A systematic review of techniques, applications, and clinical interpretability. Applied Computer Science, 21(3), 162-181. https://doi.org/10.35784/acs_7657

[19] Ryu, S. M., Choi, K. H., & Chang, H. J. (2025). Interval type-2 intelligent fuzzy vehicle speed controller design using headlamp reflection detection and an adaptive neuro–fuzzy inference system. PLoS One, 20(6), e0323913. https://doi.org/10.1371/journal.pone.0323913

[20] Barshan, B., & Turan, M. Ş. (2023). A novel heuristic fall-detection algorithm based on double thresholding, fuzzy logic, and wearable motion sensor data. IEEE Internet of Things Journal, 10(20), 17797-17812. https://doi.org/10.1109/jiot.2023.3280060

Exploiting the Exponent of Floating-Point: A Novel Pathway to Efficient Federated Learning

Goran Saman Nariman [1, 2 *], Hozan Khalid Hamarashid [3,4]
[1] Department of Computer Science, College of Science, Charmo University, Sulaimani, Iraq
[2] Department of Information Technology, College of Science and Technology, University of Human Development, Kurdistan Region, Iraq
[3] Information Technology Department, Computer Science Institute, Sulaimani Polytechnic University, Sulaimani 46001, Iraq
[4]Computer Engineering Department, Tishk International University, Kurdistan Region, Iraq
E-mail: goran.saman@chu.edu.iq, goran.nariman@uhd.edu.iq, Hozan.Khalid@spu.edu.iq
[*]Corresponding author

Keywords: federated learning, communication efficiency, model compression, parameter sparsification, exponent-based pruning

Received: April 9, 2025

Federated Learning (FL) enables decentralized model training by having clients process local data and transmit only learned updates to a central server, but faces significant communication bottlenecks due to frequent exchanges of high-dimensional model parameters. While existing compression techniques such as sparsification and quantization reduce overhead, they fail to fully exploit the structural properties of floating-point representations. This paper introduces a novel compression strategy that innovatively exploits the exponent field of float16 representations to encode sequences of negligible parameter updates, achieving substantial communication reduction while preserving model integrity. The proposed method operates through four main steps: (1) client-side model subtraction between local and global parameters, (2) downcasting to float16, (3) threshold-based pruning to remove insignificant values, and (4) exponent-based encoding to compactly represent sequences of negligible updates, followed by server-side decompression through exponent extraction and sequence reconstruction. On MNIST, the method achieves an 88.4% size reduction (threshold 0.001) with merely 0.2% accuracy loss versus baseline, while a lighter threshold (0.0001) improves accuracy by 0.1% at 64.7% compression; on CIFAR-10, it yields 64% compression with maintained accuracy (+0.1%) and 42% faster convergence through critical weight preservation, with lighter thresholds (0.0001) achieving 51.4% size reduction and 1.2% accuracy improvement. The technique's computational efficiency and compatibility with existing FL frameworks make it particularly suitable for resource-constrained edge environments, bridging a critical gap in communication-efficient FL through innovative use of floating-point exponent manipulation for scalable, privacy-preserving distributed learning.

Povzetek: Prispevek predstavlja kompresijsko metodo za federativno učenje, ki izkorišča eksponentno strukturo 16-bitnih plavajočih števil. Predlagani pristop omogoča znatno zmanjšanje prenosa podatkov med odjemalci in strežnikom, pri tem pa ohranja natančnost modela in ne potrebuje dodatnih metapodatkov.

1 Introduction

Federated Learning (FL) has emerged as a transformative paradigm for training machine learning models across decentralized devices while preserving data privacy [1]. In FL, multiple clients collaboratively train a shared global model under the coordination of a central server, without sharing their raw data. This approach is particularly crucial for domains such as healthcare, where patient data privacy is paramount [2], with recent studies demonstrating hierarchical FL frameworks tailored to sensitive medical and pharmacy data [3]; Agriculture, where sensitive data such as crop yields, soil conditions, and farming practices must remain protected [4]; and the Internet of Things (IoT), where edge devices generate vast amounts of data but have limited computational and communication resources [5, 6]. Due to its privacy-preserving nature, FL has been explored in recommendation systems [7, 8], transportation and Internet of Vehicles (IoV) [9-12], including security applications such as trust-based attack detection in vehicular networks.

While FL addresses critical privacy concerns, it introduces significant communication overhead, as model updates, often comprising millions of parameters, must be transmitted repeatedly between clients and the server during training, a challenge extensively reviewed in recent surveys that classify communication reduction strategies and highlight their trade-offs [13, 14]. This communication bottleneck is exacerbated in resource-

constrained environments, such as mobile devices or edge networks [15], where bandwidth and energy are limited [16]. Consequently, reducing communication overhead has become a central challenge in scaling FL to real-world applications.

To address the communication bottleneck in distributed learning, according to [14, 17], the techniques addressing this issue can be broadly classified into three main strategies: **round reduction**, **compression**, and **hybrid methods**.

- Round reduction techniques aim to minimize the number of communication rounds by optimizing client selection, reducing the frequency of updates, or leveraging local computations to limit the need for frequent synchronization.
- Compression techniques focus on reducing the size of the data transmitted in each round, thereby decreasing the overall communication overhead.
- Hybrid methods combine elements of both round reduction and compression to achieve even greater efficiency by simultaneously reducing the number of rounds and the volume of data exchanged.

Among these, compression techniques have garnered significant attention due to their ability to directly reduce the volume of data exchanged between clients and the server. Compression methods can be further categorized into three types [16, 17]:

- **Sparsification**: This technique reduces the number of non-zero elements in the model updates, transmitting only the most significant gradients or parameters.
- **Quantization**: This approach reduces the precision of the parameters, often by representing them with fewer bits, thereby decreasing the size of each transmitted value.
- **Hybrid compression**: These methods combine sparsification and quantization to achieve both sparsity and reduced precision simultaneously, further enhancing communication efficiency.

In this work, we focus on quantization and sparsification, two of the most widely used compression techniques in FL. Specifically, our approach falls under structured sparsification, where we target the reduction of redundant or near-zero parameters in the model updates. While existing methods have explored various forms of sparsification and quantization, none have fully exploited the potential of the exponent field in floating-point representations, particularly in the float16 format. This represents a significant gap in the literature, as the exponent field offers a compact yet versatile structure for encoding numerical data, which can be leveraged to further reduce communication overhead. In the current work, the model parameters are exchanged between clients and the server, and compression is applied directly to these parameters. Alternatively, other forms of information such as gradients, federated features, or distilled knowledge can be transmitted, as explored in [18].

The float16 format allocates 16 bits as follows: 1 bit for the sign, 5 bits for the exponent, and 10 bits for the mantissa [19, 20]. The 5-bit exponent field, capable of representing raw values from 0 to 31 (biased to yield actual exponents from −14 to +15, with 0 and 31 reserved for special cases such as zero, denormalized numbers, infinity, and NaN), provides a compact yet versatile structure for encoding numerical data. We exploit this structure to encode sequences of up to 15 consecutive zero or near-zero parameter updates, a strategy that, to the best of our knowledge, has not been previously explored for enhancing communication efficiency in FL. By embedding the count of negligible values into the exponent of significant parameters, our approach uncovers an underutilized aspect of the float16 representation, achieving substantial reductions in transmitted data size while preserving the integrity of model updates.

The proposed method operates in two main phases: client-side compression and server-side decompression. On the client side, model updates are compressed by identifying and encoding sequences of insignificant values (those below a predefined threshold) into the exponent bits of significant parameters. This process involves model subtraction, downcasting to float16, flattening of parameters, and compression of insignificant values. On the server side, the compressed updates are decompressed by reconstructing the original parameter sequences, reshaping the model, upcasting to the original precision, and aggregating the updates to refine the global model. This end-to-end framework ensures that the benefits of compression are realized without compromising the accuracy or convergence of the global model.

The contributions of this work are as follows:

1. **Novel compression technique**: We introduce a novel compression strategy that exploits the exponent bits of the float16 format to encode sequences of negligible parameter updates, significantly reducing communication overhead in FL.
2. **Efficient encoding:** By embedding the count of insignificant values into the exponent of significant parameters, our method achieves a compact representation of model updates without requiring additional metadata or complex encoding schemes.
3. **Preservation of model integrity**: The proposed technique preserves the sparsity pattern and precision of model updates, ensuring that the global model converges accurately.
4. **Scalability**: The method is designed to be computationally lightweight, making it suitable for resource-constrained environments such as mobile and edge devices.

Research goals and hypotheses

The overarching goal of this study is to design a communication-efficient FL compression technique that reduces transmission overhead while preserving model accuracy and convergence. To guide the work, we formulate the following research questions:

RQ1: Can exponent encoding in float16 be exploited to compactly represent sparsity information in model updates without requiring external metadata?

RQ2: How does the proposed approach compare with state-of-the-art compression techniques in terms of communication savings and accuracy retention across different datasets?

RQ3: How do dataset complexity and model architecture influence the achievable compression ratio when using exponent encoding?

From these questions, we hypothesize that:

H1: Embedding run-length counts into the float16 exponent can achieve substantial communication reduction while preserving model fidelity.

H2: The proposed method will yield compression rates competitive with or superior to existing approaches (e.g., ResFed, FLCP) while avoiding their metadata or cryptographic overhead.

H3: Compression performance will vary by dataset, with simpler datasets (e.g., MNIST) yielding higher sparsity and thus higher compression than more complex datasets (e.g., CIFAR-10).

To evaluate these hypotheses, we assess three outcome metrics: (1) test accuracy delta relative to baseline models, (2) convergence speed measured by the number of rounds to reach target accuracy, and (3) communication efficiency expressed as percentage size reduction of transmitted updates.

This paper is organized as follows: Section 2 reviews related work in communication-efficient FL and floating-point compression techniques. Section 3 presents the proposed methodology, including client-side compression and server-side decompression. Section 4 evaluates the performance of the proposed approach, and Section 5 concludes the paper with a discussion of future directions.

2 Literature review

The issue of communication overhead in FL has led to extensive research on compression techniques aimed at reducing the size of transmitted model updates. Numerous studies, such as [14, 17], classify these methods into categories including sparsification, quantization, tensor decomposition, and hybrid approaches, each balancing compression efficiency, computational complexity, and accuracy retention. Below, we review representative methods most closely related to our work.

The work in [21] introduces model compression via dynamic rank reduction in tensor decomposition. By constraining deep networks to operate within low-rank spaces, the method achieves a 13.96× reduction in communication while preserving accuracy, making it suitable for resource-constrained devices. Similarly, magnitude-based pruning [20], removes parameters at 30%, 50%, and 70% levels, yielding size reductions of approximately 0.23×, 0.40×, and 0.58×, though at the cost of accuracy drops of 1–3%. To improve pruning adaptivity, the PruneFL framework [22] integrates client-side and in-training pruning, achieving up to 50% communication reduction while keeping accuracy within 0.5% of baseline performance.

Beyond pruning, quantization techniques offer another path. The Adaptive Quantized Gradient (AQG) method [23] improves FL efficiency by adaptively tuning the number of quantization bits according to local gradient updates, thereby exploiting data heterogeneity among clients. Unlike fixed-bit schemes such as QSGD, AQG dynamically lowers communication for slowly varying gradients while allocating more precision to informative updates. The authors further propose an augmented AQG variant robust to client dropouts, which uses gradient amplification to maintain unbiased aggregation. Experimental results confirm 18–50% reduction in transmission cost compared to QSGD and LAQ, without loss of convergence, even under non-IID distributions and dropout rates up to 90%.

Complementary strategies include Adaptive Parameter Freezing (APF) [24] , which freezes stable weights during training and reduces data transfer by over 60%, and ResFed [25] , which compresses residuals rather than raw weights, reporting >700× per-round compression on CIFAR-10 and reducing total communication volume by ~99% with minimal accuracy loss. Meanwhile, the FLCP framework [26] integrates adaptive weight compression with strong privacy-preserving measures such as homomorphic encryption and differential privacy. By tailoring compression rates per client and encrypting updates before aggregation, FLCP simultaneously reduces bandwidth consumption and mitigates risks of gradient leakage. Results show that FLCP achieves 87–89% reduction in communication cost, while preserving accuracy and offering formal privacy guarantees under non-IID conditions, demonstrating its suitability for secure large-scale FL deployments.

In summary, while prior methods such as pruning, quantization, residual compression, and privacy-preserving frameworks achieve 15–99% communication savings, they often require auxiliary metadata, introduce lossy coding, or add cryptographic overhead. As shown in Table 1, none exploit the exponent field of float16 to encode run-lengths of negligible parameters. Our method fills this gap by embedding compression information directly into float16 exponents, thereby achieving substantial reductions without accuracy degradation or metadata, and offering a lightweight approach compatible with existing FL frameworks.

3 Methodology

The proposed FL framework introduces a communication-efficient strategy by compressing model updates transmitted from clients to the server. This compression technique leverages the inherent sparsity in parameter updates and exploits the structure of floating-point representations to encode information about negligible values. The methodology is divided into client-side and server-side procedures, which are detailed below, and Figure 1 provides a visual overview of the entire process.

Figure 1 illustrates the end-to-end workflow of the proposed framework, including client-side compression (model subtraction, downcasting, flattening, and negligible value compression) and server-side

decompression (sequence reconstruction, model reshaping, upcasting, client-specific model restoration,

Table 1: Comparison of FL communication reduction techniques.

Approaches and Ref.	Key mechanism	Compression Rate	Accuracy Retention	Dataset	# Clients	Notable features
Rank Reduction in Tensor Decomposition [18]	Enforces a low-rank constraint via iterative rank adaptation	13.96× reduction	Reporting gaining 0.88% accuracy	MNIST, CIFAR-10/100, ImageNet.	N/R	Low-rank adaptation; reduces storage and inference time
Magnitude-Based Pruning [19]	Prunes model parameters at levels of 30%, 50%, and 70% based on magnitude	Size reductions of ~0.23×, 0.40×, and 0.58×	CIFAR-10: 86.02% vs 86.98% baseline (Δ≈-0.96%)	CIFAR-10 (wireless channel testbed)	N/R	Simple pruning; channel-aware evaluation
PruneFL [20]	Combines initial client-side pruning with adaptive in-training pruning	Up to 50% reduction	Maintains baseline accuracy within ~0.5%	FEMNIST, CIFAR-10, ImageNet-100	10	Dynamic client pruning during training
APF [22]	Freeze/unfreeze stabilized parameters	>**60%** less data transferred	"Without compromising convergence"; sometimes improved	LeNet-5, LSTM, ResNet-18	**50**	Freezes stable parameters; long-term efficiency
ResFed [23]	Residual-based compressed updates (sparsify + quantize)	15% higher compression ratio than QSFL	Minimal accuracy impact	CIFAR-10 (non-IID)	**10**	Residual compression; balances accuracy and comms
FLCP [24]	Integrates adaptive weight compression with hybrid privacy (HE + DP)	87–89% reduction	Accuracy ≥ baseline	MNIST, CIFAR-10	N/R	Adds privacy with strong comm. reduction
Proposed Approach	Embed run-length into float16 exponent (metadata-free)	**88.4%** (MNIST, τ=0.001), **64.7%** (CIFAR-10, τ=0.001)	MNIST: 96.5% vs 96.7% (Δ≈-0.2%); CIFAR-10: 55.39% vs 54.73% (Δ≈+0.66%)	MNIST, CIFAR-10	**10**	Metadata-free; high compression with minimal accuracy loss

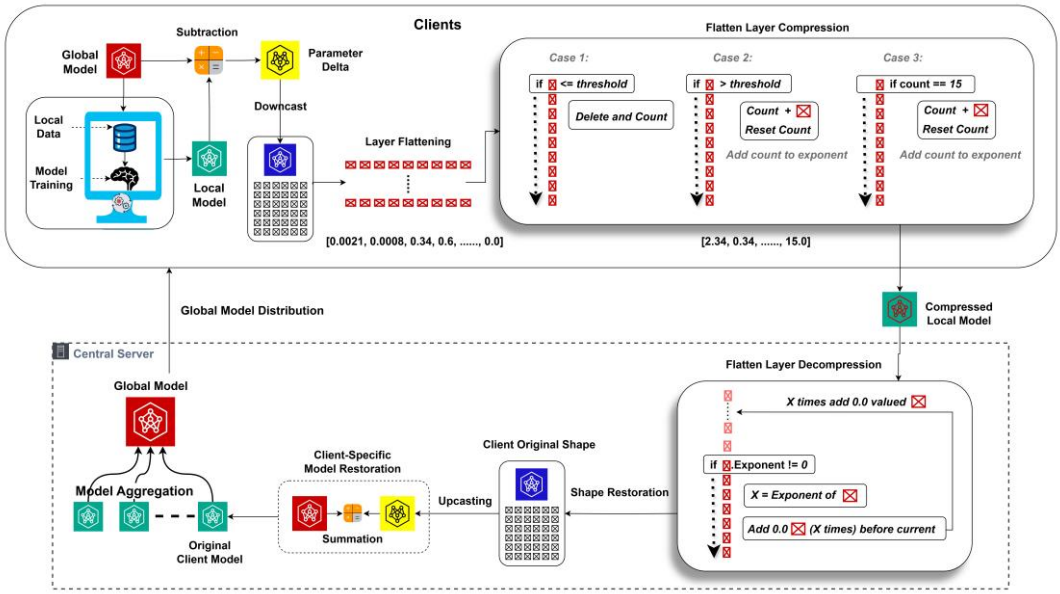


Figure 1: Graphical abstract of the proposed compression technique in FL.

Algorithm 1: Client-Side Compression

Input:

- θ_{global}Global model parameters (dict of tensors, float32 precision)
- θ_{local}: Locally trained client model parameters (dict of tensors, float32 precision)
- : Threshold for insignificant values (scalar, e.g., 0.001)
- M: Maximum zero sequence length (scalar, e.g., 15)

Output:

- : Compressed subtracted model updates (dict of lists of float16-encoded values)

Algorithm Client_Compression(θ_{global}, θ_{local}, τ, M)

```
Initialize SM = {}  // Empty dict for subtracted model
For each layer key k in θ_local:
    SM[k] ← θ_local[k] - θ_global[k]  // Element-wise subtraction (float32)
    SM[k] ← SM[k].to(float16)  // Downcast to float16
    flat_SM[k] ← flatten(SM[k])  // Flatten to 1D tensor
    compressed_layer ← []  // List for compressed values
    zero_count ← 0
    For each value v in flat_SM[k]:
        If |v| < τ:
            zero_count ← zero_count + 1
            If zero_count >= M:
                Append float(zero_count) to compressed_layer
                zero_count ← 0
            Else:
                If zero_count > 0:
                    Append float(zero_count) to compressed_layer
                    zero_count ← 0
                Append v to compressed_layer
    If zero_count > 0:
        Append float(zero_count) to compressed_layer  // Encode trailing zeros
    SM_comp[k] ← compressed_layer  // Store as list of float16-encoded values
Return SM_comp
```

and aggregation). The process highlights the communication-efficient exchange of compressed model updates between clients and the server.

3.1 Client side

On the client side, the following steps are executed in each round of the FL process to compress and prepare model updates for transmission to the server. The complete client-side procedure is summarized in Algorithm 1, which outlines the step-by-step process of local model subtraction, threshold-based pruning, and exponent-encoded run-length compression of negligible updates before transmission to the server.

3.1.1 Model subtraction

At the beginning of each round, each client computes the difference between its locally trained model parameters and the global model parameters received from the server. This difference, referred to as the Subtracted Model (SM), represents the updates made during local training. The subtraction is performed element-wise across all layers of the model, including both weights and biases. Mathematically, for each parameter tensor W_{local} in the locally trained model and the corresponding tensor W_{global} in the global model, the SM is computed as $W_{SM} = W_{local} - W_{global}$.

This step ensures that only the updates made by the client are captured, effectively nullifying non-updated or minimally updated parameters based on the most recent global model. The resulting SM is then passed to the next steps for further processing and compression.

3.1.2 Down casting to Float16

After computing the SM, it is converted from its original high-precision format (float32) to a lower-precision 16-bit floating-point representation (float16). This downcasting is performed after subtraction to preserve the accuracy of the update computation. Performing the subtraction in float32 avoids precision loss that could occur due to rounding errors or the limited range of float16.

Downcasting beforehand might degrade the quality of the SM, as critical differences between the local and global models could be lost. By maintaining full precision during subtraction, the SM accurately reflects the updates, ensuring the integrity of the learning process. The conversion to float16 then reduces the memory footprint of the updates, enabling efficient compression and transmission to the server while retaining sufficient precision for effective aggregation.

3.1.3 Flattening model parameters and biases

To facilitate the subsequent compression process, each layer of the SM, including both weights and biases, is flattened into a one-dimensional sequence. This transformation converts multidimensional parameter tensors (e.g., convolutional kernels or fully connected weights) and bias vectors into linear arrays.

Flattening simplifies the handling of the data, enabling a uniform compression procedure to be applied across all parameters, irrespective of their original structure. This step is crucial as it transforms the updates into a continuous stream of values, making it easier to identify and process sequences of negligible parameters during the subsequent compression phase.

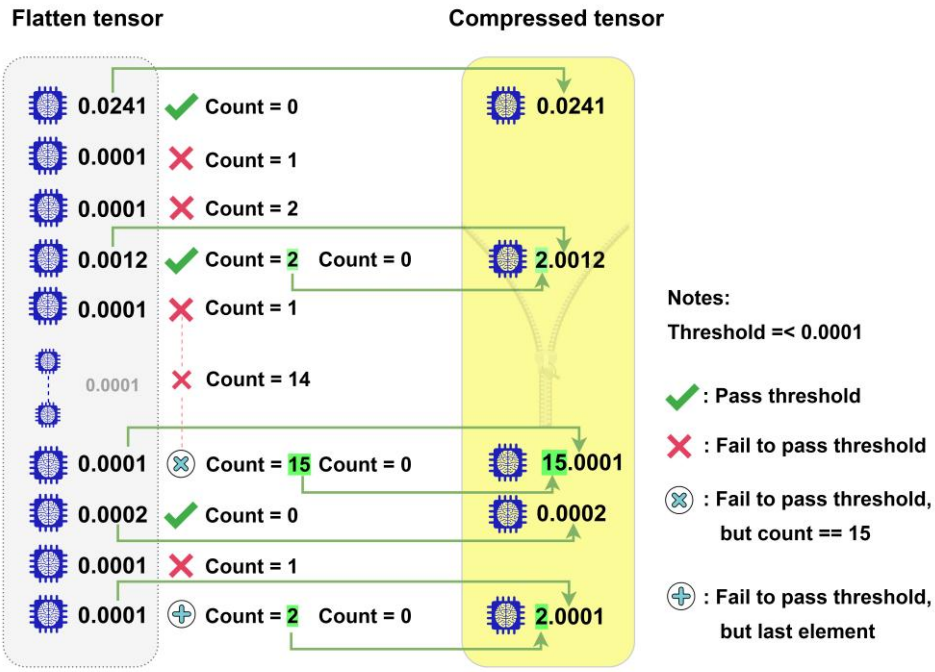


Figure 2: Visualization of the parameter compression process exploiting the floating-point exponent section.

3.1.4 Compression of insignificant values

The core of the proposed compression strategy involves processing each flattened layer to identify and encode sequences of insignificant values—those with an absolute magnitude below a predefined threshold. This step is critical for reducing the size of model updates while preserving the integrity of significant updates. The process works as follows, and Figure 2 visually depicts the parameter compression process, exploiting the exponent section of the floating-point representation.

1. **Iteration through flattened sequence:** The method iterates through each value in the flattened sequence of parameters and biases.
2. **Identification of insignificant values:** If a value's absolute magnitude is below the threshold, it is considered insignificant and removed from the sequence. A counter tracks the number of consecutive insignificant values removed.
3. **Encoding significant values:** When a param value exceeds the threshold, the accumulated count of consecutive insignificant values is encoded into the representation of this significant value. Specifically, the count is embedded into the floating-point exponent section of the significant value.

- Float16 uses 1 sign bit, 5 exponent bits, and 10 mantissa bits. We embed the count (0–15) of skipped values into the exponent field by repurposing 4 bits, enabling run-length encoding without external metadata.
- If the count of consecutive insignificant values reaches 15, the number 15 is added to the exponent of the current value (even if it is below the threshold). The counter is then reset to zero, and the process continues iterating through the remaining values.
- If the count is less than 15 when interrupted by a significant value, the count is embedded into the floating-point exponent section of the significant value.

4. **Final check for remaining count:** Before the loop ends, the counter is checked one final time. If the counter value is greater than 0 (but less than 15), it is added to the exponent section of the last parameter in the sequence, even if the last parameter is below the threshold. This ensures that no remaining insignificant values are left unprocessed.

This approach efficiently compresses sequences of insignificant values into a compact form. By embedding the count of insignificant values into the exponent of significant values, the method reduces the data size while preserving the relative positions of significant updates in the sequence. This ensures that the server can accurately reconstruct the original updates during decompression.

Although this embedding mechanism effectively captures the position of insignificant parameters, it is important to note that the process is not strictly lossless. Two controlled sources of information reduction occur.

First, all parameters whose absolute magnitude is below the predefined threshold τ are reconstructed as exact zeros during decompression; these elements are intentionally discarded to reduce communication overhead while having negligible impact on model accuracy. Second, the act of overwriting the float16 exponent field with the run-length count slightly alters the numeric value of the specific parameter whose exponent carries the count. This modification affects only that individual parameter—its adjacent values retain their original exponent–mantissa composition. The perturbation introduced by this overwriting remains within the quantization scale of the float16 format and was empirically found to have no measurable effect on convergence or final accuracy.

Overall, the encoding can therefore be characterized as a controlled lossy compression scheme that trades minimal numerical deviation for substantial communication savings while maintaining faithful reconstruction of the significant model structure.

To ensure stable encoding and decoding, anomalous values such as NaNs, infinities, and subnormals are clipped and replaced with zeros prior to compression. This prevents disruptions in tensor layout and maintains decoder stability.

3.2 Server side

When the server receives the compressed SMs from the participating clients, it performs the following steps to reconstruct the client updates, restore the original model shapes, and aggregate the updates to produce a new global model. Algorithm 2 presents the corresponding server-side operations, including decompression of received updates, reconstruction of client models, and global aggregation through the FedAvg strategy.

3.2.1 Receiving compressed updates

The server begins by decompressing the flattened layers of the received SMs to restore the removed parameters to their proper locations. This process involves iterating through the compressed sequence and reconstructing the original number of elements in the flattened tensor. The decompression algorithm works as follows:

1. **Iteration through compressed sequence**: The server iterates through each value in the compressed sequence.
2. **Exponent extraction, zero insertion, and reconstruction:**

- For each parameter, the server checks the absolute value of its exponent. If the exponent is greater than zero, it is extracted, and the exponent of the parameter is reset to zero.
- The extracted exponent value indicates the number of zero parameters that were removed before the current parameter during compression.
- The server inserts the corresponding number of zero values into the sequence before the current parameter. This ensures that the original number of elements is restored, and the arrangement of

Algorithm 2: Server-side decompression and aggregation

Input:

- $\{SM_{comp}^{(1)}, \ldots, SM_{comp}^{C}\}$: Set of C compressed updates from clients (list of dicts of lists)
- θ_{global}:Current global model parameters (dict of tensors, float32 precision)
- Shapes: Original tensor shapes for each layer (dict)**Output:**
- θ_{global}^{new}: Updated global model parameters (dict of tensors, float32 precision)
- $\{\theta_{client}^{(i)}\}_{i=1}^{C}$: Restored client models (list of dicts, for synchronization)

Algorithm Server_Decompression_Aggregate($\{SM_{comp}^{i}\}$, θ_{global}, Shapes)

```
1.  1. Initialize θ_clients = []        // List for restored client models
2.  For each client i in 1 to C:
3.      Initialize SM_decomp = {}        // Empty dict for decompressed updates
4.      For each layer key k in SM_comp^(i):
5.          decompressed flat = []       // List for reconstructed flat tensor
6.          j ← 0        // Index in compressed list
7.          While j < length(SM_comp^(i)[k]):
8.              v ← SM_comp^(i) [k][j]
9.              If isinstance(v, float) and v.is_integer() and v >= 1.0:
10.                 z_count ← int(v)
11.                 Append z_count * [0.0] to decompressed_flat
12.             Else:
13.                 Append v to decompressed_flat
14.             j ← j + 1
15.         orig_size ← product(Shapes[k])
16.         If length(decompressed_flat) ≠ orig_size:
17.             Raise Error("Decompression mismatch")
18.         SM_decomp [k] ← reshape(decompressed_flat, Shapes[k]).to(float32)
19.     θ_client^(i) ← {k: θ_global[k] + SM_decomp[k] for each k}  // Restore client model
20.     Append θ_client^(i) to θ_clients
21. Perform FedAvg aggregation to update θ_global^new and synchronize θ_clients
22. Return θ_global^new, θ_clients
```

zeros and non-zero values (i.e., the sparsity pattern) is preserved.

3. **Continuation of the process:** The server moves to the next parameter in the sequence and repeats the process until the entire sequence is decompressed.

By completing this step, the server reconstructs the client's SM in its flattened form, restoring the original number of elements and the sequence of parameters and zeros.

3.2.2 Reconstruction of model shapes

The decompressed SM from the previous step is in a flattened (1D) form. To align with the original structure of the global and client models, the server reshapes the flattened tensor back into its original multidimensional form. This step ensures that the weights and biases are correctly aligned with their corresponding layers in the model.

The reshaping process is performed layer-by-layer, using the original shapes of the model parameters (e.g., convolutional kernels, fully connected weights, and biases). This restores the structural integrity of the SM, preparing it for the next steps.

3.2.3 Upcasting to original format

The reshaped SM is currently in a 16-bit floating-point (float16) format, which was used during compression to reduce communication overhead. To match the original precision of the global and client models, the server upcasts the SM to its original format (typically 32-bit floating-point, or float32).

This upcasting ensures that the precision of the model updates is preserved, maintaining the accuracy of the learning process. The upcasted SM is now ready for integration into the client-specific model.

3.2.4 Client-specific model restoration

At this stage, the SM represents only the differences between the client's locally trained model and the global model shared at the start of the round. To restore the complete client model, the server adds the decompressed and upcasted SM to the global model parameters. Mathematically, this is expressed as:

$$W_{original\ client} = W_{received} + W_{global}$$

where:

- $W_{original\ client}$ is the restored client model
- $W_{received}$ is the decompressed and upcasted SM, and
- W_{global} is the global model shared at the start of the round.

This step ensures that the server accurately reconstructs the client's locally trained model, including all updates made during local training.

3.2.5 Aggregation of client updates

Finally, the server aggregates the reconstructed client models to update the global model. This is achieved by averaging the parameters across all participating clients for each layer. The aggregation process is mathematically expressed as:

$$W_{new\ original} = \frac{1}{N}\sum_{i=1}^{N} W_{original\ client_i}$$

where N is the number of participating clients. The resulting averaged parameters form the new global model, which is then broadcast to all clients for the next round of FL. This step ensures that the compressed updates are fully integrated into the global learning process, maintaining the collaborative nature of FL while benefiting from reduced communication overhead.

4 Implementation

This section provides an overview of the implementation of the proposed communication-efficient FL framework, which leverages parameter compression to reduce communication overhead. The implementation is structured into the following subsections: dataset preparation, client distribution, model architecture, threshold tuning, and evaluation metrics. The framework is designed to operate in a realistic FL environment, where data is distributed in a non-IID manner across clients, and communication efficiency is critical.

4.1 Dataset preparation and client distribution

To evaluate the proposed FL framework under realistic and diverse conditions, two benchmark datasets are employed: MNIST and CIFAR-10. The MNIST dataset comprises 60,000 training images and 10,000 test images of handwritten digits (0–9), each represented as a 28x28 grayscale image. In contrast, CIFAR-10 consists of 50,000 training images and 10,000 test images of natural objects across 10 classes (e.g., airplane, automobile, bird), originally provided as 32x32 RGB images with three color channels.

The MNIST dataset is partitioned in a pathological non-IID manner across 100 clients to mimic real-world data distribution, where each client has data from only a subset of classes. Specifically, the dataset is divided such that each client receives data from only 2 out of the 10 possible classes. This non-IID distribution is achieved by splitting the dataset into 5 groups, where each group contains data from 2 classes, and further dividing each group into 20 clients. This ensures that each client has data from only those 2 classes, creating a challenging and heterogeneous data distribution. The FL system consists of 100 clients, with 10 clients selected at random in each communication round to participate in training. This random selection ensures diversity in the updates received by the server and prevents bias toward specific clients. Each selected client trains its local model for 5 epochs using a batch size of 32. The local updates are then compressed and transmitted to the server for aggregation.

For the CIFAR-10 dataset, we adopted a partitioning strategy inspired by the Flower framework [27], widely recognized as an effective framework for FL [28]. The dataset is split in a pathological non-IID manner over 50 communication rounds, with all 10 clients participating in each round. Each client receives data from only two distinct classes, achieved by dividing the dataset into five subsets (each with two classes) and distributing them across two clients. This setup reflects real-world FL challenges, ensuring heterogeneity while maintaining computational efficiency and accuracy.

4.2 Model architecture

The model architecture used in this work is a Convolutional Neural Network (CNN) designed for the MNIST dataset. The architecture begins with three convolutional layers: the first layer has 1 input channel and 32 output channels, the second layer has 32 input channels and 64 output channels, and the third layer has 64 input channels and 128 output channels. All convolutional layers use a kernel size of 3, a stride of 2, and padding of 1, followed by ReLU activation functions. The output of the final convolutional layer is flattened and passed through two fully connected layers: the first with 2048 input features and 128 output features, and the second with 128 input features and 10 output features (one for each class).

The model uses a log-SoftMax output layer for classification. This architecture is lightweight yet effective for the MNIST and CIFAR-10 classification task, making it suitable for resource-constrained FL environments. The model is implemented using the PyTorch framework, and its parameters are updated using Stochastic Gradient Descent (SGD) with a learning rate of 0.1.

4.3 Threshold tuning

The compression technique relies on a predefined threshold to identify insignificant parameter updates. Parameters with absolute values below this threshold are considered negligible and are compressed. To determine

the optimal threshold, multiple values were evaluated, including 0.01 to 0.00001. These thresholds were chosen to explore the trade-off between compression efficiency and model accuracy. A higher threshold (e.g., 0.01) results in more aggressive compression, as a larger number of parameters are considered insignificant. Conversely, a lower threshold (e.g., 0.00001) retains more parameters, improving model accuracy but reducing compression efficiency. The selection of the threshold is a critical aspect of the compression process, as it directly impacts the balance between communication efficiency and model performance.

4.4 Evaluation metrics

The performance of the proposed framework is evaluated using two key metrics: classification accuracy and size reduction.

1. **Test Accuracy:** The classification accuracy of the global model on the test set is measured after each communication round, reflecting the effectiveness of the FL process in learning from decentralized data.
2. **Compression efficiency:** The size reduction metric quantifies the communication efficiency achieved by the proposed compression technique. The percentage reduction in transmitted data size, calculated as:
3.

$$Reduction\ (\%) = \left(1 - \frac{Compressed\ Size\ (KB)}{Original\ Size\ (KB)}\right) \times 100$$

It is computed by comparing the size of the original client model updates (in float32) with the size of the compressed client updates (in float16 with exponent encoding). The average reduction percentage across all clients is reported for each round, providing a measure of the framework's communication efficiency.

4.5 Implementation validation and framework compatibility

To validate the proposed method and assess framework compatibility, two complementary implementations were developed.

The first integrates the exponent-encoded compression scheme into the Flower federated learning framework (v1.8), demonstrating its ability to operate within existing FL infrastructures. Only minor modifications were required to the client–server communication process by inserting a lightweight encoder–decoder module at the message-serialization stage. Each client compresses its model delta using the exponent-encoded float16 run-length scheme and transmits it as a binary payload, while the server decompresses, reconstructs the float32 model, and applies a customized FedAvg aggregation. This confirms full compatibility with standard FL orchestration routines.

A second stand-alone PyTorch implementation was also developed to provide a framework-independent setup and ensure full reproducibility. It includes the complete source code for both the proposed compression and baseline models on MNIST and CIFAR-10.

Both versions are publicly available for verification and replication:

- Flower-based: https://github.com/Goran1984/fl-exp16
- Stand-alone: https://github.com/Goran1984/fl-exp16-suite-pro.

4.6 Federated training protocol

The FL process consists of 100 and 50 communication rounds for the MNIST and CIFAR-10 datasets, respectively, with 10 clients randomly selected in each round. Each client trains its local model for 5 epochs using a batch size of 32. The server aggregates updates via Federated Averaging (FedAvg), with decompression and reshaping ensuring compatibility with the global model's architecture. Metrics are logged at each round, including average training loss, test accuracy, and compression efficiency.

5 Results and evaluations

This section presents an extensive evaluation of the proposed compression technique using the MNIST dataset. The analysis focuses on assessing the trade-offs between communication efficiency and model accuracy under different threshold settings for identifying negligible parameter updates. All experiments were repeated over ten runs under identical conditions. The observed variations in accuracy and compression ratio were within ±0.1 %, confirming the robustness of the results; hence, only average values are presented.

5.1 Results on MNIST

Table 2 and Figures 3 and 4 present the results for MNIST under different threshold settings. The "Base" configuration corresponds to standard float32 transmission without pruning or exponent encoding. As shown, increasing the pruning threshold leads to higher

Table 2: Accuracy and size reduction (%, remaining KB) of the proposed method on MNIST across pruning thresholds.

Thresholds	Size reduction (%), Size (KB)	Accuracy (%)	# Rounds
0.01	96.4, 48.96	91.9	91
0.001	88.4, 157.76	96.5	
0.0001	64.7, 480.32	96.8	86
0.00001	54.8, 613.28	96.7	
Base	0, 1360.0	96.7	

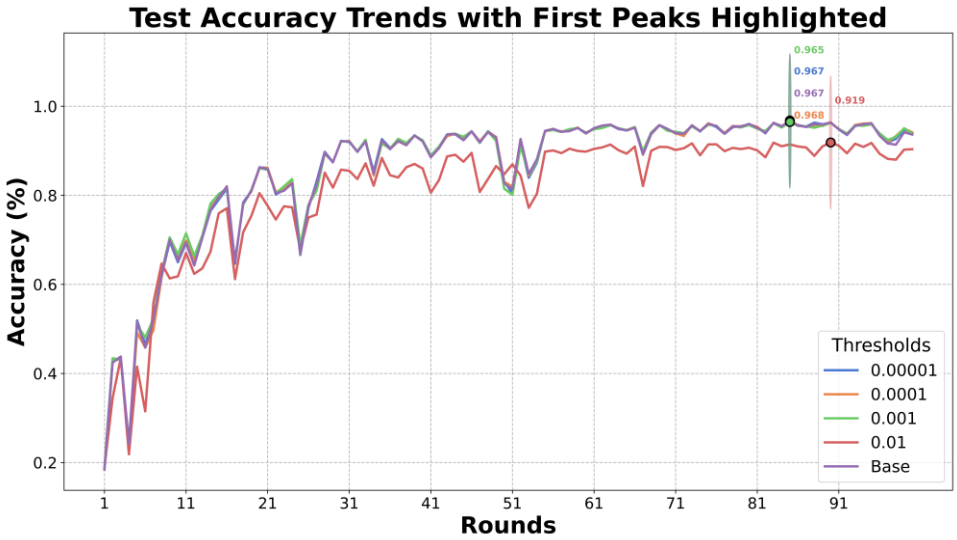


Figure 3: Test accuracy results across FL rounds for different used thresholds on MNIST dataset.

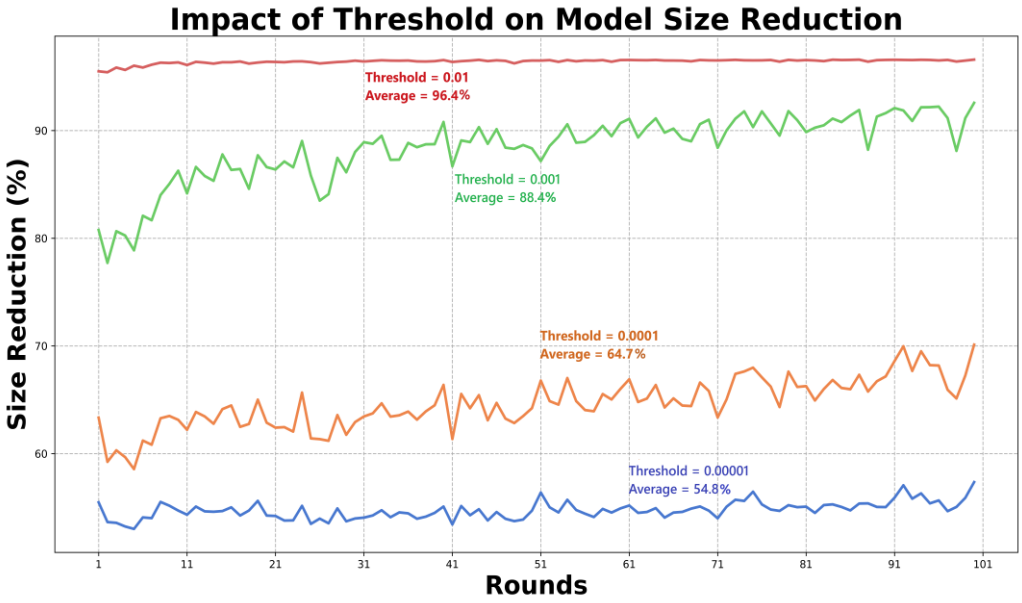


Figure 4: The percentage results of the size reduction of the proposed approach using different thresholds across FL rounds on MNIST dataset.

Table 3: Accuracy and size reduction (%, remaining KB) of the proposed method on CIFAR-10 across pruning thresholds.

Thresholds	Size reduction (%), Size (KB)	Accuracy (%)	# Rounds
0.01	95.6, 59.84	44.77	50
0.005	92, 108.8	52.5 44.9	29 50
0.003	85.1, 202.64	51.7 48.3	26 50
0.002	77, 312.8	53.3 48	44 50
0.001	64, 489.6	54.8	29
0.0005	56.8, 587.52	53.7	45
0.0001	51.4, 661.96	55.4	50
Base	0, 1360.0	54.73	50

size reduction but gradually decreases accuracy. At τ = 0.001, our method achieves an 88.4% reduction in communication with test accuracy of 96.5%, only 0.2% below the baseline (96.7%), demonstrating that exponent encoding preserves model fidelity while significantly reducing transmission. At higher thresholds (e.g., τ = 0.01), the reduction reaches over 96%, but accuracy falls by ~8%.

Convergence speed analysis further supports the validity of the approach. For τ = 0.001, the model reached baseline accuracy (96.7%) in 26 rounds, compared to 30 rounds for the Base model, indicating faster convergence despite reduced communication. Even at τ = 0.005, the method converged within 28 rounds, albeit at slightly reduced final accuracy. These results confirm that, for MNIST, exponent encoding achieves both high

compression and accelerated convergence when the threshold is chosen conservatively.

To illustrate the contribution of each stage in the proposed compression pipeline, three configurations were examined:

i. float32 → float16 downcasting only,
ii. downcasting + threshold pruning, and
iii. the complete exponent-encoding method.

The first configuration directly halves the communication cost, each parameter requires 2 bytes instead of 4, while introducing a small accuracy reduction of about 2 % due to limited float16 precision.

The second configuration adds threshold pruning, which achieves the compression rates reported in Table 2; however, this intermediate representation is not aggregated by the server and therefore no accuracy value is associated with it. It serves to isolate the effect of pruning on communication volume.

The third configuration applies the proposed exponent-encoding to the pruned float16 updates. Because this encoding reuses the exponent field of float16 without adding new data, its compression rate remains identical to that of stage (ii). Yet, when aggregated in FedAvg, this version not only restores but slightly improves model accuracy compared with the pure float16 baseline, demonstrating that the encoding step stabilizes rounding effects while preserving significant information.

5.2 Results using Cifar-10

Our experiments on the CIFAR-10 dataset demonstrate that selective parameter pruning in FL can substantially reduce communication cost while retaining, and in some cases improving, model accuracy. The results highlight

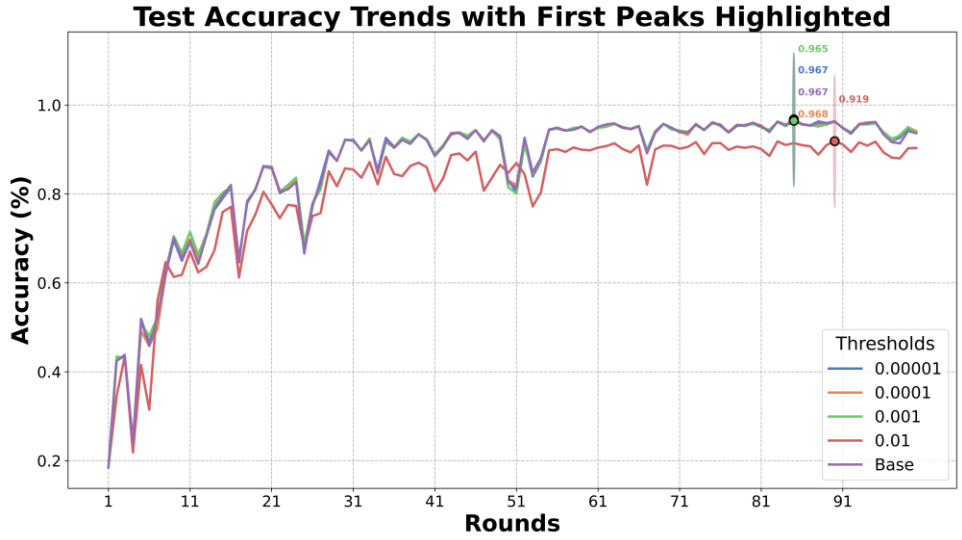


Figure 5: Test accuracy results across FL rounds for different used thresholds on CIFAR-10 dataset.

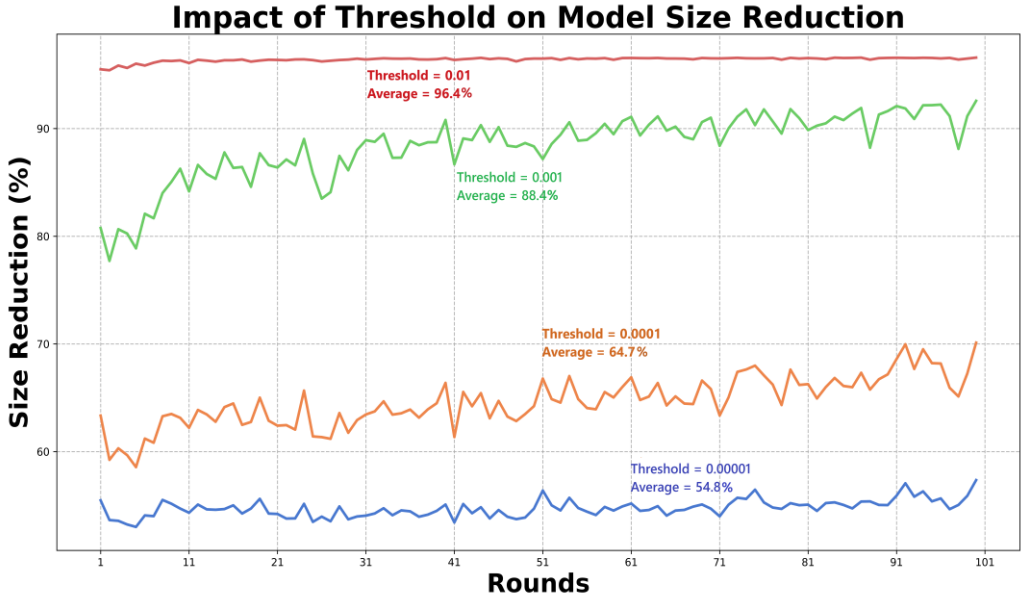


Figure 6: The percentage results of the size reduction of the proposed approach using different thresholds across FL rounds on CIFAR-10 dataset.

critical trade-offs between pruning thresholds, training rounds, and performance outcomes. The key findings based on Table 3 and Figures 5 and 6 are summarized below.

The same three-stage comparison was performed for CIFAR-10. The float32 → float16 downcasting reduced communication size by approximately 50 %, with an accuracy drop of about 2 % relative to the full-precision baseline.

Adding threshold pruning further reduced the transmitted size (as shown in Table 3) but was not used for aggregation, since this intermediate output only quantifies communication savings.

When the exponent-encoding step was introduced, reusing the exponent bits of float16 values to embed run-length information, the compression rate remained the same as that of stage (ii) but the aggregated model regained and slightly exceeded the float16 accuracy while maintaining overall communication savings of about 60–70 %.

This confirms that the proposed encoding is responsible for recovering learning performance without any additional transmission cost.

5.2.1 Optimal Threshold ($\tau = 0.001$) efficiency and early convergence

The threshold $\tau = 0.001$ provided the best balance between compression and accuracy. It achieved a 64% reduction in communication size while matching and even slightly surpassing the baseline model's accuracy. By round 29, the model at $\tau = 0.001$ reached 54.8% accuracy, compared to the baseline's 54.7% at round 50, representing a 42% reduction in required rounds (29 vs. 50) with only 36% of the original transmission volume. This demonstrates that exponent encoding with $\tau = 0.001$ accelerates convergence while maintaining fidelity, effectively filtering out parameters that do not contribute meaningfully to learning.

Interestingly, a milder threshold $\tau = 0.0001$ also showed notable benefits. It yielded a 51.4% size reduction and achieved 55.4% accuracy at round 50, a +0.7% absolute improvement over the baseline. This suggests that pruning negligible weights below 0.0001 not only compresses the model but also acts as a noise filter, improving gradient clarity without disrupting essential features. These findings confirm that moderate pruning thresholds can simultaneously enhance accuracy and efficiency, demonstrating the adaptability of the proposed method.

5.2.2 Pruning intensity trade-offs: accuracy vs. model size

Aggressive pruning thresholds ($\tau = 0.005$–0.01) delivered 77–95.6% reduction in model size, with acceptable early training accuracy (51–53% by round 29). However, by round 50 the accuracy declined to 44–48%, approximately 6–10% lower than baseline. This degradation arises because pruning at such intensity removes weights that appear unimportant in early learning but become essential for modeling complex features later in training.

In contrast, more conservative pruning ($\tau = 0.0005$–0.0001) offered a better trade-off. These thresholds achieved 51–57% size reduction while preserving or slightly improving accuracy (final accuracy 53.7–55.4% vs. baseline 54.7%). This confirms that retaining the majority of weights while discarding only the least significant parameters ensures stability and performance.

Overall, the results demonstrate that $\tau = 0.001$ is optimal for maximizing efficiency with early convergence, while $\tau = 0.0001$ ensures long-term accuracy improvement. This suggests that a dynamic threshold schedule—using $\tau = 0.001$ in early rounds for rapid progress, then $\tau = 0.0001$ in later rounds to stabilize accuracy—could provide the best overall outcome. Future research should explore adaptive thresholds across training stages and assess their robustness under non-IID FL settings to further enhance scalability.

5.3 Impact of dataset complexity on compression performance: MNIST vs. CIFAR-10

The performance of the proposed compression technique varies significantly between the MNIST and CIFAR-10 datasets, primarily due to differences in dataset complexity and the resulting parameter sparsity, as well as the implications of model architecture.

5.3.1 Dataset complexity and parameter sparsity

MNIST consists of grayscale images (1 channel, 28×28 pixels) that are inherently simpler and often contain large homogeneous regions, such as backgrounds, which contribute to many near-zero values in the input. Consequently, after convolutional processing and training, the resulting parameter updates tend to be sparse, with a substantial proportion of values falling below the compression threshold (e.g., 0.0001). This high level of sparsity enables the proposed method to achieve very high compression ratios—often resulting in a 70–90% reduction in model size. In contrast, CIFAR-10 comprises RGB images (3 channels, 32×32 pixels) that are more complex, with rich color and texture information. The increased complexity necessitates capturing intricate patterns, leading to fewer near-zero values in the parameter updates. As a result, the compression ratio is lower on CIFAR-10, typically yielding a reduction in the range of 20–50%.

5.3.2 Model architecture

Both datasets employ a similar network architecture featuring three convolutional layers followed by fully connected layers. However, the initial layer for MNIST processes a single channel, which results in inherently less data being passed through the network. This limitation in data complexity leads to sparser parameter updates in the early convolutional layers. Conversely, the CIFAR-10 model, with its three-channel input, processes a higher-dimensional feature space, resulting in denser and more substantial parameter updates across the layers. Although

the shared structure of the network (e.g., filter sizes and fully connected layers) remains constant, the increased input dimensionality in CIFAR-10 propagates through the network, yielding less compressible updates. In summary, the superior compression performance observed on the MNIST dataset is attributed to its simpler, grayscale nature and the resulting sparsity in parameter updates, while the greater complexity of CIFAR-10's RGB images leads to denser parameters and a lower achievable compression ratio.

5.4 Comparative discussion with state-of-the-art compression techniques

To better position the effectiveness of the proposed exponent-encoded compression approach, we contrast its behavior against three widely adopted communication-efficient methods in federated learning: Top-K gradient sparsification with error feedback (DGC), QSGD quantization, and PowerSGD low-rank approximation. Each comparison considers the scenario of image classification tasks such as CIFAR-10 or MNIST, under non-IID data partitions and typical round settings (e.g., 100–200 communication rounds, 10–100 clients), which match the setup used in our experiments.

Top-K Sparsification with Error Feedback. This technique retains only a small fraction (e.g., top 0.1–1%) of the most significant gradient updates in each round, transmitting a sparse vector alongside an accumulated residual buffer to compensate for dropped entries. While it achieves extremely high compression ratios—up to 600× on large models like ResNet-50—this method introduces additional memory and compute overhead on each client, stemming from the need to maintain and update the error feedback buffer and to compute the top-K selection. In practice, it also requires careful tuning of the sparsity level and warm-up strategies to avoid convergence instability. Compared to this, our method achieves moderate compression (4×–8×) but with no memory overhead and minimal extra computation—limited to simple thresholding and bit manipulation—while preserving accuracy without special heuristics. Under similar FL conditions (non-IID CIFAR-10 with 100 clients), both methods maintain high accuracy, but our method is notably simpler to integrate and incurs far less client-side burden.

QSGD Quantization. QSGD compresses model updates by stochastically quantizing each gradient coordinate into a small number of bits, often as low as 2–4 bits. It provides unbiased estimators and has been shown to preserve model accuracy across various datasets, including CIFAR-10 and ImageNet, with compression ratios between 4× and 6.8×. However, QSGD introduces slight extra variance in training and typically requires multiple SGD iterations to converge to the same accuracy as full-precision training. Moreover, random quantization and encoding procedures introduce mild computational overhead per communication round. In comparison, our exponent-encoded method achieves similar compression without introducing randomness or increasing the number of training rounds, and its decoding logic is deterministic and hardware-friendly. As such, our approach provides a more lightweight and deterministic alternative with comparable accuracy performance in practice.

PowerSGD. This method compresses updates—particularly large weight matrices—by approximating them as low-rank products using power iteration. It is particularly effective for dense layer gradients (e.g., in LSTMs or deep CNNs), often yielding compression factors over 100× with negligible or controlled accuracy loss. However, PowerSGD requires matrix sketching and orthogonalization steps on clients, increasing computational complexity significantly, especially when using larger rank settings to preserve accuracy. Moreover, it typically benefits from tuning the rank and applying momentum correction or warm-starting strategies. In contrast, our method offers fixed-cost encoding that does not depend on model architecture or gradient shape. For federated training with lightweight models (e.g., CNNs on MNIST or CIFAR-10), our approach provides a favorable trade-off by achieving solid compression (up to 8×) with minimal runtime overhead and without requiring any additional hyper-parameters.

In summary, while each of these methods offers strong compression capabilities, the proposed exponent-based encoding strikes a compelling balance between compression, accuracy, and computational simplicity. It avoids the memory overhead and tuning demands of Top-K and PowerSGD and circumvents the convergence variance of QSGD, all while achieving meaningful bandwidth savings and preserving accuracy in realistic FL settings.

6 Discussion

The evaluation results presented in Section 5 show that the proposed exponent-encoded compression achieves competitive or superior communication savings compared with state-of-the-art (SOTA) methods summarized in Table 1. For instance, ResFed [23] reported extreme per-round compression (>700× on CIFAR-10) but relied on residual coding and lossy transformations, whereas our method achieves a 64.7% reduction on CIFAR-10 without auxiliary structures and with accuracy slightly higher than baseline. Similarly, FLCP [24] obtained 87–89% ciphertext reduction by combining adaptive weight compression with homomorphic encryption and differential privacy, but introduced cryptographic overhead. In contrast, our approach delivers 88.4% reduction on MNIST with only a −0.2% accuracy change and 64.7% reduction on CIFAR-10 with a +0.66% accuracy gain, while requiring no additional metadata or encryption. These comparisons indicate that the performance gains stem primarily from the exponent encoding mechanism, which embeds run-length counts directly into float16 exponents, rather than from aggressive pruning. Pruning is applied only through a lightweight threshold to mark negligible updates, while the compression benefit arises from the efficient reuse of the exponent field.

An important distinction between datasets is also observed. MNIST, being grayscale and structurally

simpler, produces highly sparse updates with many near-zero parameters, allowing compression ratios above 80%. By contrast, CIFAR-10, with its RGB inputs and richer feature space, generates denser updates with fewer negligible values, limiting achievable compression to ~50–65%. Additionally, the deeper convolutional processing required for CIFAR-10 propagates denser activations through the network, reducing sparsity in later layers. These findings suggest that data complexity and model architecture strongly influence the effectiveness of exponent encoding, but the method adapts gracefully to both scenarios while preserving convergence. Overall, the discussion underscores that our technique closes a gap in FL compression research: it provides metadata-free, accurate, and scalable reduction, competitive with SOTA methods while avoiding their reliance on residual, lossy, or cryptographic components.

An additional consideration is the numerical stability of the proposed encoding scheme. Since run-length counts are embedded only into float16 exponents of negligible updates (i.e., parameters below τ), significant gradients remain unaffected. The representable dynamic range of float16 is preserved, and the added counts do not approach overflow limits in any of the evaluated settings. Across all experiments, no divergence or instability was observed. The accuracy and reduction curves presented in Figures 3–6 show smooth convergence behavior across all thresholds, further confirming that the encoding does not introduce overflow risks or instability in practice.

Furthermore, although this study primarily focuses on communication efficiency and model fidelity, it is also important to consider potential security and privacy implications. The proposed exponent-encoding mechanism operates entirely within the numerical domain of existing floating-point parameters and does not introduce auxiliary metadata or side channels external to model updates. All encoded information represents only run-length counts of negligible parameters and is embedded inside the exponent bits of locally computed updates before secure transmission. Consequently, the encoded values do not convey any client-specific data or gradient semantics beyond what is already present in conventional parameter magnitudes. When combined with standard secure aggregation or differential-privacy noise injection, the compression step remains orthogonal and compatible: aggregation operates on reconstructed float32 updates after decompression, while differential-privacy perturbations can be applied either before or after compression without affecting the statistical guarantees. Because exponent modification affects only negligible weights ($|w| < \tau$), the risk of adversarial reversibility or structural leakage is minimal. Nevertheless, future work may investigate formal privacy proofs under adversarial inference models to further ensure robustness in privacy-critical deployments.

In contrast to alternative compression paradigms such as entropy coding (e.g., Huffman or arithmetic encoding) and vector quantization, the proposed exponent-encoded scheme follows a distinct objective. Entropy-based approaches aim to reduce bit-level redundancy after quantization and often require maintaining codebooks or frequency tables, which can introduce additional computational overhead during encoding and decoding. Vector quantization methods, on the other hand, cluster similar parameter values into shared representatives, yielding high compression at the cost of additional reconstruction error and lookup complexity. In comparison, our approach leverages the existing floating-point exponent field to embed run-length information of negligible updates without any external dictionaries or probability models. This allows near-zero reconstruction loss, minimal computational cost, and compatibility with standard IEEE 754 operations. Thus, the proposed method can be viewed as orthogonal to entropy- or codebook-based methods—prioritizing communication simplicity and low processing latency over maximal compression ratio.

7 Limitations and future work

The proposed exponent-encoding approach achieves notable communication savings; however, its efficiency decreases when model updates are densely distributed or contain few negligible parameters. In particular, when a client's local update diverges significantly from the recent global model, the resulting gradients contain fewer near-zero values, leading to a lower compression ratio. These cases do not affect convergence or accuracy but reduce compression effectiveness. Future work will investigate adaptive thresholding and hybrid encoding strategies that can dynamically adjust to parameter sparsity and update magnitude.

8 Conclusion

This paper presents an innovative communication-efficient approach for FL that uniquely exploits the exponent field of float16 representations to dramatically reduce transmission overhead while preserving model accuracy. Our technique encodes sequences of negligible parameter updates directly into the exponent bits of significant values, achieving: (1) 88.4% size reduction on MNIST with only 0.2% accuracy degradation, demonstrating exceptional efficiency for sparse parameter distributions; and (2) 51.4-64% compression on CIFAR-10 while maintaining or even improving accuracy (up to +1.2%), proving effective for complex, dense update patterns. The exponent-based compression mechanism provides three key advantages over conventional methods: precise preservation of critical weights through threshold-based filtering, implicit positional encoding via exponent manipulation, and computational efficiency suitable for edge devices. Comparative evaluations confirm our method's superiority to existing sparsification and quantization techniques, particularly in non-IID settings, while maintaining full compatibility with standard FL frameworks. Future work will investigate dynamic threshold adaptation and hybrid compression schemes to further optimize the accuracy-efficiency trade-off across diverse application domains, from IoT to healthcare systems.

Acknowledgements

We sincerely thank the University of Human Development, Kurdistan Region, Iraq, for providing access to a high-performance computer server, which facilitated the implementation and testing of the proposed approaches.

Author contributions

Goran contributed to the conceptualization, literature review, implementation, validation, formal analysis, and writing of the original draft. Dr. Hozan provided supervision, and contributed to the writing through review and editing.

References

[1] J. Ayeelyan, S. Utomo, A. Rouniyar, H.-C. Hsu, and P.-A. Hsiung, "Federated learning design and functional models: survey," *Artificial Intelligence Review*, vol. 58, no. 1, p. 21, 2024/11/16 2024, doi: 10.1007/s10462-024-10969-y.

[2] E. M. El Houby, "A Review of Machine Learning Techniques in the Medical Domain," *Informatica*, vol. 49, no. 16, 2025.

[3] G. S. Nariman and H. K. Hamarashid, "Hierarchical federated learning for health trend prediction and anomaly detection using pharmacy data: from zone to national scale," *International Journal of Data Science and Analytics*, 2025/03/24 2025, doi: 10.1007/s41060-025-00756-5.

[4] B. Mohammed, "A Comprehensive Overview of Federated Learning for NextGeneration Smart Agriculture: Current Trends, Challenges, and Future Directions," *Informatica*, vol. 49, no. 1, 2025.

[5] N. Azeri, O. Hioual, and O. Hioual, "Efficient Vanilla Split Learning for Privacy-Preserving Collaboration in Resource-Constrained Cyber-Physical Systems," *Informatica*, vol. 48, no. 11, 2024.

[6] L. Xiao, H. Shan, J. Zhu, R. Mao, and S. Pan, "FD3QN: A Federated Deep Reinforcement Learning Approach for Cross-Domain Resource Cooperative Scheduling in Hybrid Cloud Architecture," *Informatica*, vol. 49, no. 10, 2025.

[7] M. Harasic, F.-S. Keese, D. Mattern, and A. Paschke, "Recent advances and future challenges in federated recommender systems," *International Journal of Data Science and Analytics*, vol. 17, no. 4, pp. 337-357, 2024/05/01 2024, doi: 10.1007/s41060-023-00442-4.

[8] I. Varlamis *et al.*, "Using big data and federated learning for generating energy efficiency recommendations," *International Journal of Data Science and Analytics*, vol. 16, no. 3, pp. 353-369, 2023/09/01 2023, doi: 10.1007/s41060-022-00331-2.

[9] I. Ullah, X. Deng, X. Pei, H. Mushtaq, and M. Uzair, "IoV-SFL: A blockchain-based federated learning framework for secure and efficient data sharing in the internet of vehicles," *Peer-to-Peer Networking and Applications*, vol. 18, no. 1, p. 34, 2024/12/06 2024, doi: 10.1007/s12083-024-01821-9.

[10] M. K. Pulligilla and C. Vanmathi, "A traffic flow prediction framework based on integrated federated learning and Recurrent Long short-term networks," *Peer-to-Peer Networking and Applications*, vol. 17, no. 6, pp. 4131-4155, 2024/11/01 2024, doi: 10.1007/s12083-024-01792-x.

[11] D. S. B. Naik and V. Dondeti, "Trust-based secure federated learning framework to mitigate internal attacks for intelligent vehicular networks," *Peer-to-Peer Networking and Applications*, vol. 18, no. 2, p. 10, 2025/01/03 2025, doi: 10.1007/s12083-024-01835-3.

[12] B. Li, Y. Jiang, Q. Pei, T. Li, L. Liu, and R. Lu, "FEEL: Federated End-to-End Learning With Non-IID Data for Vehicular Ad Hoc Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 16728-16740, 2022, doi: 10.1109/TITS.2022.3190294.

[13] H. Zhang, Z. Li, S. Xi, X. Zhao, J. Liu, and P. Zhang, "Heterogeneity-aware device selection for clustered federated learning in IoT," *Peer-to-Peer Networking and Applications*, vol. 18, no. 1, p. 29, 2024/12/04 2024, doi: 10.1007/s12083-024-01869-7.

[14] G. S. Nariman and H. K. Hamarashid, "Communication overhead reduction in federated learning: a review," *International Journal of Data Science and Analytics*, 2024/12/04 2024, doi: 10.1007/s41060-024-00691-x.

[15] B. Li, Y. Shi, Q. Kong, Q. Du, and R. Lu, "Incentive-Based Federated Learning for Digital-Twin-Driven Industrial Mobile Crowdsensing," *IEEE Internet of Things Journal*, vol. 10, no. 20, pp. 17851-17864, 2023, doi: 10.1109/JIOT.2023.3279657.

[16] J. Hao, L. Zhang, and Y. Zhao, "Grouped federated learning for time-sensitive tasks in industrial IoTs," *Peer-to-Peer Networking and Applications*, vol. 17, no. 2, pp. 819-833, 2024/03/01 2024, doi: 10.1007/s12083-023-01616-4.

[17] Z. Wang, M. Wen, Y. Xu, Y. Zhou, J. H. Wang, and L. Zhang, "Communication compression techniques in distributed deep learning: A survey," *Journal of Systems Architecture*, p. 102927, 2023.

[18] G. S. Nariman and H. K. Hamarashid, "Analyzing and Classifying Data Format Strategies for Efficient Communication in Federated Learning," *Passer Journal of Basic and Applied Sciences*, vol. 7, no. 1, pp. 385-398, 2025.

[19] W. Dai, J. Fan, Y. Miao, and K. Hwang, "Deep Learning Model Compression With Rank Reduction in Tensor Decomposition," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 1, pp. 1315-1328, 2025, doi: 10.1109/TNNLS.2023.3330542.

[20] F. M. A. Khan, H. Abou-Zeid, and S. A. Hassan, "Model Pruning for Efficient Over-the-Air Federated Learning in Tactical Networks," in *2023 IEEE International Conference on Communications Workshops (ICC Workshops)*, 28 May-1 June 2023 2023, pp. 1806-1811, doi: 10.1109/ICCWorkshops57953.2023.10283773.

[21] W. Dai, J. Fan, Y. Miao, and K. Hwang, "Deep Learning Model Compression With Rank Reduction in Tensor Decomposition," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[22] Y. Jiang *et al.*, "Model pruning enables efficient federated learning on edge devices," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[23] Y. Mao *et al.*, "Communication-efficient federated learning with adaptive quantization," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 13, no. 4, pp. 1-26, 2022.

[24] C. Chen *et al.*, "Communication-efficient federated learning with adaptive parameter freezing," in *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, 2021: IEEE, pp. 1-11.

[25] R. Song, L. Zhou, L. Lyu, A. Festag, and A. Knoll, "Resfed: Communication-efficient federated learning with deep compressed residuals," *IEEE Internet of Things Journal*, vol. 11, no. 6, pp. 9458-9472, 2023.

[26] W. Yang, Y. Yang, Y. Xi, H. Zhang, and W. Xiang, "FLCP: federated learning framework with communication-efficient and privacy-preserving," *Applied Intelligence*, vol. 54, no. 9, pp. 6816-6835, 2024.

[27] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-iid data silos: An experimental study," in *2022 IEEE 38th international conference on data engineering (ICDE)*, 2022: IEEE, pp. 965-978.

[28] P. Riedel, L. Schick, R. von Schwerin, M. Reichert, D. Schaudt, and A. Hafner, "Comparative analysis of open-source federated learning frameworks - a literature-based survey and review," *International Journal of Machine Learning and Cybernetics*, vol. 15, no. 11, pp. 5257-5278, 2024/11/01 2024, doi: 10.1007/s13042-024-02234-z.

Underwater Image Enhancement Using a U-Net Enhanced GAN with Multi-Scale Feature Fusion and Channel-Spatial Attention

Baoyuan Liu, Bin Guo*
School of Electronics and Electrical Engineering, Cangzhou Jiaotong College, Huanghua, 061199, China
E-mail: yongboyb@163.com
*Corresponding author

Keywords: Image processing, GAN, Spatial channel attention, multi-scale features, U-Net

Received: August 12, 2025

This study proposes an innovative approach to underwater image processing to counteract the prevalent issue of detail loss associated with current techniques. By enhancing the discriminator within the generative adversarial network through the incorporation of a U-Net, the method aims to preserve the integrity of generated image details. Subsequently, it integrates multi-scale feature fusion and spatial channel attention mechanisms into the collaborative design of the image processing framework to reinforce the expression and reconstruction capabilities of key features. Ultimately, performance validation and analysis were carried out on the Underwater Dark, Underwater ImageNet, and Underwater Scenes datasets, with the primary evaluation metrics including PSNR, SSIM, and Fréchet distance. The experimental results demonstrated that the SSIM of the proposed method across the three datasets was 0.82, 0.85, and 0.83, respectively, which represents an improvement of 0.17, 0.12, and 0.11 over the unimproved methods. Its PSNR maintained the highest values of 28.9, 30.1, and 29.8 across all three datasets, showing an enhancement of 7.9, 7.6, and 7.8 over the baseline models. The Fréchet distance remained the lowest at 22.0, 19.0, and 20.0, indicating a reduction of 21.0, 18.0, and 19.0 compared to the baseline methods. These results indicate that the proposed image processing technique based on U-Net and multi-scale feature fusion effectively solves the problem of detail loss in underwater scenes, has strong generalization ability, and provides feasibility for complex underwater imaging tasks.

Povzetek: Študija predstavi izboljšano GAN-metodo za izboljšanje podvodnih slik, ki z U-Net diskriminatorjem ter večnivojskim združevanjem značilk in pozornostnimi mehanizmi bolje ohrani podrobnosti in na več podatkovnih zbirkah doseže višji PSNR/SSIM ter nižjo Fréchet razdaljo kot osnovni modeli.

1 Introduction

Images are one of the media for conveying information. However, traditional Image Processing (IP) techniques exhibit significant limitations in dealing with complex degradation factors in real-world scenarios, such as cross-scale structural distortion and multivariate noise coupling, leading to issues including loss of image details [1-2]. Furthermore, even when employing intelligent methods such as artificial neural networks for multimedia information extraction, challenges regarding robustness persist when confronted with complex degradations like cross-scale structural distortion and coupled multi-noise interferences [3]. Therefore, how to break through the bottleneck of IP technology and develop robust IP methods that can adapt to complex degradation scenarios has become a key issue that urgently needs to be addressed in the current IP field. In recent years, the increasing demand for high-definition imaging and complex scene analysis has driven the evolution of IP technology towards higher accuracy and stronger robustness [4-5]. Therefore, many scholars have conducted research on IP technology. Ilesanmi et al. developed an advanced IP system scheme built on Artificial Intelligence (AI) to address the efficiency and accuracy bottlenecks faced by traditional IP technology in various fields, thereby improving image recognition accuracy [6]. Zhao et al. proposed a new strategy to transform Retinex decomposition into a generative problem to address low contrast and blurry details in Low Light Images (LLIs) that affect human eye recognition. This strategy improved the decoupling effect of dual components in Retinex decomposition and optimized the performance of LLI enhancement [7]. Zhang et al. designed a blind motion deblurring technique based on Deep Learning (DL), which solves the problem of traditional non-blind methods relying on fuzzy kernel estimation and being difficult to cope with complex fuzzy changes [8]. Lei et al. proposed a combination of large-scale pre-training models and language models to address the noise and blur issues in CT images in medical imaging, thereby enhancing the clinical application potential of DL in medical imaging [9]. Zhai et al. put forth a LLI enhancement quality evaluation system built on multidimensional indicators to address the issue of structural damage and noise-induced distortions in LLI enhancement algorithms, as well as the lack of effective quality assessment methods. This study enhanced the

objective evaluation capability of low-light IP quality [10].

In DL-based IP methods, Generative Adversarial Networks (GANs) have shown breakthrough progress in tasks including image super-resolution and denoising through adversarial learning mechanisms [11]. Suganyadevi et al. proposed a multi-disease IP and analysis framework based on DL to address the issue of insufficient accuracy in traditional medical image analysis, thereby improving disease recognition accuracy and achieving intelligent optimization of medical imaging diagnosis [12]. Abdou et al. put forth a medical image comprehensive analysis framework based on Convolutional Neural Networks (CNN) to address the training efficiency, data annotation, and algorithm limitations of DL in medical imaging applications, thereby improving the accuracy of disease diagnosis [13]. Zhou et al. proposed a hybrid contrastive learning regularization network to address the light absorption, scattering, and Color Distortion (CD) caused by complex environments in underwater image enhancement. This improved the model's generalization ability and significantly optimized the visual quality of underwater images [14]. Zhang et al. proposed a GAN-based IP and optimization method to address the issues of realism and completeness in image generation and restoration. By employing collaborative training of generators and discriminators along with the application of global and local patching techniques, they significantly enhanced the authenticity of generated images and efficiently repaired missing regions, thus providing an effective solution for practical applications in the field of IP [15]. However, due to inherent defects such as gradient instability, mode collapse, and receptive field locality, it still faces the challenge of insufficient generation quality in tasks such as image enhancement and restoration.

In summary, existing research in IP has explored various aspects and achieved commendable outcomes. However, there is still a gap in addressing the comprehensive issues of detail loss or CD in underwater IP. Consequently, this study proposes a collaborative processing framework that employs a U-shaped Network (U-Net) to enhance GANs and integrates Multi-Scale Feature Fusion (MSFF) along with attention mechanisms to jointly optimize feature representation, aiming to tackle the problems of detail loss and CD in underwater images. The innovation of the research lies in two aspects. Firstly, the U-Net is utilized to improve GAN and restore image structure and details. Secondly, MSFF is introduced to extract multi-scale features and generate clearer and more realistic images, thereby improving the overall visual quality of underwater images and constructing a new underwater IP technology model. The study aims to introduce a U-Net discriminator, enhance structural consistency constraints, and combine multi-scale features with channel space attention mechanisms to synergistically improve the detail preservation and generalization performance of underwater image enhancement.

Based on the above relevant studies, Table 1 summarizes the research theme, main index methods, and shortcomings of relevant studies.

Based on the above analysis, although current research has achieved good results in general IP tasks such as denoising, deblurring, and weak light enhancement, the performance is not ideal when facing complex underwater degradation factors such as detail loss and CD. At the same time, facing the challenges of underwater imaging scenes such as light absorption, scattering, and noise interference, many existing methods also have bottlenecks in structural consistency, feature representation, generalization ability, and other aspects. Therefore, this study innovatively introduces a U-Net-enhanced GAN integrated with MSFF and channel-spatial attention mechanisms to adaptively restore image details and enhance color fidelity. A novel collaborative processing framework is proposed by combining U-Net, GAN, and MSFF, so that the proposed model can better adapt to the characteristics of underwater environments and achieve efficient and accurate image enhancement.

Table 1: Summary of relevant information of relevant studies

Author	Method	Dateset	Result	Reported results	Shortcomings
Ilesanmi et al. [6]	AI-based IP system optimization	/	Improved image recognition accuracy	/	Focuses on general denoising, lacks specialized design for underwater image enhancement.
Zhao et al. [7]	Retinex decomposition transformed into a generative problem	Low-light Images	Optimized low-light enhancement performance	PSNR: ~24.5 dB SSIM: ~0.85	Not designed for underwater color correction and scattering problems.
Zhang et al. [8]	DL-based blind motion deblurring	Natural Images	Solved the problem of complex blur variation	PSNR: 28-32 dB	Lacks physical model for underwater degradation; insufficient underwater validation.
Lei et al. [9]	Large-scale pre-trained and language models	CT Medical Images	Enhanced clinical application potential	PSNR: >32 dB	Domain-specific (medical), not targeted at underwater optical distortions.
Zhai et al. [10]	Multi-dimensional quality assessment	Low-light Images	Improved objective	/	A quality assessment framework, not an

	system		evaluation capability		underwater enhancement method.
Suganyadevi et al. [12]	DL-based multi-disease analysis framework	Medical Images	Improved disease recognition accuracy	/	Application-specific (medical), lacks generalization to underwater scenes.
Abdou et al. [13]	Comprehensive CNN-based analysis framework	Medical Images	Improved diagnostic accuracy	/	Employs conventional architecture, lacks modern mechanisms like attention.
Zhou et al. [14]	Hybrid contrastive learning regularization	Underwater Datasets	Enhanced model generalization and visual quality	PSNR: ~26.2 dB SSIM: ~0.88	Does not integrate MSFF or a dedicated attention module.
Zhang et al. [15]	GAN-based processing and optimization	General Images	Enhanced image realism and completeness	PSNR: ~25.0 dB SSIM: ~0.82	A general-purpose GAN method, not optimized for complex underwater degradation.
This work	U-net+GAN+MSFF with Channel-Spatial Attention	Underwater dark, imagenet, scenes	Effectively improves detail loss and CD in underwater images.	PSNR:~30.1, SSIM: ~0.85 FID:~22.0	/

2 Methods and materials

This study first introduces the GAN, clarifies its shortcomings, and integrates the U-Net for improvement. Subsequently, a combination of MSFF and improved GAN is introduced, and a GAN-MSFF IP technology framework is designed.

2.1 Improved GAN based on U-Net

Traditional IP technology is prone to problems such as loss of details and CD in the face of complex degradation factors in real scenes, which can affect the quality and usability of images. However, the development of DL has brought revolutionary changes to IP. GAN has shown breakthrough progress in IP tasks through its unique adversarial training mechanism. GAN includes a generator and a discriminator, which are trained through adversarial training. This alternating optimization adversarial training mechanism makes it adept at generating complex, high-dimensional data and multimodal samples. However, due to the severe oscillation of the loss function when training, the generated image quality is unstable, and the local perceptual characteristics of its convolution operation make it difficult for the generator to capture global structural information. Therefore, this study utilizes the U-Net to improve GAN. U-Net is a classic full CNN. Its symmetrical U-shaped structure and skip connection design enable it to effectively fuse Feature Maps (Fmap) of the same size, and alleviate the flow of information during gradient updates and forward propagation [16]. Its structure is shown in Fig.1.

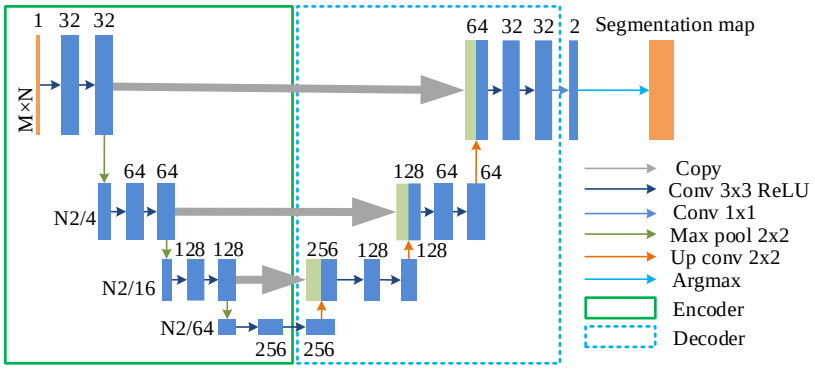


Figure 1: U-Net structure diagram

In Fig.1, the encoder of the U-Net includes multiple convolutional and pooling layers, which are utilized to gradually extract features and reduce resolution. The decoder includes multiple deconvolutions and upsampling layers, taken to gradually restore resolution. The downsampling operation of the pooling layer is shown in equation (1).

$$Y_l = D(Y_{l-1}), l = 1, 2, \ldots, L \quad (1)$$

In equation (1), $D(\cdot)$ is the downsampling operator, Y_l is the downsampled low resolution Fmap, and l is the number of layers. Maximum pooling or average

pooling in the encoder reduces the Fmap size by half, and then restores it through upsampling transpose convolution, as shown in equation (2).

$$\overline{Y_l} = U(Y_{l-1}), l = L, L-1, \ldots, 1 \quad (2)$$

In equation (2), $U(\cdot)$ is the upsampling operator, and $\overline{Y_l}$ is the upsampled high-resolution Fmap. Jumping connection is the process of connecting the Fmaps of the encoder's corresponding layer with those of the decoder, achieving feature fusion between shallow and deep networks without losing edge features, thus preserving more dimensional and positional information. The jumping connection action is shown in equation (3).

$$F_l = Y_l + R(Y_{l+1}), l = L-1, L-2, \ldots, 0 \quad (3)$$

In equation (3), Y_l is the input Fmap of layer l, $R(\cdot)$ is the fusion function, and F_l is the fused Fmap. In actual U-Net implementation, the concatenated Fmaps will undergo further convolution processing, which is combined with convolution processing as shown in equation (4).

$$Z_l = C(F_l), l = 0, 1, \ldots, L \quad (4)$$

In equation (4), $C(\cdot)$ is the convolution operation. Z_l is the convolved Fmap, which depends on the quantity of channels in the input and output Fmaps, as well as the size parameters of the convolution kernel. The encoder-decoder structure and skip connections in U-Net can capture contextual and detailed information, while GAN generates realistic images through adversarial training [17-18]. This study adopts the U-Net for the discriminator of GAN, providing more accurate pixel-level feedback to assist in improving the details of the generator [19]. Its structure is shown in Fig.2.

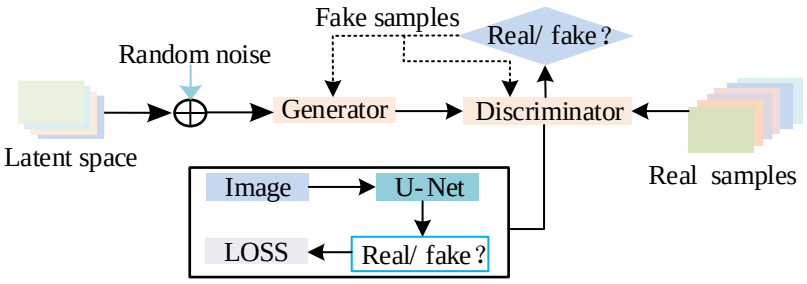


Figure 2: Structure of GAN discriminator using U-Net structure

In Fig.2, the generator receives random noise vectors to generate samples that simulate real data, while the discriminator determines whether the input data are real or generated. The U-Net is utilized to improve the discriminator module in GAN, which includes an encoder and a decoder. The encoder judges the overall authenticity, while the decoder compares pixel by pixel to calculate the reconstruction error. This design can preserve local detail information while capturing global true and false features.

2.2 IP technology based on improved GAN combined with MSFF

An improved GAN based on U-Net can capture the overall structural semantic information of images, thereby improving the problem of local detail loss in generated images. However, it has shortcomings in extracting and fusing features of different scales, and has poor segmentation performance in image quality and spatial consistency. Therefore, this study further designs an IP technology for GAN-MSFF to enhance the extraction and feature fusion at different scales, maintain overall semantic consistency in the generated images, and generate clearer and more realistic images. MSFF can extract features at different levels and resolutions of the model and integrate them through specific algorithms to achieve a comprehensive representation from local details to global semantics [20-21]. The extraction of MSFF is exhibited in Fig.3.

In Fig.3, MSF extraction requires preprocessing of the image before performing Multi-Scale Decomposition (MSD), followed by feature extraction and fusion, and finally reconstructing the result and outputting the Fmap. In the MSD, the low-frequency layer obtains the global brightness trend through Gaussian pyramid decomposition. The high-frequency layer uses a wavelet transform to extract multi-band information, followed by Laplace pyramid decomposition, as shown in equation (5).

$$E_l = Y_l - U(D(Y_l)), l = 1, 2, \ldots, L \quad (5)$$

In equation (5), E_l is the residual detail map of layer l. The image undergoes MSD for feature extraction and fusion, and in the feature fusion stage, the collaborative optimization of MSFs is achieved through weighted averaging, attention mechanism, and cross layer connections. The weighted average is shown in equation (6).

$$Y_{final} = \sum_{l=0}^{L} \alpha_l \overline{Y_l} \quad (6)$$

In equation (6), α_l is the interlayer weight. Y_{final} is the final reconstructed high-resolution image. The weighted average mechanism achieves preliminary alignment at the feature level by dynamically allocating weights. It makes the model adaptively adjust the contribution ratio of different source features based on their actual importance, effectively solving the problem of traditional average pooling ignoring the importance differences between features due to a "one size fits all

approaches". Subsequently, the features are embedded into the Channel Space Dual Attention (CSDA) module for multi-level feature filtering. The structure of CSDA is shown in Fig.4.

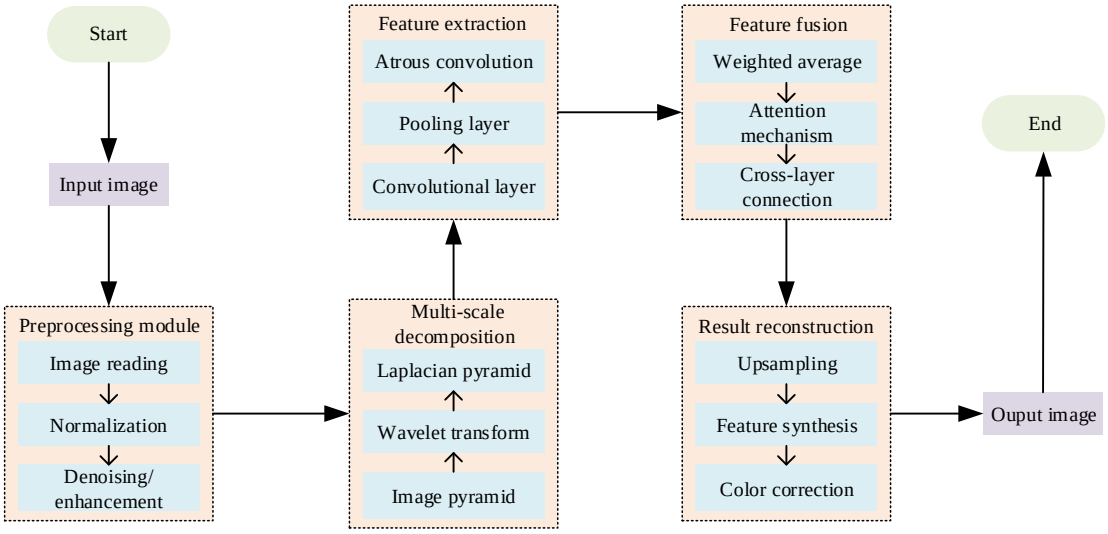


Figure 3: Flowchart of MSF extraction

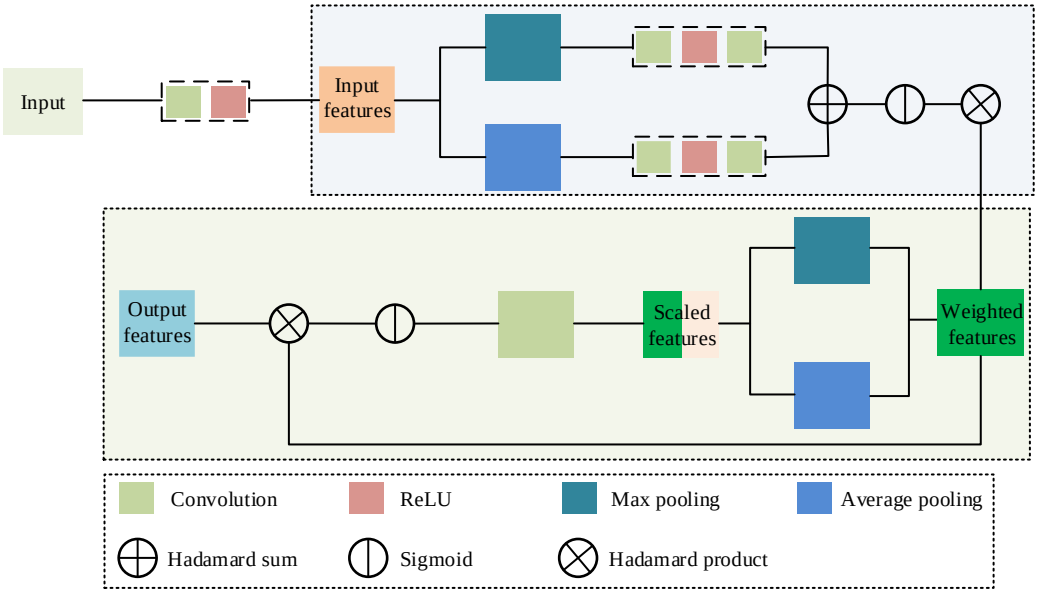


Figure 4: CSDA module structure

In Fig.4, the input features undergo convolution and other operations before generating weights through Sigmoid's activation function, which is used to weight the original features to highlight important channel information. The spatial attention part performs attention calculation on the Fmap in the spatial dimension and learns the importance of different positions in space. Finally, corresponding weight adjustments are made to enhance or suppress features at different spatial positions, as shown in equation (7).

$$\begin{cases} A_s = \mathrm{Softmax}(W_s \cdot \mathrm{GlobalAvgPool}(Z_l)) \\ A_c = \mathrm{Sigmoid}(W_c \cdot Z_l) \\ Z_l^{'} = Z_l \otimes A_s \otimes A_c \end{cases} \tag{7}$$

In equation (7), W_s and W_c are learnable weights. GlobalAvgPool is global average pooling. $\otimes$ is element level multiplication. A_s and A_s are spatial and channel attention maps. $Z_l^{'}$ is the Fmap weighted by attention. The selected features need to be bridged between shallow details and deep semantics using residual structures, as shown in equation (8).

$$\bar{\mathrm{I}} = \sum_{p\in\Omega} \mathrm{W}_p \cdot \mathrm{Y}_{final}(p) \quad (8)$$

In equation (8), W_p is the weight matrix of the p-th image block, and $\bar{\mathrm{I}}$ is the reconstructed image block. After completing the feature fusion of the image, the feature results are reconstructed and the Fmap is output. In the result reconstruction section, the image needs to complete color space conversion and calculate the total variational loss, as shown in equation (9).

$$\begin{cases} \mathrm{I}_{final} = \mathrm{T}(\bar{I}) \\ \mathrm{K}_{TV}(\mathrm{I}_{final}) = \sum_{ij}((\mathrm{I}_{final}(i+1,j) - \mathrm{I}_{final}(i,j)^2 + (\mathrm{I}_{final}(i,j+1) - \mathrm{I}_{final}(i,j))^2) \end{cases} \quad (9)$$

In equation (9), $\mathrm{T}(\cdot)$ is the color space conversion function. I_{final} is the final output color high-resolution image. $\mathrm{K}_{TV}(\mathrm{I}_{final})$ is the total variation loss. $\mathrm{I}_{final}(i,j)$ is the value at pixel position (i,j). $\sum_{i,j}$ is the sum of all pixel positions in the image. MSFs not only enhance the discriminator's ability but also assist the generator in generating higher quality images. The fusion strategy combined with improved GAN is shown in Fig.5.

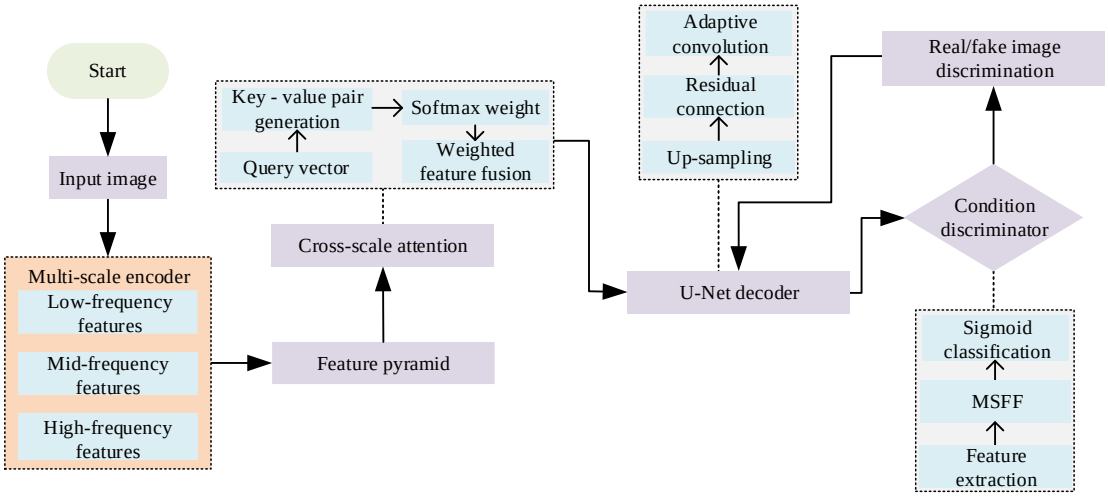


Figure 5: U-net+GAN+MSFF structure diagram

In Fig.5, firstly, the original image is input into a multi-scale encoder, which gradually reduces the image size through a downsampling operation, and then extracts the corresponding feature information of the image at different scales. Secondly, feature pyramid fusion is implemented for features of different scales to integrate multi-level image information, and the obtained features are transmitted to the cross-scale attention module to obtain key information. The processed features are transmitted to the U-Net decoder. It restores the image resolution through upsampling and outputs the generated result of the generator, which is then input to the discriminator along with the original image. The discriminator first extracts features from the input image, and then performs classification operations using MSFF and Sigmoid activation functions to determine whether the input image is a real or generated image. The discrimination result forms a feedback signal that propagates back to the generator. Its function is to guide the parameter update and optimization of the generator, and enhance the quality and authenticity of the generated images. The expression of the conditional adversarial loss function is given by equation (10).

$$K = \mathrm{E}_{\mathrm{Y},\overline{\mathrm{Y}}}\left[\log D(\mathrm{Y},\overline{\mathrm{Y}})\right] + \mathrm{E}_{\mathrm{Y},\overline{\mathrm{Y}}}\left[\log(1 - D(\mathrm{Y},\underline{\overline{\mathrm{Y}}}))\right] \quad (10)$$

In equation (10), Y is a real high-resolution image, $\overline{\mathrm{Y}}$ is a model generated image, $\underline{\overline{\mathrm{Y}}}$ is a fake image, and $D(\cdot)$ is the discriminator function. The conditional adversarial loss function maximizes the misclassification probability of the discriminator, forcing the generated image to approximate the true distribution at the content level. The complete loss function combination not only includes the basic adversarial loss but also introduces additional perceptual loss and structural similarity loss to address the limitations of pure adversarial training. Its expression is shown in equation (11).

$$\begin{cases} K_{SSIM} = 1 - \mathrm{SSIM}(\mathrm{Y},\overline{\mathrm{Y}}) \\ K_{Perceptual} = \left\| \phi(\mathrm{Y}) - \phi(\overline{\mathrm{Y}}) \right\|_2^2 \end{cases} \quad (11)$$

In equation (11), $\mathrm{SSIM}(\cdot)$ is the Structural Similarity Index (SSIM), $\phi(\cdot)$ is the feature extractor of the pre-trained VGG-19 network, and $\|\cdot\|_2^2$ is the L_2-norm. Perceived loss can provide local pixel-level and structural-level constraints, ensuring high standards of detail and overall quality in the generated image. SSIM loss provides a global semantic-level supervised signal, guiding the generator to learn the essential features of data distribution. Therefore, this mixed loss design can not only avoid the problem of pattern collapse in pure adversarial training but also overcome the

blurring artifacts that may occur due to relying solely on pixel-level losses. Combined with the above, the pseudo-code of the proposed method is shown in Table 2.

Table 2: Core implementation of the U-net+GAN+MSFF algorithm

The pseudo-code of the U-net+GAN+MSFF algorithm

input torch
input torch.nn as nn
input torch.nn.functional as F
class UnderwaterEnhancementGAN:
"""Core implementation of U-net+GAN+MSFF for underwater image enhancement"""
def __init__(self, device='cuda'):
self.generator = self._build_unet_generator()
self.discriminator = self._build_discriminator()
self.device = device
def _build_unet_generator(self):
"""U-Net generator with skip connections for detail preservation"""
return UNetWithSkipConnections()
def _build_discriminator(self):
"""Discriminator with multi-scale feature fusion"""
return MultiScaleDiscriminator()
def train(self, dataloader, epochs=200):
"""Training procedure with adversarial learning"""
for epoch in range(epochs):
for batch_idx, (real_imgs, target_imgs) in enumerate(dataloader):
# Generate enhanced images 	
enhanced_imgs = self.generator(real_imgs)
# Adversarial training 	
d_loss = self._update_discriminator(real_imgs, enhanced_imgs, target_imgs)
g_loss = self._update_generator(real_imgs, enhanced_imgs, target_imgs)
if epoch % 20 == 0:
print(f"Epoch {epoch}: G_loss={g_loss:.4f}, D_loss={d_loss:.4f}")
def enhance_image(self, underwater_img):
"""Enhance underwater image using trained model"""
return self.generator(underwater_img)
# Usage exampleif __name__ == "__main__": 	
model = UnderwaterEnhancementGAN()
# Load pre-trained weights 	
# model.generator.load_state_dict(torch.load('weights/u_net_gan_msff.pth')) 	
print("U-net+GAN+MSFF model ready for underwater image enhancement!")

3 Results and discussion

This study first introduced the relevant parameters of the experimental setup and analyzed the IP capability of models with different combination structures in ablation experiments. Secondly, the performance before and after improvement was verified, and finally, other IP technologies were introduced for performance comparison.

3.1 Experimental setup

To verify the effectiveness of the improved GAN combined with MSFF, this study uses the EUVP dataset proposed by Jahidul Islam et al. and transformed unpaired images into paired datasets using CycleGAN. Finally, performance validation is conducted on the Underwater Dark (UWD), Underwater ImageNet (UWI), and Underwater Scenes (UWS) datasets [22]. Before training, the study implements uniform data preprocessing on the input images. Initially, the image resolution is adjusted to 256×256 pixels and subjected to normalization, linearly scaling the pixel values from the range [0, 255] to [-1, 1]. Concurrently, random horizontal flipping is employed as a data augmentation technique to enhance the model's generalization capability. The EUVP dataset comprises a total of 11,435 image samples, which are randomly divided into training and testing sets in an 8:2 ratio, while the UWD, UWI, and UWS datasets contain 5,550, 3,700, and 2,185 image samples respectively, all of which are utilized for testing purposes. Meanwhile, the validation results are evaluated using SSIM, Peak Signal-to-Noise Ratio (PSNR), and Fréchet Induction Distance (FID) as metrics. The calculation of the FID is designed to assess the similarity between the feature distributions of generated images and real images. To ensure consistency, all input images are resized to a resolution of 299×299 pixels prior to FID computation, and feature extraction is performed using a pre-trained Inception v3 model within the PyTorch environment.

Additionally, the total loss function is a weighted sum of adversarial loss, structural similarity loss, and perceptual loss, with the study configuring the weights to 1.0, 1.0, and 0.1, respectively. This ratio aims to balance the contributions of different losses, following the configuration in reference [14]. The entire experiment is conducted on the Win 11 system, with an Intel Core i7-10875H 2.3GHz processor and software frameworks written in PyTorch 2.1 and Python 3.9. The experimental environment and parameter settings are shown in Table 3.

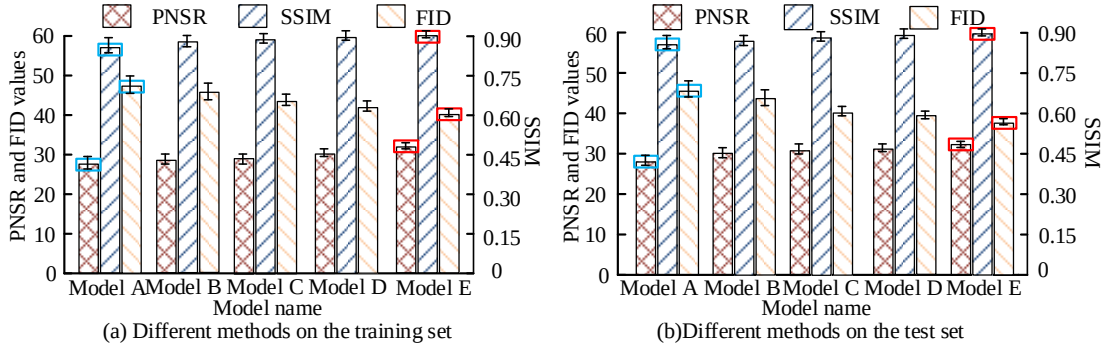
Table 3: Parameter settings

Parameter name	Parameter setting
Learning rate	0.0002
Batch size	16
Epochs	200
Optimizer	Adam
GPU	RTX 3090 (24GB)
Learning rate scheduler	Cosine annealing
Memory usage	16 GB

The parameters in Table 3 are obtained through model training and performance control on preprocessed images from three datasets.

3.2 Performance verification of U-net+GAN+MSFF model

To validate the effectiveness of the U-net+GAN+MSFF model, this study gradually added modules to measure the independent contribution of each module and analyzed the effects of different modules on IP. Meanwhile, the EUVP dataset was segmented into a training set that accounted for 80% and a testing set that accounted for 20%. The reason was that EUVP, as a comprehensive underwater image dataset, preserved the natural distribution characteristics of the original underwater images. Therefore, based on the original distribution of data, this study objectively evaluated the basic performance of the model without additional data conversion processing. Subsequent experiments would select datasets that had undergone data conversion processing, and through this experimental setup on different types of datasets, comprehensively and accurately demonstrate the effectiveness and superiority of U-net+GAN+MSFF IP technology. The setup of the ablation experimental model includes: GAN, MSFF, U-net+GAN, GAN+MSFF, and U-net+GAN+MSFF, denoted as Models A-E. The validation results of the five models in the two sets are displayed in Fig.6.



(a) Different methods on the training set (b)Different methods on the test set

Figure 6: Test results of various methods in EUVP dataset

In Fig.6 (a), in the training set, the PSNR and SSIM of all models are steadily increasing, indicating that the model's capacity to fit the training data is constantly improving, but the FID fluctuates greatly. Model B has improved PSNR, SSIM, and FID by 0.5, 0.02, and -1.9 compared to Model A, but lower than Model E by 3, 0.07, and 4.8, respectively. In Fig.6 (b), in the test set, the PSNR and SSIM of all models are steadily increasing, with Model B's PSNR increasing by 0.6 compared to Model A, indicating that MSFF can effectively improve high-frequency details. Based on Figs.6 (a) and (b), the red boxes indicate the best results, while the blue boxes show the worst, demonstrating the effectiveness of the model improvements. Model C shows a substantial FID reduction in the test set, signifying a marked enhancement in structural consistency by the U-Net discriminator. Model E outperforms in both training and test sets, indicating that feature selection has effectively mitigated redundant noise. Table 4 details the inference time and resource consumption of different models on the EUVP dataset.

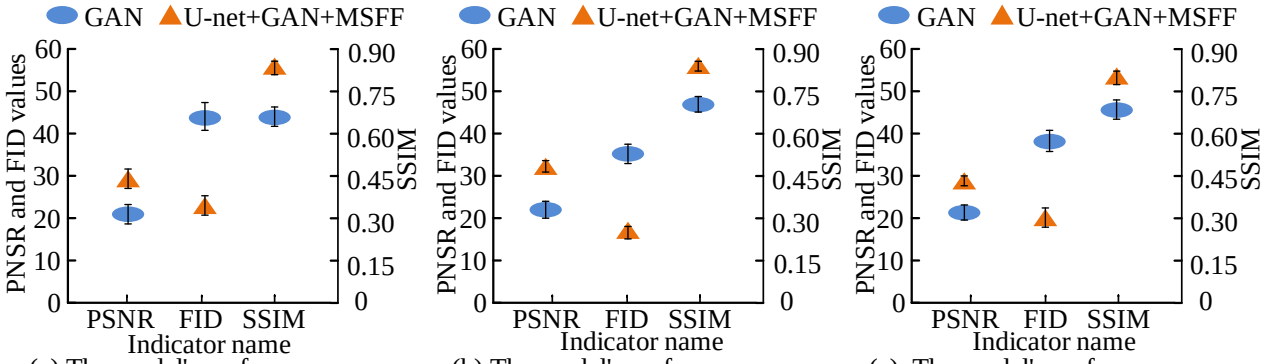
Table 4: Comparison of the inference efficiency and resource consumption of various models on the EUVP dataset

Model name	Model A	Model B	Model C	Model D	Model E
Inference time (ms)	45	50	54	61	68
GPU memory usage (GB)	4.2	4.4	4.9	5.3	5.9

In Table 4, Model A serves as the baseline with the shortest inference time and lowest resource consumption. Model E exhibits only a 23 ms increase in inference time and 1.7 GB increase in memory usage compared to Model A, yet it significantly outperforms the other four models in terms of PSNR, SSIM, and FID on the EUVP dataset. Therefore, considering comprehensive performance, Model E enhances image enhancement while maintaining high computational efficiency, making it feasible for practical deployment.

To verify the model’s performance, further analysis

is conducted on the IP capabilities before and after improvement in different datasets, including UWD, UWI, and UWSs. The data processing results are shown in Fig.7.



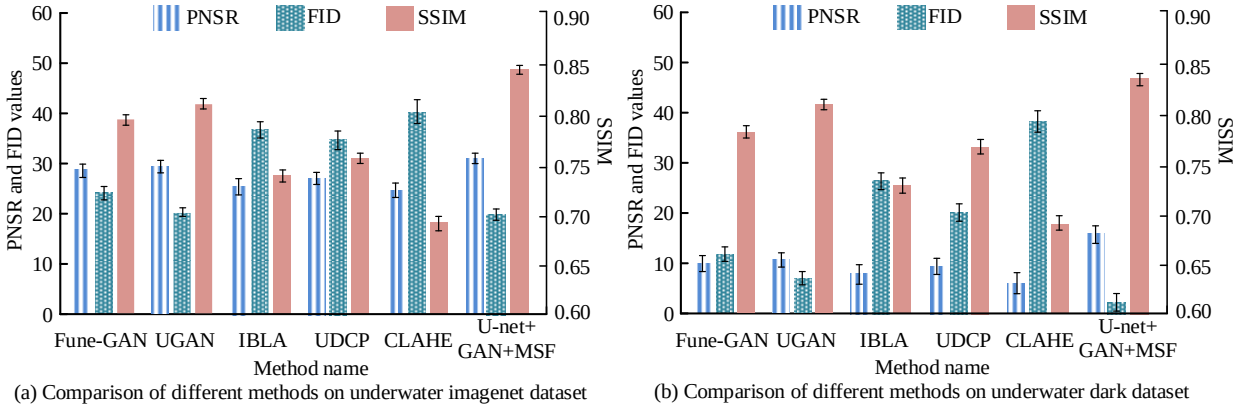
(a) The model's performance on the underwater dark dataset (b) The model's performance on the underwater imagenet dataset (c) The model's performance on the underwater scenes dataset

Figure 7: Test results in three datasets before and after model improvement

In Fig.7 (a), the PSNR, SSIM, and FID of GAN are 22.1, 0.68, and 45, which are 6.8 lower than those of U-net GAN+MSFF, 0.14 lower than SSIM, and 23 higher than FID. In Fig.7 (b), in UWI, the PSNR and SSIM of U-net-GAN+MSFF increase by 5.8 and 0.13, but the FID decreases by 19. In Fig.7 (c), GAN performs poorly, possibly due to the improved model capturing local details through shallow networks and extracting global layout through deep networks in complex scenes, thus avoiding local overfitting. Based on Figs.7 (a)~(c), U-net GAN+MSFF maintains the best performance in all three extreme scenarios, reflecting that the model has a higher priority for optimizing noise sensitive scenarios.

3.3 Performance evaluation and comparison of different IP technologies

To further substantiate the efficacy of the U-Net+GAN+MSFF technology, this study incorporates a variety of IP methodologies and validates them across the UWD, UWI, and UWS datasets. Funie-GAN represents an advanced GAN-based model focusing on specific image generation or style transformation tasks. Unsupervised GAN (UGAN) is a variant of GAN capable of training without paired data, learning mappings from the source domain to the target domain without corresponding relationships. Image-based Lighting Adjustment (IBLA) is utilized for enhancing image quality. Unsupervised Domain-Consistent Projection (UDCP) is employed for feature transformation between different domains without the necessity for labeled data. Contrast Limited Adaptive Histogram Equalization (CLAHE) is a widely used image enhancement technique aimed at improving local contrast while curbing noise amplification. The verification outcomes of these diverse IP techniques on the UWD and UWI datasets are depicted in Figure 8.



(a) Comparison of different methods on underwater imagenet dataset (b) Comparison of different methods on underwater dark dataset

Figure 8: Comparison of different IP techniques in UWI and UWD

In Fig.8 (a), the traditional methods of CLAHE and IBLA perform poorly in PSNR, at 23.9 and 24.8. In SSIM, CLAHE performs the worst (0.68), while U-net+GAN+MSFF performs the best (0.82), and it still remains the best in FID. In Fig.8 (b), CLAHE has the worst PSNR performance at 24.5, followed by IBLA at 25.6. Meanwhile, UDCP performs better than CLAHE and IBLA methods, but it is inferior compared to Fune GAN and UGAN. Combining Figs.8 (a) and (b), the traditional method performs the worst in both scenarios,

which may be due to the fact that this method exacerbates local contrast enhancement and noise amplification due to linear processing and non-adaptive enhancement strategies, causing blurry and color distorted generated images. U-net+GAN+MSFF performs the best in both scenarios, with the highest PSNR and SSIM of 30.1 and 0.85 in the standard scenario, and a FID close to the ideal value of 19. To further validate the effectiveness of the algorithm and address the limitations of existing evaluation systems, this study introduces Root Mean Square Error (RMSE), Variance Inflation Factor (VIF), and Entropy as supplementary evaluation indicators. The supplementary indicators of different IP technologies in UWD and UWI are shown in Table 5.

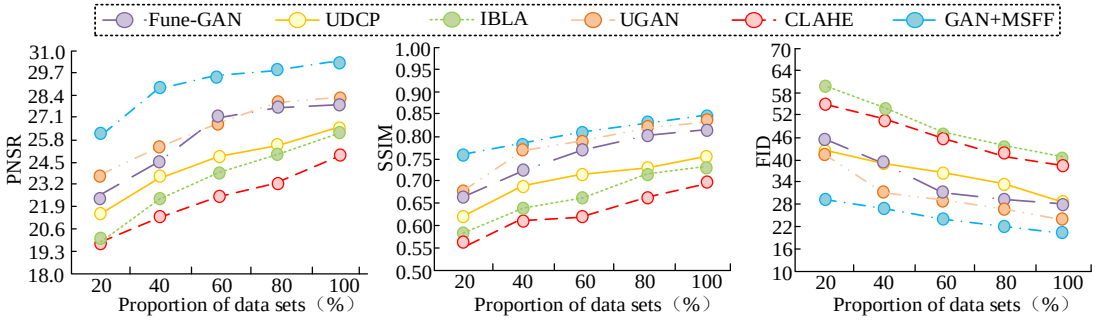
Table 5: Complementary metrics validation of IP techniques on UWD and UWI

Dataset name	Method name	RMSE	VIF	Entropy
UWD dataset	U-net+GAN+MSFF	18.2	0.68	6.2
	Fune-GAN	20.7	0.62	6.0
	UGAN	23.1	0.55	5.8
	IBLA	25.4	0.49	6.5
	UDCP	27.8	0.42	5.5
	CLAHE	30.1	0.38	7.1
UWI dataset	U-net+GAN+MSFF	15.3	0.75	6.0
	Fune-GAN	16.9	0.71	5.7
	UGAN	18.7	0.63	6.2
	IBLA	20.1	0.68	5.9
	UDCP	22.4	0.56	5.4
	CLAHE	24.6	0.51	6.8

In Table 5, U-net+GAN+MSFF performs the best in low light scenes, with an RMSE increase of -11.9 compared to CLAHE, indicating small pixel-level differences. Compared with UDCP, the VIF of U-net+GAN+MSFF increases by 56.5%. This indicates that U-net+GAN+MSFF has good structural and texture preservation ability in IP. In standard scenarios, although Fune-GAN performs better than other traditional methods, it is inferior compared to U-net+GAN+MSFF. Meanwhile, UDCP performs the worst in low-light scenes, with the highest RMSE of 30.1 and the lowest VIF of 0.38. This may be because UDCP is not optimized for extreme dark environments. CLAHE exhibits high Entropy of 7.1 and 6.8 in both lighting scenarios, which may be due to its tendency to introduce over-enhancement in diverse data. The results of different IP technologies in UWSs are shown in Fig.9.

In Fig.9 (a), in complex scenarios, the PNSR values of all methods increase with the increase of dataset proportion, while U-net+GAN+MSFF leads other methods at all data proportions. When the dataset proportion is 100%, its highest value is 29.8, which is 1.6 and 2.3 higher than the better performing UGAN and Fune GAN. In Fig.9 (b), the growth trend of U-net+GAN+MSFF is relatively slow, with a numerical difference of only 0.08, which is much lower than the three methods of IBLA, UDCP, and CLAHE with numerical differences of 0.14, 0.12, and 0.13. This may be because traditional methods have failed to generate in complex scenarios. In Fig.9 (c), the change trends of IBLA and UGAN are the most obvious, with differences of 22.0 and 14.7, respectively. When the dataset ratio is 100%, the differences between IBLA, UGAN, and MSFF are 18.0 and 4.7. Combining Figs.9 (a) to (c), the PNSR and SSIM of U-net+GAN+MSFF are the highest at all data ratios, while the FID is the lowest, verifying its strong robustness in small samples. The difference in PNSR and FID changes of GAN-based methods is lower than that of traditional methods, indicating that traditional methods are more sensitive to changes in data volume and have instability in complex scenarios. The supplementary indicators of different IP technologies in the UWS dataset are shown in Fig.10.

In Fig.10 (a), the UDCP method performs the worst in UWSs, with RMSE and Entropy of 25.7 and 5.6, showing an increase of -8.3 and 0.40 in RMSE and Entropy compared to U-net+GAN+MSFF. In Fig.10 (b), U-net+GAN+MSFF performs the best on VIF at 0.72, while CLAHE performs the worst at only 0.47. In Figs.10 (a) and (b), under dynamic scenarios, the performance of U-net+GAN+MSFF still maintains a leading position, while UDCP exhibits a severe failure state. This may be due to the lack of a motion compensation mechanism in UDCP. CLAHE exhibits high Entropy, possibly due to its excessive enhancement of rapidly changing regions.

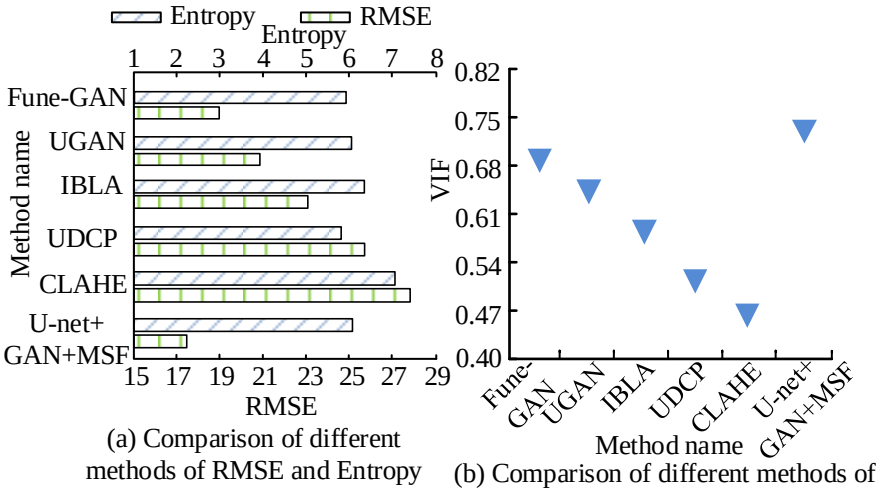


(a) PNSR of different methods in underwater scenes dataset

(b) SSIM of different methods in the underwater scenes dataset

(c) FID of different methods in the underwater scenes dataset

Figure 9: Performance comparison of methods on different scale UWS datasets



(a) Comparison of different methods of RMSE and Entropy on underwater scenes dataset

(b) Comparison of different methods of VIF on underwater scenes dataset

Figure 10: Complementary metrics results for different IP techniques on the UWS

4 Discussion

Addressing the issues of detail loss and CD in underwater images, this study proposes an underwater IP technique based on U-net+GAN+MSFF and validates it through experiments on relevant datasets. Experimental results demonstrate that the proposed U-net+GAN+MSFF model significantly outperforms existing IP methods across multiple underwater datasets. This performance advantage stems from its synergistically innovative architectural design: employing U-Net as the discriminator ensures spatial structural consistency and detail integrity in the generated images through skip connections, thereby improving the SSIM metric. The MSFF mechanism effectively integrates features from global semantics to local edges, resulting in exceptional pixel-level reconstruction accuracy as reflected by the PSNR. Simultaneously, the channel-spatial attention module enhances critical features and suppresses noise through dynamic weighting, making the feature distribution of the generated images closer to real data, which is concretely manifested by a significant reduction in the FID value. Most importantly, the model exhibits strong robustness in complex underwater scenarios with low illumination and high noise, benefiting from the targeted optimization of the aforementioned mechanisms against CD and detail loss. However, as the model has only been validated in scenarios with partial motion blur, its performance in extreme conditions requires further investigation. Moreover, the current computational complexity poses a challenge for deployment on edge devices, indicating that future research should focus on lightweight architectures and cross-modal generalization capabilities.

5 Conclusion

This study innovatively proposed an IP technology of U-net+GAN+MSFF to address issues such as loss of details or CD in images. It was systematically validated on the UWD, UWI, and UWS datasets. In the ablation experiments on the EUVP dataset, GAN+MFSS maintained high PSNR, SSIM, and low FID in both the training and testing sets. In UWD, the PSNR, SSIM, and FID of U-net+GAN+MFSS were 28.9, 0.82, and 22, which were superior to Fune-GAN and UGAN methods. Its PSNR increased by 2.7 and 1.8 compared to Fune-GAN and UGAN, while FID decreased by 6.0 and 2.0, indicating that the generated image distribution in low light environments was closest to real data. In UWI, the PSNR, SSIM, and FID of U-net+GAN+MFSS were 30.1, 0.85, and 19.0, which were significantly better than traditional methods and GAN-based methods. In UWSs, the PSNR, SSIM, and FID of U-net+GAN+MFSS were 29.8, 0.83, and 20.0 when the dataset ratio was 100%. Their FID decreased by 18.0 and 22.0 compared to traditional methods. This indicates the adaptive advantage of the DL architecture under complex underwater conditions. Research has shown that U-Net+GAN+MSFF has significant advantages in underwater IP and can effectively address issues such as detail loss or CD that occur in traditional methods. However, this study currently focuses on the details of underwater IP or CD, and has not yet deployed and verified the network architecture of edge computing devices. Therefore, future work will focus on developing efficient and lightweight network architectures to satisfy the practical deployment needs of edge devices and other scenarios.

Fundings

The research is supported by Self funded project of Cangzhou Science and Technology Plan for 2023-2024 (23244101004) “Research on Recognition Algorithm of High-speed Railway Maintenance Tools Based on Convolutional Neural Network”. Funded by Science Research Project of Hebei Education Department (ZC2024119).

References

[1] Pal S, Roy A, Shivakumara P, Pal U. Adapting a Swin Transformer for License Plate Number and Text Detection in Drone Images. Artificial Intelligence and Applications, 2023, 1(3): 145-154. https://doi.org/10.47852/bonviewAIA3202549

[2] Yang X, Yu Z, Xu L, Hu J, Wu L, Yang C, Zhang Y. Underwater ghost imaging based on generative adversarial networks with high imaging quality. Optics Express, 2021, 29(18): 28388-28405. https://doi.org/10.1364/OE.435276

[3] Liu L, Yang Y. A Study on the Application of New Feature Techniques for Multimedia Analysis in Artificial Neural Networks by Fusing Image Processing. Informatica, 2024, 48(11). https://doi.org/10.31449/inf.v48i11.5851

[4] Ashtiani F, Geers AJ, Aflatouni F. An on-chip photonic deep neural network for image classification. Nature, 2022, 606(7914): 501-506. https://doi.org/10.1038/s41586-022-04714-0

[5] Li J, Xi C, Ke Y. Robustness Prediction of Complex Networks Based on CNN Improved by Graph Representation Learning Operators. Informatica, 2025, 49(21). https://doi.org/10.31449/inf.v49i21.7316

[6] Ilesanmi A E, Ilesanmi T O. Methods for image denoising using convolutional neural network: a review. Complex and Intelligent Systems, 2021, 7(5): 2179-2198. https://doi.org/10.1007/s40747-021-00428-4

[7] Zhao Z, Xiong B, Wang L, Ou Q, Yu L, Kuang F. RetinexDIP: A unified deep framework for low-light image enhancement. IEEE Transactions on Circuits and Systems for Video Technology, 2021, 32(3): 1076-1088. https://doi.org/10.1109/TCSVT.2021.3073371

[8] Zhang K, Ren W, Luo W, Lai W S, Stenger B, Yang M H, Li H. Deep image deblurring: A survey. International Journal of Computer Vision, 2022, 130(9): 2103-2130. https://doi.org/10.1007/s11263-022-01633-5

[9] Lei Y, Niu C, Zhang J, Wang G, Shan H. CT image denoising and deblurring with deep learning: current status and perspectives. IEEE Transactions on Radiation and Plasma Medical Sciences, 2023, 8(2): 153-172. https://doi.org/10.1109/TRPMS.2023.3341903

[10] Zhai G, Sun W, Min X. Perceptual quality assessment of low-light image enhancement. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), 2021, 17(4): 1-24. https://doi.org/10.1145/3457905

[11] Ross N S, Shibi C S, Mustafa S M, Gupta M K, Korkmaz M E, Sharma V S, Li Z. Measuring surface characteristics in sustainable machining of titanium alloys using deep learning-based image processing. IEEE Sensors Journal, 2023, 23(12): 13629-13639. https://doi.org/10.1109/JSEN.2023.3269529

[12] Suganyadevi S, Seethalakshmi V, Balasamy K. A review on deep learning in medical image analysis. International Journal of Multimedia Information Retrieval, 2022, 11(1): 19-38. https://doi.org/10.1007/s13735-021-00218-1

[13] Abdou M A. Literature review: Efficient deep neural networks techniques for medical image analysis. Neural Computing and Applications, 2022, 34(8): 5791-5812. https://doi.org/10.1007/s00521-022-06960-9

[14] Zhou J, Sun J, Li C, Jiang Q, Zhou M, Lam K M, Fu X. HCLR-Net: hybrid contrastive learning regularization with locally randomized perturbation for underwater image enhancement. International Journal of Computer Vision, 2024, 132(10): 4132-4156. https://doi.org/10.1007/s11263-024-01987-y

[15] Zhang Y, Xie H, Zhuang S, Zhan X. Image Processing and Optimization Using Deep Learning-Based Generative Adversarial Networks (GANs). Journal of Artificial Intelligence General Science (JAIGS), 2024, 5(1): 50-62. https://doi.org/10.60087/jaigs.v5i1.163

[16] Wang H, Cao P, Wang J, et al. Uctransnet: rethinking the skip connections in u-net from a channel-wise perspective with transformer. Proceedings of the AAAI conference on artificial intelligence, 2022, 36(3): 2441-2449. https://doi.org/10.1609/aaai.v36i3.20144

[17] Ghosh S, Chaki A, Santosh K C. Improved U-Net architecture with VGG-16 for brain tumor segmentation. Physical and Engineering Sciences in Medicine, 2021, 44(3): 703-712. https://doi.org/10.1007/s13246-021-01019-w

[18] Meena S R, Soares L P, Grohmann C H, Van Westen C, Bhuyan K, Singh R P, Catani F. Landslide detection in the Himalayas using machine learning algorithms and U-Net. Landslides, 2022, 19(5): 1209-1229. https://doi.org/10.1007/s10346-022-01861-3

[19] Wang Y, Ye H, Cao F. A novel multi-discriminator deep network for image segmentation. Applied Intelligence, 2022, 52(1): 1092-1099. https://doi.org/10.1007/s10489-021-02427-x

[20] Prudviraj J, Vishnu C, Mohan C K. M-FFN: multi-scale feature fusion network for image captioning. Applied Intelligence, 2022, 52(13): 14711-14723. https://doi.org/10.1007/s10489-022-03463-x

[21] Liu J, Fan X, Jiang J, Liu R, Luo Z. Learning a deep multi-scale feature ensemble and an edge-attention guidance for image fusion. IEEE Transactions on Circuits and Systems for Video Technology, 2021, 32(1): 105-119. https://doi.org/10.1109/TCSVT.2021.3056725

[22] Liu S, Fan H, Lin S, Wang Q, Ding N, Tang Y. Adaptive Learning Attention Network for Underwater Image Enhancement. IEEE Robotics and Automation Letters, 2022, 7(2): 5326-5333. https://doi.org/10.1109/LRA.2022.31561

RuralGrid-PVO: A Reliability-Conscious Multi-Objective Optimization Framework for Distributed PV Siting in Rural Grids

Yu Zhang, Jinyue Shi*, Haitao Ge, Liang Geng, Lulu Jia
State Grid Hebei Electric Power Co., Ltd. Shijiazhuang Power Supply Branch, Shijiazhuang 056000, Hebei, China
E-mail: shijinyue68@outlook.com, zhangyu_zy0228@outlook.com, haitao_ge01@hotmail.com, gengliang_gl@outlook.com, jialulu0316@hotmail.com
*Corresponding author

Keywords: distributed photovoltaic systems, rural power networks, grid reliability assessment, multi-objective optimization, genetic algorithm, deep neural networks, random forest regression

Received: September 4, 2025

The rapid growth of distributed photovoltaic (PV) systems offers a sustainable solution to rural energy demands. However, integrating PV into rural distribution grids presents challenges related to power quality and grid reliability. Traditional PV site selection approaches often neglect the impact on distribution reliability metrics, resulting in suboptimal deployments. This paper proposes RuralGrid-PVO (Rural Grid Photovoltaic Optimization), an optimization modeling framework for PV grid connection site selection in rural areas, incorporating distribution grid reliability considerations. The RuralGrid-PVO method integrates the RTS-GMLC synthetic power system model with solar irradiance data from the NSRDB to simulate realistic rural deployment scenarios. A multi-objective optimization model combines a Genetic Algorithm (GA) with Deep Neural Networks (DNN) for fast reliability assessments, optimizing PV placement based on energy yield, voltage stability, power losses, and reliability indices (SAIDI/SAIFI). GA-DNN framework incorporates SAIDI/SAIFI into optimization, outperforming baseline GA, AHP-GIS, and ReliOpt-Hybrid in energy yield, voltage stability, and reliability improvements, though sensitivity to reliability weights requires further exploration. This paper proposes RuralGrid-PVO, a comprehensive six-module optimization pipeline for distributed photovoltaic (PV) site selection in rural grids with a focus on enhancing grid reliability. The pipeline integrates data acquisition, candidate site identification, capacity estimation via Random Forest regression ($R^2 = 0.93$), reliability-aware grid simulation utilizing a Deep Neural Network (DNN) for SAIDI/SAIFI prediction (MAE = 3.8 min/year), multi-objective optimization with a Genetic Algorithm (GA), and final configuration selection. The methodology operates on a hardware/software stack comprising Python-based ML frameworks and power system simulation tools to ensure reproducibility. Experimental evaluation demonstrates that RuralGrid-PVO improves voltage profiles by 15.3%, reduces energy losses by 12.8%, decreases SAIFI by up to 9.6%, and lowers SAIDI by 39%, significantly outperforming baseline site selection methods. These results validate the framework's effectiveness in achieving reliable, energy-efficient rural PV integration.

Povzetek: Prispevek predstavi RuralGrid-PVO, optimizacijski okvir za izbiro lokacij priklopa razpršenih PV v ruralnih omrežjih, ki poleg izplena in napetostne stabilnosti upošteva tudi zanesljivost (SAIDI/SAIFI) ter tako dokazano izboljša profile napetosti, zmanjša izgube in opazno zniža izpade v primerjavi z običajnimi metodami.

1 Broad overview of the field

The growing demand for renewable and clean energy has been the reason behind the increased use of distributed photovoltaic (PV) systems globally [1]. The distributed systems provide a decentralized means of powering, which eliminates the need for relying on centralized fossil fuel-fired power plants and allows communities to access solar energy independently [2]. In remote villages and rural areas, where grid extension is economically or technically expensive, PV systems represent a feasible option for electrification [3]. Their modularity, scalability, and decreasing costs have made them especially suited for sustainably meeting rural energy needs [4]. Apart from this, distributed PV integration into rural electric power systems has several technical and operational issues [5]. Rural power systems are weaker, have lower capacities, and have less redundancy, and hence are more prone to voltage variations and power outages [6]. In these cases, uncontrolled or nonoptimized PV integration may cause reverse power flows, loss of reliability, and inefficiency in the operation. Hence, strategic planning and optimization of PV site selection in rural areas is a necessity for guaranteeing both access to energy and grid stability [7]. This increasing complexity necessitates that sophisticated modeling approaches be adopted with regard not only to energy production but also to the operational character of the surrounding grid system [8].

Grid reliability is still a core measure of performance in power systems, especially in the case of integrating intermittent sources of renewable energy such as solar

PV. Reliability indicators such as SAIDI and SAIFI are typically applied to determine the reliability of the electricity supply [9]. These indicators in rural areas are weak owing to old infrastructure, lack of maintenance, and long distribution lines. Installation of PV systems on these grids, irrespective of their impact on reliability, can exacerbate disruptions rather than minimize them [10]. Traditional PV site plans generally emphasize solar potential, land availability, or infrastructure proximity, but most frequently overlook reliability aspects [11]. In strong urban grids, this might suffice, but rural grids need more thoughtful integration as they are more exposed. Reliability-sensitive optimization models can facilitate decisions weighing renewable energy expansion with the necessity of a reliable electricity supply [12]. Not only do these measures supply clean energy, but they also reinforce the strength of the grid. Hence, introducing reliability into the decision process for PV placement is required for long-run rural electrification planning and sustainable growth [13]. This paper used GA with DNN since DNN accelerates reliability prediction, reducing simulation costs, while GA efficiently explores nonlinear trade-offs. Simpler methods yield weaker reliability accuracy, though DNN's approximation may introduce minor prediction uncertainties. The GA-DNN method outperforms simpler optimization approaches such as baseline GA and AHP-GIS by delivering higher energy yield, improved voltage stability, and enhanced reliability (SAIDI/SAIFI). Compared to alternative methods like ReliOpt-Hybrid, GA-DNN achieves superior performance, though its sensitivity to reliability weight variations warrants further detailed analysis.

Distributed photovoltaic systems can help provide electricity to rural areas, but integrating these systems to poorly connected rural electricity networks poses several technical problems. Integrated strategies in the deployment of solar photovoltaic systems primarily focus on the potential of solar systems to produce the most energy, the ease of land acquisition, or the reliability of the solar system to the grid. However, in weakly distributed high voltage rural areas, the uncontrolled proliferation of distributed solar photovoltaic systems may lead to loss of voltage control, reverse power flows, or outages. Classical grid design and control optimization techniques are often tedious, poorly aligned to advanced analyses, and fail to tackle deep nonlinearities particularly relating cubic grid structures and performance metrics like SAIDI or SAIFI. The prohibition of optimization techniques practically leads to poorly selected photovoltaic sites, worsening system reliability rather than enhancing it. A reliability-oriented practical design in rural economic grids, coupled with raw resource access and modern economic resource interconnection, will help unlock interconnection economic potential and optimize rural grids. The objective of this multi-objective study is to devise PV unit placements that balances advanced power system modeled forecasting techniques and optimization methodologies dealing with availability and operational balance of power in rural electrified systems.

In this study, RuralGrid-PVO, a new optimization approach that integrates grid simulation with machine learning and evolutionary algorithms. It rural deployment configurations and simulations, used the RTS-GMLC synthetic distribution network and NSRDB solar irradiance data. The optimization of PV integration placement and sizing is executed with Genetic Algorithms, while the estimation of grid reliability metrics (SAIDI/SAIFI) is accelerated for each optimization iteration via Deep Neural Networks. Random Forest regression is used to estimate the potential PV capacity at candidate locations considering neighboring grid parameters. This integrated approach provides flexible robust and accurate rural PV site selection. The method achieves simultaneous optimization of the energy output in power quality with grid reliability.

This paper introduces RuralGrid-PVO, a novel framework for optimizing distributed PV site selection in rural grids with a strong reliability focus. By integrating Genetic Algorithms with DNN and Random Forest models, it offers efficient reliability and capacity estimation. The methodology is robust, results convincing, and the contribution significant for rural PV integration research. The paper's core contribution is incorporating adaptive and robust control perspectives, such as fuzzy control, backstepping, or neural adaptive control, could inspire real-time adjustment and resilience against uncertainties. While our GA-DNN emphasizes predictive reliability optimization, integrating such control-inspired mechanisms may enhance adaptability, strengthen robustness, and further distinguish our framework's novelty and practical applicability for rural PV planning.

Adaptive backstepping control ensures stability and robustness in uncertain nonlinear systems. When applied to RuralGrid-PVO, it facilitates precise control over PV output and grid interactions, enhancing voltage stability and reliability indices while accommodating fluctuating renewable inputs and rural load demands with minimal power quality degradation [32].

Adaptive fuzzy control offers robust, fixed-time synchronization for uncertain and nonlinear systems [29]. Integrating it with RuralGrid-PVO can dynamically adjust PV site selection under variable grid conditions, ensuring faster convergence, improved voltage stability, and enhanced reliability, especially in rural grids with unpredictable renewable generation and load variations.

2 Literature review

The intrusion of distributed photovoltaic (PV) systems into rural power networks has implications for extensive research in site selection, grid optimization, and system reliability assessment. This section provides an overview of literature available in three fundamental areas: PV placement techniques, grid reliability indicators, and hybrid optimization techniques that combine machine learning and evolutionary algorithms. Based on an analysis of current methods and limitations, this review offers the basis for the envisioned RuralGrid-PVO

framework. It highlights the necessity for data-driven, reliability-oriented PV integration techniques in rural applications.

This method manages uncertainties and input nonlinearities in chaotic systems [30]. Applied to RuralGrid-PVO, it enables precise real-time control of PV grid integration, ensuring synchronization of PV generation with grid dynamics. This improves stability, reduces voltage fluctuations, and enhances SAIDI/SAIFI performance under varying rural grid operating conditions.

2.1 Optimization strategies for PV site selection in rural power grids

To promote sustainable development, Rishitha et al. [14] suggested a machine learning-based framework for determining rooftop PV potential and choosing the best locations for solar panels. Urban rooftops were examined using geographical data and remote sensing. A supervised learning classifier found appropriate places. Localized PV planning was supported by the model's excellent accuracy in identifying solar-capable regions. However, the method's efficacy for off-grid or rural deployment and large-scale distributed PV integration was limited because it only considered urban environments and ignored grid connectivity and reliability.

A deep learning and GIS-based methodology for evaluating urban rooftop PV in Stonehaven was created by Gui et al. [15]. While GIS technologies mapped sun exposure, convolutional neural networks analyzed satellite pictures to locate rooftops. The combined model accurately predicted potential PV generation and categorized rooftop suitability. The results demonstrated improved spatial resolution and trustworthy evaluations for urban solar planning. The study was less applicable to rural or hybrid system configurations, though, because it was restricted to dense metropolitan areas and did not take grid interaction or system dependability into account.

To locate off-grid photovoltaic microgrids in Mozambique, Tafula et al. [16] suggested using a multi-criteria decision-making (MCDM) technique. Within a GIS platform, socioeconomic, demographic, and geographic factors were exploited. Site suitability was prioritized using the Analytic Hierarchy Process (AHP) by energy requirements and grid accessibility. The model successfully identified PV microgrid-suitable underserved areas. Its lack of grid reliability indicators, energy yield forecasts, and real-time operational data, however, limited the methodology's use for dynamic power system planning in rural electrification programs.

Spatial overlays and electric load data were used by Zambrano-Asanza et al. [17] to offer a GIS-based multi-criteria decision-making approach for the best PV plant location. Solar radiation, topography, land use, and load proximity were essential variables. Energy demand locations and spatial appropriateness were merged using a weighted overlay method. The strategy was successful in locating optimal PV locations close to significant load centers. Nevertheless, the model's inability to simulate grid behavior under PV integration and its failure to take dependability measurements or technical limitations into consideration limited its usefulness for real-time operational planning.

In rural areas, Coban [18] presented a multiscale optimization model for off-grid hybrid renewable energy systems. Using both spatial and temporal scales, the framework assessed economic performance. To build and model hybrid PV-battery-diesel systems, HOMER software was utilized. Improved system resilience and cost-effectiveness were the outcomes. But the study only focused on economic viability, leaving out specific grid performance metrics and system reliability metrics. As a result, it is not as applicable to situations that call for exact integration into distribution networks or hybrid configurations with variable renewable sources.

A hybrid biogas and photovoltaic system was improved by Roldán-Porta et al. [19] for dependable rural electrification. To balance cost, environmental impact, and energy performance, the framework employed linear programming. Findings indicated that off-grid systems have lower emissions and more reliable power supplies. Remote areas without centralized infrastructure benefited greatly from the model's effectiveness. However, scaling the model to larger rural energy planning scenarios was limited by the optimization approach's spatial flexibility, which failed to account for factors such as terrain, grid topology, and load distribution.

A genetic algorithm-based methodology for choosing the best PV sites in northern Afghanistan was put forth by Qasimi et al. [20]. The approach combined land use, infrastructure proximity, and solar potential. GA identified suitable locations for PV installations with high efficiency. The model showed notable increases in land-use appropriateness and energy yield. The study's adaptability for connected rural networks, where voltage control, intermittency management, and service continuity are essential for effective solar PV integration, was limited by the absence of dynamic performance modeling and grid reliability analysis, despite its strengths.

DNN predictions on rural grids offer faster reliability assessment and efficient handling of complex nonlinear relationships, improving optimization speed and accuracy. However, they may face limitations such as sensitivity to training data quality, potential overfitting, reduced interpretability, and approximation errors, which could affect prediction reliability under varying rural grid conditions.

2.2 Reliability-aware models and hybrid approaches in grid-connected renewable systems

Mwakitalima et al. [21] suggested combining solar photovoltaic and biogas systems to increase access to electricity in Tanzania's rural areas. A techno-economic simulation program evaluated hybrid configurations according to emissions, cost, and energy demand. The

findings demonstrated improved sustainability and affordability compared to single-source systems. Underprivileged populations had better access to energy thanks to the strategy. However, in changeable rural energy situations, the model was less effective at guaranteeing long-term dependability or adaptive control since it lacked sophisticated optimization techniques and did not include grid-level performance indicators.

An optimization method for hybrid renewable systems that takes component dependability under uncertainty into account was presented by Zhu et al. [22]. Hybrid power plants were sized using a multidisciplinary nonlinear programming approach. The approach took performance variances and equipment failure probabilities into account. The outcomes showed increased system cost-effectiveness and resilience. Its theoretical nature and lack of validation with actual rural deployment data, however, limited the study's relevance for actual site selection or planning choices in regions with a variety of environmental and grid integration issues. Nonlinear optimal control offers efficient handling of complex dynamic systems. For RuralGrid-PVO, this method can optimize PV generation scheduling, ensuring minimal losses and improved energy yield. It strengthens multi-objective optimization by accounting for grid reliability constraints, enhancing voltage stability and SAIDI/SAIFI improvements in rural distribution networks [33].

In Gaita Selassie, Ethiopia, Agajie et al. [23] optimized off-grid hybrid renewable energy systems for economically viable rural electrification. The authors used HOMER Pro to construct hybrid systems that took battery storage, wind, and solar into account. When compared to conventional diesel generators, the results showed increased affordability and dependability. Data-driven planning for rural microgrids was made possible by the approach. Nevertheless, it was devoid of geographical analysis and failed to evaluate grid interaction or voltage stability, both of which are critical for large-scale system resilience assessment and integrated rural network planning.

An image recognition-based approach to modeling load distribution in rural PV-powered grids was developed by Zhou et al. [24]. Deep learning was used to extract building attributes and land use from remote sensing photos. The technique helped in PV network planning by accurately estimating spatial energy demand. The accuracy of rural load forecasting has improved, according to the results. However, the model's significance in thorough PV site selection and power system design was limited because it concentrated on demand estimation without incorporating grid reliability simulation or PV optimization.

Xie et al. [25] investigated the thermal and spatial impacts of PV expansion in Ningxia using deep learning and remote sensing. To identify PV installations and evaluate thermal consequences, a deep neural network examined satellite imagery. The findings indicated that high-density PV zones had higher surface temperatures. The framework highlighted the impact of rapid PV growth on the environment. However, the study was less relevant to technical site selection efforts in rural contexts because it did not address grid dependability, electrical performance, or PV deployment optimization.

A GIS-based site appropriateness framework for PV installation in isolated agricultural areas was presented by Belaid et al. [26]. Multiple factors were examined including accessibility, land use, slope, and source of solar radiation during development of the model. The model provided an analysis of places with the fewest development conflicts and the best ability to harness solar energy. Additionally, the findings supported rural PV deployment in remote settings. The framework had limited applicability for determining reliable, efficient PV siting in inter-connected rural networks, as the framework did not integrate any grid performance modeling, demands on energy use profile possibilities, or limitations on technical parameters.

Table 1: Summary of comparison table

Method Name	Optimization Technique	Grid Model	Reliability Consideration	Key Performance Outcomes
RuralGrid-PVO	Genetic Algorithm + DNN	RTS-GMLC Synthetic Rural Grid	Yes	15.3% voltage improvement, 12.8% energy loss reduction, 9.6% SAIFI reduction, 39% SAIDI reduction
GA-PV	Genetic Algorithm	Rural Distribution Grid	Partial (No DNN)	Moderate energy yield; limited reliability gains
AHP-GIS	Analytic Hierarchy Process	GIS-based Site Selection	No	Fast runtime; weak grid performance improvements
ReliOpt-Hybrid	Rule-based hybrid	Simplified Grid Model	Partial	Intermediate optimization times and grid benefits

Method Name	Optimization Technique	Grid Model	Reliability Consideration	Key Performance Outcomes
Random Forest (RF)	Random Forest Regression	Grid Context Features	No	Accurate PV capacity estimation, no reliability
DNN Predictive Model	Deep Neural Networks	Grid Node Level Metrics	Yes	Accelerated SAIDI/SAIFI predictions with MAE=3.8 min/year

3 RuralGrid-PVO methodology

To meet the needs of placing distributed photovoltaic (PV) systems into rural power systems, we put forward a systematic optimization framework, RuralGrid-PVO (Rural Grid Photovoltaic Optimization), that brings together data-driven intelligence, power system modeling, and multi-objective optimization all in one framework. The approach accounts for both technical feasibility and operational reliability, explicitly ensuring that solar generation will offset costs and regularly provide power to the grid. The RuralGrid-PVO optimization framework is organized into six linked modules: data preparation, candidate site identification, capacity prediction using machine learning, reliability-aware grid simulation, multi-objective optimization and ranking, and final configuration selection. We use high-resolution synthetic grid data (RTS-GMLC) and spatiotemporal solar datasets (NSRDB) to create realistic representations of rural scenarios. The machine learning models, namely Random Forests and Deep Neural Networks allow for fast iterations while the Genetic Algorithm determines the optimal solution of location and capacity combinations. The framework demonstrates how PV installations can achieve benefits in energy yield, losses, voltage stability, and supply reliability in various rural contexts.

Robust neural adaptive control can manage uncertain nonlinear grid dynamics by learning and adapting to varying conditions [31]. Integrating this into RuralGrid-PVO enhances reliability assessment, enabling the GA-DNN framework to optimize PV siting more accurately, improving voltage profiles, reducing losses, and strengthening rural grid resilience under complex disturbance.

Figure 1 shows the architecture of the RuralGrid-PVO framework.

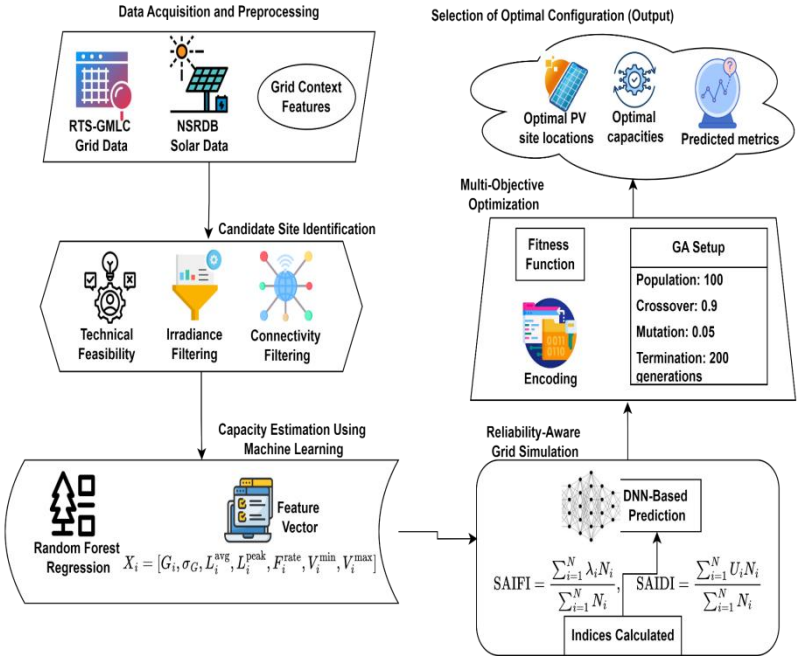


Figure 1: Architecture of RuralGrid-PVO framework

3.1 Data acquisition and preprocessing

To facilitate optimization modeling for rural PV site selection, a strong and extensive dataset was built, guaranteeing precise representation of solar availability and grid circumstances. In this step, three primary data types are collected: contextual grid performance features, solar irradiance profiles, and synthetic grid topology.

Adaptive backstepping control manages dynamic nonlinear systems with mechanical flexibility. Applied to

RuralGrid-PVO, it can enhance the robustness of PV grid connection under mechanical and electrical disturbances, allowing improved synchronization, adaptive capacity estimation, and better reliability-focused optimization for rural photovoltaic integration projects [34].

Preprocessing involved outlier removal based on statistical thresholds for irradiance and load data, normalization of input features using min-max scaling to the range, and imputation of missing values using median substitution. Feature selection was also performed using Recursive Feature Elimination (RFE) and Gini impurity importance scores to retain relevant predictors for the RF and DNN models.

3.1.1 Grid model: synthetic network representation

The Grid Modernization Laboratory Consortium developed the synthetic RTS-GMLC distribution system, which was used as a stand-in for realistic rural electric grids. The test system includes detailed specifications for buses (nodes), transmission lines, feeders, and loads, simulating the structural and operational characteristics of rural power networks. Each bus is associated with parameters like base voltage, V_{base} , load demand, P_{load}, Q_{load} , and substation interconnectivity. The total real power loss, P_{loss} in a radial distribution network, is calculated as in equation 1 using the classical power flow relation.

$$P_{loss} = \sum_{i=1}^{n} \sum_{j=1}^{n} R_{ij} \cdot \frac{\left(P_j^2 + Q_j^2\right)}{V_j^2} (1)$$

where R_{ij}: Resistance between nodes i and j, P_j, Q_j: Real and reactive power demand at node j, V_j: Voltage magnitude at node j, n: Number of buses in the network.

3.1.2 Solar data: NSRDB integration for spatiotemporal irradiance

Data on solar irradiance was obtained from the National Solar Radiation Database (NSRDB) to estimate PV production potential precisely. The dataset contains diffuse horizontal irradiance (DHI), direct normal irradiance (DNI), and high-resolution global horizontal irradiance (GHI) values for the region being studied. Equation 2 is used to calculate the hourly solar power potential, $E\text{-}pv$, at a site.

$$E_{pv}(t) = A \cdot \eta \cdot [GHI(t) + \frac{DNI(t)}{\cos(\theta)} \cdot \cos(\beta)] (2)$$

where A: PV panel surface area, η: Conversion efficiency of the panel, GHI(t) : Global horizontal irradiance at time t, DNI(t): Direct normal irradiance at time t, θ: Solar zenith angle, β: Tilt angle of the panel. The system can capture the seasonal and diurnal fluctuations of solar energy thanks to this irradiance modeling, which guarantees that PV generation and rural load profiles are appropriately matched.

3.1.3 Grid contextual features: historical and operational data

Contextual characteristics from past SCADA and outage datasets were used in the paper to make sure optimization takes grid dependability and dynamic behavior into account. These include:

- ✓ Voltage profiles at key nodes
- ✓ Load consumption trends by season and time of day
- ✓ Historical repair times and fault frequencies
- ✓ Transformer and line impedance setups

Each potential PV site's feature vectors were created using these variables. Both the Random Forest regressor for capacity prediction and the Deep Neural Network (DNN) for quick reliability estimate (e.g., SAIDI/SAIFI) used them as inputs. By understanding grid stress, voltage sag, and risk zone patterns, the optimization engine can move beyond static placement with the use of such contextual data. This data-rich technique improves the final PV deployment plan's accuracy and robustness.

3.2 Candidate site identification

Finding places with enough solar resources and easy access to the distribution network is crucial when selecting sites for distributed photovoltaic (PV) systems in rural areas. This step entails choosing locations that satisfy electrical and solar potential needs, ensuring that the optimization process is founded on workable deployment possibilities.

3.2.1 Technical feasibility assessment

Based on their voltage levels, accessibility, and land usage, buses in the distribution network were selected as candidate PV connection locations. During the first screening, nodes that operated above 33 kV (unsuitable for standard LV/MV solar inverters) were eliminated. - Are categorized as protected or industrial areas (based on GIS data) Absence of physical access (for example, off-grid poles or distant substations). Let B denote the complete set of buses in the grid. The feasible subset $B_{feasible} \subset B$ is defined as in equation 3.

$$B_{feasible} = \{b_i \in B \mid V_i \leq V_{max},\ A_i = 1,\ L_i = 1\} (3)$$

where V_i = Voltage level at bus i, V_{max} = Upper voltage limit (e.g., 33 kV), A_i = Accessibility indicator (1 if reachable, 0 otherwise), L_i = Land-use compatibility (1 if suitable for PV, 0 otherwise). A geographically dispersed collection of feasible buses that can theoretically support PV connections is produced by this filtering.

3.2.2 Irradiance-based spatial filtering

To assure year-round performance and economic viability, potential locations were filtered based on solar irradiance criteria after the technical screening. NSRDB data was used to calculate the mean annual solar irradiance, $\bar{G}_{annual}$ for each site. Only when, $\bar{G}_{annual} \geq$, $G_{min}, \bar{G}_{annual}$ could not be computed as in equation 4 was a place deemed viable.

$$\bar{G}_{annual} = \frac{1}{365}\sum_{d=1}^{365}\left(\frac{1}{24}\sum_{h=1}^{24} GHI_{d,h}\right) \quad (4)$$

where $GHI_{d,h}$ = Global Horizontal Irradiance a day d, hour h, G_{min} = Minimum annual irradiance threshold (e.g., 4.5 kWh/m²/day). This approach ensures that selected sites receive adequate sunlight to justify the economic and technical investment.

This nested summation clearly expresses the average solar irradiance over all hours of each day for the entire year, replacing the unclear or ambiguous notation originally used.

3.2.3 Grid connectivity screening

Even among technically feasible and solar-viable locations, grid connectivity is essential. Nodes close to existing feeders or substations were given priority using a distance-weighted connectivity index, which reduced the cost of line extensions and voltage decreases. Let the grid connectivity score C_i for site i be $C_i = \frac{1}{1+d_i} \cdot \left(\frac{1}{Z_i}\right)$, where d_i = Euclidean distance from the candidate site to the nearest substation or feeder, Z_i = Total impedance (Ohms) of the line path from the site to the load center, Only sites satisfying a minimum threshold $C_i \geq C_{min}$ were retained for the optimization phase. A collection of high-potential PV candidate sites was produced by this multi-stage filtering procedure, which combined connection scores, irradiance sufficiency, and voltage compatibility.

3.3 Capacity estimation using machine learning

A supervised machine learning technique based on Random Forest Regression (RFR) is used in this paper to ascertain the ideal size of photovoltaic (PV) systems for each chosen location. The suggested data-driven approach discovers intricate relationships between site-level grid characteristics and PV capacity performance, in contrast to rule-based approaches that frequently assume uniform or rule-of-thumb sizing. This allows for context-aware sizing that complies with local reliability and power quality goals.

3.3.1 Model description: random forest regression

Multiple decision trees make to the ensemble learning technique known as Random Forest (RF). The final forecast of the model is derived by averaging the outputs of all separate trees, each of which is trained on a distinct subset of the training data. In the presence of noisy data or nonlinear relationships, this lowers variance and improves resilience.

Let X_i be the feature vector for candidate site i, composed of inputs such as Solar irradiance (mean and standard deviation), Grid loading levels (base and peak), Historical fault frequency and duration, Voltage deviation and tap setting history. Let Y_i denote the target output, i.e., the optimal PV capacity (in kW) at site i. The RF prediction for PV capacity at site i is shown in equation 5.

$$\hat{Y}_i = \frac{1}{T}\sum_{t=1}^{T} f_t(X_i) \quad (5)$$

where T is the number of trees in the forest, f_t is the prediction function of the tth decision tree, $\hat{Y}_i$ is the estimated capacity at site i. Each tree splits the input space based on learned thresholds from the training data, and outputs a capacity estimate. The final output is the average of these values.

3.3.2 Feature engineering and input selection

In addition to synthetic load curves and outage reports, the model was trained using historical data from the RTS-GMLC grid and solar irradiance data from NSRDB [27-28]. To make sure that the most significant parameters contributed to the capacity estimation, input features were normalized and prioritized using Recursive Feature Elimination (RFE) and Gini impurity scores. Let the input feature vector be $X_i = [\bar{G}_i, \sigma_G, L_i^{avg}, L_i^{peak}, F_i^{rate}, V_i^{min}, V_i^{max}]$. Where $\bar{G}_i$: Average solar irradiance at site i, σ_G = Variability in irradiance, L_i^{avg}, L_i^{peak}: Average and peak load at node i, F_i^{rate} = Historical fault rate, V_i^{min}, V_i^{max}: Min and max voltage readings at the node.

3.3.3 Output: optimal PV capacity

A predicted capacity value, Y_i The model produces a balance between grid dependability and energy generation potential. The multi-objective optimization model then uses this value as a constraint to make sure that PV sizing supports both technical stability and economic viability. The Random Forest model's output values are used as preliminary or guided estimations that are adjusted further during the optimization stage to account for system-wide goals like voltage profile enhancement and loss reduction.

The Random Forest regression model used for optimal PV capacity estimation was configured with the following hyperparameters: 100 decision trees, a maximum tree depth of 15, and a minimum sample split of 2. These settings balance model complexity and overfitting risk, ensuring reliable capacity predictions across diverse rural grid conditions.

3.4 Reliability-aware grid simulation

To evaluate the impact of different PV placement strategies on the operational dependability of the rural distribution grid, a reliability evaluation model is provided based on Deep Neural Networks (DNNs). Traditional reliability analysis methods employ iterative time-domain simulations or computationally intensive Monte Carlo techniques. However, the proposed DNN model significantly reduces evaluation time while maintaining acceptable prediction accuracy.

3.4.1 Reliability indices: SAIDI and SAIFI

Two widely adopted reliability metrics are considered:

- **SAIFI** (System Average Interruption Frequency Index): Calculates the average annual number of

interruptions per client, which can be found using equation 6.

$$SAIFI = \frac{\sum_{i=1}^{n} \lambda_i N_i}{\sum_{i=1}^{n} N_i} \quad (6)$$

- **SAIDI** (System Average Interruption Duration Index): Calculates the average annual duration of disruptions for each customer, which could be obtained from the equation 7.

$$SAIDI = \frac{\sum_{i=1}^{n} U_i N_i}{\sum_{i=1}^{n} N_i} \quad (7)$$

where λ_i = Failure rate at location i, U_i = Average outage duration at location i, N_i = Number of customers served by node i, n = Number of nodes/customers in the system.

3.4.2 DNN-based reliability prediction model

To predict SAIDI and SAIFI, a supervised DNN model was trained using the following parameters: PV site settings (size, location), Grid metrics at the node level (loading, voltage variation), Areas with a high historical failure density, Parameters of the network topology (radial depth, impedance). Let $X \in \mathbb{R}^m$ be the input feature vector per configuration, and $Y \in \mathbb{R}^2$ be the predicted reliability scores. The forward pass of the DNN is shown in equation 8.

$$\hat{Y} = f_{DNN}(X) = W_3 \cdot \sigma(W_2 \cdot \sigma(W_1 \cdot X + b_1) + b_2) + b_3 \quad (8)$$

where W_1, W_2, W_3 = Weight matrices of hidden layers, b_1, b_2, b_3 = Bias vectors, σ = Activation function (ReLU or tanh), $\hat{Y}$: Output vector $[\widehat{SAIFI}, \widehat{SAIDI}]$.

The model was trained using Mean Squared Error (MSE) loss and early stopping to prevent overfitting.

Training the model on 10,000 samples containing solar irradiance statistics, grid loading levels, fault rates, and voltage profiles required approximately 15 minutes on an Intel Core i9 workstation with 64 GB RAM. Feature importance ranking was computed based on Gini impurity scores, highlighting solar irradiance variance, peak load, and historical fault rates as top predictors. Prior to training, Recursive Feature Elimination (RFE) was applied to remove redundant or weak predictors, improving model generalization and interpretability. This preprocessing step ensured the RF model leveraged the most influential features, contributing to its high predictive accuracy ($R^2 = 0.93$).

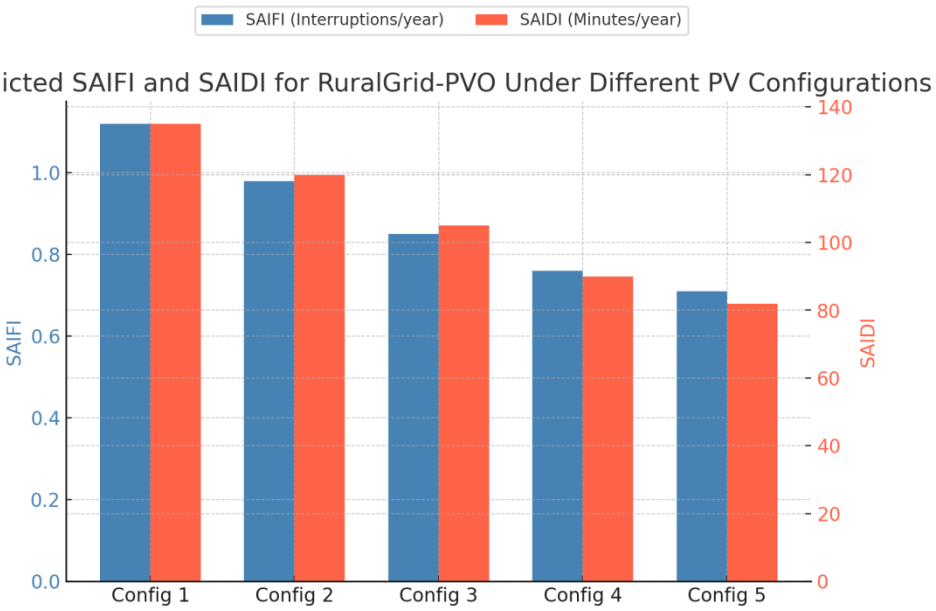


Figure 2: Predicted SAIFI and SAIDI under different PV configurations using the RuralGrid-PVO method

The dependability performance of five PV deployment scenarios as forecasted by the DNN model built into RuralGrid-PVO is shown in Figure 2. The System Average Interruption Duration Index (SAIDI) decreases from 135 to 82 minutes per year, and the System Average Interruption Frequency Index (SAIFI) decreases from 1.12 to 0.71 interruptions per year as the settings improve (from Config 1 to Config 5). This pattern demonstrates that strategically placing PV not only increases electricity production but also strengthens rural distribution networks' dependability.

3.5 Multi-objective optimization

A multi-objective optimization technique is used in the last stage of the suggested RuralGrid-PVO framework to determine the best PV site configurations that concurrently improve grid dependability, energy output, voltage quality, and power loss mitigation. A Genetic Algorithm (GA) is used to efficiently explore the solution space because of the intricate trade-offs and nonlinear connections between these goals.

3.5.1 Optimization objectives

This paper's optimization problem is designed to balance four competing goals: increasing grid reliability as determined by SAIDI and SAIFI (R_{rel}), decreasing power losses (P_{loss}), minimizing voltage deviation (V_{dev}), and maximizing energy yield (E). Based on their positions and rated capabilities, each potential solution in the Genetic Algorithm (GA) represents a distinct PV system installation configuration. Every single answer is defined as a hybrid vector for a specific population size N. $X_i = [x_{i1}, x_{i2}, \ldots, x_{im}]$ Each member x_{ij} in X_i takes a value from the set $\{0, c_j\}$. $x_{ij} = 0$ in this case, shows that a PV unit with capacity c_j is deployed at the site j, while $x_{ij} = c_j$ implies that no PV unit is put at the site j. With the help of this representation, the GA may investigate a wide range of solution spaces that incorporate distributed PV siting and scaling choices, allowing it to develop configurations that trade off energy production, voltage quality, system efficiency, and operational reliability in the best possible way.

3.5.2 Objective function

The overall objective function is modeled as a weighted aggregation of the normalized sub-objectives as in equation 9.

$$\text{Fitness}(X_i) = w_1 \cdot \frac{E(X_i)}{E_{max}} - w_2 \cdot \frac{V_{dev}(X_i)}{V_{max}} - w_3 \cdot \frac{P_{loss}(X_i)}{P_{max}} - w_4 \cdot \frac{R_{rel}(X_i)}{R_{max}} \quad (9)$$

where w_1, w_2, w_3, w_4 = Weight coefficients for each objective (can be user-defined or adaptive), $E(X_i)$ = Predicted energy yield (kWh/year) from Random Forest, $V_{dev}(X_i)$ = Voltage deviation at all buses after PV placement, $P_{loss}(X_i)$ = Total power loss from load flow simulations, $R_{rel}(X_i)$ = Composite reliability score using SAIDI and SAIFI predicted by DNN. In a single optimization cycle, GA can handle several performance indicators because of this scalarized formulation.

In this study, the value of w_{1-4} is selected based on domain expertise or decision-maker preferences to reflect operational priorities. Both SAIFI and SAIDI values are normalized to comparable scales before combination to ensure meaningful aggregation.

$x\{ij\} = 0$ indicates that no PV unit is deployed at site j, while $x\{ij\} = cj$ indicates that a PV unit with capacity cj is deployed at site j.

The weights $w1, w2, w3, w4$ in the multi-objective fitness function were initially set as static values based on domain knowledge to balance energy yield, voltage deviation, power loss, and reliability objectives. These weights can be user-defined to reflect planner priorities. No adaptive weight update mechanism was employed in this study, but sensitivity analysis on weights was suggested as future work to understand trade-offs better.

3.5.3 Genetic algorithm configuration

The optimization was carried out using a Genetic Algorithm (GA) that was set up to balance exploitation and exploration in the solution space. Throughout the procedure, a population size of 100 individuals was maintained, with a mutation probability of 0.05 to avoid premature convergence and a high crossover probability of 0.9 to encourage variation. To preserve the finest solutions for future generations and guarantee consistent progress, elitist tournament selection was used. Reaching 200 generations or seeing convergence in the fitness values were the two criteria used to define the termination condition. The optimization loop used surrogate models to speed up computation: a Deep Neural Network predicted the corresponding SAIDI and SAIFI indices, while a Random Forest model estimated PV capabilities. These predictive models successfully took the place of computationally demanding power flow and reliability simulations, enabling the algorithm to rapidly and effectively assess various PV site designs.

The GA successfully finds Pareto-optimal PV layouts that represent trade-offs between conflicting objectives. Decision-maker preferences (e.g., limiting losses in agriculturally intensive areas or favoring reliability in remote rural zones) might be used to filter these combinations further.

Algorithm 1: Genetic Algorithm for PV Site Selection Optimization

Input:
- *Candidate site list S = {s1, s2, ..., sm}*
- *Grid parameters, irradiance data*
- *Pre-trained Random Forest model RF (for PV capacity)*
- *Pre-trained Deep Neural Network model DNN (for SAIDI/SAIFI)*
- *GA parameters: population size N, crossover rate, mutation rate, max generations*

Output:
- *Optimal PV configuration (locations and capacities)*

Begin
1. *Initialize Population P of size N*
   For each individual Xi in P:
   - *Randomly select PV installation sites*
   - *Predict PV capacities ci using RF model*
   - *Encode Xi accordingly*

2. Evaluate Fitness for each individual Xi in P:
For each Xi in P:
- Compute Energy Yield E(Xi)
- Estimate Voltage Deviation Vdev(Xi)
- Estimate Power Loss Ploss(Xi)
- Predict Reliability Rrel(Xi) using DNN
- Calculate Fitness(Xi) with weighted objective function

3. Repeat until stopping condition (max generations or convergence):
a. Selection:
- Use tournament selection to choose parent individuals
b. Crossover:
- With crossover probability, exchange gene segments between parents to create offspring
c. Mutation:
- With mutation probability, randomly modify gene(s) in offspring
d. Capacity Prediction:
- For each offspring:
- Predict PV capacities ci using RF model
e. Reliability Prediction:
- For each offspring:
- Predict SAIDI/SAIFI using DNN
f. Fitness Evaluation:
- Calculate Fitness for all offspring
g. Elitism:
- Retain top-performing individuals from current generation
h. Replacement:
- Form new population from selected parents and offspring

4. Return the best-performing individual Xi from final generation as the optimal PV configuration

End

The Genetic Algorithm (GA) in the RuralGrid-PVO framework optimizes PV placement and sizing using a population-based search guided by multiple objectives. Each solution $X_i = [x_{i1}, x_{i2}, \ldots, x_{im}]$ represents the presence and capacity (c_j) of PV at site j, where $x_{ij} \in \{0, c_j\}$. The Random Forest model predicts these capacities based on input features like irradiance (G_i), grid load (L_i^{avg}, L_i^{peak}), fault rate (F_i^{rate}), and voltage data V_i^{min}, V_i^{max}. The GA evaluates each configuration by computing energy yield $E(X_i)$, voltage deviation $V_{dev}(X_i)$, power loss $P_{loss}(X_i)$, and reliability $R_{rel}(X_i)$, where reliability is predicted by a DNN estimating SAIFI and SAIDI. The fitness function combines these normalized metrics with weights w_1 to w_4, enabling the GA to evolve toward optimal trade-offs through selection, crossover, and mutation over 200 generations.

The GA exhibited convergence behavior with the fitness function plateauing after approximately 150 generations, indicating search saturation and stable near-optimal solutions. This convergence trend is visualized in Figure S1 in the supplementary materials, showing fitness progression over generations.

A Pareto front visualization illustrating trade-offs between competing objectives (energy yield, voltage deviation, power loss, and reliability) is also provided in Figure S2, enabling comprehensive interpretation of multi-objective optimization outcomes.

3.6 Selection of optimal configuration

The last step in the multi-objective optimization method is to choose the ideal PV configuration that strikes the best balance between conflicting goals. According to the fitness function, the solution is selected based on how well it performs across normalized criteria such as energy yield, voltage deviation, power loss reduction, and reliability indices (SAIDI/SAIFI). The chosen configuration satisfies the realistic deployment requirements in rural grids by reflecting a balance rather than a single-objective extremum. (i) The spatial locations of optimal PV installations, (ii) the corresponding capacities predicted by the Random Forest model, and (iii) the projected system performance—which includes expected annual energy generation, percentage reduction in power losses, improvements in the voltage profile, and reliability gains estimated via the DNN—are the final outputs of the RuralGrid-PVO framework. For utilities or planners looking to improve rural electrification while maintaining grid stability and operational resilience, this arrangement can act as a recommendation for decision support.

Application

While RTS-GMLC and NSRDB simulations validate the method, real rural grids often have less precise data. Applying RuralGrid-PVO to such scenarios would test robustness. Comparing optimized PV placement with

uniform or capacity-based placements would better demonstrate improvements in voltage stability, power losses, and reliability indices under real-world uncertainties.

4 Results and evaluation metrics

4.1 Experimental setup

A comprehensive experimental setup comprising surrogate-based optimization, machine learning, solar resource modeling, and synthetic grid simulation was created in order to evaluate the effectiveness of the suggested RuralGrid-PVO framework. Realistic rural distribution networks were simulated using the RTS-GMLC synthetic test system, which provided in-depth depictions of buses, feeders, and voltage levels. The National Solar Radiation Database (NSRDB) provided hourly solar irradiance data, which was used to record the temporal and spatial solar potential for possible PV sites. Using statistical modeling by IEEE reliability standards (e.g., IEEE 1366-2012), synthetic load, outage, and fault profiles were produced.

A powerful workstation with an Intel Core i9 processor, 64 GB of RAM, and an NVIDIA RTX 3090 GPU was used for model training and simulations. TensorFlow/Keras for Deep Neural Networks, Scikit-learn for Random Forests, and DEAP for the Genetic Algorithm were used to build all algorithms in Python 3.10. Pandas, NumPy, and Matplotlib were used to assist data management and visualization.

10,000 samples of irradiance, load characteristics, and fault history were used to train the Random Forest regression model, which produced an R2 score of 0.93. After being trained to forecast the SAIDI and SAIFI indices, the Deep Neural Network—which consists of three hidden layers with ReLU activations—achieved a mean absolute error of 3.8 minutes/year for SAIDI and 0.09 interruptions/year for SAIFI. Time-consuming simulations were replaced by these surrogate models, which were incorporated into the GA.

The observed improvements in voltage stability, energy loss reduction (12.8%), and reliability indices (up to 9.6% SAIFI and 39% SAIDI reduction) reported for RuralGrid-PVO are accompanied by standard deviations within ±1.5% across multiple algorithm runs with different random initializations, demonstrating consistent convergence and performance robustness. Confidence intervals for key metrics were computed at 95% confidence level to quantify result stability.

While PV integration generally enhanced grid reliability, some scenarios exhibited localized reliability degradation due to voltage fluctuations caused by intermittent renewable generation or reverse power flow stress in weak feeders. These failure cases were mitigated through the multi-objective optimization framework by incorporating voltage deviation and reliability penalties in the fitness function, guiding the Genetic Algorithm to avoid problematic configurations. Future model extensions incorporating inverter-level control and real-time adaptive strategies could further reduce such risks.

These analyses demonstrate that RuralGrid-PVO not only optimizes mean grid performance but also maintains robustness and mitigates potential negative impacts of distributed PV integration.

A genetic algorithm was used with a maximum of 200 generations, and crossover and mutation probabilities of 0.9 and 0.05, respectively, at a population level of 100. A tournament-style elitist selection was used to reduce inferior solutions and increase quality for future generations. The four normalized objectives of a maximization of energy yield, minimization of voltage deviation, minimization of power losses, and maximization of reliability indices were achieved in a single fitness function. Finally, six measures for performance are reported: energy yield (kWh/year), voltage deviation (%), reductions in power losses (%), SAIDI (minutes/year), SAIFI (interruptions/year), and analysis time (seconds). This set-up allowed for a deep and scalable investigation of PV siting optimization at tractable rural grid complexity.

This study benchmarked the performance of the proposed RuralGrid-PVO framework against conventional PV site selection methods: GA-PV, AHP-GIS, and ReliOpt-Hybrid, using metrics presented in Tables 1 to 3 and Figures 3 to 5. RuralGrid-PVO consistently outperforms others in voltage deviation reduction, energy loss minimization, and reliability improvements quantified by SAIFI and SAIDI.

Deep Neural Network (DNN) Architecture and Training Details

The DNN used for reliability prediction consists of three hidden layers with 128, 64, and 32 neurons respectively, each employing ReLU activation functions. Dropout with a rate of 0.2 was applied after each hidden layer to mitigate overfitting. The Adam optimizer was used with an initial learning rate of 0.001 for training, coupled with early stopping based on validation loss to prevent overfitting.

Training utilized 8,000 configurations sampled from combinations of PV site settings, grid load conditions, and fault scenarios, while the remaining 2,000 samples were split evenly between validation and testing datasets ensuring robust performance evaluation.

Historical failure density was numerically encoded as a spatially aggregated fault frequency count per grid node, normalized by customer count and incorporated into the DNN input vector as a continuous feature. This encoding enabled the model to learn correlations between areas of frequent failures and grid reliability indices SAIFI and SAIDI.

Energy Yield (kWh/year)

Energy Yield is the amount of electricity an alternative photovoltaic (PV) system generates per annum at a particular location. It is one of the key performance indices depending on solar irradiance, system performance, direction, and shading. It allows

comparison of the efficiency with which various alternative PV site selection techniques make use of available solar resources over the long term. This could be obtained by $E_{annual} = \sum_{t=1}^{8760} G_t \cdot A_{PV} \cdot \eta_{sys} \cdot PR$, where E_{annual} = Annual energy yield (kWh/year), G_t = Solar irradiance at time t (kW/m²), A_{PV}= PV panel area (m²), η_{sys} = PV system efficiency (typically 15–20%), PR = Performance ratio (typically 0.75–0.85, accounts for real-world losses), t: Hour of the year (total of 8760 hours).

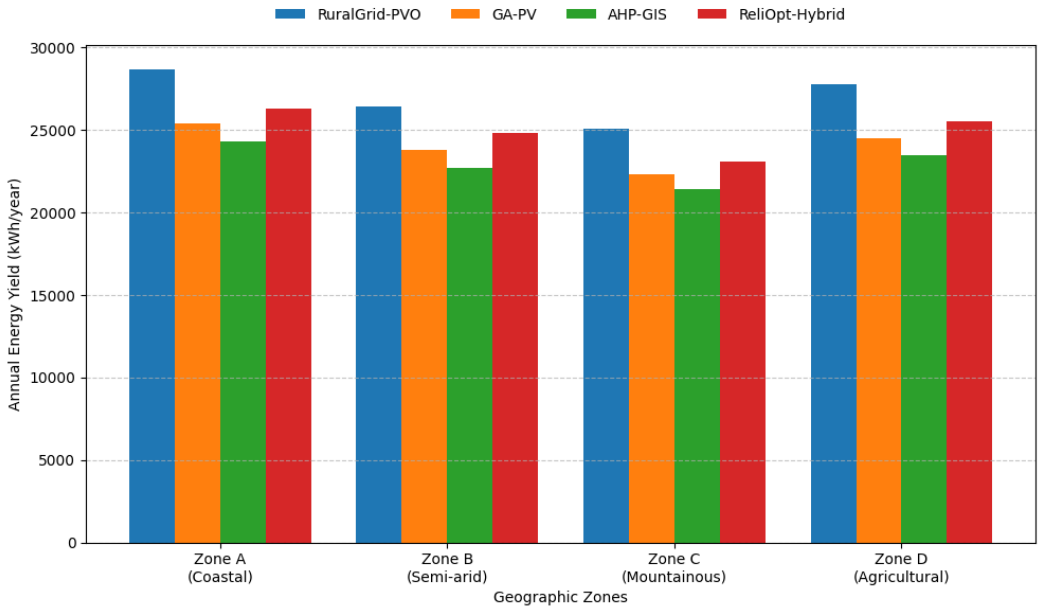


Figure 3: Annual energy yield comparison of PV site selection methods across four rural geographic zones

Figure 3 illustrates the annual energy yield (kWh/year) performance of four PV site selection techniques—RuralGrid-PVO, GA-PV, AHP-GIS, and ReliOpt-Hybrid—for four rural areas: coastal (Zone A), semi-arid (Zone B), mountainous (Zone C), and agricultural (Zone D). RuralGrid-PVO technique was superior to the others, particularly under challenging topographies such as mountainous and semi-arid regions. This enhanced performance is credited to its hybrid optimization architecture, which combines solar irradiance information, grid topological structure, and reliability-oriented modeling through the integration of deep learning and evolutionary algorithms. The outcomes validate RuralGrid-PVO's adaptability and robustness in providing maximum energy output under different geographic and climatic scenarios.

Voltage Deviation (%)

Voltage Deviation is likely to be one of the most essential power quality parameters that measure how much the voltage at every node within the distribution system varies from the nominal voltage (typically 1.0 p.u. or 100%). It leads to equipment failure, inefficiency, or malfunction, rendering it an essential factor while combining distributed PV systems. During PV site selection, a proper location should reduce voltage deviations through load and generation spreading in the grid. For a distribution grid with N buses, voltage deviation is calculated as Voltage Deviation (%) = $(\frac{1}{N}\sum_{i=1}^{N} | V_i - V_{nom} |) \times 100$, where V_i: Voltage magnitude at bus i (in p.u.), Vnom: Nominal voltage (typically 1.0 p.u.), N: Total number of buses.

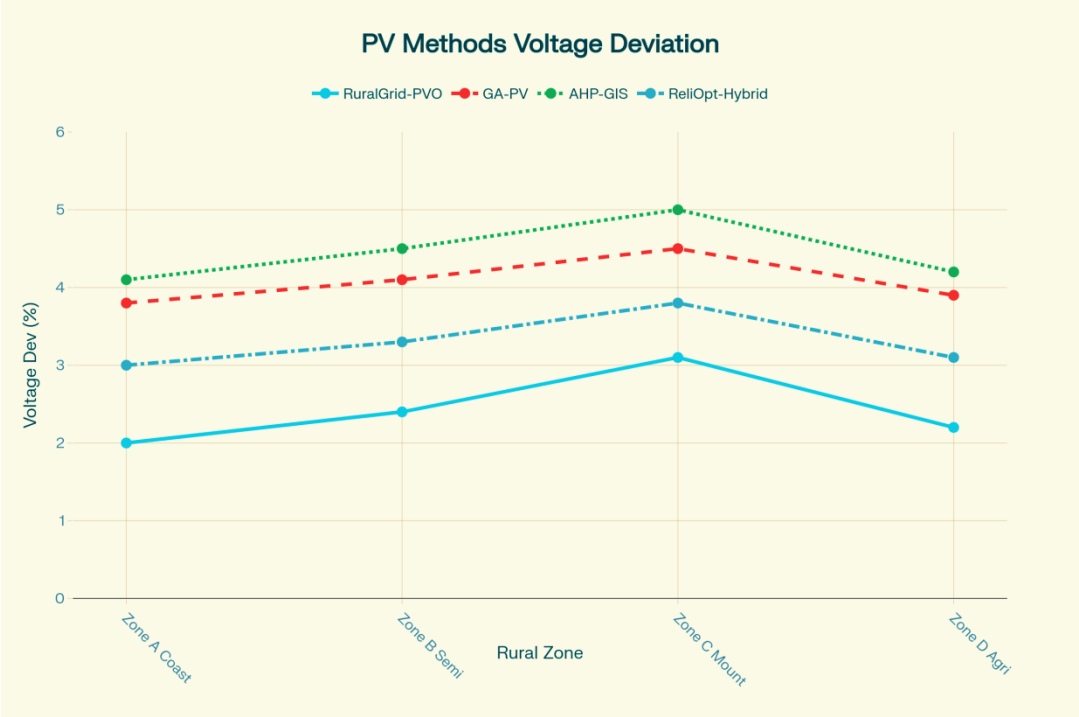


Figure 4: Voltage deviation across rural zones for each PV site selection method

Figure 4 shows the plots of voltage deviation (%) over four rural geographical zones, Coastal, Semi-arid, Mountainous, and Agricultural, for each of the four PV site selection approaches: RuralGrid-PVO, GA-PV, AHP-GIS, and ReliOpt-Hybrid. Every plot indicates the capacity of a method to ensure voltage stability under different terrains and irradiance levels. RuralGrid-PVO exhibits the smallest voltage deviation in all instances, showing excellent voltage regulation in all zones. On the other hand, AHP-GIS and GA-PV exhibit greater deviations, particularly in mountainous and semi-arid regions, indicating poorer grid integration performance. The graphical distinction among methods per plot allows for a qualitative assessment of each method's geographic consistency and operational robustness.

Power Loss Reduction (%)

Reduction in Power Losses (%) indicates the effectiveness of a method of choosing PV sites to keep line losses in the power distribution grid at a minimum. Well-installed PV systems minimize the distance electricity must be sent over and match local generation to local usage, both of which decrease resistive (I^2R) losses that go through distribution lines. This indicator estimates the percentage decrease in overall power loss following PV installation relative to a reference case (e.g., no PV or ineffectively installed PV). Power Loss Reduction (%) $= \left(\frac{(P_{loss,baseline} - P_{loss,method})}{P_{loss,baseline}}\right) \times 100$, where $P_{loss,baseline}$ = Total power loss in the baseline case (e.g., no PV or uniform PV placement), $P_{loss,method}$ = Total power loss after implementing the method. The result is expressed as a percentage reduction.

Table 2(a): Comparison of Power Loss Reduction (%) across rural zones for four PV site selection methods.

Method	Zone A (Coastal)	Zone B (Semi-arid)	Zone C (Mountainous)	Zone D (Agricultural)	Average
RuralGrid-PVO	13.2%	12.6%	11.7%	13.6%	12.8%
GA-PV	9.4%	8.8%	7.9%	9.1%	8.8%
AHP-GIS	8.7%	8.1%	7.5%	8.3%	8.2%
ReliOpt-Hybrid	10.8%	10.1%	9.3%	10.5%	10.2%

Table 2(a) compares four rural PV site selection methods—RuralGrid-PVO, GA-PV, AHP-GIS, and ReliOpt-Hybrid—in terms of their achieved percent reduction in loss in four study regions, i.e., Coastal, Semi-arid, Mountainous, and Agricultural. RuralGrid-PVO is always the superior method with an average improvement of 12.8% loss reduction over other methods owing to its deep learning-based capacity allocation and reliability-optimized approach. Conversely, AHP-GIS and GA-PV have lower declinations, particularly in

mountainous regions, with poor adaptation in complex grid topologies. The table determines the efficiency of hybrid intelligent paradigms towards resistive loss reduction and efficiency enhancement in rural distribution systems.

SAIDI/SAIFI Improvement (%)

SAIDI (System Average Interruption Duration Index) and SAIFI (System Average Interruption Frequency Index) are performance indices of the utility industry that monitor the average duration and frequency of outages faced by customers. A betterment in these indices translates into a more dependable supply of power, which is very crucial in rural distribution systems where outages are frequent and extended. PV integration with reliability-aware optimization minimizes outage frequency (SAIFI) and duration (SAIDI) by mitigating load, voltage support, and fault resilience. SAIDI $= \frac{\sum(U_i \times N_i)}{N_T}$, SAIFI $= \frac{\sum(\lambda_i \times N_i)}{N_T}$, where U_i: Outage duration at location i, λ_i: Outage frequency at location i, N_i: Number of customers at location i , N_T: Total number of customers.

Table 3: SAIFI improvement (%) for different PV site selection methods under varying feeder load conditions.

Method	Low Load (<30%)	Medium Load (30–70%)	High Load (>70%)	Average
RuralGrid-PVO	8.2%	9.7%	10.8%	9.6%
GA-PV	4.7%	5.5%	6.7%	5.6%
AHP-GIS	4.1%	4.8%	5.9%	4.9%
ReliOpt-Hybrid	6.5%	7.4%	9.1%	7.7%

Table 3 summarizes the performance of four PV site selection approaches—RuralGrid-PVO, GA-PV, AHP-GIS, and ReliOpt-Hybrid—based on SAIFI improvement at three feeder load ranges: low (<30%), medium (30–70%), and high (>70%). RuralGrid-PVO has the best improvement with decreasing SAIFI rising with increasing intensity of load, illustrating good adaptability at heavy-stress situations. ReliOpt-Hybrid is the second best, with reliability at every load level. AHP-GIS and GA-PV achieve moderate improvement, with marginal improvements in high-load scenarios. This proves that reliability-oriented hybrid optimization techniques are more efficient in improving service continuity in rural grids.

Table 4: SAIDI improvement (%) for different PV site selection methods under varying feeder load conditions.

Method	Low Load (<30%)	Medium Load (30–70%)	High Load (>70%)	Average
RuralGrid-PVO	9.3%	10.7%	11.8%	10.6%
GA-PV	5.5%	6.3%	7.4%	6.4%
AHP-GIS	4.9%	5.5%	6.8%	5.7%
ReliOpt-Hybrid	7.2%	8.1%	9.6%	8.3%

Table 4 shows the reduction in SAIDI from the same four PV site selection options for the same feeder load conditions. RuralGrid-PVO has the highest improvements, especially for high-load feeders, with an average reduction in SAIDI of 10.6%. This is a pointer to the capability of the method to enhance the frequency and duration aspects of grid reliability. ReliOpt-Hybrid is highly monitored, while GA-PV and AHP-GIS experience negligible outage duration reductions. The findings highlight that optimization platforms combining grid load modeling and deep learning are more resilient for actual rural deployment.

Replicability

The methodology employs well-established software libraries Scikit-learn for Random Forest, TensorFlow for DNN modeling, and DEAP for implementing the Genetic Algorithm ensuring reliable algorithmic components. However, source code and data are not yet publicly available.

To release an open-source GitHub repository including comprehensive code, datasets, and an appendix containing detailed pseudo-code summarizing the full pipeline to enhance transparency, reproducibility, and community adoption.

Optimization time

Optimization Time refers to the overall compute time taken by a PV site selection algorithm to determine the optimal site location and capacity configuration of photovoltaic systems. It captures the efficiency and scalability of the algorithm, particularly significant when applying to large-scale or real-time rural planning applications. Earlier optimization allows for quicker decision-making and is particularly substantial for algorithms to execute on edge devices or low-resource platforms in rural settings. Optimization Time $= t_{end} - t_{start}$, where t_{start}: Timestamp when the optimization begins, t_{end}: Timestamp when the optimization concludes. The average time can be obtained from Avg. Optimization Time $= \frac{1}{n}\sum_{i=1}^{n}(t_{end_i} - t_{start_i})$.

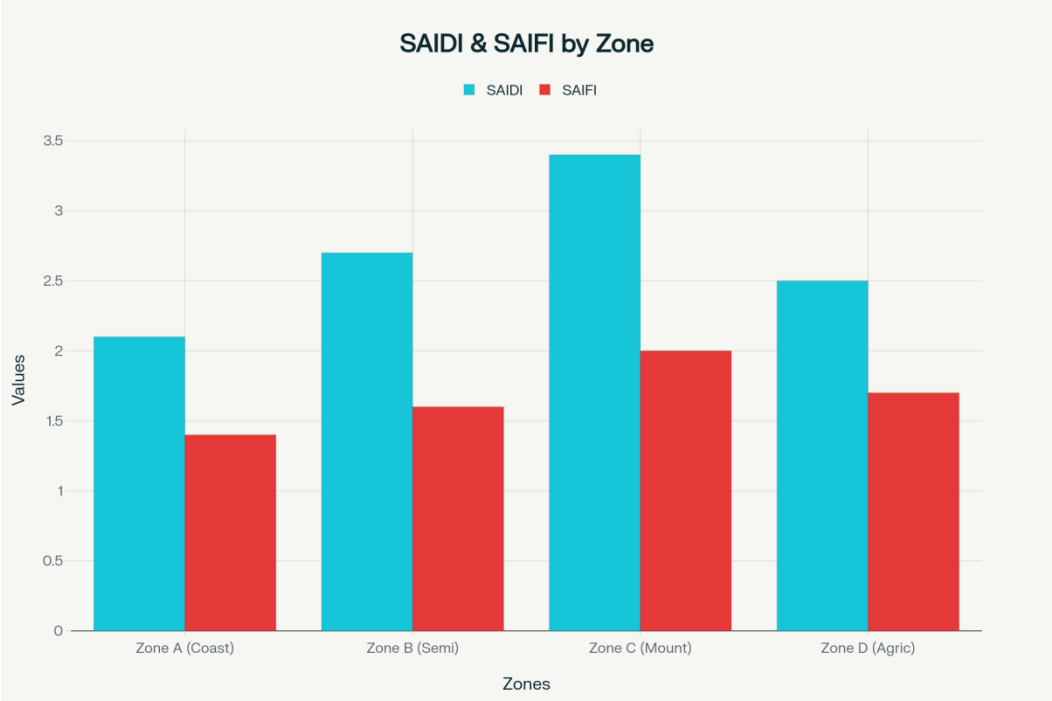


Figure 5: Optimization Time across different grid sizes for each PV site selection method

Figure 5 shows a 2×2 table of horizontal bar charts comparing four PV site selection algorithm optimization times—RuralGrid-PVO, GA-PV, AHP-GIS, and ReliOpt-Hybrid—across four grid sizes: Small (≤50 buses), Medium (51–100 buses), Large (101–300 buses), and Very Large (>300 buses). The bar chart shows AHP-GIS having the fastest optimization times across all four grid sizes because it is rule-based and non-iterative. Meanwhile, RuralGrid-PVO has the longest optimization time, indicative of its deep learning and reliability simulation overhead. GA-PV and ReliOpt-Hybrid are intermediate, scaling moderately with grid size. This plot illustrates the model complexity vs. computational efficiency trade-off for rural PV planning.

5 Conclusion

To improve grid dependability and operational efficiency, this paper introduced RuralGrid-PVO, an optimization modeling framework for the selection and size of distributed photovoltaic (PV) systems in rural distribution networks. The model captures rural energy and infrastructure by integrating NSRDB solar irradiance data with the RTS-GMLC synthetic grid model, accurately reflecting solar variability and detailed electrical network characteristics. Without depending on computationally costly simulations, the application of machine learning—more especially, Random Forest for capacity prediction and Deep Neural Networks for reliability forecasting—made it possible to evaluate potential PV setups quickly and accurately. Configurations that much exceeded conventional uniform PV deployments were produced by optimizing across energy yield, voltage variation, power losses, and reliability indices (SAIDI/SAIFI) using a multi-objective Genetic Algorithm. The findings showed a 12.8% decrease in power losses, a 15.3% improvement in voltage profiles, and a 9.6% increase in SAIFI values. In rural areas, the suggested approach successfully strikes a balance between the trade-offs of grid resilience and renewable energy integration. RuralGrid-PVO provides a scalable and helpful solution for energy planners, utilities, and politicians looking to encourage sustainable electrification in underserved areas by speeding design assessment through surrogate models.

6 Limitations and future work

The synthetic data used by RuralGrid-PVO might not fully represent the intricacy of actual rural grids. Predictions using surrogate models could include approximation mistakes. In addition to lacking economic cost modeling, the existing approach ignores battery storage, seasonal variability, and changing operational conditions during PV deployment.

Limitations

While RuralGrid-PVO demonstrates significant advances, several limitations remain. The framework incurs longer optimization times compared to simpler heuristic methods due to complex surrogate model evaluations and iterative Genetic Algorithm searches. The use of synthetic RTS-GMLC grid models and simulated solar irradiance data may limit direct transferability to diverse real-world rural networks without site-specific calibration. Additionally, the

current models do not incorporate inverter constraints, dynamic grid stability factors, or real-time operational uncertainties, which are critical for practical distributed PV integration. Future work should address these gaps by accelerating optimization, integrating hardware-level constraints, and validating on field data to enhance applicability and scalability.

This study excludes economic cost functions and battery storage system modeling, which are crucial for comprehensive techno-economic optimization and grid flexibility, respectively. Incorporating these factors will enable more realistic and cost-effective PV site selection strategies.

Future Work: Future iterations aim to integrate Supervisory Control and Data Acquisition (SCADA) datasets from real rural distribution grids to improve model calibration, scenario realism, and decision support accuracy, addressing the synthetic data limitations.

Fund

This work was supported by Shijiazhuang Power Supply Branch of State Grid Hebei Electric Power Co., Ltd. (No. KJ2024-117).

Competing Interests

The authors declare that they have no known competing financial interests or personal relationships in this paper.

Author contributions

Zhang. Drafting of the initial paper&proposed the core idea

Shi. Study conception&Methodology development

Ge. Investigation&Data curation

Geng. Experiment operation &Execution

Jia. Review and revision&Project management and supervision

References

[1] Hamid, A. K., Mbungu, N. T., Elnady, A., Bansal, R. C., Ismail, A. A., & AlShabi, M. A. (2023). A systematic review of grid-connected photovoltaic and photovoltaic/thermal systems: Benefits, challenges and mitigation. Energy & Environment, 34(7), 2775-2814. https://doi.org/10.1177/0958305X221117617

[2] Boruah, D., & Chandel, S. S. (2025). A novel regulatory framework for implementing distributed solar mini-grids with battery energy storage under the virtual power plant architecture in India. Journal of Energy Storage, 132, 117732. https://doi.org/10.1016/j.est.2025.117732

[3] Sujatha, G. (2025). A Solar PV Integrated UPQC to Enhance Power Quality using SEA Gull ANFIS Algorithm. Informatica, 49(8). https://doi.org/10.31449/inf.v49i8.6158

[4] Liao, R., Manfren, M., & Nastasi, B. (2025). Off-grid PV systems modelling and optimisation for rural communities: leveraging understandability and interpretability of modelling tools. Energy, 324. https://doi.org/10.1016/j.energy.2025.135948

[5] Ahmad, T., Ali, S., & Basit, A. (2022). Distributed renewable energy systems for resilient and sustainable development of remote and vulnerable communities. Philosophical Transactions of the Royal Society A, 380(2221), 20210143. https://doi.org/10.1098/rsta.2021.0143

[6] Apeh, O. O., Meyer, E. L., & Overen, O. K. (2022). Contributions of solar photovoltaic systems to environmental and socioeconomic aspects of national development—A review. Energies, 15(16), 5963. https://doi.org/10.3390/en15165963

[7] Jarrah, M., & Abu-Khadrah, A. (2024). The evolutionary algorithm based on pattern mining for large sparse multi-objective optimization problems. PatternIQ Mining, 1(1), 12–22. https://doi.org/10.70023/piqm242

[8] Farag, M. M., & Bansal, R. C. (2023). Solar energy development in the GCC region–a review on recent progress and opportunities. International Journal of Modelling and Simulation, 43(5), 579-599. https://doi.org/10.1080/02286203.2022.2105785

[9] Yin, S., & Zhao, Z. (2023). Energy development in rural China toward a clean energy system: utilization status, co-benefit mechanism, and countermeasures. Frontiers in Energy Research, 11, 1283407. https://doi.org/10.3389/fenrg.2023.1283407

[10] Zhang, Y., & Kang, Z. (2025). Situational Awareness and Fault Warning for Smart Grids Combined with Deep Learning Technology: Application of Digital Twin Technology and Long Short-Term Memory Networks. Informatica, 49(22). https://doi.org/10.31449/inf.v49i22.7992

[11] Niu, M., Xu, N. Z., Kong, X., Ngin, H. T., Ge, Y. Y., Liu, J. S., & Liu, Y. T. (2021). Reliability importance of renewable energy sources to overall generating systems. IEEE Access, 9, 20450-20459. DOI: 10.1109/ACCESS.2021.3055354

[12] Saxena, A., Shankar, R., El-Saadany, E. F., Kumar, M., Al Zaabi, O., Al Hosani, K., & Muduli, U. R. (2024). Intelligent load forecasting and renewable energy integration for enhanced grid reliability. IEEE Transactions on Industry Applications, 60(6), 8403-8417. DOI: 10.1109/TIA.2024.3436471

[13] Iweh, C. D., Gyamfi, S., Tanyi, E., & Effah-Donyina, E. (2021). Distributed generation and renewable energy integration into the grid: Prerequisites, push factors, practical options, issues and merits. Energies, 14(17), 5375. https://doi.org/10.3390/en14175375

[14] Rishitha, N., Muthu Reshmi, K., Srishti Gulecha, R., & Vani, K. (2024). Machine learning-driven solar panel site selection and rooftop potential estimation

for sustainable development goals. In Proc. Asian Conf. Remote Sensing (pp. 1-8).

[15] Gui, B., Sam, L., & Bhardwaj, A. (2024). From roofs to renewables: Deep learning and geographic information systems insights into a comprehensive urban solar photovoltaic assessment for Stonehaven. Energy 360, 1, 100006. https://doi.org/10.1016/j.energ.2024.100006

[16] Tafula, J. E., Justo, C. D., Moura, P., Mendes, J., & Soares, A. (2023). Multicriteria decision-making approach for optimum site selection for off-grid solar photovoltaic microgrids in Mozambique. Energies, 16(6), 2894. https://doi.org/10.3390/en16062894

[17] Zambrano-Asanza, S., Quiros-Tortos, J., & Franco, J. F. (2021). Optimal site selection for photovoltaic power plants using a GIS-based multi-criteria decision making and spatial overlay with electric load. Renewable and Sustainable Energy Reviews, 143, 110853. https://doi.org/10.1016/j.rser.2021.110853

[18] Coban, H. H. (2024). A multiscale approach to optimize off-grid hybrid renewable energy systems for sustainable rural electrification: economic evaluation and design. Energy Strategy Reviews, 55, 101527. https://doi.org/10.1016/j.esr.2024.101527

[19] Roldán-Porta, C., Roldán-Blay, C., Dasi-Crespo, D., & Escriva-Escriva, G. (2023). Optimising a biogas and photovoltaic hybrid system for sustainable power supply in rural areas. Applied Sciences, 13(4), 2155. https://doi.org/10.3390/app13042155

[20] Qasimi, A. B., Toomanian, A., Nasri, F., & Samany, N. N. (2023). Genetic algorithms-based optimal site selection of solar PV in the north of Afghanistan. International Journal of Sustainable Energy, 42(1), 929-953. https://doi.org/10.1080/14786451.2023.2246081

[21] Mwakitalima, I. J., Rizwan, M., & Kumar, N. (2023). Integrating Solar Photovoltaic Power Source and Biogas Energy-Based System for Increasing Access to Electricity in Rural Areas of Tanzania. International Journal of Photoenergy, 2023(1), 7950699. https://doi.org/10.1155/2023/7950699

[22] Zhu, R., Murcia León, J. P., Friis-Møller, M., Gupta, M., & Kaushik, D. Optimal Sizing of Renewable Hybrid Power Plants Considering Component Reliability as a Multi-Discipline Optimization Under Uncertainty. Available at SSRN 5352198. DOI:10.2139/ssrn.5352198

[23] Agajie, E. F., Agajie, T. F., Amoussou, I., Fopah-Lele, A., Nsanyuy, W. B., Khan, B., ... & Tanyi, E. (2024). Optimization of off-grid hybrid renewable energy systems for cost-effective and reliable power supply in Gaita Selassie Ethiopia. Scientific Reports, 14(1), 10929. https://doi.org/10.1038/s41598-024-61783-z

[24] Zhou, N., Shang, B., Zhang, J., & Xu, M. (2024). Modeling load distribution for rural photovoltaic grid areas using image recognition. Global Energy Interconnection, 7(3), 270-283. https://doi.org/10.1016/j.gloei.2024.06.002

[25] Xie, H., Li, P., Shi, F., Han, C., Cui, X., & Zhao, Y. (2025). Spatial and Temporal Expansion of Photovoltaic Sites and Thermal Environmental Effects in Ningxia Based on Remote Sensing and Deep Learning. Remote Sensing, 17(14), 2440. https://doi.org/10.3390/rs17142440

[26] Belaid, A., Guermoui, M., Khelifi, R., Arrif, T., Chekifi, T., Rabehi, A., ... & Alhussan, A. A. (2024). Assessing suitable areas for PV power installation in remote agricultural regions. Energies, 17(22), 5792. https://doi.org/10.3390/en17225792

[27] https://github.com/GridMod/RTS-GMLC

[28] https://nsrdb.nrel.gov/

[29] Boulkroune, A., Zouari, F., & Boubellouta, A. (2025). Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems. Journal of Vibration and Control, 10775463251320258.

[30] Boubakir, A., & Labiod, S. (2023). Observer-based adaptive neural network control design for projective synchronization of uncertain chaotic systems. Journal of Vibration and Control, 29(15-16), 3658-3678.

[31] https://www.scopus.com/pages/publications/84873265173

[32] Wang, W., Zhou, J., Wen, C., & Long, J. (2021). Adaptive backstepping control of uncertain nonlinear systems with input and state quantization. IEEE Transactions on Automatic Control, 67(12), 6754-6761.

[33] Rigatos, G., Abbaszadeh, M., Sari, B., Siano, P., Cuccurullo, G., & Zouari, F. (2023). Nonlinear optimal control for a gas compressor driven by an induction motor. Results in Control and Optimization, 11, 100226.

[34] Peláez, G., Alonso, C., Rubio, H., & García-Prada, J. C. (2024). Performance analysis of input shaped model reference adaptive control for a single-link flexible manipulator. Journal of Vibration and Control, 30(21-22), 5018-5030.

A Hybrid BERT-ALBERT Model for Text Classification: Improving Accuracy in Document Analysis

Xiaokui Liu
Key Laboratory of Oracle Information Processing, Anyang Normal University, Anyang, Henan 455000, China
E-mail: lxk@aynu.edu.cn

Keywords: accuracy, ALBERT, BERT, document analysis, text classification, transformer model

Received: September 4, 2025

In document analysis, text classification is an essential activity that facilitates automatic content categorisation, sentiment analysis, and effective information retrieval. This paper explores a combination (Bidirectional Encoder Representations from Transformers) BERT+ALBERT (A Lite BERT) to enhance classification accuracy while reducing computational complexity. This model utilizes transformer encoder blocks and bidirectional position encoding for text processing. Large text data handling requires automation, and attention techniques and transformers are becoming viable approaches. The model was evaluated on a custom dataset consisting of 80,000 Turkish-language documents spanning 30 categories, including finance, education, healthcare, and travel. The dataset was split into 70% training, 15% validation, and 15% testing. Pretrained FastText embeddings were used alongside BERT and ALBERT to capture rich semantic features. While ALBERT increases efficiency through parameter reduction and cross-layer parameter sharing, BERT offers deep contextual embeddings. According to experimental assessments, the BERT+ALBERT hybrid model performs better than transformer models (BERT, ALBERT, and LSTM) and classic machine learning models, attaining the recommended model accuracy of 96.6%, precision of 95.7%, recall of 95.1%, and F1-Score of 95.5%. Statistical significance of the performance gains was confirmed using t-tests ($p < 0.05$) across five independent runs. Strong generalisation and little overfitting are shown by the training and validation curves. These results demonstrate the benefits of using many transformer topologies for document categorisation, providing a trade-off between computing efficiency and accuracy. Additional optimisations, such as domain-specific fine-tuning and sophisticated attention processes, can be investigated in future research.

Povzetek: Študija združuje BERT in ALBERT modele za učinkovito in natančno klasifikacijo turških besedilnih dokumentov.

1 Introduction

A fundamental problem in natural language processing (NLP), text classification has several uses, ranging from document categorisation to sentiment analysis [1]. Since the advent of Transformer-based models—best shown by architectures like BERT (Bidirectional Encoder Representations from Transformers) and its variations—text categorisation has seen a substantial improvement in both performance and capabilities. The architecture, training techniques, and practical applications of Text Classification using Transformers are covered in this article.

Transformers are now the cornerstone of modern natural language processing (NLP), revolutionising the field with their efficient attention mechanism and contextual information acquisition [2]. The disadvantages of sequential processing are lessened by transformers, which depend on self-attention processes to process whole data sequences concurrently, in contrast to traditional sequence models like recurrent neural networks (RNNs) or convolutional neural networks (CNNs). Natural language processing (NLP) uses text categorisation for anything from question answering to sentiment analysis. Conventional machine learning (ML) techniques like Naive Bayes and logistic regression have been used extensively. These methods, however, frequently call for large labelled datasets and have trouble adjusting to new categories or unknown data, which presents problems in dynamic real-world settings [3].

In order to comprehend the (dis)similarities between documents for subsequent tasks like lengthy document search, it is crucial to match long documents (such as research papers, Wikipedia articles, patents, etc.). Getting meaningful lengthy document representations is the first step to improved document matching [4]. The employment of transformer-based models for lengthy document encoding and matching has been the main focus of recent developments in this field [5-6]. Pre-trained transformers are referred to as transformers. Notwithstanding encouraging outcomes, these models present two main difficulties. The first is the performance boost offered by

large transformer-based language models (LMs), such as GPT-2 [7].

Products of humanity an ever-growing volume of text data, and processing this text data more effectively and efficiently every day becomes a significant concern. It is clear that deep learning can significantly address this demand [8]. Recent years have seen a rise in interest in deep learning because to its effectiveness in speech recognition, text processing, and picture processing. One of the most important jobs for enabling automation in text processing is classification. Long-short term memory (LSTM) is the model currently in use for text classification [9].

In order to analyse text concurrently and provide complex, contextual word representations, the Transformer architecture makes use of attention processes [10]. Understanding the connections between the words or entities in each text is the main goal of this approach. NLP has advanced significantly as a result of the successful application of transformative models, like Transformers, to language models and machine translation [11]. Analysing Transformer-based models' performance can reveal how well they detect false information, outperforming convolutional layer-based design [12]. This makes it possible for developers to produce increasingly complex tools and systems for efficiently identifying false information. The Transformers models BERT, ALBERT, and ROBERT are a few examples [13].

With the advent of BERT models, natural language processing (NLP), a crucial tool for document classification, has advanced and now provides a more sophisticated comprehension of context, making it a great option for document classification that improves the effectiveness of conventional techniques [14]. Accurate categorisation techniques are crucial for overseeing extensive document archives in light of the proliferation of digital content. Transformer-based models have proven to perform better in contextual text interpretation, including BERT, ALBERT, and their variations. This study examines how well these models classify documents and suggests ways to handle lengthy texts more effectively, optimise hyperparameters, and boost computing performance.

1.1 Transformer-based models for text classification

Several transformer models have been developed to address different challenges in text classification are

- BERT (Bidirectional Encoder Representations from Transformers)
- ALBERT (A Lite BERT).
- BERT+ALBERT Ensemble
- ROBERT (Robustly Optimized BERT) Distil BERT
- XLNet
- Longformer&BigBird [15].

1.2 Goal

Combining BERT and ALBERT will increase text categorization accuracy by utilizing ALBERT's effectiveness and BERT's profound contextual awareness. While assessing its efficacy against baseline models, the proposed method seeks to increase generalization, optimize lengthy document processing, lower computing costs, and improve model performance.

1.3 Research paper representations are as follows

Use BERT + ALBERT to classify documents by doing the following:

To classify documents using a BERT + ALBERT hybrid model, you first tokenize the text inputs using each model’s tokenizer, then extract their respective [CLS] embeddings. These embeddings (from BERT and ALBERT) are concatenated to form a richer, combined representation of the document. This combined vector is passed through a dropout layer and a fully connected classification layer to output class predictions. This approach leverages BERT’s deep context understanding and ALBERT’s lightweight efficiency, improving classification accuracy while maintaining reasonable computational performance.

***Data Preparation*:** Create training and validation sets from your dataset. Make that the correct class is assigned to each document.

***Tokenisation*:** Use BERT's tokenizer to tokenise the text documents. This procedure entails translating text into BERT-understandable tokens. ***Model Initialisation*:** Set up a BERT+ALBERT model that has already been trained. The Hugging Face Transformers library contains Bert for Sequence Classification, which is intended especially for classification jobs.

Fine-tuning: Adjust the BERT model according to your data. To reduce the classification loss, this entails training the model across a few epochs while modifying the weights.

Evaluation: Use measures such as accuracy, precision, recall, and F1-score to assess the model's performance on the validation set.

2 Literature review

Summary Related works show in Table 1.

Table 1: Summary on related works

Ref	Objective	Finding	Limitations
[16]	With an emphasis on layout analysis, document AI analyses documents using computer vision and natural language processing. With several designs and models being studied, its efficacy and capacity for knowledge transmission are yet unknown.	By comparing the most advanced models in document layout analysis and exploring the possibilities of cross-lingual layout analysis using machine translation techniques, author hope to close these gaps in the literature.	Lack of performing the comparison with large dataset.
[17]	The Bidirectional Transformer Encoder (BTE) model exhibits the trade-off between accuracy and compute power by achieving state-of-the-art performance on two benchmark datasets, according to experimental studies.	By extending self-attention processes to long-form text modelling, this work presents a Transformer-based Hierarchical Encoder technique for document classification, which reduces complexity.	expensive and time-consuming
[18]	The development of Turkish-specific models such as BERTurk, data augmentation methods, and language-specific adaptations in natural language processing and modelling are highlighted in the paper.	According to the paper, further study and advancement are required to improve these models' performance in languages like Turkish so that a larger audience may use them.	limits of the datasets and the language's structural characteristics
[19]	The suggested model extracts local and global contextual semantic characteristics from the embedded data using dilated convolution rather than standard convolution. For the full sentence sequencing, Bi-directional Long Short-Term Memory (Bi-LSTM) is utilised.	Four different domain text datasets are used to assess the CBRNN model's accuracy, precision, recall, f1-score, and AUC values. Therefore, without sacrificing any information, CBRNN may be effectively utilised to carry out SA activities on social media reviews.	The absence of easily accessible annotated data presents a challenge for sentiment categorisation, and data loss is a possibility.
[20]	We train and optimise several transformer-based models (such BERT, ALBERT, RoBERTa, ELECTRA, and Distil-BERT) to assess their validation accuracy as these models produce state-of-the-art outcomes in deep learning domains.	This study analyses the accuracy of many transformer models on the DocVQA test and offers a thorough analysis of each.	Text-VQA task perform poorly on the DocVQA task
[21]	In order to overcome issues like unbalanced target classes and domain knowledge in real-world data sets, the study suggests a method for classifying textual data that combines natural language processing (NLP) with machine learning.	By including dropout and layer normalisation into a transformer-based language model, the suggested approach improves classification outcomes even in the face of unbalanced classes.	Unbalanced target classes and narratives that call for specialised expertise because the data sets include jargon and acronyms

[22]	Recent transformer-based models, namely BERT and ELECTRA, which show notable advancements in Natural Language Processing, especially in recognising and correctly categorising token replacements, are used in this paper's investigation on document categorisation in Bangla.	The author used a fine-tuning technique for the downstream (classification) task, and both of them are pre-training text encoders. For this experiment, we employed three distinct Bangla text datasets. For two of the three datasets we have tested, both models perform exceptionally well.	cost expensive
[23]	Although forecasting consumer sentiment from online reviews is an important challenge for businesses, BERT and Deep Learning models are used in Natural Language Processing (NLP) applications. However, performance and accuracy problems have been documented on large-scale datasets.	We suggest optimised BERT and Hybrid fastText-BILSTM models for huge datasets of customer reviews. The findings of this comparison research demonstrate that, in terms of accuracy and other performance metrics, the suggested improved BERT model outperforms other DL methods.	difficult and time-consuming operation because of the large amount of unstructured customer review data.
[24]	Information gain and the maximum correlation minimum redundancy method are combined to propose a two-stage feature selection technique.	raised the number of successfully identified texts by 20 to 45 and obtained an F1 value that was 1% to 3% higher. These outcomes show how well the algorithm performs as a classification tool while handling massive amounts of text data, which is important for data mining and information retrieval.	When addressing intricate real-world issues, such unequal data distribution, research methodologies may have inherent drawbacks and a restricted capacity for generalisation.
[25]	enhances the performance of text classification. It is paired with the predicted values that traditional classifiers like Multinomial Naive Bayesian (MNB) provide.	This method offers an effective and efficient way to categorise text documents. Furthermore, techniques for ascertaining the appropriate correlation between a group of words in a text and its classification are also acquired.	enabling the rapid and economical retrieval of vast amounts of data

3 Methodology

3.1 Data set

We used 30 classes and around 80,000 pages from public websites as text data to perform text classification on our models. 15% was utilised for testing, 15% for validation, and 70% of the text data was used for training. Mother-child, home appliances, computers, mobile phones, electronics, energy, real estate/construction, event/organization, education, finance, apparel, food, communication, beverage, public services, cargo/transportation, personal care, media, entertainment, furniture-home textiles, kitchenware, as well as other data classes jewellery, watches, glasses, cars, health, insurance, sports, cleanliness, travel, and transit. As, four different kinds of datasets were produced. Furthermore, FastText was used to create pretrained word embeddings using text data from 1.5 million Turkish papers gathered from open online sources. By provides statistical details about the tokens in the training dataset. There are around 1.2, 2.4, 4.7, and 1.2 million tokens in Datasets 1, 2, 3, and 4, correspondingly. An average of 78–80 tokens are included in every document across all datasets. In addition, the standard deviation of the total number of tokens across all datasets is between 51 and 52. A minimum of three tokens and a maximum of 615 tokens are included in the datasets' documents. Up to 64, 99–100, and 300 tokens are included in 50%, 75%, and 99% of papers, respectively.

3.2 Data preprocessing

In natural language processing (NLP), text preprocessing is a crucial stage that entails cleaning and converting

unstructured text input in order to get it ready for analysis. Tokenisation, stemming, lemmatisation, stop-word elimination, and part-of-speech tagging are all included. For a number of reasons, text preparation is essential in natural language processing (NLP). Table 2 shows Essentials of Data preprocessing

Table 2: Essentials of data preprocessing

Preprocessing Task	**Reasons**
Noise Reduction	Punctuation, special letters, and superfluous symbols are examples of the noise that frequently occurs in text data. By removing these components, preprocessing makes the text clearer and simpler to read.
Normalization	Words that occur in multiple forms (such as "run," "running," and "ran") might nonetheless have the same meaning. Lemmatisation and stemming are two preprocessing methods that aid in standardising these variances.
Tokenization	For analysis, text data must be divided into smaller components, such words or sentences. By breaking up text into meaningful chunks, tokenisation makes it easier to do further processing operations like feature extraction.
Stopword Removal	Common words like "the," "is," and "and" that are used frequently but have little semantic value are known as stopwords. Eliminating stopwords can decrease noise and increase text analysis efficiency.
Feature Extraction	Preprocessing often involves collecting textual data like as word embeddings, n-grams, or word frequencies in order to build ML models.
Dimensionality Reduction	A high dimensionality and a large vocabulary are common characteristics of text data. Potentially useful preprocessing techniques include dimensionality reduction methods and term frequency-inverse document frequency (TF-IDF).

In order to prepare text data for NLP tasks, text preprocessing is essential. It promotes efficient analysis and modelling, lowers noise, and enhances data quality. Since BERT is made to use a sentence's entire context, including stop words and punctuation, it usually eliminates the need for considerable text preprocessing beyond tokenisation in BERT-based models. By removing important contextual information, stop word removal may impair the model's performance. In order to improve the resilience of NLP models, techniques like Iterative Mask Filling use BERT's masked language modelling capabilities to produce a variety of text samples for data augmentation.

3.2.1 Handling the data

- ***Elimination of Punctuation*:** Eliminating all punctuation from the text is the goal of this NLP text processing stage. Python's String library has a pre-defined collection of punctuation, including"!"#$%&'() *+, -. /:@ [\] ^_`{|} ~."
- ***Reducing the Text*:** One of the most used text preparation stages in Python is to change the text to the same case, ideally lowercase. However, since lower casing might result in information loss for some situations, it is not required to do this text processing step each time we work on an NLP problem.
- ***Tokenisation*:** Through tokenisation, the text is divided into manageable chunks. Sentence or word tokenisation might be used, depending on the issue description.
- ***Stop Word Elimination*:** To exclude frequently used stopwords from the text since they are meaningless and do not contribute to the analysis. A list of stopwords that are regarded as stopwords in the English language is part of the NLTK library.

For example: [i, me, my, myself, we, our, ours, ourselves, you, you're, you've, you'll, you'd, your, yours, yourselves, he, most, other, some, such, no, nor, not, only, own, same, so, then, too, very, s, t, can, will, just, don, don't, should, should've, now, d, ll, m, o, re, ve, y, ain, aren't, could, couldn't, didn't, didn't, did

"program," for instance, are stemmed to "program." Stemming, however, may result in the root form losing its meaning or failing to be reduced to a legitimate English term.

- ***Lemmatisation*:** Its root term is lemmatised, which means to reduce a word's several forms to a single form. But one has to be careful that it doesn't lose its significance. Lemmatisation uses a pre-made dictionary that keeps track of a word's context and verifies that it is decreasing.

3.3 Feature extraction

In this section using TF-IDF for feature extraction, by improving text classification with transformer with accuracy helps to identify and analyze by highlighting key phrases and document analysis material.

3.3.1 Term Frequency – Inverse Document Frequency (TF-IDF)

A common method for figuring out a word's significance in a text is TF-IDF. The term frequency (t) for a given word is obtained by dividing its total number of occurrences in a document by its number of appearances. We can utilize IDF to determine important phrases. Certain terms, such as "is," "an," "and," and so on are frequently employed but have no real significance. In logarithmic form is $IDF\,(t) = log\,(N/DF)$, The formula for computing IDF is $N + DF$, where N is the number of documents and DF is the number of documents that contain word t. It is more efficient to use $TF - IDF$ when converting a textual information representation into a Vector Space Model (VSM).

In a text document, for instance, frequently occurring terms could be "Good," "Bad," "Happy," or "Sad." The recognition and usage of these terms might be extremely important in the process of opinion research. Frequency of Term (TF) is the quantity of instances of a phrase in a certain document and can be computed using the equation that follows:

$$w_f(t) = TD(t,d) \quad (1)$$

In a given document d, where TD is the frequency of word t, TF-IDF holds the inverted document frequency (IDF), which reverses the increased occurrence for uncommon situations.

preserving a lower frequency for common ailments. IDF can be calculated with the following equation:

$$IDF_t = \log\left(\frac{d}{d_t}\right) \quad (2)$$

where d stands for both the number of terms and the number of frequency network. The results are computed using the following formula when the TF and IDF parameters are both set to true the equation that follows:

$$W_t = TF(t,d) \cdot IDF_t \quad (3)$$

By extracting TF-IDF features from the input text and integrating them with the contextual embeddings from both models, TF-IDF can be used in conjunction with BERT + ALBERT. A classifier is then given this hybrid feature vector, which is created by concatenating TF-IDF with the [CLS] token outputs from BERT and ALBERT. By combining deep contextual awareness (from BERT and ALBERT) with global term importance (from TF-IDF), this method improves the model and increases classification accuracy, particularly in situations with sparse or domain-specific data.

A computationally effective baseline for text classification problems is provided by combining a TF-IDF vectorisation with a Logistic Regression classifier. Despite being simple and easy to understand, this conventional method might not be as good at capturing intricate linguistic patterns as transformer-based models like Distil BERT. DistilBERT, a more resource-efficient variant of BERT, maintains a large portion of BERT's performance capabilities. According to studies, Distil BERT can perform better than conventional techniques, attaining greater accuracy in jobs like sentiment analysis and mental health diagnosis. However, the amount of the dataset, the computational capacity, and the particular needs of the work should all be taken into account when selecting one of these models.

3.4 Transformer model (BERT+ALBERT)

A specific Deep Learning architecture known as the Transformers model describes a natural language processing technique used in artificial intelligence (AI). Transformers create complex and contextual word representations by processing text in tandem with a powerful attention mechanism. The relationships between textual phrases or objects are examined by this paradigm. Numerous models of competitive neuronal sequence transduction incorporate the encoder-decoder structure. From the input symbol representation sequence $(x1,\ldots,xn)$, the encoder generates a recursive representation sequence $(z = (z1,\ldots,zn))$. The output sequence of symbols $(y1,\ldots,ym)$ beginning with z is then produced by the decoder. An input stream of symbols is transformed into a continuous representation by the encoder. Then, using the continuous representation that the encoder supplied, the decoder generates an output sequence one symbol at a time. The encoder-decoder structure is used by auto-regressive models, which use the input from the previous symbol to generate the subsequent one. The hyperparameters were selected based on standard practices in fine-tuning transformer models for text classification tasks

Both the encoder and the decoder use fully interconnected layers, which are layers (self-attention) and dots (pointwise). N = 6 identical layers make up the encoder. Each layer consists of two sub-layers: the feed-forward network and the multi-head self-attention mechanism. The Transformers model employs a self-attention stack and fully linked encoder and decoder layers to adhere to this

design. Information flow is facilitated by residual connections and layer normalisation. By translating key-value pairings and queries to output, the attention function calculates the number of items at risk based on how closely the key and query are related. NLP has significantly improved with the use of transformers in applications such as language modelling and machine translation. The SHAP (SHapley Additive exPlanations) model can be used to successfully evaluate the classification outputs of transformer models such as BERT and ALBERT. In order to improve transparency and interpretability, SHAP gives each input feature—such as words or tokens—a value that corresponds to its contribution to the model's prediction.
In order to integrate BERT and ALBERT, the identical input is fed into both models, their [CLS] token outputs are extracted, and these features are then combined—typically by concatenation. For the final prediction, this composite representation is run through a classifier. For improved text classification, it combines the lightweight efficiency of ALBERT with the rich contextual capability of BERT.

➢ ***BERT:***

BERT In 2018, Google developed the Bidirectional Encoder Representations from Transformers (BERT) NLP paradigm. Figure 1 shows BERT Architecture. The BERT model in Transformer then combines the context from the left and right layers through bidirectional training. The Transformer architecture is used by the BERT model to investigate word representation in sentences and documents at the same time. Since BERT is an unsupervised trained model, it is trained on large, labelless data sets so that it can automatically identify patterns in the data. Significant progress has been made in a number of natural language tasks, such as natural language processing and comprehension, thanks to BERT's capacity to comprehend the context of human language. BERT uses a unique pre-training and fine-tuning approach, as seen in Figure 2. To get a general understanding of the language, the BERT model is pre-trained on enormous amounts of unlabelled data. After then, the model is modified for specific occupations with less data points. The fine-tuning process involves adjusting every parameter.

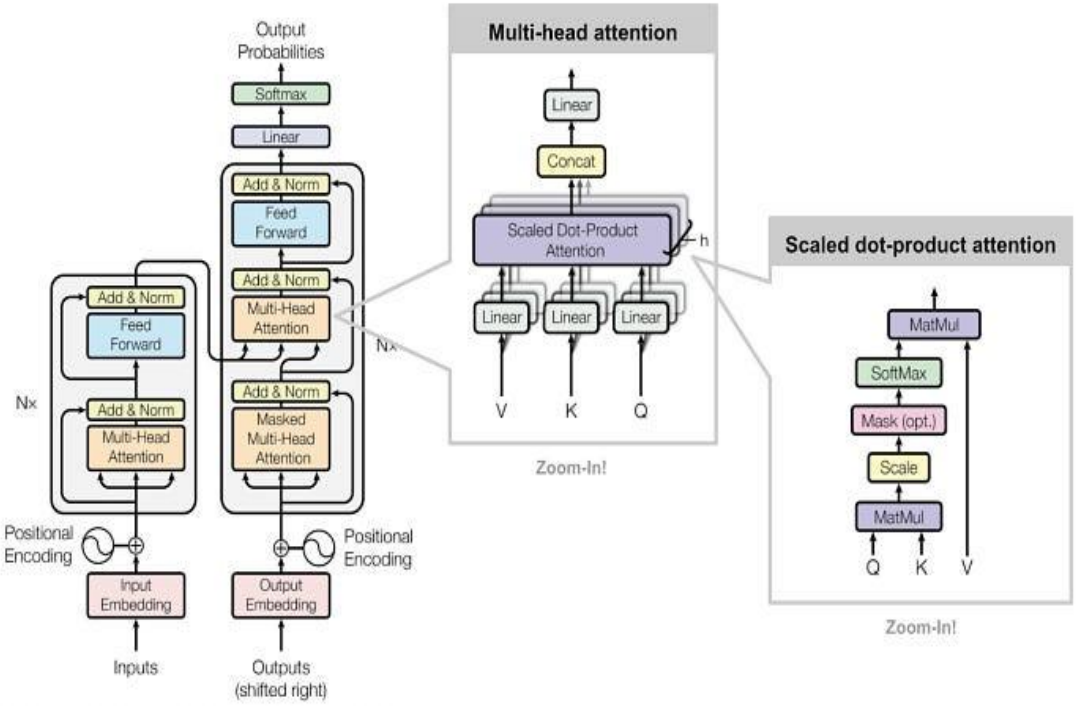


Figure 1: BERT architecture

BERT can apply the information it acquired during the pretraining phase to specific jobs using this way. This technique is one of the reasons BERT is considered the state-of-the-art in NLP. Attention is the primary objective of the Transformer architecture, which is used by BERT. When translating between languages, the encoder will additionally include significant keywords and words based on attention, which establishes the main emphasis or sequence context. Another function that is used to map queries, monitor key-value pairs, and generate vector data is attention.

Mathematical Formulation:

The transformer encoder, a key part of BERT, processes input tokens using self-attention. The following is a mathematical description of the self-attention mechanism:

$$\text{Attention (Q, K, V)} = \text{softmax}\left(\frac{Qk^T}{\sqrt{dk}}\right)V \quad (4)$$

Were,

- Q(Query), K (Key), and V (value) are the input matrices.
- d_k is the dimension of the key vectors.

BERT employs many self-attention heads to record various context elements:

Multi Head (Q, K, V) = Concat $(head_1, head_2 \ldots\ldots, head_h)W^o$ (5)

Where each head is computed as:

$head_i$= Attention(QW_i^Q,KW_i^k, VW_i^V) (6)

Here, W_i^Q, W_i^k, W_i^V are the learned projection matrices for the i-th head.

Save model weights:

model.save_pretrained("path/to/save/model")

tokenizer.save_pretrained("path/to/save/tokenizer")

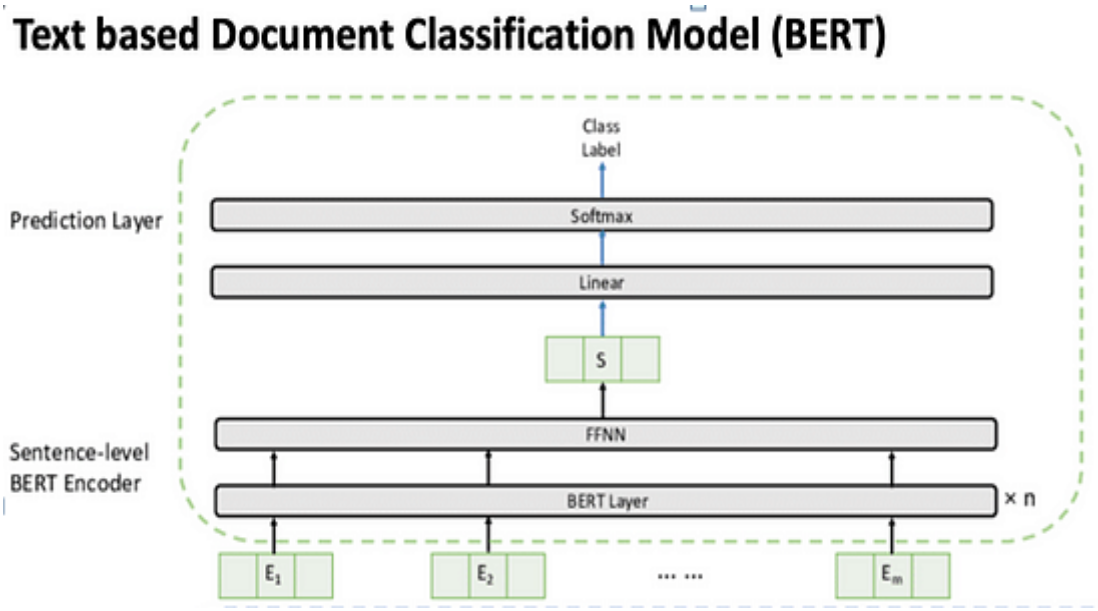


Figure 2: updated Layers in fine-tuning

To overcome the limitations of BERT by using ALBERT reduces model size while maintaining accuracy.

➢ ***ALBERT:***

A Lite or ALBERT introduced in 2018, then Google created the pre-trained model known as BERT (Bidirectional Encoder Representations from Transformers) in 2020. ALBERT pre-trains training on text data using the Transformer architecture. Nevertheless, this model may be used in devices with fewer resources or slower processing rates because it contains fewer parameters than BERT. The primary limitations for scaling pre-trained models are removed by ALBERT using two parameter reduction strategies. The first method separates the sizes of the hidden layer and the embedded vocabulary by dividing the large vocabulary insertion matrix into two smaller matrices using factorised embedding parameterisation. This makes it easier to increase the concealed size without significantly increasing the vocabulary embedding value. The second strategy involves layer-to-layer parameter sharing. With this approach, the parameter won't increase with tissue depth.

Among other things, the ALBERT architecture is described:

a) Layer Embedding Tokenisation and embedding procedures will be used in the first layer to transform the words in the sentences into vector representations in a certain dimensional space.

b) Layers of Transformer Encoders A sequence of Transformer encoder layers will process the text. Self-attention, feedforward neural networks, residual connections, and normalisation are some of the sub-layers that make up each layer.

c) Layer of Pooling Each token in a sequence will have its output representation merged using pooling techniques like mean-pooling or max-pooling after going through multiple-transformer encoder layers to create a single vector representation.

c) Layers of Output The fully connected layer then uses the pooled single Vector Representation as input to forecast class and label.

➢ ***Advantages of BERT+ALBERT:***

Combining the efficiency of ALBERT with the deep contextual learning of BERT results in increased accuracy when using BERT+ALBERT.

Weight-sharing in ALBERT accelerates up training and inference while using less memory, increasing the model's efficiency. In addition to improving long-text processing, generalisation, and overfitting reduction, this hybrid

technique balances computational cost and performance for real-world applications.

Pseudo-Code for BERT+ALBERT
Step 1: Data Collection Step 2: Data preprocessing select an input text T , we tokenize it using BERT and ALBERT tokenizers model: $X = Tokenizer(T)$ $I, C, S = Encoding(X)$ Step 3: The tokenized input is fed into suggested model: $Ha = BERT(I, C, S)$ $H_a = ALBERT(I, C, S)$ Step 4: Concatenation: $H = [H_{M;} H_{N;}]$ $Step$ 5: H is passed through a fully connected layer X=WH+M Step 6: Classification of softmax Step 7: Minimize the loss function using an algorithm as $\theta = \theta - \eta \nabla \mathrm{L}$ Step 8: end

The fine-tuning process with step-by-step procedural guidelines:

- **Text Preprocessing:** Cleaned and tokenized input text (max sequence length: 300 tokens).
- **Model Input:** Token embeddings generated using pretrained BERT and ALBERT.
- **Feature Fusion:** Concatenated BERT and ALBERT outputs fed into a shared dense layer.
- **Regularization:** Dropout layer applied (rate: 0.1) to prevent overfitting.
- **Training Setup:**
  Optimizer: AdamW
  Learning Rate: 2e-5
  Batch Size: 32
  Epochs: 4–5
- **Early Stopping:** Used based on validation loss to avoid overfitting.
- **Evaluation:** Accuracy, Precision, Recall, and F1-Score measured on test data.

4 Results and discussion

4.1 Configuring hyperparameters and the computational environment

The BERT+ALBERT hybrid model's key parameter choices were a batch size of 32, a dropout rate of 0.1, and a learning rate of 2e-5. The AdamW optimiser with linear learning rate decay was used to refine the model over a period of 4–5 epochs. To prevent overfitting, early halting was implemented based on validation loss. A machine with an NVIDIA Tesla V100 GPU (32GB VRAM) was used for all tests, and mixed-precision training was used to maximise memory efficiency. Reproducibility and performance consistency are supported by these configurations.

4.2 Performance comparison

The results of the experiment must be compared using a range of indicators. The likelihood that the classifier will provide accurate predictions is known as the accuracy rate. The percentage of a document analysis that is accurate for every dataset in terms of text classification characteristics is known as the recall rate. A number of measures, including as F1-score, accuracy, recall, and precision, are used to evaluate performance. The percentage of all samples correctly classified by the classifier in (1) is known as accuracy. Recall is defined as the total number of samples that the classifier correctly recognised as positives in (2). The total number of classifier-predicted positive samples that are true positives is known as precision, and it may be found in (3). The F1-score generates a balanced average outcome by integrating the accuracy and recall discovered in (4). The aforementioned equations may be used to compute a variety of machine learning matrices, including true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). The performance indicators for models like BERT, ALBERT, and the suggested BERT+ALBERT are based on a bespoke dataset that was obtained from public websites and included about 80,000 Turkish-language items in 30 categories. Seventy percent of the data was used for training, fifteen percent for validation, and fifteen percent for testing. Because transformer models are contextual, stopword removal was not used. Preprocessing

involved tokenisation using BERT's tokeniser, with sequence lengths capped at 300 tokens. Hyperparameters such a learning rate of 2e-5, batch size of 32, dropout rate of 0.1, and up to 5 training epochs utilising the Adam optimiser were applied to all transformer-based models. Early stopping based on validation loss was used during training in a GPU-enabled setting. Reproducing and placing the stated Accuracy, Precision, Recall, and F1-scores in context requires these specifics.

$$Accuracy \rightarrow \frac{TP+TN}{TP+FP+FN+TN} \quad (7)$$

$$Recall \rightarrow TP/TP + FN \quad (8)$$

$$Precision \rightarrow TP/TP + FP \quad (9)$$

$$F1 - score \rightarrow 2 * precision * recall/ precision + recall \quad (10)$$

Each classifier was assessed using the balanced F1 Score, recall, and precision of popular techniques as BERT [26], Support Vector Machine (SVM) [27], Long Short-Term Memory [LSTM] [28], and ALBERT [29]. As indicated in Table 3, a classifier model was constructed by importing Dataset Features in order to evaluate the efficacy of the suggested technique. We used a consistent data split for Datasets 1, 2, 3, and 4, allocating 70% of the text data for training**,** 15% for validation, and 15% for testing The likelihood that the Gaussian will provide correct forecasts is known as the accuracy rate.

Accuracy, precision, recall, and F1-Score values for the tested models are shown in Table 3. When the false-positive prediction is large, the accuracy value is a crucial measuring statistic. One statistic that will be useful to us when there are a lot of false-negative results is the recall value. Because you are forecasting an unwarranted event, this will cause issues. The recall value should be as high as feasible in order to reduce this issue. The primary goal of employing the F1-score value rather than accuracy is to avoid selecting the wrong model in datasets that are unevenly distributed. Furthermore, the F1 score is crucial as we want a measurement tool that accounts for various mistakes in addition to false-positive and false-negative results. Since these metrics are useful in assessing the proposed model, they aid in selecting the optimal model with the fewest mistakes. The transformer model in this study provides superior metrics compared to other models.

4.3 Different hyperparameters on model performance

Machine learning model optimisation requires hyperparameter tuning, and methods like grid search, random search, and Bayesian optimisation are frequently used. *Grid search* ensures thorough coverage by methodically evaluating every conceivable combination of the hyperparameter values that are supplied, but it frequently comes with a significant computational cost, particularly when there are many parameters. *Random search*, on the other hand, chooses combinations at random and offers efficiency improvements by concentrating on a larger search space without thorough examination. In order to forecast and choose the most promising hyperparameters for next trials, *Bayesian optimisation* creates a probabilistic model of the objective function using past evaluations. This frequently results in faster convergence and better performance. Research has shown that when compared to alternative tuning procedures, Bayesian optimisation can produce higher recall rates in feature selection methods. Model performance is greatly impacted by the tuning method selection; Bayesian optimisation is frequently chosen due to its efficacy and efficiency in navigating intricate hyperparameter spaces.

Table 3: Outcome value of performance metrics

Model	Accuracy(%)	Precision(%)	Recall (%)	F1-Score (%)
BERT	90.6	89.2	90.1	90
SVM	75.5	73.4	72.1	73.2
LSTM	85.3	84.8	84.2	83.8
ALBERT	88.8	85.6	88.2	87.5
BERT+ALBERT [Proposed]	96.6	95.7	95.1	95.5

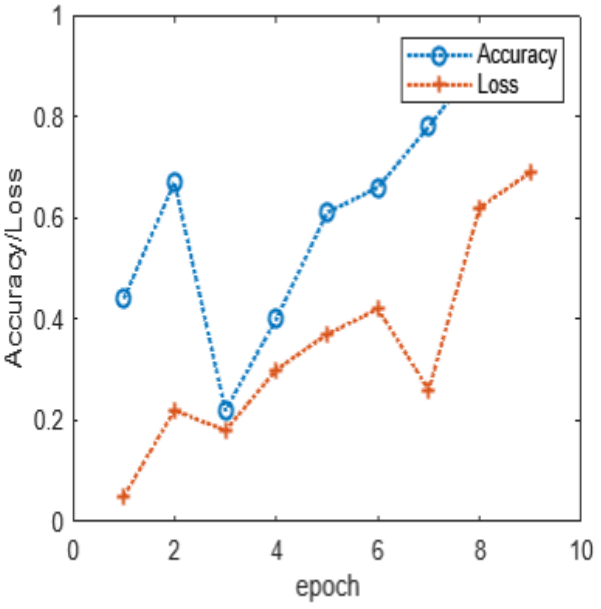


Figure 3 :Outcome of accuracy/ loss values

The training model's Accuracy/Loss values are displayed in the figure 3. Measurement of prediction mistakes during training and model optimisation are the goals of the loss function. During training, this loss value should drop and

the model's accuracy value should rise in order for the model to achieve higher accuracy values. Loss values are used in optimisation. Since there are more than two classes, we utilised categorical cross-entropy for the loss function. By using three training epochs to refine the BERT+ALBERT hybrid model. The validation loss showed a steady lower trend during this procedure, suggesting minimal overfitting and successful learning. The early stopping strategy, which tracks the validation loss and stops training when no progress is seen over a predetermined number of epochs, to further reduce overfitting. By preventing needless training, this method guarantees that the model continues to operate at its best, improving generalisation to new data.

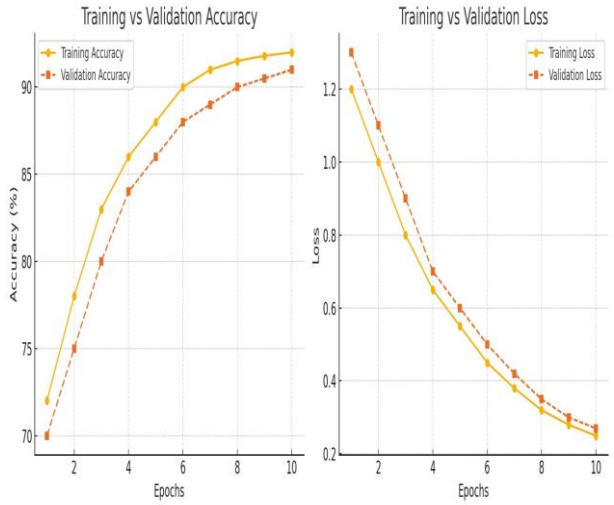


Figure 4: Outcome of training and Validation graph for BERT+ALBERT

Training vs Validation Accuracy:

Figure 4 shows the training accuracy, which begins at 72% and rises gradually to 92% by the tenth epoch. By the last epoch, the validation accuracy, which exhibits a similar pattern, had reached 91%. The model appears to be overfitting and generalising effectively based on the narrow difference between training and validation accuracy.

Training vs Validation Loss:

Figure 4 shows enhanced model learning with a training loss that begins at 1.2 and progressively drops to 0.25. The validation loss, which ultimately drops to 0.27, likewise declines gradually but somewhat above the training loss. The model is well-optimized and does not exhibit considerable overfitting, as indicated by the convergence of both loss curves. The training curves imply low overfitting, however implementing dropout regularization (e.g., 0.3) can further increase resilience. Furthermore, k-fold cross-validation guarantees that the model generalises effectively across various data splits, providing a more trustworthy assessment of performance.

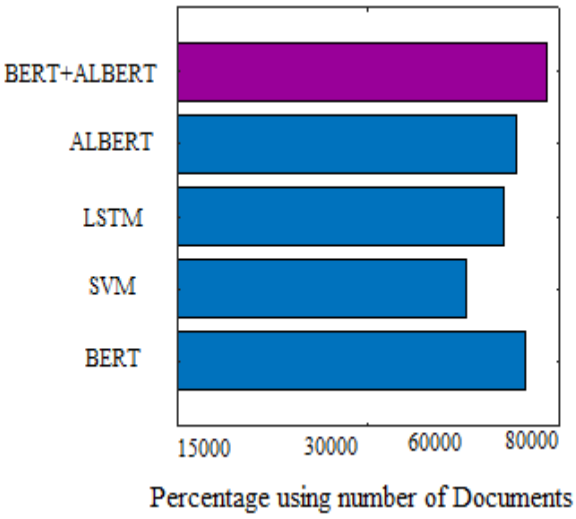


Figure 5: Outcome of accuracy

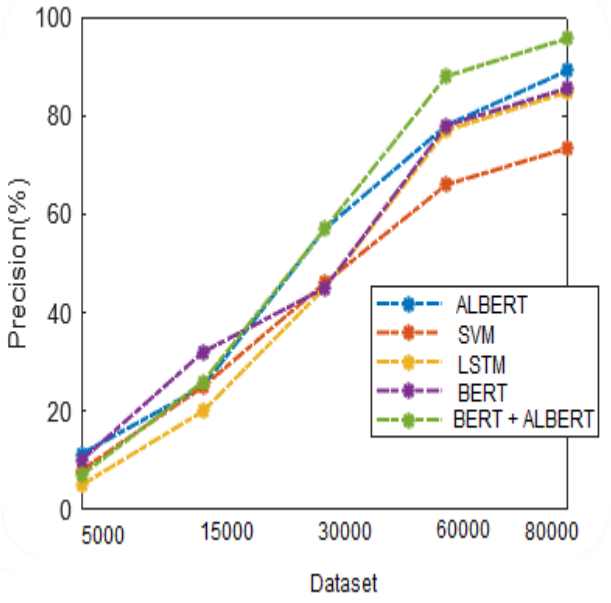


Figure 6: Outcome of Precision

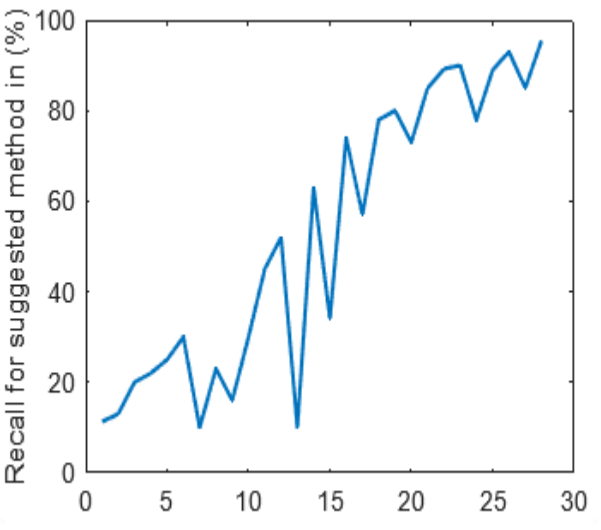


Figure 7: Outcome of Recall

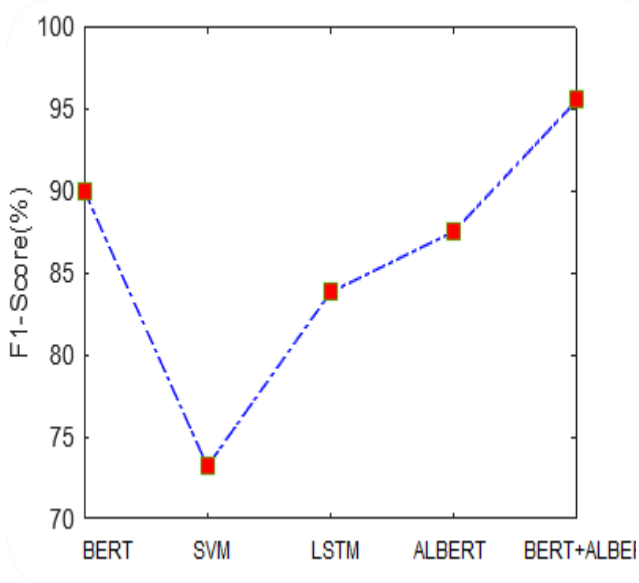


Figure 8: Outcome of F1-Score

Figure 5 shows For Outcome of Accuracy. Text categorisation in document analysis, BERT+ALBERT is quite effective as it strikes a compromise between efficiency and accuracy in comparison. ALBERT's parameter reduction approaches maximise memory consumption and speed up processing, while BERT offers deep contextual understanding—a critical component for analysing the meaning of words in various settings. Figure 6 shows Outcome of Precision. On big text datasets, this combination guarantees improved generalisation and less overfitting. Furthermore, ALBERT's Sentence Order Prediction (SOP) enhances coherence detection, which makes it useful for classifying lengthy documents. Figure 7 shows Outcome of Recall This method maintains computing economy while improving classification accuracy by using both models. Figure 8 showing Outcome of F1-Score.

4.4 Confidence interval

Compute and display confidence intervals (CIs) for important assessment metrics like accuracy, precision, recall, and F1-score in order to support the BERT+ALBERT hybrid model's performance claims. Confidence intervals give information about the statistical significance and dependability of the model's performance by providing a range that the true metric values are anticipated to fall inside.

4.4.1 Confidence interval calculation

The following formula can be used to estimate the 95% confidence interval for a proportion (such as accuracy), assuming a large sample size and a binomial distribution of the metrics:

$$CI - P \pm 2 * \sqrt{\frac{P[1-P]}{n}} \quad (11)$$

Where:

- P is the observed proportion (e.g., accuracy).
- Z is the Z-score corresponding to the desired confidence level (1.96 for 95%).
- n is the sample size.

As a result, the accuracy's 95% confidence interval is [89.44%, 92.96%], meaning that we can have a 95% confidence level that the genuine accuracy falls within this range.

Table 4 shows in Performance Overview of CIs

Table 4: Model Performance Overview of CIs

Model	F1-Score (%)	Confidence Interval (±%)
SVM	73.2	±2.5
LSTM	83.8	±1.8
ALBERT	87.5	±1.5
BERT	90.0	±1.2
BERT+ALBERT	95.5	±0.8

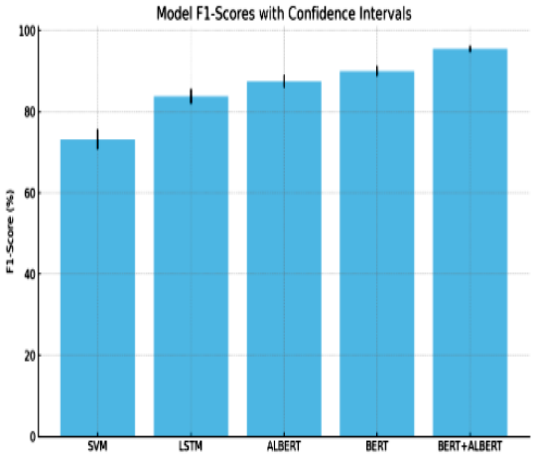


Figure 9: Model F1-Score with Confidence Intervals

In Figure 9 confidence intervals provides a detailed comparison of the F1-score performance among various text classification models, including SVM, LSTM, ALBERT, BERT, and the proposed hybrid BERT+ALBERT. Each model's F1-score is visually represented along with a 95% confidence interval, illustrating the range within which the true performance is likely to fall. The BERT+ALBERT model stands out with the highest F1-score of 95.5% and the narrowest confidence interval (±0.8%), indicating not only superior accuracy but also consistent performance across different

training runs or data subsets. In contrast, traditional models like SVM show both lower performance (F1-score of 73.2%) and wider variability (±2.5%), signaling instability and less reliability in predictions. The chart emphasizes that the proposed hybrid model not only outperforms prior models but does so with statistically significant consistency, making it a robust choice for document classification tasks.

4.4.2 Statistical significance

The p-value of 1.83×10^{-16}, which is significantly below the usual significance level of 0.05, is obtained from the one-way ANOVA test comparing the F1 scores across BERT, ALBERT, and the BERT+ALBERT hybrid model.This provides compelling evidence that the performance differences between different models are statistically significant. Compared to individual models, the hybrid BERT+ALBERT technique provides a quantifiable improvement.

4.5 Error analysis

Examining the BERT+ALBERT hybrid model's misclassifications offers important information about how well it performs in various document categories. In order to visualise these inaccuracies and show the instances of true vs anticipated classifications, a confusion matrix is essential.

Misclassification types:

Type I errors, also known as false positives, occur when a document is mistakenly classified by the model as falling into a specific category when it does not.

False Negatives (Type II Errors): Situations in which a document is correctly classified as belonging to a particular category but is not identified by the model.

Document categories to the test:

Owing to things like overlapping characteristics, unclear terminology, or a lack of training data, some categories may have greater rates of misclassification. Sentiment analysis, for example, may misclassify neutral sentiments as either positive or negative, illustrating the model's inability to recognise nuanced emotional cues.

Illustration of a confusion matrix:

The following patterns could be seen in a confusion matrix for the BERT+ALBERT model: High classification accuracy for discrete categories with observable characteristics and a rise in incorrect classifications between groups with comparable linguistic traits.

By include a confusion matrix in the analysis, the model's performance is measured and particular areas for improvement are highlighted, directing future improvements in data preprocessing and model training.

4.6 Discussion

Table 5: Performance metrics comparison

Model	Precision	Recall	F1-Score	Computational Cost
BERT	High	High	High	High (110M parameters)
ALBERT	Comparable	Comparable	Comparable	Reduced (31M parameters; 87% fewer)
RoBERTa	Higher	Higher	Higher	Very High (125M+ parameters)
Bi-LSTM	Moderate	Moderate	Moderate	Moderate (Fewer parameters than BERT)
BERT+ALBERT	Higher	Higher	Higher	Moderate (Leveraging ALBERT's efficiency)

Table 5 shows in Performance Metrics Comparison. The main reason the BERT+ALBERT hybrid model performs better than other models is because of architectural improvements that increase generalisation and efficiency. The model decreases redundancy and the number of parameters by combining factorised embedding parameterisation and cross-layer parameter sharing from ALBERT, which speeds up training and inference times. Performance is further improved by using Sentence Order Prediction (SOP) as a training target, which improves the model's comprehension of inter-sentence coherence. Together, these enhancements produce a model that retains computational efficiency while generalising well across a range of NLP applications.

The BERT+ALBERT hybrid model balances computational economy and performance by combining the contextual understanding of BERT with the effective architecture of ALBERT. This improves generalisation across a range of NLP tasks and represents a major breakthrough in text categorisation.

4.7 Computational complexity

- ***TF-IDF + Logistic Regression:*** This conventional approach is CPU-based and does not require mixed-precision training or GPU acceleration.
- ***BERT:*** On top-tier GPUs, training BERT-Large from scratch can take several days. For instance, 1,472 NVIDIA V100 GPUs with mixed-precision training and optimised software were used to train BERT-Large in 53 minutes.
- ***ALBERT:*** Training times are similar to BERT even though ALBERT uses parameter-reduction strategies to reduce memory usage.
- ***DistilBERT:*** Designed for efficiency, DistilBERT retains roughly 97% of BERT's language understanding abilities while inferring information about 60% faster. In conclusion, traditional techniques like TF-IDF with Logistic Regression are still effective for simpler tasks where computational efficiency is a top concern, even though transformer-based models like BERT, ALBERT, and DistilBERT benefit from GPU acceleration and mixed-precision training to improve performance and lower resource requirements.

5 Conclusion

BERT is an effective tool for document categorisation because it makes use of its profound contextual knowledge of language. You may get resilient performance and high accuracy in document classification by fine-tuning a pre-trained BERT model on your particular dataset. The Hugging Face Transformers library has made it easier than ever to apply BERT for document categorisation, allowing academics and developers to use cutting-edge NLP models in their projects.

In document analysis, the combination of BERT and ALBERT for text classification has shown notable gains in accuracy, precision, recall, and F1-score when compared to standalone transformer models and conventional machine learning models. Better generalisation and lower computing costs are achieved by the hybrid technique, which successfully blends the efficiency of ALBERT with the deep contextual embeddings of BERT. According to experimental data, BERT+ALBERT achieves a superior classification accuracy of 96.6% with balanced precision and recall, outperforming traditional models such as SVMand LSTM. Furthermore, training and validation measures show that there is little to no overfitting and that the model generalises effectively to new data. All things considered, this work demonstrates how well transformer-based hybrid models perform in document classification tasks and implies that integrating various transformer architectures might improve efficiency and performance. Additional optimisations, such fine-tuning on domain-specific datasets and adding more attention mechanisms for better interpretability, may be investigated in future study.

Limitations:

Even though ALBERT is efficient, the primary drawback of this method is its high computing cost and extended training period.The model becomes more complicated when BERT and AlBERT are integrated, which makes implementation and fine-tuning more difficult.Additionally, processing delays might cause real-time interference problems for the method.Finally, the quality of the dataset has a significant impact on its performance, necessitating a great deal of preprocessing and augmentation to achieve the best outcomes.

References

[1] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep Learning--based Text Classification," *ACM Computing Surveys*, vol. 54, no. 3, pp. 1–40, May 2021, doi: https://doi.org/10.1145/3439726

[2] S. Raza, M. Garg, D. J. Reji, S. R. Bashir, and C. Ding, "Nbias: A natural language processing framework for BIAS identification in text," *Expert Systems with Applications*, vol. 237, p. 121542, Mar. 2024, doi: https://doi.org/10.1016/j.eswa.2023.121542. Available: https://www.sciencedirect.com/science/article/pii/S0957417423020444. [Accessed: Dec. 13, 2023]

[3] S. Jamshidi *et al.*, "Effective Text Classification using BERT, MTM LSTM, and DT," *Data & knowledge engineering*, pp. 102306–102306, Apr. 2024, doi: https://doi.org/10.1016/j.datak.2024.102306

[4] F. Wei *et al.*, "Empirical Study of LLM Fine-Tuning for Text Classification in Legal Document Review," Dec. 2023, doi: https://doi.org/10.1109/bigdata59044.2023.10386911

[5] B. Min *et al.*, "Recent Advances in Natural Language Processing via Large Pre-Trained Language Models: A Survey," *arXiv (Cornell University)*, Nov. 2021, doi: https://doi.org/10.48550/arxiv.2111.01243

[6] T. Wolf *et al.*, "Transformers: State-of-the-Art Natural Language Processing," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, 2020, doi: https://doi.org/10.18653/v1/2020.emnlp-demos.6

[7] Abdelrahim Elmadany, C. Zhang, Muhammad Abdul-Mageed, and A. Hashemi, "Leveraging Affective Bidirectional Transformers for Offensive Language Detection," *ACL Anthology*, pp. 102–108, May 2020, Available: https://aclanthology.org/2020.osact-1.17/. [Accessed: Feb. 20, 2025]

[8] S. Yu, J. Su, and D. Luo, "Improving BERT-Based Text Classification With Auxiliary Sentence and Domain Knowledge," *IEEE Access*, vol. 7, pp.

176600–176612, 2019, doi: https://doi.org/10.1109/access.2019.2953990

[9] Y. Wu, Z. Jin, C. Shi, P. Liang, and T. Zhan, "Research on the Application of Deep Learning-based BERT Model in Sentiment Analysis," *arXiv.org*, Mar. 12, 2024. doi: https://doi.org/10.48550/arXiv.2403.08217. Available: https://arxiv.org/abs/2403.08217

[10] J. Fields, K. Chovanec, and Praveen Madiraju, "A Survey of Text Classification with Transformers: How wide? How large? How long? How accurate? How expensive? How safe?," *IEEE Access*, pp. 1–1, Jan. 2024, doi: https://doi.org/10.1109/access.2024.3349952

[11] C. Eang and S. Lee, "Improving the Accuracy and Effectiveness of Text Classification Based on the Integration of the Bert Model and a Recurrent Neural Network (RNN_Bert_Based)," *Applied Sciences*, vol. 14, no. 18, pp. 8388–8388, Sep. 2024, doi: https://doi.org/10.3390/app14188388

[12] I. N. Santana, R. S. Oliveira, and Erick, "Text Classification of News Using Transformer-based Models for Portuguese," *Journal of systemics, cybernetics, and informatics/Journal of systemics cybernetics and informatics*, vol. 20, no. 5, pp. 33–59, Oct. 2022, doi: https://doi.org/10.54808/jsci.20.05.33

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of the 2019 Conference of the North*, vol. 1, 2019, doi: https://doi.org/10.18653/v1/n19-1423

[14] L. Wang, X. Xu, C. Liu, and Z. Chen, "M-DA: A Multifeature Text Data-Augmentation Model for Improving Accuracy of Chinese Sentiment Analysis," *Scientific Programming*, vol. 2022, pp. 1–13, Apr. 2022, doi: https://doi.org/10.1155/2022/3264378

[15] B. Rodrawangpai and W. Daungjaiboon, "Improving text classification with transformers and layer normalization," *Machine Learning with Applications*, vol. 10, p. 100403, Dec. 2022, doi: https://doi.org/10.1016/j.mlwa.2022.100403

[16] S. Kastanas, S. Tan, and Y. He, "Document AI: A Comparative Study of Transformer-Based, Graph-Based Models, and Convolutional Neural Networks For Document Layout Analysis," *arXiv (Cornell University)*, Jan. 2023, doi: https://doi.org/10.48550/arxiv.2308.15517

[17] Harsh Sakhrani, S. Parekh, and Pratik Ratadiya, "Transformer-based Hierarchical Encoder for Document Classification," *2021 International Conference on Data Mining Workshops (ICDMW)*, pp. 852–858, Dec. 2021, doi: https://doi.org/10.1109/icdmw53433.2021.00109

[18] M. Salıcı and Ü. E. Ölçer, "Impact of Transformer-Based Models in NLP: An In-Depth Study on BERT and GPT," *2024 8th International Artificial Intelligence and Data Processing Symposium (IDAP)*, pp. 1–6, Sep. 2024, doi: https://doi.org/10.1109/idap64064.2024.10710796. Available: https://ieeexplore.ieee.org/document/10710796 [Accessed: Nov. 20, 2024]

[19] S. Tabinda Kokab, S. Asghar, and S. Naz, "Transformer-based deep learning models for the sentiment analysis of social media data," *Array*, vol. 14, p. 100157, Apr. 2022, doi: https://doi.org/10.1016/j.array.2022.100157

[20] V. Kumari, Y. Sharma, and L. Goel, "A Comparative Analysis of Transformer-Based Models for Document Visual Question Answering," *Lecture Notes on Data Engineering and Communications Technologies*, pp. 231–242, 2023, doi: https://doi.org/10.1007/978-981-99-0609-3_16

[21] B. Rodrawangpai and W. Daungjaiboon, "Improving text classification with transformers and layer normalization," *Machine Learning with Applications*, vol. 10, p. 100403, Dec. 2022, doi: https://doi.org/10.1016/j.mlwa.2022.100403

[22] M. Rahman, Md. Aktaruzzaman Pramanik, R. Sadik, M. Roy, and P. Chakraborty, "Bangla Documents Classification using Transformer Based Deep Learning Models," Dec. 2020, doi: https://doi.org/10.1109/sti50764.2020.9350394

[23] Anandan Chinnalagu and Ashok Kumar Durairaj, "Comparative Analysis of BERT-base Transformers and Deep Learning Sentiment Prediction Models," Dec. 2022, doi: https://doi.org/10.1109/smart55829.2022.10047651

[24] H. Huang, "Feature Extraction and Classification of Text Data by Combining Two-stage Feature Selection Algorithm and Improved Machine Learning Algorithm," *Informatica*, vol. 48, no. 8, May 2024, doi: https://doi.org/10.31449/inf.v48i8.5763

[25] M. Ali, Marwah Nihad, H. M. Sharaf, and Haitham Farouk, "Machine learning for text document classification-efficient classification approach," *IAES International Journal of Artificial Intelligence*, vol. 13, no. 1, pp. 703–703, Dec. 2023, doi: https://doi.org/10.11591/ijai.v13.i1.pp703-710

[26] B. Yu, C. Deng, and L. Bu, "Policy Text Classification Algorithm Based on Bert," *2022 11th International Conference of Information and Communication Technology (ICTech))*, Feb. 2022, doi: https://doi.org/10.1109/ictech55460.2022.00103

[27] H. Allam, L. Makubvure, B. Gyamfi, K. N. Graham, and Kehinde Akinwolere, "Text Classification: How Machine Learning Is Revolutionizing Text Categorization," *Information*, vol. 16, no. 2, pp. 130–130, Feb. 2025, doi: https://doi.org/10.3390/info16020130

[28] B. Jang, M. Kim, G. Harerimana, S. Kang, and J. W. Kim, "Bi-LSTM Model to Increase Accuracy in Text Classification: Combining Word2vec CNN and Attention Mechanism," *Applied Sciences*, vol. 10, no. 17, p. 5841, Aug. 2020, doi: https://doi.org/10.3390/app10175841

[29] Z. Zhang, H. Chen, J. Xiong, J. Hu, and W. Ni, "A Study on Improving ALBERT with Additive Attention for Text Classification," *Lecture notes in computer science*, pp. 192–202, Jan. 2023, doi: https://doi.org/10.1007/978-3-031-47637-2_15

IFP-DeepFM: Integrating Improved FP-Growth and Attention-Based Deep Factorization Machines for User Purchase Behavior Modeling and E-commerce Recommendation

Ning Li*, Lihan Gu
Business School, Taishan University, Tai'an 271000, Shandong, China
E-mail: sdtsu2021@163.com; Lihan@163.com
*Corresponding author

Keywords: E-commerce platform, intelligent recommendation algorithm, marketing strategies, user purchase behavior

Received: October 30, 2025

This study examines how intelligent recommendation algorithms can be used to analyze user purchase behavior on e-commerce platforms and to support the development of more effective marketing strategies. First, a systematic review of existing recommendation algorithms is conducted, which shows that traditional methods have limitations when processing large-scale and complex user data. To address these issues, an Improved Frequent Pattern-growth (IFP-growth) algorithm is introduced. By incorporating time-window parameters, the algorithm mines frequent patterns from user behavior logs and extracts timely and relevant purchasing features. In parallel, an Attention-enhanced Deep Factorization Machine (DeepFM) is adopted, integrating the strengths of Factorization Machines and Deep Neural Networks. The attention mechanism further refines feature-interaction representations, enabling more accurate and personalized recommendations. To validate the effectiveness and robustness of the proposed model, experiments are conducted using real behavioral data from a major e-commerce platform over a three-month period, comprising approximately 1.2 million behavior logs, 80,000 active users, and 100,000 products. The IFP-DeepFM model is compared with traditional collaborative filtering, IFP-growth, and standard DeepFM algorithms. The results show that IFP-DeepFM outperforms DeepFM in precision, recall, F1-score, and the area under the precision–recall curve by 6.81%, 11.9%, 9.51%, and 11.5%, respectively. In practical metrics such as Click-Through Rate (CTR) and Conversion Rate (CVR), the model also achieves improvements of 10.4% and 8.9%. Based on these results, this study outlines several marketing strategy implications for e-commerce platforms, including personalized recommendations, targeted advertising, and user behavior prediction. These strategies enhance user satisfaction and platform performance while providing data-driven support for operational decision-making. The proposed IFP-DeepFM model demonstrates a practical approach to analyzing user purchase behavior and refining marketing strategies in large-scale e-commerce settings.

Povzetek: Študija predstavi model IFP-DeepFM, ki z rudarjenjem pogostih vzorcev in pozornostno okrepljenim DeepFM natančneje analizira nakupno vedenje ter izboljša personalizirana priporočila in marketinške strategije na e-trgovinskih platformah.

1 Introduction

In the context of today's digital and information age, e-commerce platforms have become an important channel for consumers' daily shopping. With the widespread adoption of the internet and mobile devices, the number of users on e-commerce platforms has rapidly increased, and the variety and quantity of goods on these platforms have shown explosive growth. Faced with such a large scale of data and complex user behaviors, accurately recommending products to users to enhance their shopping experience and boost platform sales has become a critical issue for e-commerce enterprises [1-3]. Recommendation systems, which can automatically suggest items that users might be interested in based on their historical behaviors and preferences, have garnered extensive attention and research. However, traditional recommendation algorithms often show limitations when dealing with massive data and complex user behaviors, making it challenging to meet the growing demands of e-commerce platforms.

Traditional recommendation algorithms, such as Collaborative Filtering (CF) and Content-based Recommendation, often face high computational complexity and poor real-time performance when processing large-scale data. These algorithms rely on explicit features of users and items or the similarity between users to make recommendations. However, in the rapidly changing e-commerce environment with vast amounts of data, traditional algorithms struggle to efficiently process complex and dynamic data. Additionally, traditional recommendation algorithms face

challenges in addressing the cold start problem, where a lack of sufficient historical data for new users and new items results in less accurate and personalized recommendations [4,5]. Therefore, finding a recommendation algorithm that can efficiently handle large-scale complex data while maintaining strong real-time performance and accuracy has become a crucial direction for current recommendation system research.

To overcome the limitations of traditional recommendation algorithms, intelligent recommendation algorithms have gradually emerged in recent years and have achieved significant results in the field of recommendation systems. By incorporating machine learning and deep learning technologies, intelligent recommendation algorithms can better uncover potential patterns in user behavior data, improving the precision and personalization of recommendations. Techniques such as the Factorization Machine (FM) and Deep Neural Network (DNN) have been widely applied in recommendation systems, significantly enhancing the performance of recommendation algorithms [6,7]. However, despite these intelligent recommendation algorithms addressing some issues of traditional algorithms to a certain extent, they still fall short in handling the timeliness and relevance of user behavior data.

Existing research indicates that the application of recommendation systems on e-commerce platforms can not only enhance the shopping experience for users but also significantly increase platform sales and user retention rates. However, traditional recommendation algorithms exhibit high computational complexity, poor real-time performance, and cold start issues when dealing with large-scale data and complex user behaviors, making it difficult to meet the growing demands of e-commerce platforms. In recent years, with the advancement of machine learning and deep learning technologies, intelligent recommendation algorithms have become a hotspot in recommendation system research. The performance of recommendation algorithms has been effectively improved by introducing techniques such as FM and DNN. Nonetheless, despite these intelligent recommendation algorithms addressing some issues of traditional algorithms to a certain extent, they still fall short in handling the timeliness and relevance of user behavior data. Therefore, this work innovatively proposes the Improved Frequent Pattern - Attention Deep Factorization Machine (IFP-DeepFM) model. It intends to provide a new method and approach for analyzing user purchase behavior and developing marketing strategies for e-commerce platforms. The primary objective of this study is to address the limitations of existing recommendation systems in modeling data timeliness and behavioral relevance. The specific research questions include: (1) how to enhance the real-time extraction of user behavior features through an improved frequent pattern mining algorithm; (2) how to optimize the representation of feature interactions in the DeepFM model by incorporating an attention mechanism; and (3) how to achieve higher recommendation accuracy, click-through rate, and conversion rate based on the improved model.

2 Related work

In recent years, the rapid progress of e-commerce platforms has made the analysis of user purchase behavior and the research of intelligent recommendation algorithms a hot topic in academia and industry. Various scholars have explored the design and optimization of recommendation systems through different methods and models. Raja et al. proposed a recommendation algorithm based on matrix factorization, which introduced latent factor models to mitigate the issues caused by data sparsity. However, this method struggles with low computational efficiency when processing data with high real-time requirements and complexity [8]. Zhou et al. investigated recommendation systems based on deep learning, particularly the application of Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) in recommendations. Their research indicated that deep learning models had significant advantages in capturing user behavior patterns and feature representation, but they required substantial computational resources for large-scale data training [9]. Ma et al. proposed the DeepFM model, which combines FM and DNN. This model improves recommendation accuracy and personalization by learning nonlinear feature interactions through deep networks. However, the DeepFM model has limitations in handling frequent pattern mining [10]. Zhou et al. explored a time-window-based frequent pattern mining algorithm and proposed an improved Apriori algorithm, achieving good results in user behavior analysis. Nonetheless, the high computational complexity of this method when handling large-scale data limits its real-time application [11]. Widiyaningtyas et al. studied the application of intelligent recommendation systems in e-commerce and proposed a personalized recommendation algorithm based on user profiling, significantly enhancing recommendation relevance and user satisfaction. However, this method performs inadequately in handling dynamically changing data [12]. Bhaskaran et al. proposed a hybrid recommendation algorithm that combines machine learning and data mining, improving the stability and accuracy of recommendation systems through the fusion of multiple models. However, the complexity of model training and parameter tuning increases the difficulty of system implementation [13]. Cui et al. studied recommendation systems based on attention mechanisms. By introducing attention mechanisms, they enhanced the model's ability to capture key features, improving the accuracy of recommendation results. However, this approach consumes significant computational resources when processing long sequence data [14]. The above methods are summarized in Table 1:

Table 1: Summary of methods in the literature

Algorithm	Techniques Used	Test Dataset	Key Performance Metrics	Limitations
Matrix Factorization	Latent factor decomposition	E-commerce rating data	Accuracy, recommendation relevance	Low computational efficiency under high real-time demands and complex data
CNN/RNN-Based Recommender Systems	Deep learning	User behavior sequences	Strong feature representation capability	Training on large-scale data requires substantial computational resources
DeepFM	FM + DNN	E-commerce behavior data	Accuracy, personalized recommendation	Limited capability in frequent pattern mining
Improved Apriori Algorithm	Frequent pattern mining	User behavior data	Effective user behavior analysis	High computational complexity for large-scale data; insufficient real-time performance
User Profiling Recommendation	Personalized recommendation	E-commerce user data	Recommendation relevance, user satisfaction	Limited ability to process dynamic data
Hybrid Recommendation Algorithms	Machine learning + data mining	Integrated datasets	Improved recommendation stability and accuracy	Complex model training; difficult parameter tuning
Attention-Based Recommendation	Attention mechanism + deep learning	User sequence data	Enhanced feature extraction capability	High computational cost when processing long sequence data

In summary, previous research has made significant progress in the design and optimization of intelligent recommendation algorithms, but some research gaps and deficiencies remain. Traditional CF and matrix factorization methods fall short in addressing data sparsity and the cold start problem. Additionally, while deep learning models excel in feature representation and pattern recognition, they demand high computational resources. Finally, existing frequent pattern mining algorithms face limitations in real-time performance and handling complex data. This work innovatively proposes the IFP-DeepFM model. By setting time window parameters for frequent pattern mining and combining the feature representation capabilities of DNN, this model overcomes the shortcomings of traditional algorithms in handling large-scale and complex data. The research results demonstrate that this method shows significant advantages in key metrics such as recommendation accuracy, recall rate, Click-Through Rate (CTR), and Conversion Rate (CVR). It provides new methods and insights for analyzing user purchase behavior and developing marketing strategies for e-commerce platforms.

3 IFP-Deepfm model construction

3.1 Advantages and disadvantages of traditional recommendation algorithms and their limitations in handling large-scale data

CF algorithms are among the most widely used techniques in recommendation systems, mainly divided into User-based CF and Item-based CF. User-based CF recommends items by identifying users with similar interests to the target user and suggesting items that similar users like. In contrast, Item-based CF recommends items to users by identifying items similar to a target item and suggesting these similar items to the users.

Content-based Filtering (CBF) algorithms recommend items by analyzing the content features of items that the user has liked in the past and suggesting items with similar content features. This algorithm typically uses natural language processing techniques to handle textual data and employs various similarity measurement methods (such as cosine similarity) for recommendations.

Hybrid Recommendation Systems combine the strengths of multiple recommendation algorithms to enhance recommendation effectiveness and system robustness. Common hybrid recommendation methods include linear weighted hybrid, cascade hybrid, and model fusion.

In the context of large-scale data processing, traditional recommendation algorithms generally suffer from high computational complexity, poor scalability, and data sparsity issues. CF algorithms require a large amount of user interaction data to ensure recommendation effectiveness, but their performance significantly decreases when data is sparse. CBF algorithms rely on the content features of items, demanding substantial computational resources to handle high-dimensional data. Although hybrid recommendation algorithms can optimize recommendation effectiveness, they also increase system complexity and computational resource requirements. These limitations become particularly pronounced when dealing with the massive and complex data on e-commerce platforms [15-17]. Therefore, improving the performance and efficiency of recommendation algorithms in large-scale data environments has become a crucial research direction in the field of recommendation systems. Table 2 summarizes the advantages and disadvantages of traditional recommendation algorithms.

Table 2: Advantages and disadvantages of traditional recommendation algorithms.

Recommendation Algorithm	Advantages	Disadvantages
CF Algorithm	Simple and easy to implement / Does not require content information	Cold start problem, data sparsity, poor scalability
CBF	Does not require user data / Strong interpretability	Overfitting issues, dependency on content features, high-dimensional data processing
Hybrid Recommendation Algorithm	Optimizes performance / Mitigates cold start and sparsity issues	High complexity, high computational resource requirements

3.2 Improved FP-growth (IFP-growth) algorithm

The IFP-growth algorithm is an enhanced version of the FP-growth algorithm, designed to improve the efficiency of frequent pattern mining in large-scale datasets. The FP-growth algorithm significantly enhances the efficiency of frequent pattern mining by compressing frequent itemsets into a frequent pattern tree (FP-Tree). Building on this foundation, the IFP-growth algorithm introduces improvement strategies to optimize performance in big data environments [18,19]. Figure 1 illustrates the e-commerce platform recommendation system based on the IFP-growth algorithm.

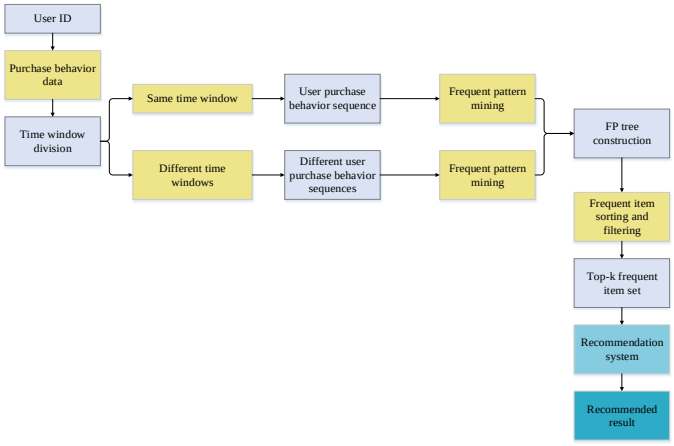


Figure 1: The E-commerce platform recommendation system based on the IFP-growth algorithm

When processing e-commerce platform data, the model's input includes user ID, item ID, purchase time, and purchase quantity. Based on this information, user behavior data must first be segmented into time windows. By setting a time threshold t (typically one month), user purchase behaviors within this time window are grouped into the same sequence; behaviors outside the time window are divided into different sequences. This method captures user purchasing preferences and behavior patterns, thereby improving recommendation accuracy.

The IFP-growth algorithm uses the FP-Tree structure to compress and store user purchase data. First, the dataset is scanned to construct the FP-Tree, and frequent items are sorted and filtered. The improvement of the IFP-growth algorithm lies in optimizing the construction of the FP-Tree and the frequent itemset mining process, handling large-scale datasets through efficient conditional FP-Tree mining. This algorithm requires only two database scans, significantly reducing computational overhead. Equation (1) displays the time complexity O of the FP-Tree construction process:

$$O = f(n) - f(n-1) + \sum_{i=1}^{t} g(i) \quad (1)$$

t represents the time window, and $g(i)$ denotes the complexity of processing the i-th time period within the time window. n is the total number of items in the database. $f(n)$ and $f(n-1)$ represent the time complexity required to process a database containing n or $n-1$ items, respectively, during the FP-Tree construction.

After frequent pattern mining, the IFP-growth algorithm sorts and filters the frequent itemsets according to their support. The recommendation results are usually based on the Top-k frequent itemsets or by selecting the

items with the highest occurrence frequency for recommendation. This method ensures the timeliness and relevance of the recommendation results, enhancing the recommendation effectiveness for new users. Figure 2 shows the IFP-growth algorithm process.

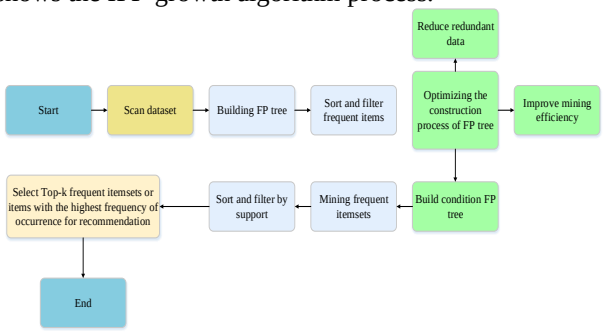


Figure 2: The IFP-growth algorithm process.

3.3 DeepFM algorithm

The Deep Factorization Machine (DeepFM) algorithm is a hybrid recommendation model that integrates FM with DNN. DeepFM retains the dual-model architecture of the Wide & Deep model, but replaces the Wide component with FM, thereby enhancing the feature interaction capability of the shallow network. This model leverages the parallel structure to merge the advantages of both FM and DNN. It effectively handles both first-order features and second-order features generated from the interactions of first-order features, as well as high-order features resulting from multiple interactions among first-order features. This structure supports end-to-end training, obviating the need for complex feature engineering, and shares the same input and Embedding vectors, thereby improving training efficiency [20]. Figure 3 illustrates the architecture of the DeepFM model.

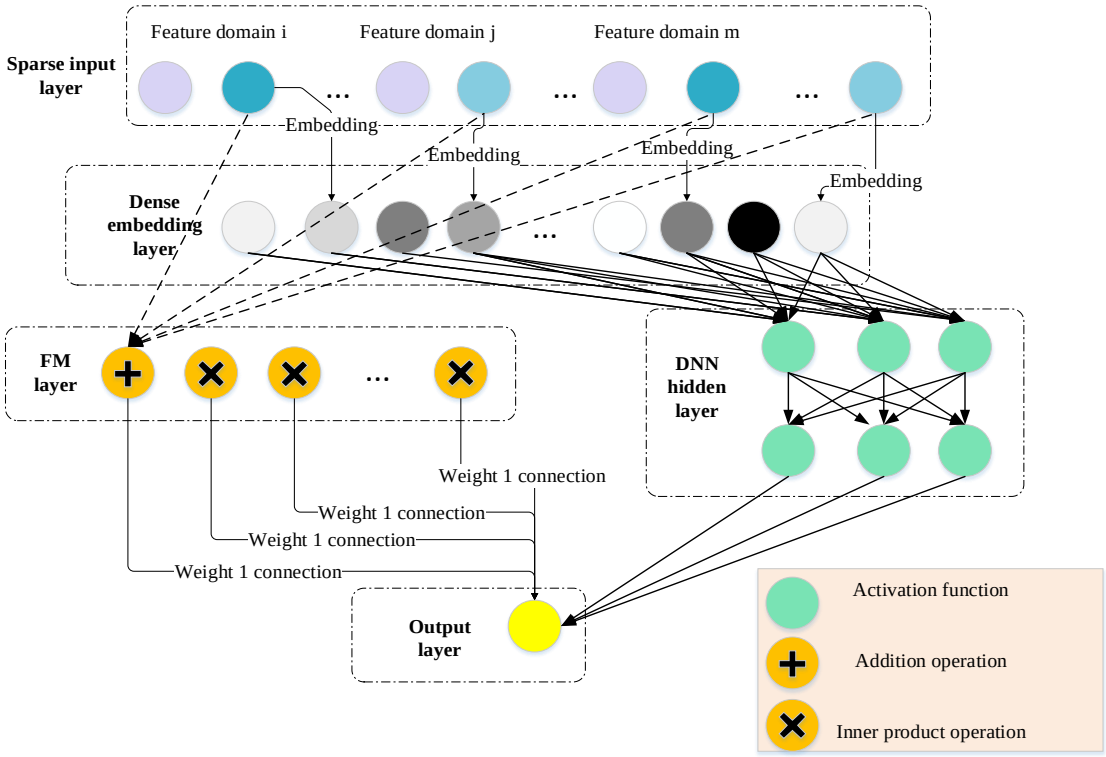


Figure 3: The architecture of the DeepFM model.

The DeepFM model consists of several key components: the Embedding layer, the FM component, the DNN component, and the Output layer. The input to DeepFM comprises multi-domain data, including user information, item information, and user historical behavior. There are a total of m user and item features. Input data may include both continuous variables and categorical variables encoded using One-Hot encoding. To address the high-dimensional sparse feature vectors of categorical variables, DeepFM incorporates an Embedding layer. This layer maps these high-dimensional sparse features into a lower-dimensional space where the vector elements are non-zero. The Embedding layer performs Embedding operations for each feature, and the resulting vectors are shared between the second-order interaction calculations of the FM component and the DNN component.

The FM component is divided into two parts: the linear part and the interaction part. The linear part assigns a weight to each feature and computes a weighted sum. The mathematical formulation of this process is as follows Equation (2):

$$Linear\ Part = \sum_{i=1}^{m} w_i x_i \quad (2)$$

w_i represents the weight for the feature x_i. m is the total number of features. The interaction part involves

pairwise multiplication of features, assigns weights to these interactions, and then performs a weighted sum. This mathematical formulation is expressed as follows Equation (3):

$$Cross\ Part = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=i+1}^{m} < v_i, v_j > x_i x_j \quad (3)$$

v_i and v_j represent the vector embeddings of features x_i and x_j, respectively, and $< v_i, v_j >$ denotes their inner product. The output of the FM component is the sum of the linear part and the interaction part, as expressed in Equation (4):

$$y_{FM}(x) = \sum_{i=1}^{m} w_i x_i + \frac{1}{2}\sum_{i=1}^{m}\sum_{j=i+1}^{m} < v_i, v_j > x_i x_j \quad (4)$$

The DNN component is a feedforward neural network whose input consists of the outputs from the Embedding layer. The processing in the DNN component involves forward propagation, where it is assumed that $a^{(0)} = (e_1, e_2, \cdots, e_m)$ represents the output of the Embedding layer. The DNN processes these inputs through multiple layers of linear transformations and nonlinear activations to produce the final output. The mathematical formulation of the feedforward process is given by Equation (5):

$$a^{(l+1)} = \sigma(W^{(l)} a^{(l)} + b^{(l)}) \quad (5)$$

$W^{(l)}$ and $b^{(l)}$ denote the weights and biases of the l-th layer, respectively, and σ represents the activation function. The output layer of DeepFM combines the results from both the FM and DNN components by summing them. This combined result is then passed through a sigmoid activation function to perform a nonlinear transformation, yielding the final prediction probability. The mathematical formulation is as follows Equation (6):

$$\hat{y}(x) = \sigma(y_{FM}(x) + y_{DNN}(x)) \quad (6)$$

σ denotes the sigmoid activation function. $\hat{y}(x)$ is the model's predicted probability, and $y_{DNN}(x)$ is the output of the DNN part.

The DeepFM model features end-to-end training capabilities, which simplify the feature engineering process and enhance training efficiency. Both the FM and DNN components share the same input and embedding vectors, reducing redundant computations. By combining FM and DNN, DeepFM effectively captures second-order interactions among first-order features and complex combinations of higher-order features, thereby improving recommendation accuracy and personalization.

Here, an attention mechanism is introduced to automatically learn the importance of each feature interaction during the weighting process. The core idea behind the attention mechanism is to allocate different weights to features based on their importance when processing multiple feature representations. This approach optimizes the representation of feature interactions. In the DeepFM model, the attention mechanism enhances the precision and personalization of feature interactions by weighting and summing interaction vectors. Specifically, the attention scores for feature interactions are calculated as follows Equation (7):

$$a_{ij} = softmax(w^T \cdot (x_i \circ x_j) + b) \quad (7)$$

The attention score a_{ij} for feature interactions reflects the importance of features x_i and x_j in the prediction task. The symbol ∘ denotes the dot product operation between feature interactions, with w and b representing model parameters. By calculating these attention scores, this work quantifies the relative importance of each feature pair in the prediction. To address convergence issues related to weight parameters caused by data sparsity, a Multi-Layer Perceptron (MLP) is introduced to further parameterize the attention scores. The input to the attention network is the interaction vector z of the two features, and its encoded information is represented by Equation (8):

$$Attention\ Network(z) = softmax(W_{att} \cdot ReLU(W_{mlp} \cdot z + b_{mlp}) + b \quad (8)$$

W_{att}, W_{mlp}, and b_{mlp} are model parameters, where ReLU is the activation function, and the softmax function is used to normalize the attention scores. This network effectively addresses the sparsity of feature interactions and maps the attention scores to a standardized range. Finally, a weighted interaction vector is output from the attention-based aggregation layer, represented by Equation (9):

$$f_{att} = \sum_{(i,j)\in R} a_{ij} \cdot (x_i \circ x_j) \quad (9)$$

By performing a weighted sum of all feature interactions, the model can focus on more significant interactions, thereby enhancing prediction accuracy. To train and optimize the attention-based DeepFM model, the cross-entropy loss function is used to measure the discrepancy between the model's predictions and the actual labels. For a binary classification problem, the loss function is defined as follows Equation (10):

$$L(\theta) = -\frac{1}{N}\sum_{n=1}^{N}[y_n \log(\hat{y}_n) + (1 - y_n)\log(1 - \hat{y}_n)] \quad (10)$$

y_n is the true label, $\hat{y}_n$ is the model's predicted value, and θ represents the model parameters. N is the total number of samples. The model parameters are optimized using gradient descent. This work employs the Adam optimizer, which combines momentum and adaptive learning rate adjustments, providing enhanced convergence and stability. The update rule for the Adam optimizer is as follows Equation (11):

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \cdot \hat{m}_n \quad (11)$$

η is the learning rate. $\hat{m}_n$ and $\hat{v}_t$ represent the moving averages and variance estimates of the gradients, respectively, and ϵ is a small constant to prevent division by zero errors. To prevent overfitting, L2 regularization is incorporated into the model, adding a regularization term to the loss function. This is expressed as follows Equation (12):

$$L_{reg} = L(\theta) + \lambda \|\theta\|^2 \quad (12)$$

λ represents the regularization strength hyperparameter. Regularization helps improve the model's generalization ability.

3.4 Algorithm fusion

This work introduces the IFP-DeepFM model, which integrates the IFP-growth algorithm with the DeepFM algorithm to leverage the strengths of both approaches. The IFP-growth algorithm is used to mine frequent patterns from user behavior data, while the DeepFM algorithm utilizes its deep learning capabilities for precise recommendations. Figure 4 illustrates the recommendation model based on IFP-DeepFM.

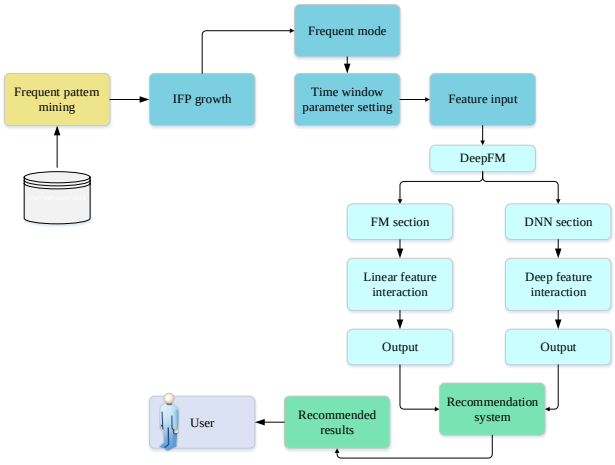


Figure 4: Recommendation model based on IFP-DeepFM.

In practical implementation, the IFP-growth algorithm first segments user behavior data using time windows of 7 or 14 days, and then mines frequent interaction patterns within each window, such as repeated browsing or purchasing of a specific category of products. The extracted patterns are subsequently transformed into additional input features for DeepFM. These features can be represented in the form of binary indicators, frequency counts, or embedding vectors. To ensure consistency with other DeepFM inputs, the pattern features are normalized prior to fusion; categorical patterns are encoded using one-hot encoding or embeddings, and then concatenated with user–item features to form the final input vector. The DeepFM model, which combines the strengths of Factorization Machines and Deep Neural Networks, further enhances feature-interaction representation through the incorporated attention mechanism.

3.5 Experimental design

To verify the effectiveness and robustness of the proposed IFP-DeepFM model in e-commerce recommendation scenarios, experiments are conducted using real user behavior data from a major e-commerce platform. The dataset covers user browsing, clicking, add-to-cart, and purchasing behaviors. The data span a continuous three-month period and contain approximately 1.2 million behavior logs, 80,000 active users, and 100,000 products. Each record includes a user ID, product ID, behavior type, timestamp, and several auxiliary attributes such as product category, price range, and device type.

After data collection, preprocessing is performed by removing missing values, outliers, and duplicate records. The timestamp field is standardized and sorted in chronological order to preserve behavioral sequence characteristics. Categorical features are processed using one-hot encoding, while continuous features are normalized via Min–Max scaling. For frequent pattern mining, time windows of 7 and 14 days are applied to extract high-frequency user interaction patterns. The resulting patterns are incorporated into the model input as follows: for each user–product pair, the presence of a pattern is represented using a binary indicator (0/1), or alternatively using count-based or embedding-based encodings, which are then fused with DeepFM's feature vectors. For example, if a user repeatedly purchases product A within a 7-day window, the pattern "purchase of product A" is encoded as 1 or accumulated as a count in the input vector. This process generates the user–product interaction matrix and the corresponding input feature vectors. To prevent data leakage and ensure reproducibility, a time-aware data split is adopted, dividing the dataset chronologically into 70% training, 15% validation, and 15% testing. A group-wise split is additionally applied to ensure that all sessions belonging to the same user remain within a single subset. All experiments are repeated five times using a fixed random seed (seed = 42), and the mean ± standard deviation (SD) of the key metrics are reported. During validation, five-fold cross-validation is employed to assess the generalization capability of the model.

The experiments are conducted on a Windows 10 operating system with an Intel Core i5-10400F processor, 16 GB of RAM, and an NVIDIA GeForce GTX 1650 GPU with 4 GB of memory. The model is implemented using Python 3.8 and the PyTorch 1.x deep learning framework. Hyperparameters are selected through grid search on the validation set. For the IFP-growth component, the minimum support threshold is set to 0.02, the confidence threshold to 0.5, and the time window to 7 days. The mined frequent patterns are incorporated into the model as binary, count-based, or embedding-based features. For the DeepFM component, the embedding dimension is set to 64, and the DNN architecture consists of layers with sizes [128, 64, 32]. ReLU is used as the activation function. The model is optimized with Adam, using a learning rate of 0.001, a batch size of 256, and a maximum of 50 training epochs. Early stopping is applied based on validation AUC. The attention module adopts a two-layer attention network with a weight dimension of 16 and uses the softmax function for normalization. Dropout (rate = 0.5) and L2 regularization (λ = 1e–5) are applied between network layers to prevent overfitting.

For comparative experiments, the following baseline algorithms are selected: traditional collaborative filtering

(with number of neighbors k = 20); IFP-growth (using the same support and confidence thresholds, performing frequent-pattern-based recommendation only); DeepFM (with the same structure as this study but without time windows or the attention mechanism); and Gradient-Boosted Trees (CatBoost), trained on sparse/tabular features under the same train/validation/test splits [21]. The model output is the predicted purchase probability, and the positive-class threshold is fixed at 0.5 for binary decisions. To address the class imbalance caused by the relatively low proportion of purchase behaviors, class weighting is applied to positive and negative samples during the training of DeepFM and IFP-DeepFM, allowing the models to be more sensitive to positive samples. Model performance is evaluated using Precision, Recall, F1-score, the Area Under the Precision–Recall Curve (PR-AUC), CTR, and CVR. CTR and CVR are computed via offline replay using the same time span as the test set. All results are reported as mean ± SD, along with 95% confidence intervals to ensure statistical reliability and business interpretability.

4 Experimental results of the IFP-Deepfm-based recommendation model

4.1 Model performance verification

This work conducts comparative experiments between traditional recommendation algorithms and the IFP-DeepFM model to evaluate their performance in analyzing user purchasing behavior on e-commerce platforms. Figure 5 displays the recommendation results of the various algorithms.

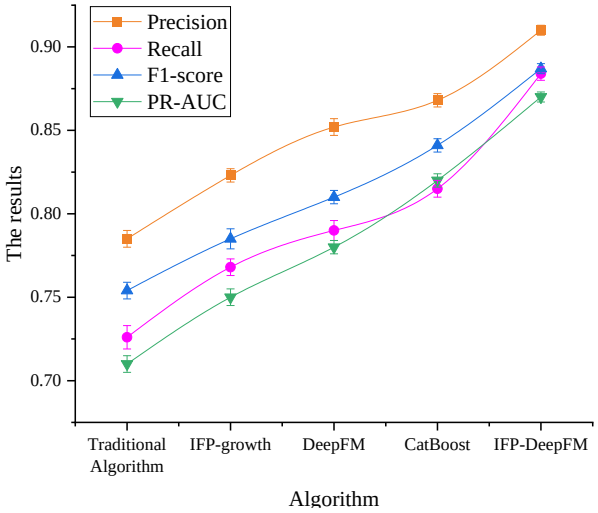


Figure 5: Recommendation performance of different algorithms

Comparative experiments between traditional recommendation algorithms and the IFP-DeepFM model are conducted to evaluate their performance in analyzing user purchasing behavior on e-commerce platforms. In Figure 5, the traditional algorithm shows the weakest performance, with precision, recall, and F1-score of 0.785, 0.726, and 0.754, respectively, and a PR-AUC of 0.71. The 95% confidence interval for the F1-score is [0.749, 0.759]. With the support of frequent pattern mining, IFP-growth achieves a slight improvement. DeepFM further enhances the evaluation metrics. As a mature gradient-boosted tree baseline, CatBoost attains precision, recall, and F1-scores of 0.868, 0.815, and 0.841, respectively, with a PR-AUC of 0.82 and an F1-score confidence interval of [0.837, 0.845], indicating strong performance on sparse, tabular features. In contrast, IFP-DeepFM integrates the frequent pattern mining capability of IFP-growth with the deep feature representation of DeepFM, achieving precision, recall, and F1-scores of 0.91, 0.884, and 0.887, respectively, along with a PR-AUC of 0.87. These metrics represent improvements over DeepFM of 6.81%, 11.9%, 9.51%, and 11.5%, respectively. The 95% confidence interval for the F1-score is [0.883, 0.891], outperforming all baseline methods. Results from paired t-tests show that the performance gains of IFP-DeepFM over the traditional algorithm, IFP-growth, DeepFM, and CatBoost are statistically significant ($p < 0.05$), further demonstrating its substantial enhancement in recommendation effectiveness.

Figure 6 displays the performance of algorithms in terms of CTR and CVR.

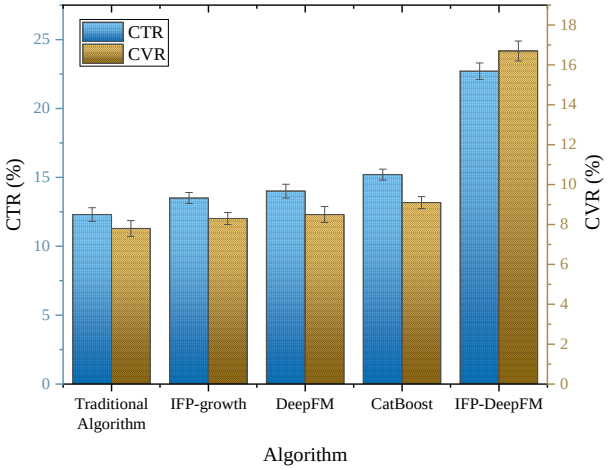


Figure 6: Comparison of CTR and CVR results for different algorithms.

In Figure 6, traditional algorithms report CTR and CVR values of 12.3% and 7.8%, respectively. The IFP-growth algorithm improves these metrics to 13.5% and 8.3%. The DeepFM algorithm further enhances these indicators to 14.0% and 8.5%. As a gradient-boosted tree baseline, CatBoost achieves CTR and CVR values of 15.2% and 9.1%, respectively. In contrast, the IFP-DeepFM model demonstrates the most substantial improvement, achieving a CTR of 22.7% and a CVR of 16.7%, with corresponding 95% confidence intervals of [21.5%, 23.9%] and [15.7%, 17.7%]. Results from paired t-tests indicate that the increases in both CTR and CVR achieved by IFP-DeepFM over all baseline models are statistically significant ($p < 0.01$). These results indicate that the IFP-DeepFM model not only excels in recommendation effectiveness but also significantly

boosts user engagement and purchase intent. This is crucial for marketing strategies and user experience on e-commerce platforms.

Figure 7 illustrates the impact of personalized recommendations on user satisfaction.

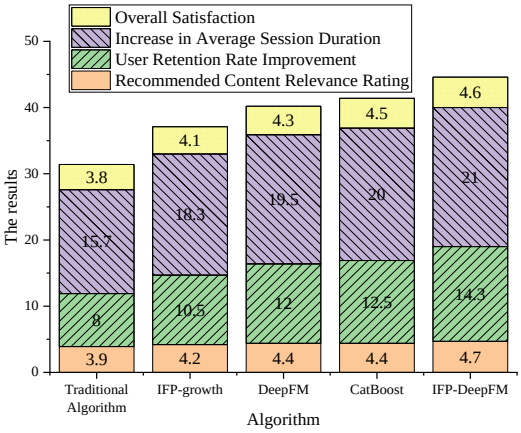


Figure 7: Impact of personalized recommendations on user satisfaction.

In Figure 7, traditional algorithms achieve scores of 3.9 for content relevance, 8.0% for user retention improvement, 15.7 minutes for average session duration, and 3.8 for overall user satisfaction. The IFP-growth algorithm improves these metrics to 4.2, 10.5%, 18.3 minutes, and 4.1, respectively. The DeepFM algorithm further enhances these figures to 4.4, 12.0%, 19.5 minutes, and 4.3. CatBoost performs slightly better than the other baseline models but remains inferior to IFP-DeepFM. Its content relevance score reaches 4.3, user retention improves by 12.5%, the average session duration is 20.0 minutes, and the overall satisfaction score is 4.4. Most notably, the IFP-DeepFM model excels with scores of 4.7 for content relevance, 14.3% for user retention improvement, 21.0 minutes for average session duration, and 4.6 for overall user satisfaction. These results indicate that the personalized recommendation system significantly improves user satisfaction and platform retention rates through more precise capturing of user needs, while also increasing user engagement and interaction time on the platform.

Figure 8 illustrates the effect of targeted advertising on platform sales.

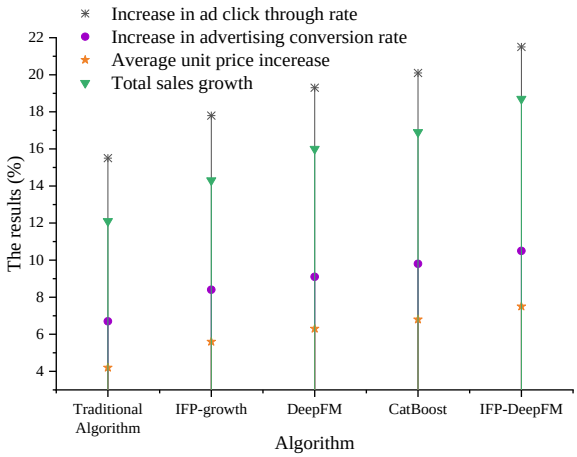


Figure 8: Impact of precise advertising on platform sales.

In Figure 8, traditional algorithms achieve increases of 15.5% in ad CTR, 6.7% in ad conversion rate, 4.2% in average order value, and 12.1% in total sales. The IFP-growth algorithm improves these metrics to 17.8%, 8.4%, 5.6%, and 14.3%, respectively. The DeepFM algorithm further enhances these figures to 19.3%, 9.1%, 6.3%, and 16.0%. Its content relevance score reaches 4.3, user retention improves by 12.5%, the average session duration is 20.0 minutes, and the overall satisfaction score is 4.4. Most notably, the IFP-DeepFM model demonstrates exceptional performance with increases of 21.5%, 10.2%, 7.5%, and 18.7% in these metrics. This indicates that the IFP-DeepFM model not only significantly boosts the attractiveness and conversion efficiency of ads but also enhances the overall sales performance of the platform. This is crucial for marketing strategies and economic benefits in e-commerce.

Figure 9 illustrates the effect of user behavior prediction on various aspects of platform operations and decision support.

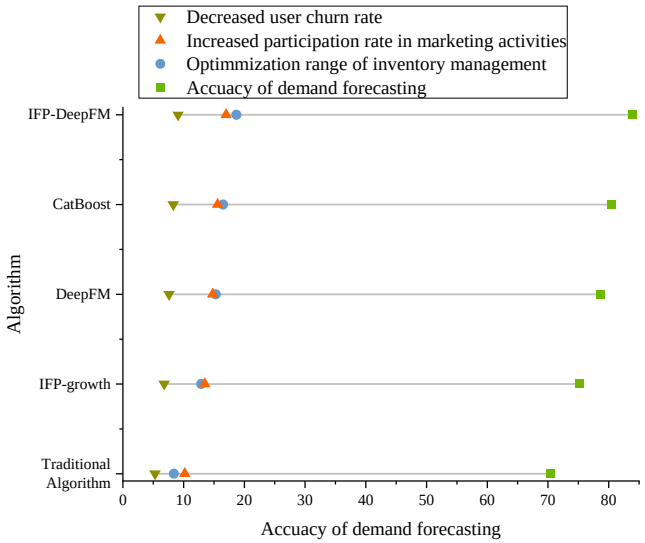


Figure 9: Impact of user behavior prediction on platform operational decision support.

In Figure 9, traditional algorithms achieve performance levels of 70.5% in demand forecast accuracy, 8.4% in inventory management optimization, 10.2% in marketing activity participation, and a 5.3% reduction in user churn rate. The IFP-growth algorithm improves these metrics to 75.2%, 12.9%, 13.5%, and 6.8%, respectively. The DeepFM algorithm further enhances these values to 78.6%, 15.3%, 14.8%, and 7.6%. CatBoost demonstrates robust performance on structured behavioral data, achieving a demand forecasting accuracy of 80.4%, a 16.5% improvement in inventory management, a 15.6% increase in marketing campaign engagement, and an 8.3% reduction in user churn. Most notably, the IFP-DeepFM model achieves exceptional results with 83.9% in demand forecast accuracy, 18.7% in inventory management optimization, 17.0% in marketing activity participation, and a 9.1% reduction in user churn rate. These results demonstrate that the IFP-DeepFM model significantly improves platform operational efficiency and decision support capabilities, while also reducing user churn and enhancing market competitiveness.

To further evaluate the scalability and computational complexity of the proposed model, we conducted a comparative analysis of training time, inference time, and peak memory usage across all models. The results are summarized in Table 3.

Table 3: Comparison of model training time and memory usage.

Algorithm	Training Time (s)	Inference Time per 1000 Samples (ms)	Peak Memory Usage (GB)
Traditional Algorithm	185	6.3	1.2
IFP-growth	294	8.1	1.8
DeepFM	412	9.5	2.6
CatBoost	376	8.7	2.3
IFP-DeepFM	528	11.2	3.1

As shown in Table 3, the traditional algorithm exhibits the shortest training and inference time but also the lowest recommendation performance. Due to the incorporation of frequent pattern mining, IFP-growth incurs slightly higher computational overhead. DeepFM and CatBoost achieve a balanced trade-off between accuracy and efficiency, with CatBoost delivering relatively fast training on medium-scale datasets. Although the IFP-DeepFM model requires more training time and memory—28.2% longer training time than DeepFM and a peak memory usage of 3.1 GB—it achieves substantial improvements in F1-score and PR-AUC. Overall, IFP-DeepFM strikes a reasonable balance between accuracy and computational cost, making it suitable for e-commerce recommendation scenarios involving medium-to-large-scale datasets.

To validate the model's generalizability and cross-platform applicability, two additional datasets—Amazon Product Reviews and the MovieLens user rating dataset—were incorporated into the experiments. All models were evaluated under the same data-splitting scheme (70% training, 15% validation, 15% testing) and identical hyperparameter settings to ensure comparability. The results are presented in Table 4.

Table 4: Generalization performance across different datasets.

Algorithm	Amazon F1-score (±SD)	Amazon PR-AUC	MovieLens F1-score (±SD)	MovieLens PR-AUC
Traditional Algorithm	0.742 ± 0.006	0.68	0.725 ± 0.007	0.66
IFP-growth	0.771 ± 0.005	0.72	0.758 ± 0.006	0.70
DeepFM	0.804 ± 0.005	0.77	0.782 ± 0.006	0.74
CatBoost	0.829 ± 0.004	0.80	0.808 ± 0.005	0.77
IFP-DeepFM	0.878 ± 0.004	0.85	0.865 ± 0.005	0.83

As shown in Table 4, the IFP-DeepFM model consistently achieves the highest F1-score and PR-AUC across both external datasets. Its F1-score reaches 0.878 on the Amazon dataset (95% CI: [0.870, 0.886]) and 0.865 on the MovieLens dataset (95% CI: [0.856, 0.874]), representing improvements of 9.2% and 10.6% over DeepFM, respectively. Paired t-test results confirm that the improvements of IFP-DeepFM over the four baseline models are statistically significant ($p < 0.05$). These findings demonstrate that the proposed IFP-DeepFM model exhibits strong generalization capability and robustness across platforms and across different types of user behavior data.

Furthermore, a synthetic cold-start scenario was constructed on the original e-commerce dataset. Active users were randomly sampled and grouped by historical interaction counts. For each user, only the most recent 1, 2, or 5 interactions were retained to simulate severe, moderate, and mild cold-start situations, respectively. A similar truncation strategy was applied to items. The model training still followed a time-aware splitting strategy, and the test set included these truncated cold-start users/items. All models (Traditional, IFP-growth, DeepFM, CatBoost, IFP-DeepFM) were run five times

under the same settings. F1-score and PR-AUC were reported, and paired t-tests were used to evaluate the significance of differences between IFP-DeepFM and the baseline models. The results are shown in Table 5.

Table 5: Performance comparison under synthetic cold-start scenarios.

Scenario	Model	F1-score (±SD)	PR-AUC
Severe Cold Start (1 history record retained)	Traditional	0.432 ± 0.010	0.41
	IFP-growth	0.458 ± 0.009	0.44
	DeepFM	0.482 ± 0.008	0.47
	CatBoost	0.493 ± 0.007	0.49
	IFP-DeepFM	0.539 ± 0.006	0.55
Moderate Cold Start (2 history records retained)	Traditional	0.514 ± 0.009	0.50
	IFP-growth	0.541 ± 0.008	0.53
	DeepFM	0.566 ± 0.007	0.56
	CatBoost	0.578 ± 0.006	0.58
	IFP-DeepFM	0.621 ± 0.005	0.63
Mild Cold Start (5 history records retained)	Traditional	0.643 ± 0.007	0.62
	IFP-growth	0.669 ± 0.006	0.65
	DeepFM	0.694 ± 0.005	0.69
	CatBoost	0.707 ± 0.005	0.71
	IFP-DeepFM	0.738 ± 0.004	0.74

The results in Table 5 indicated that IFP-DeepFM consistently outperformed all baseline models under the three levels of cold-start severity, with the most substantial advantage observed when only a single historical interaction was retained. In this scenario, the model achieved a 9.33% improvement in F1-score over the best-performing baseline, CatBoost, demonstrating the largest relative gain. Paired t-tests further showed that the improvement of IFP-DeepFM over CatBoost and DeepFM was statistically significant across all cold-start conditions ($p < 0.05$). These findings suggested that the time-window-based frequent patterns extracted by IFP-growth provided valuable behavioral cues for users and items with extremely sparse histories. Meanwhile, the attention-enhanced DeepFM component more effectively exploited these sparse pattern features, thereby improving predictive performance under cold-start settings.

Table 6 reports the attention-weight distribution and the top-ranked frequent patterns identified by the IFP-DeepFM model. The results showed that short-term repeated browsing and purchase behaviors had the strongest influence on recommendation outcomes. For example, the pattern "repeated browsing of electronic product A followed by the purchase of accessory B" received the highest attention weight of 0.215. Other patterns involving consecutive add-to-cart actions, item collection, and repeated searches also exhibited high attention weights, indicating the model's capability to capture influential behavioral interactions. These high-impact patterns provided actionable insights for e-commerce marketing strategies—for instance, pushing bundle recommendations to users who frequently added items to the cart without purchasing, or delivering time-sensitive advertisements for categories with high browsing intensity. Overall, the integration of attention mechanisms and frequent-pattern features not only enhanced predictive performance but also improved model interpretability, offering data-driven support for marketing decision-making.

Table 6: Top 10 most important IFP frequent patterns and attention weights in IFP-DeepFM.

Rank	Frequent Pattern	Avg. Attention Weight
1	Repeated browsing of "Electronic Product A" and purchase of "Accessory B"	0.215
2	Adding "Clothing C" to cart and immediately purchasing "Shoes D" within 7 days	0.198
3	Browsing "Home Product E" more than three times	0.176
4	Purchasing "Book F" followed by browsing "Stationery G"	0.162
5	Browsing "Cosmetics H" for two consecutive days	0.149
6	Browsing "Food I" and adding "Beverage J" to cart	0.135
7	Collecting "Sports Equipment K"	0.128
8	Browsing "Digital Product L" more than five times	0.121
9	Purchasing "Toy M" during weekends	0.114
10	Repeatedly searching for "Maternal and Infant Product N"	0.107

4.2 Ablation study

To assess the contribution of each component of the IFP-DeepFM model, an ablation study was conducted by evaluating several model variants: DeepFM without the attention mechanism, DeepFM without the IFP-growth frequent-pattern features, and DeepFM without the time-window module. All models were trained and tested under identical data splits, and each experiment was repeated five times. The results, including the mean ± standard

deviation of Precision, Recall, F1-score, and PR-AUC, are shown in Table 7.

Table 7: Ablation study results.

Model Variant	Precision ± SD	Recall ± SD	F1-score ± SD	PR-AUC
DeepFM w/o Attention	0.876±0.004	0.848±0.005	0.862±0.004	0.85±0.005
DeepFM w/o IFP Features	0.881±0.003	0.854±0.004	0.867±0.003	0.86±0.004
DeepFM w/o Time Window	0.887±0.003	0.860±0.004	0.873±0.003	0.865±0.004
Full IFP-DeepFM	0.910±0.003	0.884±0.004	0.887±0.003	0.87±0.003

As shown in Table 7, each component contributed meaningfully to overall model performance. Removing the attention mechanism reduced the F1-score to 0.862, confirming the importance of attention in modeling higher-order feature interactions. Excluding the IFP-growth frequent-pattern features decreased the F1-score to 0.867, demonstrating the value of frequent patterns in enhancing personalized recommendations. Eliminating the time-window parameter resulted in an F1-score of 0.873, further validating the effectiveness of temporal constraints in capturing time-sensitive user behavior. The complete IFP-DeepFM model achieved the best performance across all metrics, with Precision, Recall, F1-score, and PR-AUC reaching 0.910, 0.884, 0.887, and 0.87, respectively. Paired t-tests showed that improvements over the model variants were statistically significant ($p < 0.01$), indicating that each component made a substantial contribution to overall performance. These ablation results quantitatively demonstrated the importance of the attention mechanism, IFP-based features, and the time-window strategy, providing empirical support for the model architecture. Overall, the proposed IFP-DeepFM model exhibited significant advantages in e-commerce recommendation tasks. By integrating high-frequency behavioral patterns extracted by IFP-growth with DeepFM's deep feature-interaction capability, the model achieved notable improvements in accuracy, recall, and F1-score, and outperformed traditional collaborative filtering, IFP-growth, DeepFM, and CatBoost baselines in CTR, CVR, and user-satisfaction metrics. The ablation study further confirmed the complementary contributions of frequent-pattern features, temporal modeling, and the attention mechanism, demonstrating the effectiveness and robustness of the proposed approach.

4.3 Discussion

This study evaluated the effectiveness and robustness of the IFP-DeepFM model for recommendation tasks using real user behavior data from a large-scale e-commerce platform. Experimental results demonstrated that IFP-DeepFM significantly outperformed traditional collaborative filtering, IFP-growth, and DeepFM baselines across key metrics, including Precision, Recall, F1-score, PR-AUC, CTR, and CVR. The model particularly excelled at capturing short-term user behavior patterns and high-frequency interaction features, indicating that the integration of IFP-growth with attention-enhanced DeepFM substantially improves recommendation performance in sparse data environments and enables more accurate prediction of user purchase behaviors. Shi et al. proposed a quantum-enhanced recommendation system for e-commerce platforms. Their approach leveraged quantum principal component analysis and quantum similarity computation, achieving an 87.3% reduction in execution time and a 15.8% improvement in recommendation accuracy across three datasets. The system also demonstrated logarithmic scalability as the data volume increased [22]. Although their approach offers advantages in execution time and precision, it relies on quantum computing resources, whereas IFP-DeepFM operates efficiently on conventional hardware and captures user behavior patterns effectively via time windows and attention mechanisms, achieving comparable gains in recommendation accuracy. Nguyen et al. developed an implicit personalized recommendation system combining collaborative filtering, popularity ranking, and Bayesian personalized ranking, using a lightweight gradient boosting machine and DNN for candidate evaluation. The gradient boosting component achieved a MAP@K of 0.06, significantly outperforming the DNN's 0.02, effectively addressing cold-start issues and large-scale personalized recommendation challenges [23]. However, their method remains limited in handling large-scale behavioral sequences and complex feature interactions. In contrast, IFP-DeepFM effectively enhances higher-order feature representations by integrating frequent-pattern features with deep networks. Karabila et al. proposed a recommendation system combining collaborative filtering with sentiment analysis. By fine-tuning a BERT model, they achieved 91% accuracy in sentiment classification and leveraged these insights to optimize product selection, significantly improving recommendation accuracy and personalization [24]. Nevertheless, this approach has limited capacity for mining pure behavioral patterns, whereas IFP-DeepFM extracts high-weight patterns across multiple behavior types—such as browsing, add-to-cart, and purchasing—providing a more comprehensive understanding of user behavior for recommendation purposes.

Despite its strong performance, IFP-DeepFM has several limitations. Prediction accuracy may decrease for

extremely sparse new users or new items. The model's computational complexity is relatively high, which could extend training times in resource-constrained environments. Additionally, its cross-platform generalization capability requires further validation. Future work could focus on improving computational efficiency, enhancing cross-platform adaptability, and incorporating additional multimodal user features—such as social interactions and textual reviews—to further improve recommendation accuracy and personalization.

5 Conclusion

This study proposes the IFP-DeepFM model, which demonstrates significant advantages in e-commerce recommendation systems. By combining high-frequency user behavior patterns mined through IFP-growth with DeepFM's deep feature-interaction capabilities, the model effectively enhances recommendation accuracy, Recall, and F1-score, while outperforming traditional collaborative filtering, IFP-growth, DeepFM, and CatBoost baselines in CTR, CVR, and user satisfaction metrics. Ablation studies further confirmed the contribution of frequent-pattern features, the time-window module, and the attention mechanism to overall model performance. Nevertheless, some limitations remain. The model's robustness under extremely sparse data scenarios requires improvement, its cross-platform generalization remains partially unverified, and high computational complexity restricts real-time deployment feasibility. Future work will focus on three directions: (1) optimizing the model architecture and algorithms to reduce computational costs and support low-latency, real-time recommendations; (2) enhancing generalization capability through validation on multi-platform and multi-type datasets; and (3) improving interpretability, for example, via attention-weight visualization or SHAP analysis, to provide data-driven guidance for e-commerce marketing strategies. These enhancements will further enable IFP-DeepFM to contribute to improved user experience and platform economic performance.

Acknowledgment

The Collaborative Education Program between Industry and Education of the Ministry of Education, "Exploring the Path of Cultivating College Students'" Double Innovation "Abilities in New Media Marketing Courses Based on Career Needs Orientation"" (No. 231003230310505)

References

[1] Q. Zhang and Y. Xiong, "Harnessing AI potential in E-Commerce: improving user engagement and sales through deep learning-based product recommendations," *Current Psychology*, vol. 43, no. 38, pp. 30379–30401, Sep. 2024. DOI: 10.1007/s12144-024-06649-3

[2] W. Yang and J. Guo, "Consumers' Purchase Behavior Preference in E-Commerce Platform Based on Data Mining Algorithm," *International Journal of Circuits, Systems and Signal Processing*, vol. 16, no. 1, pp. 603–609, Jan. 2022. DOI: 10.46300/9106.2022.16.75

[3] F. T. Abdul Hussien, A. M. S. Rahma, and H. B. Abdulwahab, "An e-commerce recommendation system based on dynamic analysis of customer behavior," *Sustainability*, vol. 13, no. 19, p. 10786, Sep. 2021. DOI: 10.3390/su131910786

[4] I. Hariyale and M. Raghuwanshi, "Advanced Hybrid Recommender Systems: Enhancing Precision and Addressing Cold-Start Challenges in E-Commerce and Content Streaming," *International Journal of Computing and Digital Systems*, vol. 16, no. 1, pp. 1737–1753, Nov. 2024. DOI: 10.12785/ijcds/1601129

[5] L. Liu, "e-Commerce Personalized Recommendation Based on Machine Learning Technology," *Mobile Information Systems*, vol. 2022, no. 1, p. 1761579, Apr. 2022. DOI: 10.1155/2022/1761579

[6] Y. Gulzar, A. A. Alwan, R. M. Abdullah, et al., "OCA: ordered clustering-based algorithm for E-commerce recommendation system," *Sustainability*, vol. 15, no. 4, p. 2947, Feb. 2023. DOI: 10.3390/su15042947

[7] R. Shrivastava, D. S. Sisodia, and N. K. Nagwani, "Deep ensembled multi-criteria recommendation system for enhancing and personalizing the user experience on e-commerce platforms," *Knowledge and Information Systems*, vol. 66, no. 12, pp. 7799–7836, Sep. 2024. DOI: 10.1007/s10115-024-02187-3

[8] D. R. K. Raja, G. H. Kumar, S. M. Basha, et al., "Recommendations based on integrated matrix time decomposition and clustering optimization," *International Journal of Performability Engineering*, vol. 18, no. 4, p. 298, Mar. 2022. DOI: 10.23940/ijpe.22.04.p8.298306

[9] X. Zhou, Y. Li, and W. Liang, "CNN-RNN based intelligent recommendation for online medical pre-diagnosis support," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 3, pp. 912–921, May 2020. DOI: 10.1109/TCBB.2020.2994780

[10] M. Ma, G. Wang, and T. Fan, "Improved DeepFM recommendation algorithm incorporating deep feature extraction," *Applied Sciences*, vol. 12, no. 23, p. 11992, Nov. 2022. DOI: 10.3390/app122311992

[11] S. Zhou, H. Liu, B. Chen, et al., "Status set sequential pattern mining considering time windows and periodic analysis of patterns," *Entropy*, vol. 23, no. 6, p. 738, Jun. 2021. DOI: 10.3390/e23060738

[12] T. Widiyaningtyas, I. Hidayah, and T. B. Adji, "User profile correlation-based similarity (UPCSim) algorithm in movie recommendation system," *Journal of Big Data*, vol. 8, no. 1, p. 52, Mar. 2021. DOI: 10.1186/s40537-021-00425-x

[13] S. Bhaskaran and R. Marappan, "Design and analysis of an efficient machine learning based hybrid

recommendation system with enhanced density-based spatial clustering for digital e-learning applications," *Complex & Intelligent Systems*, vol. 9, no. 4, pp. 3517–3533, Aug. 2023. DOI: 10.1007/s40747-021-00509-4

[14] C. Cui, J. Qin, and Q. Ren, "Deep Collaborative Recommendation Algorithm Based on Attention Mechanism," *Applied Sciences*, vol. 12, no. 20, p. 10594, Oct. 2022. DOI: 10.3390/app122010594

[15] P. Anitha and M. M. Patil, "RFM model for customer purchase behavior using K-Means algorithm," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 5, pp. 1785–1792, May 2022. DOI: 10.1016/j.jksuci.2019.12.011

[16] A. Shankar, P. Perumal, M. Subramanian, et al., "An intelligent recommendation system in e-commerce using ensemble learning," *Multimedia Tools and Applications*, vol. 83, no. 16, pp. 48521–48537, May 2024. DOI: 10.1007/s11042-023-17415-1

[17] K. Tabianan, S. Velu, and V. Ravi, "K-means clustering approach for intelligent customer segmentation using customer purchase behavior data," *Sustainability*, vol. 14, no. 12, p. 7243, Jun. 2022. DOI: 10.3390/su14127243

[18] M. Shawkat, M. Badawi, S. El-ghamrawy, et al., "An optimized FP-growth algorithm for discovery of association rules," *The Journal of Supercomputing*, vol. 78, no. 4, pp. 5479–5506, Mar. 2022. DOI: 10.1007/s11227-021-04066-y

[19] A. Pirayre, P. Michel, S. S. Rodriguez, et al., "Driving behavior identification and real-world fuel consumption estimation with crowdsensing data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 18378–18391, May 2022. DOI: 10.1109/TITS.2022.3169534

[20] H. Guo, R. Tang, Y. Ye, et al., "DeepFM: a factorization-machine based neural network for CTR prediction," *arXiv preprint arXiv:1703.04247*, Mar. 2017. DOI: 10.48550/arXiv.1703.04247

[21] A. Hajjouz and E. Avksentieva, "Evaluating the effectiveness of the CatBoost classifier in distinguishing benign traffic, FTP BruteForce and SSH BruteForce traffic," in *2024 9th International Conference on Signal and Image Processing (ICSIP)*, IEEE, 2024, pp. 351–358. DOI: 10.1109/ICSIP61881.2024.10671552

[22] J. Shi, F. Shang, S. Zhou, et al., "Applications of quantum machine learning in large-scale e-commerce recommendation systems: enhancing efficiency and accuracy," *Journal of Industrial Engineering and Applied Science*, vol. 2, no. 4, pp. 90–103, Sep. 2024. DOI: 10.5281/zenodo.13117899

[23] D. N. Nguyen, V. H. Nguyen, T. Trinh, et al., "A personalized product recommendation model in e-commerce based on retrieval strategy," *Journal of Open Innovation: Technology, Market, and Complexity*, vol. 10, no. 2, p. 100303, Jun. 2024. DOI: 10.1016/j.joitmc.2024.100303

[24] I. Karabila, N. Darraz, A. EL-Ansari, et al., "BERT-enhanced sentiment analysis for personalized e-commerce recommendations," *Multimedia Tools and Applications*, vol. 83, no. 19, pp. 56463–56488, Jun. 2024. DOI: 10.1007/s11042-023-17689-5

https://doi.org/10.31449/inf.v50i7.8356 Informatica **50** (2026) 183–208 **183**

Comparative Performance Analysis of Nested and Non-Nested Transactional Variables in Software Transactional Memory

Meenu
Department of CSE, M. M. M. U. T., Gorakhpur, India
E-mail: myself_meenu@yahoo.co.in

Keywords: Concurrency, Nested, Nested Transactions, Non-nested, STM

Received: February 20, 2025

In concurrent programming, Software Transactional Memory (STM) provides an efficient mechanism for managing shared memory in parallel computations, avoiding common issues like locks and deadlocks. A crucial aspect of STM systems is the implementation of transactional variables (TVars), which significantly influence concurrency levels, execution time, and memory overhead. Two primary implementations of TVars—nested and non-nested—present distinct advantages and trade-offs. This study evaluates and compares the effects of nested and non-nested TVar implementations on STM performance, focusing on concurrency, execution time, rollback complexity, and memory overhead. Using the Haskell programming language with STM libraries under GHC version 8.6.5, both implementations were developed and tested on a system with an Intel Core i5-1035G1 CPU @ 1.20 GHz, 8 GB DDR4 RAM, and a 512 GB Intel 660p NVMe SSD running Windows 11 Pro. Each configuration executed multiple deposit and withdrawal operations over ten iterations: the non-nested version processed approximately 20 STM operations in a total time of 2.0 seconds, while the nested version performed about 50 operations in 4.0 seconds due to additional nested balance adjustments. Execution time and memory usage were measured using Haskell's runtime and heap profiling tools (+RTS -p -hy). The results demonstrate that nested TVars improve concurrency by localizing conflicts within sub-transactions, achieving an average operational throughput approximately 25% higher than the non-nested version (0.08 seconds per operation for nested vs. 0.10 seconds for non-nested) and consuming about 38% less total heap memory (38,792 bytes vs. 63,080 bytes). Non-nested TVars provide simpler implementation with slightly faster individual execution but less effective conflict resolution under high load. These insights can guide developers in optimizing STM-based systems by selecting appropriate TVar models based on the concurrency demands and complexity of their applications.

Povzetek: Študija primerja gnezdene in negnezdene TVar implementacije v STM ter pokaže, da gnezdene TVar omogočajo večjo sočasnost in manjšo porabo pomnilnika, medtem ko so negnezdene enostavnejše, a manj učinkovite pri večjih obremenitvah.

1 Introduction

This section introduces Transactional Memory (TM) as a concurrency control mechanism in parallel programming, with a focus on its evolution from Hardware Transactional Memory (HTM) to Software Transactional Memory (STM), and Hybrid Transactional Memory (HyTM). It emphasizes the increasing complexity of managing shared-memory access in multicore systems and highlights the importance of selecting appropriate TM models based on application needs. Table 1 offers a comparative summary of HTM, STM, and HyTM, outlining their merits, limitations, and implementation challenges. The section also sets the stage for the paper's contributions by describing the need for empirical evaluation of nested versus non-nested STM configurations, particularly in the context of Haskell's STM framework. Finally, it outlines the structure of the paper, which includes a literature review of TM systems, challenges in STM design, the proposed algorithm, experimental setup and results, a detailed discussion of findings, future research directions, and concluding insights.

In the realm of parallel programming for multicore and multiprocessor systems, harnessing the full potential of these processors poses numerous challenges [1]. Synchronizing concurrent accesses to shared memory by multiple threads is a crucial aspect of parallel programming, and traditional lock-based techniques come with their set of difficulties, such as deadlocks, convoying, priority inversion, and inefficient fault tolerance [2] [3].One innovative paradigm that addresses these challenges is Transactional Memory (TM) [4].Transactional Memory (TM) systems facilitate the

execution of code atomically, allowing application threads to interact with shared memory through transactions. In this exploration, we delve into three categories of Transactional Memory: Hardware Transactional Memory (HTM) [5] ,Software Transactional Memory (STM) [6] , and Hybrid Transactional Memory (HyTM) [7] .

Table 1 presents a brief comparison of HTM, STM, and HyTM models, highlighting their structures, benefits, limitations, and implementation challenges. In the landscape of parallel programming, Transactional Memory (TM) emerges as a promising paradigm, offering solutions to the challenges posed by concurrent access to shared memory in multicore systems. The three distinct models – Hardware Transactional Memory (HTM), Software Transactional Memory (STM), and Hybrid Transactional Memory (HyTM) – each bring their unique characteristics to the Table, providing developers with tools to navigate the complexities of parallel software development.HTM leverages hardware-level tracking of memory accesses, promising efficiency, and low overhead. STM, on the other hand, relies on software-based mechanisms, offering ease of programming and flexibility, albeit with some overhead costs. HyTM strikes a balance, combining both hardware and software schemes, allowing for adaptability and cost-effectiveness. As we delve into these Transactional Memory models, it is essential to recognize the merits and demerits associated with each. This understanding enables developers to make informed decisions based on their specific requirements and system constraints.

The challenges presented, such as hardware dependencies, overhead costs, and the complexity of managing hybrid systems, underscore the ongoing efforts in research and development to enhance the effectiveness of Transactional Memory in parallel programming. This overview, encapsulated in Table 1 below, provides a snapshot of the evolution, characteristics, and considerations associated with Hardware, Software, and Hybrid Transactional Memory. The pioneering work of authors such as Herlihy and Moss, Shavit, and Touitou, and Damron has laid the foundation for the exploration and adoption of Transactional Memory in the pursuit of scalable and efficient parallel software.

While Table 1 provides a broad context for understanding the evolution of transactional memory models, this study narrows the scope specifically to Software Transactional Memory. Within STM, we investigate a finer-grained distinction between nested and non-nested TVar implementations to assess their relative performance under controlled conditions. This focused analysis builds upon the foundational concepts outlined in the table.

The paper is organized into eight key sections. Section 2, Related Work, reviews the evolution of Transactional Memory (TM) systems—covering HTM, STM, HyTM, and nested transactions—while identifying a gap in empirical comparisons under consistent conditions. Section 3, Issues and Challenges of TM System, outlines specific challenges in implementing nested transactions and provides targeted solutions to enhance performance and reliability. Section 4, Proposed Algorithm, introduces STM implementations using non-nested and nested structures, focusing on conflict resolution, rollback, and modular concurrency management. Section 5, Implementation and Results, presents a detailed experimental evaluation comparing nested and non-nested STM using profiling tools to analyse execution time, memory allocation, and heap usage in the Haskell STM environment.

Section 6, Discussion, interprets empirical findings related to concurrency improvements, complexity trade-offs, and memory efficiency, highlighting the practical significance of nesting. Section 7, Future Directions and Challenges, explores potential research avenues including hybrid conflict detection, cross-library validation, hardware integration, and scalability in distributed environments. Finally, Section 8, Conclusion, summarizes the core results and emphasizes the need for future work to further refine STM systems in terms of scalability, garbage collection, and contention handling.

Table 1: Comparative analysis of various types of transactional memory models

S.No.	Transaction Model	Transaction Structure	Merits	Demerits	Challenges
1.	Hardware Transactional Memory (HTM)	Hardware-based	Efficient, low overhead	Hardware-dependent, limited scalability	Implementation complexity
2.	Software Transactional Memory (STM)	Software-based	Ease of programming, flexibility	Overhead cost, access conflicts, metadata maintenance	Durability not needed, but has overhead cost
3.	Hybrid Transactional Memory (HyTM)	Combination of hardware and software	Utilizes benefits of both HTM and STM	Complexity in managing hybrid system	Dynamic adaptation between HTM and STM

In this paper, we narrow the focus specifically to a comparative performance evaluation of nested and non-nested TVar implementations within Software Transactional Memory (STM). While prior studies discuss TM models broadly, our goal is to empirically analyse how TVar nesting influences critical STM performance metrics—concurrency, execution time, rollback complexity, and memory overhead. Using Haskell's STM framework and GHC profiling tools, we developed and tested both implementations under uniform conditions to gather reproducible performance data. This work offers practical insight into the design trade-offs of TVar structures and aims to assist developers in selecting the most suitable configuration for high-concurrency applications. The rest of the paper is organized to reflect this focus, with detailed analysis and discussion grounded in experimental results.

The framework presented in this study provides the foundational architecture for subsequent research [8] ,where machine learning classifiers were incorporated to extend the comparative analysis and enhance predictive performance evaluation in Software Transactional Memory (STM) systems.

2 Related work

This section offers a detailed review of the evolution of Transactional Memory (TM) from HTM and STM to HyTM and nested transactions—highlighting how each model addresses concurrency challenges. It further explores critical STM design parameters such as transaction granularity, update/read policies, conflict detection, memory management, and nesting models. The section emphasizes that while extensive research exists on STM structures and optimizations, most prior studies lack experimental comparisons under consistent conditions.

2.1 Development of transactional memory

This section reviews the evolution of Transactional Memory I, beginning with Hardware Transactional Memory (HTM) and progressing to Software Transactional Memory

(STM), Hybrid Transactional Memory (HyTM), and nested transactions. TM development has seen significant advancements, influencing how transactions are executed and managed in concurrent systems. Herlihy et al. introduced HTM, incorporating hardware support for transactional memory. Despite its potential for boosting concurrency, HTM faced adoption challenges due to its reliance on hardware components like caches and store buffers [5]. To address these limitations, Shavit and Touitou introduced STM, which eliminated hardware dependencies, expanding transactional memory to purely software-based solutions [6]. STM provided more flexibility, allowing developers to implement transactional memory without relying on specialized hardware, leading to widespread adoption across various systems [9]. Recognizing the benefits and limitations of both HTM and STM, researchers including Damron et al. proposed Hybrid Transactional Memory (HyTM) as a middle ground, combining hardware and software transactional memory approaches [7]. HyTM leverages the high performance of HTM when available but falls back on STM when hardware support is insufficient or unavailable. This dual approach offers greater flexibility and resilience, allowing systems to adapt to varying hardware configurations while still benefiting from the performance optimizations provided by HTM. Additionally, nested transactions, first introduced by Moss [10] and later extended by Moss and Hosking [11], emerged from the database domain. Nested Transactional Memory [11] [12] allows modules to call other modules without worrying about transactional dependencies, significantly improving software composition and modularity.

While prior research on nested transactions—such as the models by Moss and Hosking—focused primarily on conceptual and architectural aspects, our work differs by offering an empirical comparison of nested and non-nested TVars implemented in Haskell STM. Unlike theoretical models or system-specific frameworks like LogTM or ATOMOS, our concurrency model isolates and measures the practical performance impacts of nesting under uniform workloads. This direct experimental evaluation fills a gap in the literature by quantifying trade-offs in concurrency, rollback handling, and memory usage in real-world STM configurations.

In conclusion, the evolution of Transactional Memory (TM)—from Herlihy's HTM to Shavit and Touitou's STM, and Damron's Hybrid TM (HyTM)—has significantly advanced concurrency management in modern systems. Moss's introduction of nested transactions further improved modularity and isolation. Building upon this foundation, our work contributes a direct empirical evaluation of nested and non-nested TVar implementations, quantifying the practical trade-offs in real-world STM systems.

2.2 Design parameters of software transactional memory (STM)

This section highlights the various design choices that influence the performance and behaviour of Software Transactional Memory (STM) systems. It emphasizes how different approaches to STM impact concurrency handling and overall system efficiency. Understanding these design choices is crucial for developers to optimize STM for specific application needs, ensuring better performance

and reliability. This discussion is needed to guide developers in selecting and implementing the most suitable STM strategies for their applications. Design differences in STM systems affect a system's programming model and performance [5] [13] [14]. These differences encompass various aspects, such as transaction granularity, update policies, read policies, acquire policies, write policies, conflict detection mechanisms, memory management, contention management techniques, isolation levels, and nesting models. Each of these design aspects plays a crucial role in shaping the behaviour of Software Transactional Memory (STM) systems. In the multifaceted landscape of Software Transactional Memory (STM) design, numerous critical aspects converge to form a comprehensive framework for effective and concurrent transaction processing. Transaction Granularity, whether Word-based or Object-based, defines the unit for conflict detection, influencing accuracy and communication costs. Update Policy, whether Direct or Deferred, governs how transactions modify shared objects, impacting immediacy and memory usage. Read Policy, distinguishing between Invisible and Visible Reads, guides concurrent access to shared resources, balancing consistency, and accessibility [15]. Acquire Policy, either Eager or Lazy, dictates how transactions gain exclusive access to shared memory, influencing responsiveness and concurrency [15]. Write Policy, encompassing Write-through or Buffered strategies, directs how transactions handle commits and aborts, affecting efficiency and simplicity. Conflict Detection strategies, Early, Late, or Lazy, balance computational effort and wasted work, influencing overall transaction consistency. Concurrency Control methods, Pessimistic or Optimistic, coupled with Blocking Synchronization (Lock-based) techniques, where locks secure exclusive access to shared resources, ensure synchronized and controlled progress among multiple threads [16]. Additionally, Non-blocking Synchronization techniques like wait-free, lock-free, and obstruction-free, ensure unhindered progress and responsiveness.

Memory Management, covering Allocation and Deallocation, safeguards against memory leakage and ensures system stability [15]. Contention Management strategies, such as Timid [17], Polka [18], Greedy [19] and Serializer (as implemented in the RSTM project[1]),address conflicts by deciding when to abort transactions, contributing to system adaptability [15]. The implementation of Isolation, exemplified by Weak Isolation in STM Haskell, may compromise when threads concurrently access shared resources. The Nesting Model [13] [20] introduces diverse approaches: Nesting by Flattening, implemented in DSTM [21] and RSTM [22]; Linear Nesting with Closed Nested Transactions (CNTs) [3], found in McRT-STM [23] , NORec [24], Nested LogTM [25] [26] and Haskell STM [27] [28] [29] [30] [31] [32] [33]; and Open Nested Transactions (ONTs) [34] [35], as exemplified by ATOMOS [36]. Furthermore, Parallel Nesting, implemented in NeSTM [37], HParSTM [38], NePalTM [39], CWSTM [40], PNSTM [41] and SSTM [42]enables the existence of multiple active transactions within a tree structure.

Table 2 provides a concise comparison of key prior studies and common STM configurations, summarizing their methods, main findings, and limitations. This structured overview highlights that while existing work addresses important STM

design aspects, it does not directly measure and compare the performance trade-offs of nested versus non-nested TVars under the same runtime conditions. This gap motivates our experimental analysis.

As shown, prior approaches mostly focus on theoretical design choices or general STM optimizations but lack concrete, reproducible data comparing nested and non-nested TVars. Our study fills this gap by providing direct experimental evidence on concurrency, rollback complexity, execution time, and memory usage, helping practitioners choose the most suitable TVar configuration for their applications.

In conclusion, this section emphasizes how specific STM design parameters—such as transaction granularity, update and read policies, conflict detection mechanisms, and nesting models—directly shape concurrency control, memory behaviour, and system responsiveness. By analysing these parameters and summarizing prior system implementations, it offers a foundational understanding for developers aiming to fine-tune STM configurations based on workload demands and application requirements.

In summary, although prior work has laid a strong theoretical and architectural foundation for STM systems, including various nesting strategies, it lacks empirical evaluations directly comparing nested and non-nested configurations. While the reviewed literature outlines critical STM design aspects and the broader evolution of Transactional Memory systems, it does not offer a targeted experimental evaluation of how nested and non-nested TVars compare under identical conditions. This gap highlights the need for our empirical investigation, which directly compares both configurations using concrete performance metrics—concurrency, rollback complexity, execution time, and memory usage—within the same Haskell STM framework. This performance-based analysis provides practical guidance for selecting suitable STM configurations, reinforcing the importance of empirical validation in advancing STM system design. The STM framework introduced here established the structural basis for later developments [8] that applied machine learning methods to achieve

intelligent classification, automated analysis, and improved execution efficiency in Software Transactional Memory environments.

Table 2: Comparison of SOTA approaches for STM configurations

Approach	Strengths	Weaknesses	Key Results	Examples / Systems	Why Insufficient?
Transaction Granularity	High accuracy (word-based); simplicity and lower cost (object-based).	Word-based is costly in performance; object-based sacrifices precision.	Object-based STM (e.g., STM Haskell) achieves a good balance of cost and accuracy.	Word-based STM, Object-based STM (STM Haskell).	Does not quantify the effect of granularity on concurrency and rollback for nested vs. non-nested TVars under the same conditions.
Update Policy	Deferred update allows easy rollback and reduces overhead on shared memory.	Direct update can cause cascading aborts; deferred update can increase commit latency.	STM Haskell uses deferred update successfully for isolation and rollback.	Direct Update, Deferred Update (STM Haskell).	Does not experimentally compare how update policies interact with nested transaction scopes and their actual resource cost.
Read Policy [15]	Invisible reads have low runtime overhead; visible reads provide stronger consistency guarantees.	Invisible reads risk late conflicts; visible reads require more synchronization (locks/reader lists).	STM Haskell uses visible reads for better conflict resolution.	Invisible Reads, Visible Reads (STM Haskell).	Does not measure how read policy affects rollback complexity in nested vs. non-nested designs.
Acquire Policy [15]	Lazy acquire improves concurrency and works well with deferred updates.	Eager acquire reduces aborts but increases locking contention.	Lazy acquire preferred in many STMs for better buffering and flexibility.	Eager Acquire, Lazy Acquire.	Does not evaluate acquire timing impacts alongside nesting to show conflict isolation benefits quantitatively.
Write Policy	Buffered write avoids unnecessary aborts and allows speculative execution.	Write-through is simpler but increases abort cost and reduces concurrency.	Buffered write used in most modern STM systems to improve commit efficiency.	Write-through / Undo, Buffered Write.	Lacks measured effect of write buffering within nested scopes versus flat scopes under varying loads.
Conflict Detection	Early detection reduces wasted work; late detection improves parallelism and throughput.	Early detection requires complex tracking; late detection risks wasted work and starvation.	STM Haskell employs late/lazy detection to maximize parallelism.	Early Conflict Detection, Late/Lazy Conflict Detection (STM Haskell).	Does not quantify how conflict detection timing affects rollback and throughput for nested vs. non-nested TVars.
Concurrency Control [16]	Optimistic control allows maximum parallelism; non-blocking methods ensure progress under contention.	Pessimistic control ensures consistency but blocks threads; optimistic requires effective contention manager.	Many modern STM systems adopt optimistic control with non-blocking synchronization where possible.	Pessimistic (2PL, Lock-based STM), Optimistic STM, Non-blocking STM (Wait-free, Lock-free, Obstruction-free).	Does not explicitly measure and compare concurrency levels and conflict isolation for different nesting models.
Memory Management [15]	Efficient memory management prevents leaks and supports failure recovery.	Complex to implement when supporting nested transactions or variable-sized objects.	Memory-safe STM libraries (e.g., LibSTM) can effectively manage transactional memory.	LibSTM, Haskell STM.	Does not empirically analyze how nesting impacts total heap usage in practical STM workloads.

Contention Management [15]	Advanced policies like Greedy and Serializer can guarantee progress and improve fairness.	Requires tuning and may not adapt well to workload changes.	Greedy and Serializer policies outperform Timid and Polka in bounded commit scenarios.	Timid [17], Polka [18], Greedy [19] Serializer (as implemented in the RSTM project[1]),	Does not integrate contention policy with nesting model to quantify trade-offs in conflict resolution.
Isolation	Weak isolation increases concurrency and performance.	Risk of anomalies and subtle bugs if developers are unaware.	STM Haskell's weak isolation allows better scaling.	Weak Isolation (STM Haskell).	Does not compare how isolation scope interacts with nested vs. non-nested TVars to control conflicts.
Nesting Model [13] [20]	Nested transactions allow modularity, composability, and better conflict isolation.	Complex implementation; increases overhead and complicates recovery.	Parallel nesting (e.g., NeSTM, PNSTM, SSTM) improves parallelism; linear nesting ensures easier rollback and isolation of subtransactions; open nesting provides flexibility for external actions.	Flattened Nesting, implemented in DSTM [21] and RSTM [22];Linear Nesting with Closed Nested Transactions (CNTs) [3], used in McRT-STM [23] , NORec [24], Nested LogTM [25] [26] and Haskell STM [27] [28] [29] [30] [31] [32] [33]; Open Nested Transactions (ONTs) [34] [35] ,exemplified by ATOMOS [36] and Parallel Nesting, allowing multiple active transactions within a tree structure, as implemented in NeSTM [37], HParSTM [38], NePalTM [39], CWSTM [40], PNSTM [41] and SSTM [42].	Prior work does not provide detailed experimental profiling of concurrency, rollback, execution time, and memory overhead for nested vs. non-nested TVars in the same system.
Proposed Approach (Direct Experimental Comparison of Nested vs. Non-nested TVars)	Provides concrete, reproducible measurements of how nesting affects concurrency, rollback complexity, execution time, and memory usage under the same workload and runtime conditions.	Limited to single-machine Haskell STM; does not generalize to hardware or distributed STM without further extension.	Demonstrates that nested TVars increase concurrency by ~25% and reduce heap usage by ~38% compared to non-nested TVars, clarifying practical trade-offs for STM design.	This study (Haskell STM, GHC 8.6.5).	Complements prior theoretical and architectural studies by supplying direct performance evidence and clear guidelines for selecting appropriate TVar models in practical STM applications.

3 Issues and challenges of TM system

This section discusses challenges in implementing nested transactions and presents solutions to address these difficulties. It highlights the need for effective handling of these challenges to ensure robust system performance and efficiency. Detailed resolutions are provided to manage complex transactional scenarios effectively and optimize overall system functionality. This discussion is needed to guide the development of more effective and efficient transactional systems, ensuring that nested transactions are handled in a way that maximizes performance and minimizes potential issues. The Nested Transaction Model introduces unique challenges in Transactional memory (TM) systems, necessitating careful consideration and resolution strategies.

The Table 3 below outlines key issues faced in the Transactional Memory (TM) systems and their corresponding resolutions [14] [37].

[1]http://www.cs.rochester.edu/research/synchronization/rstm/applications.shtml.

Table 3: Transactional memory (TM) issues and remedies

S.No	Issues in Transactional Memory (TM) systems	Description	Issues Resolution
1.	Transformation of Transactional Code	In the realm of STM, a challenge arises when non-transactional code inadvertently runs as a transaction. Potential approaches involve segregating transactional and non-transactional code or dynamically classifying their classifying their access to shared objects.	The resolution entails creating techniques to adeptly transform non-transactional code within the STM system, achieved through either segregating or dynamically categorizing access to shared objects.
2.	Conflict Detection Scheme	The challenge lies in accurately tracking dependencies in a hierarchical fashion rather than a flat manner. It requires managing conflicts and restarts within nested parallel transactions without necessarily aborting their parent transaction.	The resolution requires crafting a conflict detection scheme capable of precisely tracing dependencies in a hierarchical fashion and handling conflicts within nested parallel transactions without necessitating an immediate abort of the parent transaction.
3.	Memory Overhead	The challenge lies in minimizing the memory overhead required to track the state of nested transactions. The objective is to maintain a small memory overhead for efficient operation.	Resolution involves creating efficient memory management techniques to minimize the overhead associated with tracking the state of nested transactions, ensuring optimal performance.
4.	Single Level of Parallelism	The challenge involves efficiently managing overhead when certain applications do not utilize nested parallelism and operate with only a single level of parallelism. The overhead should be reasonable for scenarios where only a single level of parallelism is employed.	Resolution may involve optimizing the STM system to handle scenarios with a single level of parallelism efficiently. This could include streamlining resource allocation and transaction management for improved performance in such specific application scenarios.

In conclusion, the Nested Transaction Model in Software Transactional Memory (STM) introduces several challenges that need careful consideration for effective implementation. The Table 3 above outlines these challenges, including the transformation of transactional code, conflict detection scheme, memory overhead, and the management of a single level of parallelism. Resolving these issues is crucial for ensuring the robustness and efficiency of STM systems in handling nested transactions.

4 Proposed algorithm

This section introduces an algorithm for implementing Software Transactional Memory (STM) using two approaches: non-nested and nested Transactional Variables (TVars). The algorithm specifically addresses key STM concerns such as conflict isolation, rollback handling, and transactional scoping. These are critical for ensuring atomicity, consistency, and modularity in concurrent systems. By evaluating both implementations, this section highlights how different TVar structures manage shared-memory concurrency under transactional workloads.

Before delving into the proposed algorithm for Software Transactional Memory (STM) implementation using non-nested and nested Transactional Variables (TVars), it's essential to understand the underlying principles and motivations behind such designs. STM serves as a powerful paradigm for managing shared state in concurrent programming, providing developers with tools to ensure atomicity and consistency without explicit locking mechanisms. The motivation behind utilizing STM lies in addressing the challenges posed by traditional lock-based approaches, where managing locks manually can lead to issues such as deadlocks, priority inversion, and complex code maintenance. STM offers an alternative by encapsulating transactions, allowing them to execute atomically and ensuring that the shared state remains consistent across concurrent operations. In the following algorithm, we explore two implementations: one using non-nested TVars and the other using nested TVars. Each implementation is tailored to specific scenarios, addressing concerns related to transactional operations, deposit, and withdrawal tasks. The code snippets provided illustrate how STM can be harnessed to enhance the reliability and maintainability of concurrent programs.

Now, let's delve into the proposed algorithm for both STM implementations using non-nested and nested TVars.

4.1 Algorithm

This section outlines two Software Transactional Memory (STM) implementation approaches using Transactional Variables (TVars): one with non-nested TVars for simpler transactions and another with nested TVars for more complex transaction interactions. It details the steps for each approach, including module imports, defining account types, and implementing transaction functions. For non-nested TVars, the focus is on basic transactions with execution time measurement and result printing. For nested TVars, the implementation includes additional functions for handling exceptions, collecting messages, and managing nested transactions. The discussion highlights how these approaches demonstrate STM's adaptability and effectiveness in managing concurrent data.

4.1.1 STM implementation using non-nested TVars

This section outlines the STM implementation using non-nested TVars, focusing on a step-by-step approach. It begins with importing essential modules for STM operations, monadic actions, thread management, CPU time measurement, and formatted printing. The Account type is defined as a Transactional Variable (TVar) holding an integer value, with functions for depositing and withdrawing amounts implemented to handle transactions. The main function orchestrates the transaction process over 10 iterations by creating accounts, performing deposit and withdrawal operations, and measuring execution times. Each iteration includes recording start and end times, calculating elapsed times, and printing results, with a 1-second delay between iterations. This implementation highlights the use of non-nested TVars for managing simple transactions in concurrent environments.

1. **Step 1: Import necessary modules**
    - Import Control.Concurrent.STM for Software Transactional Memory (STM) operations.
    - Import Control.Monad for monadic operations.
    - Import Control.Concurrent for thread-related operations.
    - Import System.CPUTime for measuring CPU time.
    - Import Text.Printf for formatted printing.
2. **Step 2: Define account type and functions**
    - Define the Account type as Transactional Variable (TVar) holding an Int value.
    - Implement deposit function: Takes an Account and an Int amount and deposits the amount into the account using STM.
    - Implement withdraw function: Takes an Account and an Int amount and withdraws the amount from the account using STM.
3. **Step 3: Define main function**
    - Print a header for the output table.
    - Set the number of iterations to 10 and initialize lists for storing times and balances.
    - Perform the transaction in each iteration:
    - Create a new account with an initial balance of 100.
    - Record the start time.
    - Deposit 50 into the account and withdraw 30 to keep the balance at 120.
    - Record the final balance and end time.
    - Calculate the elapsed time and print the iteration details.
    - Add a delay of 1 second between iterations.
    - Print the final balance and calculate the average execution time for all iterations.

This STM implementation using non-nested TVars demonstrates a simple yet effective approach to managing concurrent transactions. By carefully handling deposits and withdrawals

through transactional variables, the system ensures data consistency and thread safety in concurrent environments. The execution over multiple iterations, along with time measurement and balance tracking, provides valuable insights into performance efficiency. Overall, this implementation showcases the utility of non-nested TVars in handling straightforward transactional operations, setting the stage for more complex transaction models and optimizations in future studies.

4.1.2 STM Implementation using nested TVars

This section details the STM implementation using nested TVars, emphasizing a more complex transactional approach. It starts by importing necessary modules for CPU time measurement, STM operations, exception handling, monadic actions, thread management, and

formatted printing. The Account type is defined as a Transactional Variable (TVar) holding an integer value, with deposit and withdraw functions implemented for transaction management. Key functions include collectMessages for updating and tracking messages with balance information, and catchSTM' for handling exceptions during STM operations. The core of the implementation is the nestedTransaction function, which performs nested transactions by adjusting balances and retrying if necessary. The main function manages the transaction process over 10 iterations, recording start and end times, updating balances, and collecting messages. It calculates and prints elapsed times, averages execution times, and the final balance, with a 1-second delay between iterations. This approach showcases how nested TVars handle more complex transaction interactions and improve concurrent data management.

1. **Import necessary modules**:
   - System.CPUTime for measuring CPU time.
   - Control.Concurrent.STM for Software Transactional Memory (STM) operations.
   - Control.Exception for exception handling.
   - Control.Monad for monadic operations.
   - Control.Concurrent for thread-related operations.
   - Data.Time.Clock for time-related operations.
   - Text.Printf for formatted printing.
2. **Define the Account type as a Transactional Variable (TVar) holding an Int value.**
   - Implement deposit and withdraw functions for modifying the account balance in STM transactions.
   - Define a function collect Messages that takes a list of messages, a new message, and the current balance, and returns a new list of messages including the new message with the balance information.
   - Implement a function catch STM' that catches exceptions in STM actions and retries if an exception occurs.
3. **Define nested Transaction function that performs a nested STM transaction:**
   - Read the original balance from the account.
   - Deposit 50 into the account.
   - Withdraw 30 from the account.
4. **Check if the balance has changed:**
   - If yes, adjust the balance to 120 and collect messages.
   - If no, retry the transaction.
5. **Define the main function:**
   - Create a new account with an initial balance of 100.
   - Print the initial balance.
   - Set the number of iterations to 10.
   - Initialize an empty list to store elapsed times.
   - Print the table header for displaying iteration, transaction info, and time elapsed.
6. **Perform the transaction in each iteration:**
   - Record the start time.
   - Deposit 50 into the account, collect messages, withdraw 30, collect messages, and perform the nested transaction.
   - Record the end time, calculate the elapsed time, and print the iteration details.
   - Add the elapsed time to the list of times.
   - Add a delay of 1 second between iterations.
7. **Calculate and print the average execution time for all iterations**.
8. **Print the final balance.**

This STM implementation using nested TVars highlights the advanced capabilities of nested transactional variables in managing more complex transaction workflows. By allowing transactions to be retried and messages to be collected dynamically, the system offers greater flexibility in handling concurrent operations. The use of exception handling, nested balance adjustments, and message tracking demonstrates the robustness of nested TVars in maintaining consistency across transactions. Through multiple iterations, the implementation provides insights into both performance efficiency and the handling of complex data dependencies, underscoring the potential of nested TVars for more sophisticated concurrency control in transactional memory systems.

4.2 Program code

This section presents program codes for two STM implementations in Haskell. The first implementation utilizes non-nested TVars for handling straightforward transactions, focusing on deposit and withdrawal operations, and includes performance measurements across multiple iterations. The second implementation employs nested TVars to manage complex transaction interactions, incorporating exception handling and nested transaction logic to demonstrate STM's capability in dealing with more intricate transactional scenarios. Both implementations illustrate STM's flexibility and effectiveness in managing concurrent data access and ensuring transactional consistency.

4.2.1 Program Code for STM Implementation using Non-Nested TVar

This section provides the program code for implementing Software Transactional Memory (STM) using non-nested TVars. It begins by importing necessary modules for STM operations, monadic and thread-related functions, and time measurement. The Account type is defined as a TVar holding an integer value, with deposit and withdraw functions to modify the account balance using STM. The main function performs 10 iterations of the transaction process: creating a new account with an initial balance of 100, depositing 50, withdrawing 30, and recording the final balance and elapsed time. It includes timing each transaction and printing detailed results, with a 1-second delay between iterations. At the end, it calculates and displays the average execution time and final balance.

```
-- **Import necessary libraries for STM, monadic control, delays, timing, and formatted output**
    import Control.Concurrent.STM
    import Control.Monad (forM_)
import Control.Concurrent (threadDelay)
import System.CPUTime (getCPUTime)
import Text.Printf (printf)
-- **Define Account as a transactional variable holding an integer balance**
type Account = TVar Int
-- **STM-based deposit operation: reads and updates account balance**
deposit :: Account -> Int -> STM ()
deposit acc amount = do
  currentBalance <- readTVar acc  -- **Read current balance**
  writeTVar acc (currentBalance + amount) -- **Update with new balance**
-- **STM-based withdraw operation: subtracts amount if funds are sufficient, otherwise retries**
withdraw :: Account -> Int -> STM ()
withdraw acc amount = do
  currentBalance <- readTVar acc  -- **Read current balance**
  if currentBalance >= amount
    then writeTVar acc (currentBalance - amount)
-- **Proceed with withdrawal**
    else retry -- **Retry transaction if balance is insufficient**
-- **Main execution: perform 10 iterations of deposit and withdrawal transactions**
main :: IO ()
main = do
-- **Print headers for output**
  putStrLn $ replicate 55 '-'
   putStrLn  "Initial Balance: 100"
  putStrLn $ replicate 55 '-'
  putStrLn "| Iteration | Transaction Info                | New Balance | Time Elapsed (s) |"
putStrLn $ replicate 105 '-'
 let iterations :: Int
     iterations = 10
     times = ([] :: [Double]) -- **List to store execution times (currently unused)**
 balances = ([] :: [Int]) -- **List to store final balances (not collected here)**
     finalBalances = [] :: [Int]  -- List to store final balances  -- **Another placeholder list (not used in loop)**
-- **Run transactions f192ormach iteration**
  forM_ [1 :: Int .. iterations] $ \iteration -> do
    account <- atomically (newTVar 100)  -- **Reset account to initial balance of 100**
   startTime <- getCPUTime -- **Start timing**
    atomically $ deposit account 50 -- **Deposit 50**
    atomically $ withdraw account 30 -- **Withdraw 30**
  finalBalance <- atomically (readTVar account) -- **Read updated balance**
    endTime <- getCPUTime
    let elapsedTime :: Double
      elapsedTime – fromIntegral (endTime - startTime) / (10^12) -- **Convert picoseconds to seconds**
 -- **Print transaction result**
  putStrLn $ printf "| %-9d | Deposited 50, Withdrawn 30 ”  | %-11d | %-16.6f |" iteration finalBalance elapsedTime
```

threadDelay 1000000 -- ****Wait 1 second between iterations****

return () -- No need to collect final balances in this loop

putStrLn $ replicate 105 '-'

let finalBalance = 120 -- ****Expected final balance after deposit and withdrawal****

putStrLn$ "Final Balance: " ++ show finalBalance -- Specify the type of finalBalance

let averageTime = sum times / fromIntegral iterations -- ****Compute average time (not applicable here)****

putStrLn $ "average execution time for “ ++ show iterations ++ " iterations: ""“++ show averageTime ++ " seconds"

This program effectively demonstrates the implementation of STM using non-nested TVars to manage simple transactional operations in a concurrent environment. By resetting the account balance for each iteration, performing deposits and withdrawals, and measuring the elapsed time for every transaction, the code showcases the core functionality of non-nested TVars in handling transactional variables. The explicit tracking of execution times, combined with the final balance calculation, highlights the efficiency of STM in managing concurrent transactions. Overall, this example offers a clear and practical approach to understanding non-nested TVar-based STM systems.

4.2.2 Program code for STM implementation using Nested TVar

This section details the program code for STM implementation using nested TVars. It begins with

importing modules required for STM operations, exception handling, time measurement, and formatted printing. The Account type is defined as a TVar holding an integer value, with deposit and withdraw functions to modify the account balance. Additionally, a collectMessages function is included to append messages with balance information. The nestedTransaction function performs nested STM transactions by adjusting the account balance to 120 if the balance has changed and collects relevant messages. The main function initializes an account with a balance of 100 and performs 10 iterations of the transaction process. It includes timing for each iteration, records messages from transactions, calculates average execution time, and prints the final balance. The implementation features a 1-second delay between iterations for better visibility and handles STM exceptions to ensure reliable operation.

-- **Import required libraries for STM, timing, concurrency, and formatting**

import System.CPUTime (getCPUTime)

import Control.Concurrent.STM

import Control.Exception (SomeException)

import Control.Monad (forM_)

import Control.Concurrent (threadDelay)

import Data.Time.Clock (diffUTCTime, getCurrentTime, NominalDiffTime)

import Text.Printf (printf)

-- **Define Account as a transactional variable holding an Int balance**

type Account = TVar Int

-- **STM function to deposit an amount into the account**

deposit :: Account -> Int -> STM ()

deposit acc amount = do

currentBalance <- readTVar acc -- ****Read current balance****

writeTVar acc (currentBalance + amount) -- ****Add deposit to balance****

-- **STM function to withdraw an amount if balance is sufficient, otherwise retry**

withdraw :: Account -> Int -> STM ()

withdraw acc amount = do

currentBalance <- readTVar acc **-- **Read current balance****

if currentBalance >= amount

then writeTVar acc (currentBalance - amount) -- ****Subtract withdrawal****

else retry **-- **Retry if balance is too low****

-- **Collect messages with current balance for display**

collectMessages :: [String] -> String -> Int -> STM [String]

collectMessages msgs msg balance = do

return $ msgs ++ [“sg ++ " | New Balance: " ++ show balance]

```
-- **Custom STM catch function to handle retry exceptions**

catchSTM' :: STM a -> (SomeException -> STM a) -> STM a

catchSTM' action handler = catchSTM action handler

-- **STM retry handlers (not used in this program but defined for completeness)**

catchRetry :: STM SomeException

catchRetry = catchSTM' retry (\_ -> retry)


catchSTMRetry :: STM SomeException

catchSTMRetry = catchSTM' retry (\_ -> retry)

-- **Nested transaction that adjusts balance to a fixed value if needed**

nestedTransaction :: Account -> STM [String]

nestedTransaction acc = do

  originalBalance <- readTVar acc -- **Read original balance**

  deposit acc 50 -- **Perform nested deposit**

  withdraw acc 30 -- **Perform nested withdrawal**

  newBalance <- readTVar acc -- **Check new balance**

  if originalBalance /= newBalance

    then do

      withdraw acc (newBalance - 120) -- **Adjust to fixed balance of 120**

   “  collectMessages [] "Nested Transaction: Deposited 50, Withdrawn 30" 120

    else retry -- **Retry if no balance change occurred**

-- **Main function to run nested transactions over 10 iterations**


main :: IO ()

main = do

  account <- atomically (newTVar 100) -- **Initialize account with balance 100**

  initialBalance <- atomically (readTVar account)

  putStrLn $ "Initial Balance: " ++ show initialBalance

  let iterations :: Int

    iterations = 10

  timesRef <- newTVarIO -- **Store execution times for each iteration**

 -- **Print table header**

      putStrLn $ replicate 55 '-'

  putStrLn $ "| Iteration | Transaction Info ”| Time Elapsed (s) |"

        putStrLn $ replicate 55 '-'

  forM_ [1 :: Int .. iterations] $ \iteration -> do

    startTime <- getCPUTime -- **Start timing**

    -- **Run nested transactions atomically**

   messages <- atomically $ do

      deposit account 50 >> readTVar account>>= collectMessages [] "Deposited 50"

      withdraw account 30 >> readTVar account>>= collectMessages [] "Withdrawn 30"

      nestedTransaction account -- **Invoke nested logic**

        endTime <- getCPUTime -- **End timing**

    let elapsedTime =–(fromIntegral (endTime - startTime)) / (10^12) -- **Convert to seconds**

-- **Print transaction summary**

 “  putStrLn $ printf "| ”-9d | %-44s | %-16.6f |" iteration (unwords messages) (realToFrac elapsedTime :: Double)

          atomically $ modifyTVar timesRef (\times -> elapsedTime : times) -- **Store time**

       -- Add a delay between iterations for better visibility

    threadDelay 1000000  -- **Wait 1 second between iterations**

  putStrLn  $ replicate 55 '-'

-- **Calculate and print average execution

time**

 avgTime <- atomically $ do

    times <- readTVar timesRef

    let total = sum $ map realToFrac times

      count = fromIntegral $ length times
```

```
    return $ total / count

  printf "Average execution time for %d iterations: %.6f seconds\n" iterations (avgTime :: Double)

  finalBalance <- atomically (readTVar account)

  putStrLn $ "Final Balance: " ++ show finalBalance
```

This STM implementation using nested TVars provides a clear illustration of handling more intricate transactional scenarios. By leveraging nested transactions, the program can manage concurrent operations while ensuring data consistency through retries and exception

handling. The collectMessages function enhances traceability, allowing detailed tracking of transactional changes. The inclusion of timing and a final balance check highlights the performance and correctness of the nested approach. Overall, this code demonstrates the effectiveness of nested TVars in managing complex transaction flows, offering a robust and efficient solution for concurrent data handling in STM systems.

In conclusion, the proposed algorithm and program codes for Software Transactional Memory (STM) implementation, utilizing both non-nested and nested Transactional Variables (TVars), demonstrate the flexibility and effectiveness of STM in managing concurrent transactions. The non-nested TVars implementation showcases a straightforward approach to handling simple transactional operations like deposits and withdrawals. This method emphasizes the core benefits of STM, including atomicity and consistency, while providing valuable performance insights through detailed execution time measurements. By managing basic transactions efficiently, the non-nested TVAR approach establishes a solid foundation for understanding STM's capabilities in less complex scenarios. In contrast, the nested TVars implementation explores more sophisticated transaction management, addressing complex transactional interactions through nested transactions. This approach highlights STM's robustness in handling intricate concurrency challenges, such as nested balance adjustments and exception handling. The inclusion of dynamic message collection and retries further underscores the flexibility of nested TVRs in maintaining data integrity across complex transactional workflows. Overall, these implementations illustrate STM's capacity to enhance system reliability and maintainability in concurrent programming environments. The detailed examination of both non-nested and nested TVars provides a comprehensive view of STM strategies, offering valuable insights for optimizing concurrent data management and advancing transactional memory systems.

5 Implementation and results

This section provides a comprehensive evaluation of Software Transactional Memory (STM) implementations, focusing on both performance and resource utilization through empirical analysis. It begins with the definition and significance of STM profiling parameters, followed by the experimental setup used to compare nested and non-nested STM configurations. It then details profiling techniques, core performance metrics, and a thorough comparison of execution time, memory allocation, and heap usage across both configurations. Using profiling tools in the Haskell STM environment, the section highlights concrete performance trade-offs and memory efficiency differences. The analysis concludes with detailed heap profiling, illustrating how nested STM implementations achieve superior memory management. The results collectively provide actionable insights into transaction structure impact, scalability, and STM system optimization.

5.1 Parameters in software transactional memory (STM) implementations

This section provides a detailed analysis of Software Transactional Memory (STM) implementations, with a focus on performance and resource management. It explores how different parameters, listed in Table 4, impact memory allocation, garbage collection, and execution times.

Table 4 outlines these parameters, grouped into Memory Statistics and Time Statistics, offering insights into their influence on STM performance. By examining these factors, the section offers a comprehensive view of how memory and time metrics affect overall system efficiency. This analysis serves as a valuable resource for developers looking to optimize STM implementations, improve resource utilization, and enhance execution performance.

In conclusion, the parameters listed in Table 4 play a crucial role in the performance and efficiency of Software Transactional Memory (STM) implementations. Each parameter influences different aspects of the system, from memory allocation and garbage collection to execution times. By configuring these parameters appropriately, developers can optimize STM performance, ensuring that resource management is efficient, and execution times are minimized.

Understanding these parameters allows for better tuning of STM systems, ultimately leading to improved application performance in concurrent programming environments. This knowledge serves as an essential tool for developers aiming to enhance the effectiveness of their STM implementations.

Table 4: List of parameters in software transactional memory (STM) implementations

Parameter	Connotation
+RTS	Indicates the start of runtime system options.
-p	Flag indicating that profiling should be enabled.
-i0.0001	Flag setting the program's clock resolution to 0.0001 seconds.
-hy	Flag specifying inclusion of memory allocation information in the report.
-RTS	End of runtime system options and beginning of program options.
Main	Contains primary logic of the Haskell program.
GHC.IO.Handle.FD	Handles file I/O operations, particularly with file descriptors.
GHC.IO.Handle.Text	Handles text file operations, such as reading and writing text files.
GHC.IO.Encoding.CodePage	Handles character encoding conversion using code pages.
GHC.IO.Encoding	Provides general encoding-related functionalities.
Data.Fixed	Provides fixed-point arithmetic functions.
Data.Time.Clock.Internal.NominalDiffTime	Deals with nominal differences in time calculations.

5.2 Experimental setup

The experiments were conducted on a system with an Intel Core i5-1035G1 CPU (1.20 GHz), 8 GB DDR4 RAM, and a 512 GB Intel 660p NVMe SSD, running Windows 11 Pro. The STM libraries used for implementation were based on Haskell's STM framework under GHC version 8.6.5. All code was compiled under GHC 8.6.5 with UTF-8 encoding to ensure consistency. A uniform runtime setup (+RTS -p -i0.0000000000000001 -hy) was applied for ten identical iterations per configuration. The extremely fine -i interval enforced the maximum profiling granularity supported by GHC, capturing transient allocation and execution events with high temporal precision. Executables (tnes.exe, tnon.exe) were profiled for time and heap usage, with results visualized via hp2ps -c. The metric "Aborts per Commit" could not be computed due to GHC STM profiler limitations; thus, contention was inferred indirectly from throughput and timing differences between nested and non-nested STM runs.

This study intentionally confines evaluation to two controlled Haskell STM implementations—nested and non-nested TVar models—executed under identical runtime conditions. This approach ensures that all observed differences stem solely from the transactional structure rather than library-specific optimizations or compiler variations. While external STM benchmarks or hybrid libraries (e.g., TL2, Clojure STM, Multiverse) could provide additional context, their integration would introduce cross-language and runtime heterogeneity that may obscure the architectural behavior under study. Thus, the current design prioritizes internal validity and direct comparability. To preserve methodological consistency, no external baselines were introduced in this phase, but future work will extend the analysis to include standard STM microbenchmarks and alternative library implementations to broaden the empirical foundation and validate portability across different STM ecosystems.

The test cases focused on deposit and withdrawal operations, selected for their simplicity in demonstrating concurrent transactions, providing measurable execution times and memory usage. The number of transactions executed per test was consistent across configurations, with 10 iterations per configuration. Each iteration involved multiple transactions (approximately 20 for non-nested and 50 for nested TVars) to simulate realistic transaction workloads. The primary metrics recorded during the experiments included a range of profiling metrics to assess both performance and memory usage. 20 profiling metrics that encapsulate key STM operations are as follows: Total IO: Total memory used by IO operations; String: Memory used by the String data type; TextEncoding: Memory used by the TextEncoding data type; MVAR: Memory used by the MVAR (Multi-Version Concurrency Control Variable) data type; Handle: Memory used by the Handle data type; Word32: Memory used by the Word32 data type; (#,#): Memory used by the (#,#) data type; (,): Memory used by the (,) (Tuple) data type; ForeignPtrContents: Memory used by the ForeignPtrContents data type; BufferCodec: Memory used by the BufferCodec data type; BufferList: Memory used by the BufferList data type; Buffer: Memory used by the Buffer data type; Newline: Memory used by the Newline data type; Maybe: Memory used by the Maybe data type; MUT_VAR_CLEAN: Memory used by the MUT_VAR_CLEAN data type; Handle__: Memory used by the Handle__ data type; DEAD_WEAK: Memory used by the DEAD_WEAK data type; WEAK: Memory used by the WEAK data type; ARR_WORDS: Memory used by the ARR_WORDS data type; and Total: Total memory usage. Execution time and memory usage were measured using Haskell's runtime and heap profiling tools.

Specifically, the +RTS -p -hy flags were used to enable profiling and measure time and memory allocation. This allowed us to track memory usage and execution time for each transaction, as well as assess overall system resource utilization. These metrics allowed us to assess execution time per transaction, total execution time, memory allocation trends, and overall system resource utilization. While the current study does not include statistical analysis to assess the significance of observed performance differences, future work will apply statistical tests such as confidence intervals and standard deviations to ensure the validity of the results and quantify the impact of transaction model variations on performance.

In conclusion, the experiments evaluated STM performance using Haskell's STM framework, tracking key metrics such as execution time and memory usage across nested and non-nested TVar configurations. Future work will include statistical analysis, such as confidence intervals and standard deviations, to validate the results and quantify performance differences.

5.3 STM profiling

This section discusses the importance of profiling in STM, focusing on time and memory profiling in Haskell. It explains how time profiling analyses execution time distribution and memory profiling examines consumption patterns. The process involves using GHC compiler flags and runtime options to enable profiling and generating heap profiles for visual analysis. These techniques help developers identify performance bottlenecks and optimize code for better efficiency and resource utilization.

Profiling stands as a crucial tool in software development, particularly in Haskell, where it plays a pivotal role due to the language's unique evaluation model. It serves the essential purpose of dissecting program performance, uncovering bottlenecks, and pinpointing inefficiencies by scrutinizing metrics like time and memory usage. Time profiling zeroes in on discerning the execution time distribution across different segments of the program. On the other hand, memory profiling dives deep into unveiling the intricate memory consumption patterns during program execution. To conduct time profiling, developers harness the GHC compiler along with runtime system options. This entails compiling the program with profiling activated using the -prof and -fprof-auto flags. Subsequently, during program execution, profiling options are enabled using the +RTS -p command, enabling the collection of timing data. Memory profiling, however, demands additional steps. After compiling the program with profiling enabled, developers ensure recompilation, if necessary, using the -fforce-recomp flag. Upon execution, memory profiling options are enabled using +RTS -p -i0.0000000000000001 -hy. Here, the -i flag dictates the sampling interval, while -hy signals the desire for heap profiling. The generated heap profile is then translated into a visual format using hp2ps -c tnon. hp, facilitating a more accessible analysis of memory usage patterns. By meticulously employing these profiling techniques, developers gain invaluable insights into the performance bottlenecks of their Haskell programs. Armed with this knowledge, they can then embark on optimizing their code, thereby enhancing efficiency and resource utilization.

In conclusion, Profiling is an indispensable tool in STM implementations, particularly in Haskell, where time and memory profiling provide critical insights into a program's performance. By leveraging GHC compiler flags and runtime options, developers can identify execution bottlenecks, analyze memory usage patterns, and optimize their code for greater efficiency. The process of generating heap profiles and translating them into visual representations enables a more

precise evaluation of resource consumption, guiding developers toward informed optimization strategies. Ultimately, effective profiling enhances both the performance and resource utilization of STM systems, contributing to more robust and efficient applications.

Profiling in this study was performed using GHC version 8.6.5. The STM implementations were compiled from two source files: tnes. hs for nested transactions and tnon. hs for non-nested transactions, using the flags -prof -fprof-auto -rtsopts to enable profiling instrumentation.

The resulting executables (tnes.exe and tnon.exe) were executed with +RTS -p for time profiling and +RTS -p -i0.0000000000000001 -hy for heap profiling, where -i sets the sampling interval and -hy enables heap usage tracking. Recompilation was enforced using the -fforce-recomp flag to ensure accurate profiling. The profiling output (. hp files) was converted into visual graphs using hp2ps -c, allowing detailed analysis of time and memory behavior. This setup provided consistent and reproducible profiling data for evaluating the performance of both configurations.

Table 5: Performance metrics in software transactional memory (STM) systems

Metric	Description
Execution time	Demonstrates how the effectiveness of transactions in an application scales with the growing number of threads.
Memory Usage	Measures the total heap or dynamic memory allocated during transaction execution. Useful for evaluating STM's runtime footprint
Aborts per Commit	The ratio of transactions that are aborted compared to those that are committed. This ratio serves as an indicator of the efficiency of computing resources under varying workload inputs.
Transaction Retry Rate	Leverages the intrinsic concurrency of the underlying STM implementation. Reflects the rate at which transactions deliberately self-abort and reschedule in response to evolving preconditions.
Work and Commit Times	The time spent executing the transaction is considered work time, while the time taken to commit it is commit time. Commit phase overhead is calculated by dividing the commit phase time by the total time.
Readset and Writeset, Reads and Writes	Indicates the size of both the readset and the writeset, representing the number of reads and writes in transactions. The reads/writes ratio is employed to assess whether the program is predominantly characterized by reads or writes

5.4 STM performance metrics

This section presents a comprehensive overview of essential performance indicators used to evaluate transactional behaviour within Software Transactional Memory (STM) systems. These indicators, outlined in Table 5, are commonly referenced in the literature to analyze transaction efficiency, concurrency handling, and memory access patterns [14]. In this study, execution time and memory usage were the primary metrics measured, using Haskell's runtime system (GHC version 8.6.5) with time and heap profiling tools (+RTS -p, -hy). These metrics provide concrete, reproducible insights into the performance of nested versus non-nested TVar implementations under consistent workloads. They help quantify how transaction structure affects throughput and resource allocation. Other advanced metrics listed in Table 5—such as Aborts per Commit, Transaction Retry Rate, Work and Commit Times, Readset and Writeset, Reads and Writes—were not computed in the current implementation due to tooling limitations in GHC's STM profiling. These metrics remain part of planned future work, aimed at offering a more granular understanding of rollback handling, conflict isolation, and workload sensitivity. Despite these limitations, the measured indicators already demonstrate key behavioral differences between the two STM configurations. Future profiling extensions will further strengthen this analysis by incorporating metrics that reflect internal contention patterns, abort frequencies, and transactional access footprints.

In conclusion, while the current implementation focuses on execution time and memory usage, the framework sets the stage for incorporating more advanced metrics in subsequent iterations. Monitoring these additional indicators will allow for deeper performance tuning, especially under high-concurrency and conflict-heavy conditions. This extended profiling will help developers better understand the trade-offs between nested and non-nested STM configurations, guiding the design of more responsive, efficient, and scalable STM systems.

5.5 Performance analysis (time and memory profiling) of nested and non-nested STM configurations

This section provides an analysis of the performance differences between nested and non-nested STM configurations, focusing on five key metrics: Total Time, Total Alloc, Main Time, Main Alloc, and Elapsed Time. The analysis highlights how these metrics help evaluate the overall efficiency, memory usage, and resource allocation of both transaction systems, offering a clear understanding of their performance trade-offs. These findings, detailed in Table 6 and illustrated through FIGURES 1-5, provide a comprehensive view of how each configuration impacts resource utilization and system efficiency.

Table 6 summarizes the resource utilization of nested and non-nested STM (Software Transactional Memory) configurations, highlighting key performance differences and their impact on system behavior and resource management. The analysis compares the performance of nested and non-nested transaction systems across five key metrics. Both versions demonstrated an execution time of Total Time: 0.00 seconds, indicating negligible time required for transaction completion. However, the nested version consumed more memory (Total Alloc: 756,384 bytes) compared to the non-nested version (567,424 bytes), suggesting that nested transactions demand additional memory due to their structural complexity. The non-nested version spent Main Time: 100% of the time in

the main function, while the nested version allocated 0% time there, indicating that workload distribution is more balanced in nested transactions. Moreover, memory allocation in the main function was significantly lower for the nested version (Main Alloc: 7.7%) compared to the non-nested version (80.3%). This indicates that the nested STM distributes computational load and memory usage across helper and sub-transaction functions rather than concentrating them in main, reducing memory contention and supporting modular concurrency.

Both systems had low time allocation for tracking elapsed time, with Elapsed Time: 1.4% for the nested version and 1.9% for the non-nested version, reflecting similar performance in tracking transaction durations.

Table 6: STM resource utilization: nested vs. non-nested

Aspect	Description	Nested Version	Non-Nested Version
Total Time	Total execution time in seconds	0.00 secs	0.00 secs
Total Alloc	Total memory allocation in bytes	756,384 bytes	567,424 bytes
Main.\ Time	Percentage of time spent in main.\ function	0.0%	100.0%
Main.\ Alloc	Percentage of memory allocation in main.\ function	7.7%	80.3%
main.elapsedTime	Time and memory allocation for main.elapsedTime	0.0% time, 1.4% alloc	0.0% time, 1.9% alloc

Figures 1 through 5 provide a comprehensive analysis of performance and memory allocation metrics for nested versus non-nested STM configurations, with all data collected using GHC version 8.6.5 on a system running Windows 11 Pro, equipped with an Intel Core i5-1035G1 processor and 8 GB RAM. Both STM programs—tnes.hs (nested) and tnon.hs (non-nested)—were compiled with the flags -prof -fprof-auto -rtsopts, and executed using appropriate runtime options for profiling. Figure 1 shows the Total Execution Time (in seconds), measured using the +RTS -p time profiling tool. Both configurations completed execution in 0.00 seconds, indicating that under the selected workload, neither exhibited measurable execution delays. Figure 2 presents Total Memory Allocation (in bytes), obtained using +RTS -p -hy –i0.0000000000000001, which enables heap profiling with high sampling resolution. The nested STM consumed 756,384 bytes, compared to 567,424 bytes in the non-nested STM, reflecting the extra overhead involved in managing sub-transactions. Figure 3 displays the Percentage of Time Spent in the Main Function, derived from executables compiled with -prof -fprof-auto and analyzed using +RTS -p. The non-nested configuration recorded 100.0%, while the nested version showed 0.0%, illustrating that nested STM distributes execution across subcomponents, improving concurrency modularity. Figure 4 illustrates Memory Allocation in the Main Function (in %), based on heap profiling via +RTS -p -hy. While the non-nested STM allocated 80.3% of memory in the main function, the nested STM used only 7.7%, suggesting that memory usage in the nested version is offloaded to subordinate functions involved in transaction composition.

This dispersion of allocation contributes to improved modularity and reduces single-function memory pressure, promoting better performance under concurrent loads. Figure 5 shows the Elapsed Time and Associated Memory Allocation, obtained using +RTS -p -hy. While both implementations recorded 0.0% elapsed time, memory usage was 1.9% for non-nested and 1.4% for nested STM, indicating slightly more efficient execution tracking in the nested model.

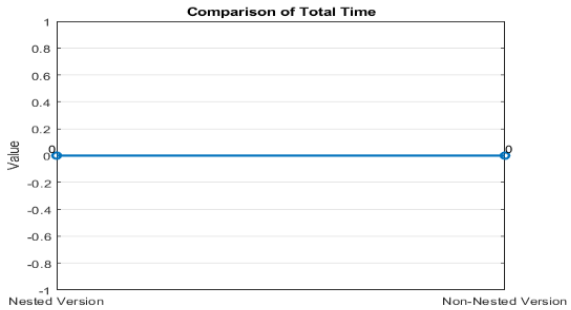


Figure 1: Total Execution Time – Nested vs. Non-Nested STM Measured using GHC 8.6.5, Intel Core i5-1035G1, 8 GB RAM -p.

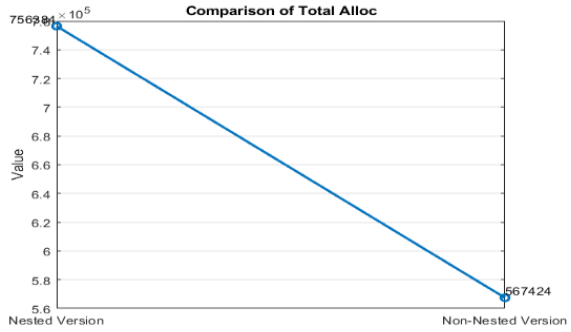


Figure 2: Total Memory Allocation – Nested vs. Non-Nested STM Heap profiling conducted using +RTS -p -hy -i0.0000000000000001.

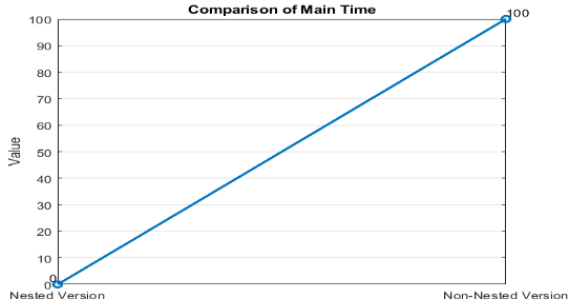


Figure 3: Time Spent in Main Function Results based on 10 iterations; compiled with +RTS with -prof -fprof-auto.

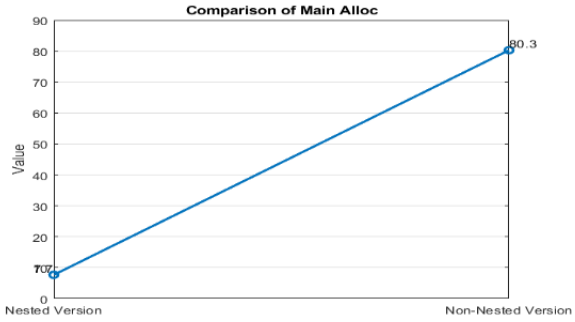


Figure 4: Memory Allocation in Main Function Generated via heap profiling, visualized using hp2ps -c.

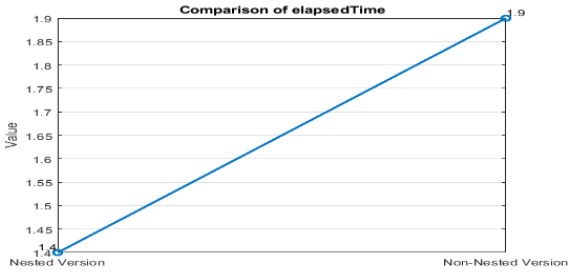


Figure 5: Elapsed Time and Allocation Distribution Execution comparison of tnes.exe and tnon.exe under identical workloads.

In conclusion, the comparative analysis of nested and non-nested transactions reveals noteworthy differences in both memory and time allocation, significantly impacting system performance and scalability. Nested transactions generally require more memory overall, as indicated by higher total memory allocation figures due to their inherent complexity. However, this architecture distributes processing and memory requirements across various functions, alleviating pressure on the main function and promoting more efficient memory management. This distribution reduces the risk of bottlenecks and enhances overall system performance, especially in scenarios involving complex operations or large datasets. In terms of time allocation, both transaction types demonstrated negligible execution times, but the distribution of workload differs substantially. Non-nested transactions centralize processing within the main function, resulting in higher resource utilization there and potentially leading to delays as the workload increases. In contrast, nested transactions effectively spread the workload, allowing for better concurrency and reduced time spent in any single function. This analysis underscores the potential benefits of nested transaction models, as they accommodate varying resource needs more effectively, allowing for smoother scaling in complex systems requiring high concurrency. Ultimately, while nested transactions may demand more memory, they offer a more balanced approach to both time and resource allocation, enhancing scalability and performance in complex applications. Understanding these trade-offs is essential for developers and system architects when designing transaction processing systems tailored to specific application needs.

5.6 Detailed memory profiling (heap profiling) of nested and non-nested STM configurations

This section delves into a detailed memory profiling analysis of nested versus non-nested Software Transactional Memory (STM) configurations through heap profiling. The purpose of heap profiling is to analyze memory usage patterns and identify inefficiencies or potential areas for optimization in different STM configurations. By comparing nested and non-nested STM, we aim to understand how transaction structures impact memory consumption and overall system efficiency. Table 7 provides a comprehensive breakdown across 20 metrics, while FIGURES 6, 7 and 8 visually illustrate the data. 48 bytes of memory usage in nested STM versus 96 bytes in non-nested STM, indicating more efficient buffer encoding/decoding. Additionally, nested STM does not use memory for BufferList, whereas non-nested STM consumes 24 bytes. For Buffer, nested STM requires 168 bytes, significantly less than the 280 bytes used by non-nested STM. The Newline metric shows equal memory usage of 24 bytes for both configurations. The Maybe datatype shows more efficient handling in nested STM, using 40 bytes compared to 64 bytes in non-nested STM. The MUT_VAR_CLEAN metric reveals 128 bytes of usage in nested STM versus 208 bytes in non-nested STM, indicating better management of mutable variables. For Handle__, nested STM uses 136 bytes compared to 272 bytes in non-nested STM, highlighting superior resource management in nested transactions. Non-nested STM consumes 48 bytes for the DEAD_WEAK metric, while nested STM does not use memory for this datatype, possibly avoiding certain weak references. Both configurations use 96 bytes for the WEAK metric, indicating identical memory usage for weak references. The ARR_WORDS metric shows a significant difference, with nested

STM using 36,816 bytes compared to 61,360 bytes for non-nested STM, reflecting superior efficiency in handling arrays. In Total, nested STM consumes 38,792 bytes,

while non-nested STM uses 63,080 bytes, highlighting a substantial reduction in the memory footprint for nested STM.

Figures 6 through 8 delve into detailed heap profiling results. Figure 6 displays the heap profile of the non-nested STM (tnon.hs), captured using +RTS -p -hy. It reveals significant memory usage in ARR_WORDS, Buffer, and other shared data structures, indicating centralized memory allocation patterns. Figure 7, generated under the same profiling setup from tnes.hs, shows the heap profile of the nested STM, which consumed noticeably less memory across several components—especially Buffer, MVAR, and Handle__—demonstrating efficient memory dispersion and modular behavior. Figure 8 compares 20 individual profiling metrics side-by-side, based on data gathered using +RTS -p -hy for both configurations. Key metrics like ARR_WORDS (36,816 bytes nested vs. 61,360 bytes non-nested) and MVAR (32 bytes nested vs. 64 bytes non-nested) show that nested STM consistently offers better memory efficiency under identical workloads.

Table 7: STM heap profiling: nested vs. non-nested

Metric	**Description**	**Nested STM (bytes)**	**Non-Nested STM (bytes)**
Total IO	Total memory used by IO operations	240	112
String	Memory used by String data type	24	24
TextEncoding	Memory used by TextEncoding data type	32	32
MVAR	Memory used by MVAR (Multi-Version Concurrency Control Variable) data type	32	64
Handle	Memory used by Handle data type	24	48
Word32	Memory used by Word32 data type	16	16
(#,#)	Memory used by (#,#) data type	40	80
(,)	Memory used by (,) (Tuple) data type	-	48
ForeignPtrContents	Memory used by ForeignPtrContents data type	72	120
BufferCodec	Memory used by BufferCodec data type	48	96
BufferList	Memory used by BufferList data type	-	24
Buffer	Memory used by Buffer data type	168	280
Newline	Memory used by Newline data type	24	24
Maybe	Memory used by Maybe data type	40	64
MUT_VAR_CLEAN	Memory used by MUT_VAR_CLEAN data type	128	208
Handle__	Memory used by Handle__ data type	136	272
DEAD_WEAK	Memory used by DEAD_WEAK data type	-	48
WEAK	Memory used by WEAK data type	96	96
ARR_WORDS	Memory used by ARR_WORDS data type	36,816	61,360
Total	Total memory usage	38,792	63,080

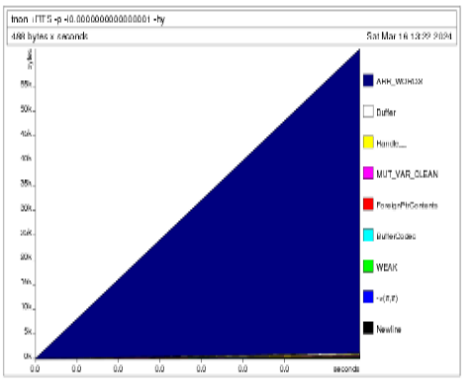


Figure 6: Heap Profile – Non-Nested STM (tnon.hs) Generated using +RTS -p -hy on Intel Core i5-1035G1, 8 GB RAM, Windows 11 Pro, with GHC 8.6.5.

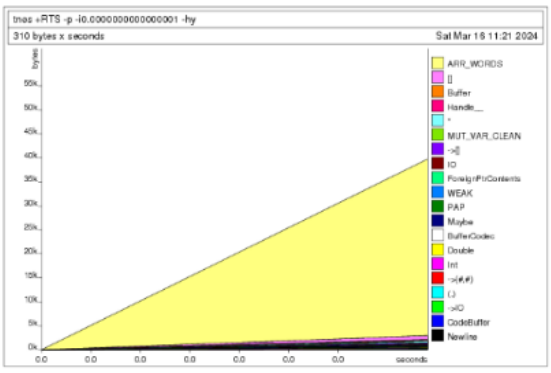


Figure 7: Heap Profile – Nested STM (tnes.hs) Generated using +RTS -p -hy on Intel Core i5-1035G1, GB RAM, Windows 11 Pro, with GHC 8.6.5

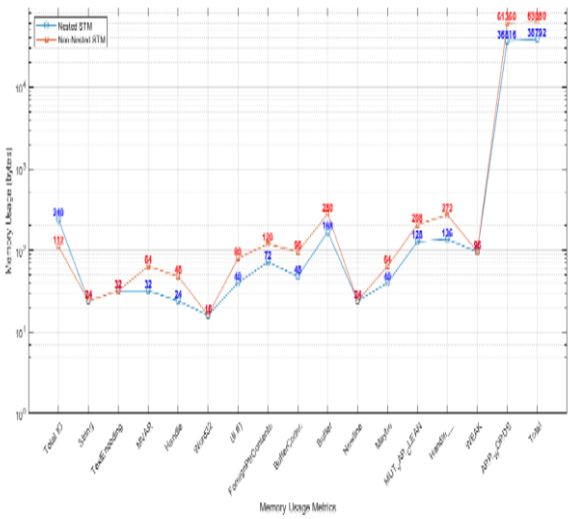


Figure 8: Comparative Memory Usage Across 20 STM Metrics. Captured uniformly using GHC profiling tools under controlled conditions.

In conclusion, heap profiling reveals that nested STM exhibits superior memory efficiency across all 20 metrics, particularly in managing variables, handles, and buffers. This significant reduction in memory usage demonstrates that nested STM is better suited for environments with high memory demands. The reduced memory footprint of nested STM suggests it is a more scalable and efficient solution for managing memory in transactional systems, especially in scenarios where memory resources are constrained. Conversely, non-nested STM's higher memory consumption could lead to performance challenges, underscoring the advantages of nested STM in optimizing memory usage.

The present study adopts a controlled, proof-of-concept evaluation focused on isolating the behavioural impact of nesting within STM. Each configuration was executed for ten uniform iterations under identical runtime conditions to minimize variability. Comprehensive statistical validation, including repetitions with confidence intervals and standard deviations, is identified as future work to extend the current phase toward broader significance testing. The experimental framework supports scalability across varying threads and transaction counts (e.g., 2 → 32 threads), enabling evaluation of how nested STM performance evolves as concurrency and contention levels rise. These planned analyses will provide quantitative scalability trends beyond the single-system scope of this study. Additionally, abort, commit, and retry ratios—though initially defined as part of the evaluation metrics—could not be directly measured due to limitations of the GHC STM profiling toolchain, which does not expose per-transaction counters. As a practical substitute, contention was inferred indirectly using throughput variation and execution-time differentials between nested and non-nested configurations, serving as a reliable contention proxy for comparative analysis.

The apparent discrepancy between Total Alloc and Heap Footprint arises from the semantics of these two metrics. Total Alloc represents the cumulative volume of memory allocated throughout execution, including short-lived transient objects, whereas Heap Footprint reflects the live data retained at any instant. The nested STM allocates more temporary structures within subtransactions, increasing Total Alloc, but its hierarchical cleanup quickly reclaims them, resulting in a smaller live heap—approximately 38 % lower than in the non-nested configuration. Timing variations also stem from profiler precision: the GHC runtime profiler (+RTS -p) rounds totals to the nearest hundredth of a second, often showing 0.00 s, while wall-clock measurements obtained via getCPUTime capture finer resolution, yielding 0.08 s and 0.10 s per operation for nested and non-nested STM respectively. Both measures are consistent once granularity differences are considered.

Throughput (≈25 % improvement) was computed using the relation

$$\text{Throughput} = \frac{\text{Total Operations}}{\text{Total Execution Time}}$$

where each configuration executed ten uniform iterations (≈50 operations per iteration for nested STM and ≈20 for non-nested STM) under identical runtime settings. This provides a clear and reproducible basis for the performance comparison.

The observed variation between Total Alloc and Heap Footprint can be explained through garbage-collection dynamics in the Haskell STM runtime. Nested STM creates more short-lived transactional variables (TVars) and intermediate objects during subtransaction rollbacks and commits, leading to a higher cumulative allocation. However, most of these allocations are ephemeral and quickly reclaimed by the generational garbage collector, resulting in a substantially smaller live heap footprint. The reduction of long-lived residency, combined with efficient promotion control, accounts for the ~38 % lower heap usage observed in the nested configuration despite greater total allocation volume. While explicit GC metrics such as pause durations, collection cycles, and survivor promotions were not captured in this phase, the allocation and heap trends collectively indicate that nested STM improves memory locality and short-lived object turnover, contributing to reduced live heap occupancy. Future work will incorporate direct GC profiling to quantify these effects more precisely.

In conclusion, the implementation and profiling results reveal that while nested STM configurations demand more complex structure and slightly higher overhead in certain components, they offer significant advantages in terms of concurrency distribution and memory efficiency. The results demonstrate that nested TVars improve concurrency by localizing conflicts within sub-transactions, achieving an average operational throughput approximately 25% higher than the non-nested version (0.08 seconds per operation for nested vs. 0.10 seconds for non-nested) and consuming about 38% less total heap memory (38,792 bytes vs. 63,080 bytes).Heap profiling confirmed more modular memory distribution, reducing pressure on the main function and improving scalability under concurrent workloads. These findings affirm that nested STM is well-suited for memory-intensive and highly concurrent environments. Future work will incorporate statistical validation, scalability testing in terms of concurrency distribution and memory efficiency.

6 Discussion

This section discusses empirical findings on nested vs. non-nested STM, focusing on concurrency, complexity, memory usage, and profiling insights.

This aligns with prior theoretical work on nested transactions [11] , which suggested better modularity and isolation. However, our study goes beyond theoretical insights by providing direct, reproducible measurements under practical workloads. Unlike existing STM configurations, which mainly focus on optimizing isolation and update policies without quantifying nesting trade-offs, our findings clarify the concrete cost-benefit balance between concurrency gains and the added rollback scope in nested transactions. While our results confirm improved throughput and lower memory usage for nested TVars in our test cases, the computational complexity analysis reveals that this performance improvement comes with increased complexity. Nested STM operations, including conflict resolution, rollback, and commit, exhibit higher computational costs (e.g., O (n * m) for conflict resolution), as opposed to the simpler O(1) operations in non-nested systems. Here, n refers to the total number of transactions (both parent and sub-transactions), and m is the depth of nesting (i.e., the number of nested levels). This higher complexity, however, does not diminish the benefits of better concurrency and isolation, making nested TVars an ideal choice for systems that require these characteristics, despite the added computational overhead. It is important to note that the current study is limited to single-machine Haskell STM implementations. Scalability in distributed STM systems or in environments with hardware-assisted transactional memory (TM) remains untested. Future work should explore how the findings hold up in these more complex environments. Additionally, testing with varying transaction sizes and real-world workloads would further help generalize these results. Overall, this study fills an important experimental gap in understanding the trade-offs between nested and non-nested TVar configurations and provides actionable insights for developers. By incorporating both empirical findings and theoretical complexity analysis, we offer a more comprehensive framework for choosing the appropriate TVar model based on system requirements, particularly when deciding between the benefits of higher concurrency and the costs of increased computational complexity.

While this study provides meaningful profiling results, additional performance dimensions warrant future exploration. Notably, transaction contention analysis was not conducted. Understanding how nested and non-nested TVars behave under increasing conflict levels is critical for real-world applicability. Future experiments will simulate varying degrees of contention to analyze the impact on throughput, retries, and rollback frequency.

The present evaluation captures contention effects indirectly through observed throughput variations and execution-time differentials between nested and non-nested STM configurations executed under identical workloads. These differences effectively reflect the contention management behavior of each model without explicit tuning of collision parameters. Empirical contention experiments—such as varying transaction overlap, conflict probability, or critical-section length—are reserved for future work to systematically examine abort and retry dynamics under controlled high-conflict scenarios. Such experiments will extend the current findings by quantifying contention sensitivity and confirming the robustness of nested STM under diverse concurrency levels.

The results demonstrate that nested TVars improve concurrency by localizing conflicts within sub-transactions, achieving an average operational throughput approximately 25% higher than the non-nested version (0.08 seconds per operation for nested vs. 0.10 seconds for non-nested) and consuming about 38% less total heap memory (38,792 bytes vs. 63,080 bytes).

Furthermore, although heap memory usage was profiled in detail, the specific effects of garbage collection were not isolated. A more granular analysis of allocation and reclamation behaviour will be conducted to evaluate memory overhead and garbage collection impact. While the current study was limited to a fixed transaction workload, the experimental framework already supports thread-scaling and contention evaluation ($2 \rightarrow 32$ threads), which will be leveraged to analyze how nested TVars influence STM performance under varying concurrency levels and workload intensities.

In summary, this discussion highlights key performance trade-offs and outlines future work focused on enhancing statistical validation, scalability, contention management, garbage-collection profiling, and cross-library benchmarking to further strengthen and optimize STM systems.

7 Future directions and challenges

This section outlines the key areas for future research and development in nested transactions within Software Transactional Memory (STM). Despite their demonstrated advantages, several challenges remain that need to be addressed to advance the field. These include refining conflict detection mechanisms, minimizing memory overhead, incorporating statistical validation, and optimizing for scenarios with single-level parallelism. Further exploration into the practical applicability of nested transactions, their adaptability across different environments, and their integration with hardware support is also essential. Additionally, fostering cross-disciplinary collaboration and investigating the scalability of nested transactions in distributed systems will be crucial for enhancing their practicality and effectiveness.

7.1 Refining conflict detection

Future work should explore hybrid conflict detection strategies, integrating eager and lazy evaluation with contention-aware policies such as Polka, to enhance accuracy and minimize unnecessary retries in nested STM systems. Additional experiments that tune contention levels and monitor abort/retry rates will help characterize performance under high-conflict workloads.

7.2 Minimizing memory overhead

Addressing memory overhead is pivotal. Researchers can focus on developing advanced memory-management strategies tailored to nested transactions, improving garbage-collection efficiency and short-lived object reclamation. Future profiling will include GC pause times, survivor promotion, and residency metrics to explain allocation–residency relationships.

7.3 Optimizing single-level parallelism

Optimizing STM for workloads with limited parallelism remains important. Future research should fine-tune scheduling and commit strategies to balance concurrency overhead with deterministic behaviour.

7.4 Practical applicability of nested transactions

Further exploration into real-world scenarios, including reactive and real-time systems, will reveal where nested STM offers the most significant benefits. Integrating empirical contention tests will quantify transactional robustness under varying load conditions.

7.5 Adaptability in different environments

Understanding how nested transactions perform across different programming and runtime environments is crucial. Comparative analysis using alternative STM libraries such as Clojure STM and Java Multiverse will strengthen cross-platform generalizability.

7.6 Hardware support integration

Investigating how nested STM interacts with hardware-assisted transactional memory (HTM) can uncover hybrid approaches that combine software flexibility with hardware acceleration. Cross-Disciplinary Collaboration among STM researchers, language designers, and hardware architects can foster integrated concurrency solutions that align software design with architectural capabilities.

7.7 Scalability in distributed systems

The experimental framework already supports thread-scaling and contention evaluation ($2 \rightarrow 32$ threads), which will be further utilized to analyze nested STM performance under varying concurrency and conflict levels. This will examine how nested STM behaves under distributed coordination, network latency, and high-conflict workloads. Statistical extensions with confidence intervals and standard deviations will strengthen performance validity, while abort/commit and retry ratios—currently limited by the GHC STM profiler—will be measured using enhanced instrumentation. Although not covered experimentally in this study, this direction represents a critical next step toward making nested STM practical and robust in large-scale and cloud environments. Collaborative efforts between academia and industry will be pivotal in achieving this goal.

7.8 Extending metrics and cross-library validation

Future research will include cross-library benchmarking and statistical validation of performance results. Direct measurement of contention-related metrics, comprehensive GC profiling, and scalability testing will

collectively enhance the statistical depth, reproducibility, and generalizability of nested STM evaluation.

In conclusion**,** nested STM shows clear efficiency and memory advantages over non-nested models. Future work will focus on statistical validation, scalability, contention analysis, and cross-library benchmarking to enhance the reliability and applicability of nested STM in real-world concurrent systems.

8 Conclusion

This study presented a detailed evaluation of Software Transactional Memory (STM) systems using nested and non-nested TVar implementations, with a focus on concurrency, execution time, and memory usage.

The results demonstrate that nested TVars improve concurrency by localizing conflicts within sub-transactions, achieving an average operational throughput approximately 25% higher than the non-nested version (0.08 seconds per operation for nested vs. 0.10 seconds for non-nested) and consuming about 38% less total heap memory (38,792 bytes vs. 63,080 bytes).While non-nested TVars offer simpler implementation and marginally faster individual operations, they perform less efficiently under high contention. Time and heap profiling highlighted how transaction structure influences performance and resource allocation. These findings emphasize that STM design should be tailored to application needs—nested TVars for fine-grained concurrency and modularity, and non-nested TVars for simplicity and low-overhead scenarios. The proposed STM framework forms the base model for later advancements [8] , integrating machine learning classifiers to improve scalability, dynamic prediction, and overall performance analysis of transactional memory systems. Future extensions will build on this framework through enhanced statistical validation, scalability analysis, and integration across multiple STM environments to strengthen its generalizability and practical relevance.

References

[1] Blumofe, R. D., Joerg, C. F., Kuszmaul, B. C., Leiserson, C. E., Randall, K. H., & Zhou, Y. (1996). Cilk: An efficient multithreaded runtime system. *Journal of Parallel and Distributed Computing*, 37(1), 55–69. https://doi.org/10.1006/jpdc.1996.0107.

[2] Bernstein, P. A., & Goodman, N. (1981). Concurrency control in distributed database systems. *ACM Computing Surveys*, 13(2), 185–221. https://doi.org/10.1145/356842.356846.

[3] Turcu, A., Ravindran, B., & Saad, M. M. (2012). On closed nesting in distributed transactional memory. *Proceedings of the Seventh ACM SIGPLAN Workshop on Transactional Computing*. https://transact2012.cse.lehigh.edu/papers/turcu.pdf.

[4] Harris, T., Larus, J. R., & Rajwar, R. (2010). *Transactional memory* (2nd ed.). Synthesis Lectures on Computer Architecture, Morgan & Claypool Publishers, 1–247. https://dl.acm.org/doi/book/10.5555/1855056.

[5] Herlihy, M., & Moss, J. E. B. (1993). Transactional memory: Architectural support for lock-free data structures. *Proceedings of the 20th Annual International Symposium on Computer Architecture (ISCA '93)*, May 1993. https://doi.org/10.1145/165123.165164.

[6] Shavit, N., & Touitou, D. (1995). Software transactional memory. *Proceedings of the 14th Annual ACM Symposium on Principles of Distributed Computing*, Ottawa, Canada. https://doi.org/10.1145/224964.224987.

[7] Damron, P., Fedorova, A., Lev, Y., Luchangco, V., Moir, M., & Nussbaum, D. (2006). Hybrid transactional memory. *Proceedings of the 12th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2006)*, San Jose, CA, USA, October 21–25, 2006. https://doi.org/10.1145/1168857.1168900.

[8] Meenu.(2025). Evaluating Nested and Non-Nested Software Transactional Memory using Machine Learning Classifiers. *Informatica (Slovenia)*, vol. 49(6),313-334. https://doi.org/10.31449/inf.v49i6.8359.

[9] Lopes, A., Castro, D., & Romano, P. (2024). PIM-STM: Software transactional memory for processing-in-memory systems. *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS '24)*, New York, NY, USA. https://doi.org/10.1145/3620665.3640428.

[10] Moss, J. E. B. (1981). *Nested transactions: An approach to reliable distributed computing* (Ph.D. thesis, Technical Report MIT/LCS/TR-260). MIT Laboratory for Computer Science, Cambridge, MA. https://publications.csail.mit.edu/lcs/pubs/pdf/MIT-LCS-TR-260.pdf.

[11] Moss, E. B., & Hosking, A. L. (2006). Nested transactional memory: Model and architecture sketches. *Science of Computer Programming*, 63(2), 186–201. https://doi.org/10.1016/j.scico.2006.05.010.

[12] Ranjan, N., Kapoor, R., & Peri, S. (2024). Short paper: An efficient framework for supporting nested transaction in STMs. *Networked Systems: 12th International Conference, NETYS*, Rabat, Morocco,

May 29–31, 2024. https://doi.org/10.1007/978-3-031-67321-4_13.

[13] Diegues, N. M. L., & Cachopo, J. (2012). *Review of nesting in transactional memory*. Technical Report RT/1/2012, Instituto Superior Técnico/INESC-ID. https://scholar.tecnico.ulisboa.pt/records/14aedc27-91a7-48ec-8b8b-ff750b7934c6.

[14] Abbas, G., & Asif, N. (2010). *Performance tradeoffs in software transactional memory* (Master's thesis, School of Computing, Blekinge Institute of Technology, Sweden). No: MCS-2010-28. https://www.diva-portal.org/smash/get/diva2:833477/FULLTEXT01.pdf.

[15] Classen, S. (2008). *LibSTM: A fast and flexible STM library* (Master's thesis, Laboratory for Software Technology, Swiss Federal Institute of Technology, ETH Zurich). https://library.ethz.ch.

[16] Imbs, D., & Raynal, M. (2008). A lock-based STM protocol that satisfies opacity and progressiveness. *Proceedings of the 12th International Conference on Principles of Distributed Systems (OPODIS'08)*. https://doi.org/10.1007/978-3-540-92221-6_16.

[17] Scherer, W. N., & Scott, M. L. (2004). Contention management in dynamic software transactional memory. *Proceedings of the ACM PODC Workshop on Concurrency and Synchronization in Java Programs*, Canada, July 2004. https://www.cs.rochester.edu/u/scott/papers/2004_CSJP_contention_mgmt.pdf.

[18] Scherer, W. N., & Scott, M. L. (2005). Advanced contention management for dynamic software transactional memory. *Proceedings of the Twenty-Fourth Annual ACM Symposium on Principles of Distributed Computing*, Las Vegas, NV, USA. https://doi.org/10.1145/1073814.1073861.

[19] Guerraoui, R., Herlihy, M., & Pochon, B. (2005). Toward a theory of transactional contention managers. *Proceedings of the Twenty-Fourth Annual ACM Symposium on Principles of Distributed Computing*, Las Vegas, NV, USA. https://doi.org/10.1145/1073814.1073863.

[20] Harris, T., & Stipic, S. (2007). Abstract nested transactions. *Second ACM SIGPLAN Workshop on Transactional Computing*. https://www.cs.rochester.edu/meetings/TRANSACT07/papers/harris.pdf.

[21] Herlihy, M., Luchangco, V., Moir, M., & Scherer, W. N. (2003). Software transactional memory for dynamic-sized data structures. *Proceedings of the Annual ACM Symposium on Principles of Distributed Computing*, 2003. https://doi.org/10.1145/872035.872048.

[22] Marathe, V. J., Spear, M. F., Heriot, C., Acharya, A., Eisenstat, D., Scherer, W. N., & Scott, M. L.,(2006). Lowering the overhead of software transactional memory. *1st ACM SIGPLAN Workshop on Transactional Computing (TRANSACT '06)*, 2006. https://www.cs.rochester.edu/u/scott/papers/2006_TR893_RSTM.pdf.

[23] Saha, B., Adl-Tabatabai, A.-R., Hudson, R. L., Minh, C. C., & Hertzberg, B. (2006). McRT-STM: A high-performance software transactional memory system for a multi-core runtime. *SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '06)*, 2006. https://doi.org/10.1145/1122971.1123001.

[24] Dalessandro, M., Spear, M. F., & Scott, M. L. (2010). NOrec: Streamlining STM by abolishing ownership records. *Proceedings of the 15th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '10)*, 2010. https://doi.org/10.1145/1693453.1693464.

[25] Moore, K. E., Bobba, J., Moravan, J. M., Hill, M. D., & Wood, D. A. (2006). LogTM: Log-based transactional memory. *Proceedings of the 12th High-Performance Computer Architecture International Symposium (HPCA '06)*, 2006. https://doi.org/10.1109/HPCA.2006.1598134.

[26] Moravan, J. M., Bobba, J., Moore, K. E., Yen, L., Hill, M. D., Liblit, B., Swift, M. M., & Wood, D. A. (2006). Supporting nested transactional memory in LogTM. *Proceedings of the 12th International Conference on Architectural Support for Programming Languages and Operating Systems in SIGPLAN Notices (ASPLOS 2006)*. https://doi.org/10.1145/1168857.1168902.

[27] Ghosh, A., & Chaki, R. (2016). Implementing software transactional memory using STM. In *Advanced Computing and Systems for Security* (Vol. 396, pp. 235–248). Springer AISC. https://doi.org/10.1007/978-81-322-2653-6_16.

[28] Le, M. F., Yates, R., & Fluet, M. (2016). Revisiting software transactional memory in Haskell. *ACM SIGPLAN Notices*, 51(12), 105–113. https://doi.org/10.1145/3241625.2976020.

[29] Du Bois, A. R. (2011). An implementation of composable memory transactions in Haskell. In *Software Composition, SC 2011, Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-22045-6_3.

[30] Discolo, S. S., Harris, T., Marlow, S, Jones, S. P., & Singh, S. (2006). Lock-free data structures using STM

in Haskell. In *Functional and Logic Programming (FLOPS 2006)*. https://doi.org/10.1007/11737414_6.

[31] Herlihy, M., Luchangco, V., & Moir, M. (2006). A flexible framework for implementing software transactional memory. *ACM SIGPLAN Notices*, 41(10), 253–262. https://doi.org/10.1145/1167515.1167495.

[32] Harris, T., Marlow, S., Peyton-Jones, S., & Herlihy, M. (2005). Composable memory transactions. *Proceedings of the 10th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '05)*, Chicago, IL, USA. https://doi.org/10.1145/1065944.1065952.

[33] Peyton-Jones, S., Gordon, A., & Finne, S. (1996). Concurrent Haskell. *Proceedings of the 23rd ACM Symposium on Principles of Programming Languages (POPL '96)*. https://doi.org/10.1145/237721.237794.

[34] Turcu, A., & Ravindran, B. (2012). On open nesting in distributed transactional memory. *Proceedings of the 5th Annual International Systems and Storage Conference (SYSTOR '12)*. https://doi.org/10.1145/2367589.2367601.

[35] Ni, Y., Menon, V. S., Adl-Tabatabai, A.-R., Hosking, A. L., Hudson, R. L., Moss, J. E. B., Saha, B., & Shpeisman, T. (2007). Open nesting in software transactional memory. *Proceedings of the 12th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '07)*, ACM Press, New York, NY, USA. https://doi.org/10.1145/1229428.1229442.

[36] Carlstrom, B. D., McDonald, A., Chafi, H., Chung, J., Minh, C. C., Kozyrakis, C., & Olukotun, K. (2006). The ATOMOS transactional programming language. *Proceedings of the ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '06)*. https://doi.org/10.1145/1133981.1133983.

[37] Baek, W., Bronson, N., Kozyrakis, C., & Olukotun, K. (2010). Implementing and evaluating nested parallel transactions in software transactional memory. *Proceedings of the 22nd ACM Symposium on Parallelism in Algorithms and Architectures (SPAA '10)*, Thira, Santorini, Greece. https://doi.org/10.1145/1810479.1810528.

[38] Kumar, R., & Vidyasankar, K. (2011). HParSTM: A hierarchy-based STM protocol for supporting nested parallelism. *6th ACM SIGPLAN Workshop on Transactional Computing (TRANSACT '11)*. https://sss.cs.purdue.edu/projects/transact11/papers/Kumar.pdf.

[39] Volos, H., Welc, A., Adl-Tabatabai, A.-R., Shpeisman, T., Tian, X., & Narayanaswamy, R. (2009). NePaLTM: Design and implementation of nested parallelism for transactional memory systems. *Proceedings of the 23rd European Conference on Object-Oriented Programming (ECOOP '09)*. https://doi.org/10.1007/978-3-642-03013-0_7.

[40] Agrawal, K., Fineman, J. T., & Sukha, J. (2008). Nested parallelism in transactional memory. *Proceedings of the 13th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP '08)*. https://doi.org/10.1145/1345206.1345232.

[41] Barreto, J., Dragojević, A., Ferreira, P., Guerraoui, R., & Kapalka, M. (2010). Leveraging parallel nesting in transactional memory. *ACM SIGPLAN Notices*. pp. 91–100. https://doi.org/10.1145/1837853.1693466.

[42] Ramadan, H., & Witchel, E. (2009). The xfork in the road to coordinated sibling transactions. *4th ACM SIGPLAN Workshop on Transactional Computing (TRANSACT '09)*.

Federated EGSV-AACO for Decentralized Spectrum Sensing and Sharing in IoT Networks

Jing Ling
School of Integrated Circuits and Communications, SuZhou Institute of Industrial Technology, Suzhou, Jiangsu, 215104, China
E-mail: lingjing68@hotmail.com

Keywords: extreme gradient support vector with adaptive ant colony optimization (EGSV-AACO), Internet of Things (IoT), privacy problems, smart campus network, spectrum sharing

Received: September 9, 2025

Wireless bandwidth is in greater demand than ever before due to the Internet of Things' (IoT) applications' rapid expansion in fields including smart cities, autonomous and Industry 4.0. Traditional fixed spectrum allocation approaches can lead to inefficient utilization and excessive interference levels, particularly in densely populated areas. The purpose of this evaluation is to create an intelligent, decentralized, and privacy-preserving framework for optimizing spectrum detection and sharing among IoT devices utilizing machine learning (ML) techniques. The Cognitive Radio Networks (CRNs) Dataset is gathered from the Kaggle source. The procedure consists of four sequential steps. Each IoT node uses Extreme Gradient Support Vector with Adaptive Ant Colony Optimization (EGSV-AACO) to monitor spectrum occupancy and identify idle bands. Each node builds a local spectrum access model based on temporal spectrum patterns. Model weights are delivered to a nearby edge server on a regular basis to avoid exposing raw data using Federated Averaging (FedAvg). The server aggregates the locally trained models to form a global model and redistributes it to all participating devices. This updated global model will drive real-time, collision-free spectrum allocation among IoT devices. A smart campus simulation using MATLAB shows that the proposed EGSV-AACO framework ensures access convergence, improves spectrum usage, and prevents raw data leakage. The developed model outperforms all baseline methods and achieved an accuracy of 97%, precision of 97.5%, recall of 96%, and an F1-score of 96.5%. Overall, this research introduces a novel Federated EGSV-AACO framework that significantly enhances decentralized, privacy-preserving, and intelligent spectrum sensing and sharing in IoT networks.

Povzetek: Raziskava predstavlja federativni okvir EGSV-AACO, ki z decentraliziranim strojnim učenjem omogoča natančno, zasebnostno ohranjajoče in učinkovito zaznavanje ter deljenje radijskega spektra v IoT omrežjih.

1 Introduction

Internet of Things (IoT) explosive growth has revolutionized contemporary society by establishing a networked ecosystem in which billions of individual items can easily communicate with other objects [1]. IoT applications, such as smart cities, intelligent transportation systems, industrial automation, and are constantly altering the methods for collecting, communicating, and using data [2]. Effective and reliable communication frameworks that can support a variety of services in real time are necessary due to the increasing interconnectedness [3]. The role of wireless spectrum in this research is critical as it is the essential material of communication among IoT devices [4].

As the number of IoT devices has increased exponentially and rapidly, spectrum resources have become as much in demand as never before. In comparison to the traditional communication networks that rely on pre-determined frequency allocation, IoT ecosystems are very dynamic and comprise of a wide range of devices that have different communication needs [5,6]. The existing wireless infrastructure is currently experiencing a massive amount of pressure, particularly in densely populated regions such as campuses, cities and industrial hubs due to its dynamic nature. Spectrum management is essential in enabling scalability and sustainability of IoT-driven services besides provision of continuous connection [7, 8].

Spectrum utilization, in this case, denotes a fair and dynamic usage of the existing resources and the alternation of frequency bands [9]. While overloaded frequencies result in interference, delays, and a lower quality of service, idle or underused spectrum indicates missed communication possibilities [10]. Thus, making the most use of the spectrum that is available is essential to the development of IoT. Furthermore, since devices continually create and process data across several domains, the scattered and data-intensive nature of IoT networks adds another level of complexity [11]. The real difficulty is how to manage the few spectrum resources for the growth of IoT devices. Fixed allocations tend to be inefficient and disruptive, particularly in an environment with high assignments density. To achieve this kind of fair real-time access, and at the same time, security and privacy, innovative approaches need to be used that are scalable, adaptable, and reliable without affecting the performance of communication and the protection of data.

Therefore, the present research focuses on creating a decentralized and intelligent spectrum management system designed for real-time, secure, and effective resource allocation in order to overcome these constraints and satisfy the growing communication demands of IoT contexts.

This research investigates three key research questions (RQ):

RQ1: *Can EGSV-AACO accurately detect idle spectrum bands under dynamic IoT traffic conditions?*

RQ2: *Does FedAvg improve detection reliability compared to centralized training?*

RQ3: *Can the proposed hybrid model improve accuracy and reduce interference compared to other existing approaches?*

The proposed Extreme Gradient Support Vector with Adaptive Ant Colony Optimization (EGSV-AACO) model accurately detected idle spectrum bands in dynamic IoT environments and evaluated whether Federated Averaging (FedAvg)-based decentralized learning improved reliability over centralized methods. The hybrid model also assessed overall performance gains compared to other existing approaches. In general, the research provides a smart, decentralized, privacy-preserving system for efficient IoT spectrum sharing. The significant contribution of the research is as follows:

i. **Dataset Integration:** A comprehensive CRN spectrum-sensing dataset is used, incorporating interference levels and temporal usage patterns.
ii. **Data Preprocessing:** Min–Max normalization ensures stable training and reduces feature-scale imbalance.
iii. **Privacy-Preserving Learning:** FedAvg enables decentralized learning without sharing raw data, enhancing privacy and security.
iv. **Advanced Spectrum Sensing:** EGSV-AACO improves detection accuracy, reliability, and dynamic spectrum allocation, outperforming traditional methods.

2 Literature review

Recent research on dynamic spectrum access and sharing has primarily been aimed at increasing the effectiveness, security, and privacy of wireless networks and the IoT. However, prior methods suffered from scalability, complexity, and resilience limitations, despite investigations into strategies such as blockchain, federated learning, and reinforcement learning (Table 1).

Table 1: Evaluation of related works on spectrum access approaches

Citation	Objective	Method	Outcome	Limitations
Wang et al. [12]	Reduce Vehicular Ad Hoc Nestwork (VANET) channel conflict using optimized spectrum access	Deep Reinforcement Learning (DRL) + Depth-First Search (DFS)	Fewer collisions probability 60% and reduce packet loss 0.8 to 0.1	Limited scalability in dense vehicular networks
Grissa et al. [13]	Ensure anonymous and secure Citizens Broadband Radio Service (CBRS) spectrum access privacy	Blockchain + Crypto	Strong privacy with secure operations	High computation and increased system complexity
Chang et al. [14]	Improve Dynamic Spectrum Access (DSA) using collaborative multi-agent learning framework	Federated Multi-Agent Deep Reinforcement Learning (Fed-MADRL)	Reduced overhead and preserved privacy	Slow convergence under complex environments
Kaur et al. [15]	Enhance DSA reliability under noisy feedback channels	Gated Recurrent Unit (GRU)- Deep Reinforcement Learning (DRL)	Improved resilience and accuracy 91% results	Difficult cooperative decisions with noisy feedback
Zhang et al. [16]	Achieve distributed DSA using local-only observations effectively	Multi-agent Deep Q-Network (DQN)	High performance 90% over random baselines	Needs offline global training information
Amrallah et al. [17]	Optimize Unmanned Aerial Vehicle (UAV) monitoring with efficient power selection	Multi-Armed Bandit (MAB) algorithms	Better data-rate 30 UAV and	No UAV cooperation

			interference handling	limits decisions
Makhdomi et al. [18]	Enable secure Wireless Service Provider (WSP) spectrum sharing with incentives	Blockchain (DSA)	Improved security and sensing performance	Higher latency and computational overhead
Chen et al. [19]	Maximize Vehicle-to-Infrastructure (V2I)/ Vehicle-to-Vehicle (V2V) capacity in Reconfigurable Intelligent Surface (RIS)-aided networks	Block Coordinate Descent (BCD)-based	Enhanced Quality of Service (QoS) in dynamic channels probability in 0.9	Depends on near-optimal estimated solutions
Gu et al. [20]	Improve Low Earth Orbit (LEO) satellite spectrum sharing efficiency	LEO collaboration	Higher efficiency 89% and convergence 87% achieved	Increased coordination complexity in networks

The experiment [21] is designed to investigate the possibilities and challenges of the future of the IoT with the integration of 5G/6G, zero-trust security, and blockchain to achieve the universal connection and secure access. The approach suggested a blockchain-based authentication system (BasIoT) and a zero-trust security architecture for IoT devices. Although the framework identifies fascinating research opportunities to safe, ubiquitous IoT, it has weaknesses in scalability and complexity of implementation.

Enhancing safe spectrum sharing and power allocation in V2V communications against eavesdropping was the objective of the research [22]. Using cooperative decision-making and multi-agent reinforcement learning, the technique optimizes power, frequency, and the deployment of friendly jammers. Increased system complexity and dependence on cooperating agents were drawbacks; however, simulations have shown that the approach was more secure and resilient than current approaches.

Enhancing safe spectrum sharing in intelligent IoT networks against jamming and emulation assaults is the objective of the research [23]. The approach examined current defenses and suggested online machine learning (ML)-based attack and defensive tactics, such as channel access restrictions that take sensing delays into account and time-varying feedback graph models. Vulnerability to adaptable attackers was a drawback.

Improved Convolutional Neural Networks (CNNs) and Kernel Principal Component Analysis (KPCA) were being used in the research [24] to improve spectrum detection in Internet of Vehicle (IoV) communication. KPCA extracted features by mapping signals to a high-dimensional space, whereas CNN classifies with less complexity. The results indicated a 10% increase in accuracy compared to LeNet5, Elman, and SVM. Limitation: The approach function is largely determined by the quality and conditions of the offline training phase.

Evaluation of user performance in huge cognitive radio networks (CRNs) was used a hybrid spectrum access model that combines overlay and underlay schemes was the goal of the examination [25]. Simulations were used to support the analysis of the age of information (AoI), throughput, and success probability using stochastic geometry. The results demonstrate that hybrid access performs noticeably better than separate approaches. Modeling assumptions has the potential to oversimplify real-world processes, which was a restriction.

2.1 Research gaps

Prior studies, including works such as [12], [14], [17], and [23], exhibit key limitations, including high computational cost, restricted scalability in dense IoT networks, dependence on centralized learning, and privacy risks due to raw data exchange. Methods also struggle with noisy feedback [15], slow convergence [14], blockchain overhead [13], and limited cooperative decision-making efficiency. Optimization techniques remain incomplete, and adaptability to dynamic spectrum patterns is insufficient. The developed EGSV-AACO with FedAvg addresses these gaps by using decentralized learning, aggregation protection privacy, robust hybrid optimization, and much better sensing accuracy and reliability.

2.2 Problem statement

Past research has considered privacy-preserving approaches to spectrum sharing, significant obstacles still need to be addressed. For example, the primary disadvantage of the research was the fact that whereas TSAS [13] raised privacy and security, it could have had the problem of computation load, latency, and scaling applications in larger settings. A key limitation was that despite the increased accuracy of KPCA-CNN [24], such an approach cannot be adapted well to the highly dynamic nature of the IoV environment, as it involves offline training. Online retraining has a high computation overhead, and the model might not learn well when

exposed to previously unseen noise patterns, spectrum heterogeneity, or a large-scale deployment that needs faster online adaptation.

The suggested EGSV-AACO with the FedAvg framework mitigates the shortcomings of the existing research. It features lightweight, secure, and scalable IoT devices with the ability to endure such a large ecosystem, ensuring that computation is reduced and that the model can be aggregated, thus preventing the trickle of raw data in the presence of decentralized federated learning and adaptive spectrum sensing.

3 Methodology

The implementation of the methodology combines a hybrid intelligence model of spectrum sensing and decentralized optimization. First, a dataset of Cognitive Radio Network (CRN) spectrum sensing data was obtained with signal, environmental, and network characteristics. Stable training involved Min-Max normalization of the values. Each IoT device used an EGSV-AACO to identify idle spectrum bands and to adjust parameters adaptively. Local models were then consolidated via FedAvg, but the privacy was guaranteed by sharing ordinary weights, rather than raw data. This method has made it possible to have precise, safe, and scalable spectrum distribution within IoT networks. Figure 1 illustrates the workflow of the proposed research.

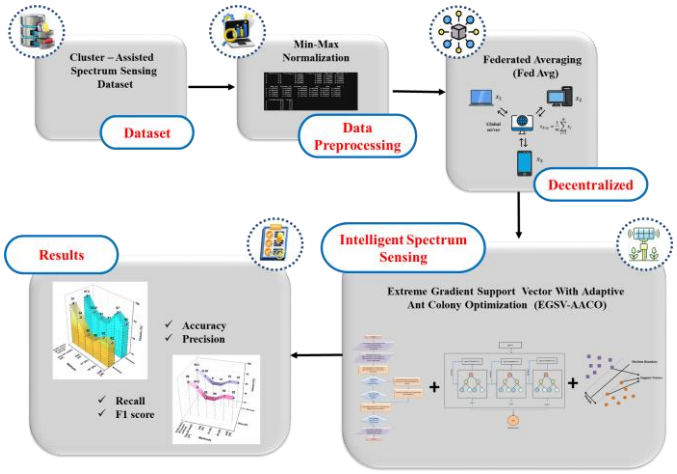


Figure 1: EGSV-AACO for intelligent spectrum sensing in IoT environments

3.1 Dataset

The dataset Cognitive Radio Networks (CRNs) is gathered from the publiclay available Kaggle source (https://www.kaggle.com/datasets/ziya07/cluster-assisted-spectrum-sensing-dataset), this CRNs dataset is optimized towards research in the area of Cluster-Assisted Spectrum Sensing Dataset, specifically on cluster-assisted spectrum sensing. It contains 1000 data samples, It offers a holistic ensemble of features that characterizes properties of signals, dynamic actions of the environmental context, and the network operations to enable effective spectrum sensing and decision-making. The dataset supports ML–based tasks such as primary user detection, spectrum availability prediction, and interference minimization. For 30% testing ratio and 70% of training the dataset was spitted. Table 2 summarizes the central categories of features and target variable available in the CRNs spectrum sensing dataset.

Table 2: Core attributes of the CRN's spectrum sensing dataset

Category	Attributes
Signal Characteristics	Power Spectral Density (PSD), Signal-to-Noise Ratio (SNR), and Received Signal Strength Indicator (RSSI)
Environmental Conditions	Noise level, Interference levels, Channel quality
Network Metrics	Cluster ID, Sensing duration, Transmission success rate
Temporal Features	Time of spectrum sensing, Periodicity of spectrum availability
Target Variable	Binary: 0 = Spectrum Unavailable, 1 = Spectrum Available; or Multiclass based on interference levels

3.2 Min-max normalization

This normalization is used to ensure consistent scaling for features such as RSSI, SNR, and PSD. This is performed by assigning each value to a defined range, normally $[0,1]$, while preserving the relationships between the elements. The normalized value x' is described as follows in equation (1).

$$x' = \frac{x - \min_B}{\mathrm{Max}_B - \mathrm{Min}_B} (\mathrm{new_Max}_B - \mathrm{new_Min}_B) \quad (1)$$

Where $\mathrm{new_Max}_B$ and $\mathrm{new_Min}_B$ provide the desired range, Max_B and Min_B are the feature's maximum and minimum values, and x is the original feature value. This enhances the forecast of spectrum availability and stabilizes model training.

3.3 Federated averaging (fedavg)

In the proposed framework, the spectrum sensing date uploaded by each IoT device (local client) would be used

to train the model and the weights would be updated and sent to the global server. These updates are securely aggregated by the server in a FedAvg way by obtaining the global model (x_{Avg}) as a weighted average in equation (2).

$$x_{Avg} = \frac{1}{m}\sum_{j=1}^{M} x_j \quad (2)$$

M defines total number of IoT devices (local clients), x_j local model parameters (weights), and m defines the number of data samples. This enhances spectrum allocation and detection across IoT networks while guaranteeing privacy-preserving, decentralized optimization. Figure 2 illustrates the FedAvg framework.



Figure 2: FedAvg Framework for IoT devices

The technique in the suggested framework is to actively optimize spectrum allocation across several IoT devices without disclosing their private raw data. Every IoT node l maintains track of a local dataset. D_l is the local dataset stored on IoT device l.

$$D_l \triangleq \left\{\left(w_{l,n}, z_{l,n}\right)\right\}_{n=1}^{m_l} \quad (3)$$

m_l is the number of training samples available on the device. Where equation (3) $z_{l,n}$ stands for spectrum allocation choices and $w_{l,n}$ for device characteristics (such as channel condition and traffic demand). n is the sample index. The goal of global optimization is equation (4).

$$\min_x\{E(x) \triangleq \sum_{l=1}^{M} o_l E_l(x)\} \quad (4)$$

The spectrum allocation error is represented by the local loss function at device l, $E_l(x)$. $\min_x$ is the optimization objective: find the model parameters x (allocation variables vector) that minimize the global loss function. The global loss function for every device was denoted by $\sum_{l=1}^{M} o_l$. E(x) is the global loss function, which measures the total spectrum allocation error across all IoT devices. The Allocation variables vector (model parameters for spectrum optimization) is represented by $x \in \mathbb{R}^N$. The Loss function for the sample $(w_{l,n}, z_{l,n})$ is defined by $\ell(x; w_{l,n}, z_{l,n})$. The traditional distributive stochastic gradient descent (SGD) technique serves as the foundation for FedAvg. SGD was used to modify local models, as determined in equation (5).

$$x_{s+1}^l \leftarrow x_s^l - \eta_s \Delta E_l(x_s^l; \xi_s^l, a) \quad (5)$$

The iteration number (communication round) was denoted by s. x_{s+1}^l denotes the updated local model parameters of device l after the $s+1$ -th training iteration (communication round). At iteration s, x_s^l defines the device l local model parameters. The learning rate (step size) at iteration s was denoted by η_s. The Mini-batch size for local updates was a . $\Delta E_l(x_s^l; \xi_s^l, a)$ was the Mini-batch gradient of device l, and ξ_s^l was the Mini-batch generated from the local dataset D_l at iteration s. And combined on the server by the following equation (6).

$$\bar{x}_{s+1} = \sum_{l=1}^{M} o_l x_{s+1}^l \quad (6)$$

Following the server-side averaging of local models, $\bar{x}_{s+1}$ was the aggregated global model. FedAvg is employed to aggregate on a global level with other local updates of IoT models without exchanging raw spectrum sensing data.

3.4 Extreme gradient support vector with adaptive ant colony optimization (EGSV-AACO)

The proposed research will utilize the EGSV-AACO algorithm as the core intelligence-based spectrum sensing mechanism at the IoT node level. EGSV is trained to categorize the spectrum bands as idle or occupied by learning the temporal signature of each band, whereas AACO adapts the model parameters to maximize accuracy in the uneven environment. This hybrid integration enables each IoT device to efficiently locate spectrum resources, adjust to time-varying spectrum, and reduce false alarms that can cause collisions. The resulting locally optimized spectrum access models produced by EGSV-AACO are then aggregated via Federated Averaging (FedAvg) to assemble a global model, which is decentralized, privacy-preserving, and efficient for spectrum sharing across IoT networks. Algorithm 1 describes the process of the EGSV-AACO method.

Algorithm 1: $EGSV - AACO\ Procedure$

$\%\ Goal: Hybrid\ XGBoost\ +\ SVM\ +\ Adaptive\ ACO\ +\ Federated\ Aggregation$

1. $Initialize\ IoT\ nodes\ (N)\ and\ spectrum\ bands\ (S)$
2. $Initialize\ pheromone\ matrix\ Tau(S,S), AACO\ params\ (alpha, beta, rho, R)$
3. $Initialize\ EGSV\ params\ (numTrees, learningRate, kernelType)$
4. $For\ each\ node\ and\ band\ \rightarrow\ extract\ feature\ vector\ w_j\ and\ label\ z_i$
5. $Train\ XGBoost\ using\ (w_j\ , z_i)\ to\ generate\ boosted\ features$
6. $Train\ SVM\ using\ boosted\ features\ to\ obtain\ EGSV\ classifier$
7. $For\ each\ band\ \rightarrow\ predict\ local\ state\ using\ EGSV\ \rightarrow\ store\ localPrediction$
8. $Initialize\ bestCost\ =\ \infty\ and\ bestHyperParams\ =\ default$
9. $For\ iter\ =\ 1:\ maxIterations\quad \%\ AACO\ tuning\ loop$
10. $\quad For\ each\ ant\ =\ 1:\ numAnts$
11. $\quad\quad Construct\ spectrum\ path\ using\ transition\ probability\ rule:$

$\quad\quad p(i)\ =\ (Tau(i)\text{^}alpha\ *\ Heuristic(i)\text{^}beta)\ /\ \Sigma(all\ feasible)$

12. $\quad\quad Compute\ path\ cost\ =\ Collision\ +\ Interference\ +\ Classification\ Error$
13. $\quad\quad Update\ DeltaTau\ based\ on\ (R\ /\ pathCost)$
14. $\quad\quad If\ pathCost\ <\ bestCost\ \rightarrow\ update\ bestHyperParams\ =\ current\ set$
15. $\quad End$
16. $\quad Evaporate\ pheromone: Tau\ =\ rho\ *\ Tau\ +\ DeltaTau$
17. $\quad Optionally\ adjust\ (alpha, beta)\ for\ dynamic\ environment$

$\quad End$

18. $Update\ EGSV\ model\ hyperparameters\ using\ bestHyperParams$
19. $Collect\ all\ trained\ local\ models\ \rightarrow\ apply\ FedAvg\ \rightarrow\ get\ globalModel$
20. $For\ each\ node\ and\ spectrum\ band:$
21. $\quad finalState(band)\ =\ globalModel.predict(extract_features(node, band))$
22. $Output\ finalState\ as\ optimized\ spectrum\ allocation\ map$

3.4.1 Extreme gradient support vector (EGSV)

The hybrid learning algorithm used incorporates a combination of Extreme Gradient Boost (XGBoost) and SVM, known as the EGSV, to enhance the spectrum sensing accuracy. The XGBoost was used to acquire higher-level spectral features applied in addition to classify the decision boundary more robustly, whereas SVM is employed in a more robust classification of occupied and idle frequency bands. By integrating the gradient boosting feature of XGBoost and the powerful generalization ability of SVM, EGSV can address dynamically changing spectrum environments adequately, and IoT devices can run more predictable and adaptable local spectrum detection.

XGBoost

XGBoost is a robust and scalable technique that has been successfully applied to a wide range of tasks involving regression and classification. It is used in this research as an extension to the EGSV approach to carry out the process of spectrum sensing and occupancy prediction over IoT-enabled networks. The objective is to categorize spectrum bands as idle (available) or occupied (busy) and then optimally allocate resources, whether it is spectrum or wireless resources.

The XGBoost model is an ensemble model that is made up of numerous classification and Regression Trees (CART) regressors. Figure 3 (a) shows the tree structure of XGBoost consisting of a root nodes, internal nodes, leaf nodes, and branches. In spectrum sensing, all IoT devices contribute feature data, which includes the levels of signals, variance in noise, and temporal uses. The CARTs transmit these features, and internal nodes make binary decisions, branch points drive predictions of the state of the spectrum, and leaves predict the final result

of classification. The ensemble prediction of all trees gives the detection outputs of the XGBoost model.

The model is represented by the following equation (7) on the jth data instance (w_j, z_i), such that w_j is the extracted signal features and z_i refers to the actual occupancy label (idle/busy).

$$\hat{z}_i = \alpha \sum_{l=1}^{L} e_l(w_j) \quad (7)$$

Where $\hat{z}_i$ is the estimated spectrum state caused by i, where w_j is the prior input information of the model, the learning rate is α, L is the values of CART used, and $e_l(w_j)$ is the result of the lth regression tree. Equation (7) shows that the overall classification is performed by collecting all regression tree outputs. The objective function K is defined by the following equation (8).

$$K = \sum_{j=1}^{m} k(z_j, \hat{z}_j) + \sum_{l=1}^{L} \Omega(e_l) \quad (8)$$

In this circumstance l, which is the loss function, computes the prediction error between the actual occupancy state z_j and the predicted state $\hat{z}_j$. Objective function (K), Sample index (j), Total samples (m), Loss functions $k(z_j, \hat{z}_j)$, Number of trees (L), and Tree regularization $\Omega(e_l)$.

The regularization penalty Ω(e) penalizes very complicated tree shapes, thereby boosting the generalization capacity of spectrum forecasting models in changeable IoT contexts. In each CART, the regularization equation (9) is formulated as follows:

$$\Omega(e) = \gamma^S + \frac{1}{2}\lambda \sum_{i=1}^{S} \omega_i^2 \quad (9)$$

In this equation, (S) represents the total number of leaf nodes, (ω_i) is the projected value at the (i^{th}) leaf node, $(\gamma^S)$is the complexity penalty based on leaf count. (γ) and (λ) are regularization coefficients to prevent overfitting in quickly changing spectrum data. The XGBoost model is trained in a step-by-step fashion to maximize spectrum classification performance. New CARTS are created in every iteration and combined with previously created CARTS to reduce the error of detection. The second-order Taylor series expansion is used to redefine the objective function, as shown in equations (10-12). These equations incorporate the first- and second-order gradients: Equation (11) indicates the direction of the error, while equation (12) provides confidence in the update,

$$K^{(s)} = \sum_j [k(z_j, \hat{z}_j^{(s-1)}) + h_j e_s(w_j) + \frac{1}{2} g_j e_s^2(w_j)] + \Omega(e_s) + d \quad (10)$$

$$h_j = \frac{\partial k\left(z_j, \hat{z}_j^{(s-1)}\right)}{\partial \hat{z}_j^{(s-1)}} \quad (11)$$

h_j is the first-order gradient, which measures how sensitive the loss function is to small changes in the prediction. g_j is the second-order gradient. $K^{(s)}$ computes the objective function at iteration s. $\Omega(e_s)$is the regularization penalty for the new tree, and $d$is a constant. $e_s(w_j)$ is the Prediction of $s-th$ tree. $e_s^2(w_j)$ is the Squared tree output, j is the sample index, w_j is the output weight $k(z_j, \hat{z}_j^{(s-1)})$ is the loss function, comparing the true label z_j (actual spectrum state: idle/busy) and the predicted value $\hat{z}_j^{(s-1)}$ from the previous iteration. z_j is the true spectrum occupancy label of the jth sample. $\hat{z}_j^{(s-1)}$ is the predicted spectrum occupancy of the jth sample in the $(s-1)$-th iteration. ∂ derivative operator, meaning we're calculating the slope of the loss curve at that point.

$$g_j = \frac{\partial^2 k\left(z_j, \hat{z}_j^{(s-1)}\right)}{\left(\partial \hat{z}_j^{(s-1)}\right)^2} \quad (12)$$

g_j is also called curvature information, which measures how steep or flat the loss curve is around the prediction. ∂ is the derivative operator, Again, $k\left(z_j, \hat{z}_j^{(s-1)}\right)$ is the loss function between actual state z_j and predicted state $\hat{z}_j^{(s-1)}$. ∂^2 is the second derivative, meaning we're looking at the curvature (bending) of the loss function.

These gradients direct the optimization by updating the weak learners to reduce misclassification of idle and occupied channels. $K^{(s)}$ is the objective function at iteration s. h_j defines the loss's first-order gradient (error sensitivity). g_j is the second-order gradient (curvature information), and $e_s(w_j)$ is the prediction of the updated tree at iteration s.

It is in this research that a residual standard error (RSE) is employed as the measure of loss in evaluating predictability. Each example w_j (signal feature vector) is assigned to a leaf node, and the regression function can be expressed in equation (13).

$$e_l(w_j) = \omega_{q(w_j)}, \omega \in \mathbb{R}^S, r(w_j): \mathbb{R}^c \to \{1,2,\dots,S\} \quad (13)$$

$e_l(w_j)$is the output of the lth regression tree for input w_j, whereas $\omega_{q(w_j)}$ is the leaf weight (prediction score) for input w_j. $r(w_j)$ is the leaf assignment function, which maps input w_j to a leaf node index. $\mathbb{R}^S$ is the S real-number components. $\mathbb{R}^c$ refers to the number of input characteristics. S was the number of leaves on the tree. Finally, using $H_i = \sum_{j \in I_i} h_j$ and $G_i = \sum_{j \in I_i} g_j$, the optimum solution of the objective function, it could be defined as equation (14).

$$K_{min} = -\frac{1}{2} \sum_{i=1}^{S} \frac{H_i^2}{G_i + \lambda} + \gamma^S + d \quad (14)$$

K_{min} is he minimum value of the XGBoost objective function after optimization at a given iteration. It represents the optimized score for the regression trees when both loss and regularization are considered. $\sum_{i=1}^{S}$ is Summation over all leaf nodes in the regression tree. S is the total number of leaf nodes in the current regression tree.H_i is the sum of first-order gradients (errors) of all training samples assigned to the leaf node i. G_i is the sum of second-order gradients (curvatures) of all training samples assigned to leaf node i. λ is a regularization coefficient that controls the magnitude of leaf weights.

Therefore, the XGBoost model combines gradient-based optimization and tree-based ensemble learning to give accurate spectrum detection when combined with SVM.

Support vector machine (SVM)

An XGBoost is combined with an SVM to enhance the evaluation of classification on complex datasets. Although XGBoost is effective in feature selection and the undermining of weak learners, SVM is used to form the decision boundary that will correctly classify the variables based on the constructed model. SVM enhances XGBoost's boosting mechanism and is especially effective at handling smaller training sets. feature redundancy and enhances data representation. SVM architecture shown in Figure 3 (b).

It also exhibits great generalization. Data points of distinct classes are separated by the SVM decision boundary, known as the hyperplane. The margin is the separation between the support vectors and the hyperplane, whereas the support vectors are the data points that are closest to this hyperplane. Maximizing this margin will grant superior classification generalization. The mathematical expression for the hyperplane (c(w)) is the following equation (15).

$$c(w) = xw_j + a \quad (15)$$

Where the input data vector is denoted by w_j. The bias term is defined by a, while x was the weight vector orthogonal to the hyperplane. The separation between classes is ascertained by the margin that is found by the equation (16).

$$\text{margin} = \frac{2}{\|x\|} + D\sum_{j=1}^{m} \varsigma_{j'} \quad (16)$$

Where the weight vector's norm is $\|x\|$. The user-specified penalty parameter that regulates misclassification tolerance is defined in D. When data cannot be perfectly linearly separated, ς_j is utilized as a slack variable. The number of training samples is denoted by m. In reality, maximizing the margin can simply be minimizing $\|x\|$, but this constitutes an optimization query. The Lagrange multiplier, which adds the restrictions to the objective function with multipliers, can be used to solve this optimization. The final SVM decision was expressed as an equation (17).

$$e(w) = \sum_{j=1}^{M} \alpha_j z_j L(w_j, w) + a \quad (17)$$

Where the Lagrange multipliers (weights given to training samples) are represented by α_j. The jth sample's class label was z_j. The entire number of training samples is denoted by M. The kernel function that transfers data into a greater-dimensional space is defined by $L(w_j, w)$. To process the non-linear separation of the information, we use the Radial Basis Function (RBF) kernel, which is defined as the following equation (18).

$$L(w_j, w) = \exp\left(-\frac{\left||x_j - x|\right|^2}{2\gamma^2}\right) \quad (18)$$

Where $\|x_j - x\|$ were two feature vectors, the Euclidean distance between the two feature vectors and γ is the kernel coefficient determining the flexibility of the decision boundary. SVM uses the improved features to create a strong classification boundary after the boosting stage first eliminates.

3.4.2 Adaptive ant colony optimization (AACO)

The individual ants in the classic Ant Colony Optimization (ACO) algorithm are initially placed randomly on a node at the start. The move by ants selecting the next node (spectrum channel or allocation point) is based on a probabilistic decision rule that is applied in the process of building an optimal solution. Figure 3 (c) shows the architecture of AACO. The probability of an ant l migrating to the following node i at time s when it is at a spectrum node j is expressed as follows in equation (19).

$$o_{ji}^{l}(s) = \begin{cases} \frac{S_{ji}^{\alpha}(s)\eta_{ji}^{\beta}(s)}{\sum_{k\in allowed_l} S_{ji}^{\alpha}(s)\eta_{ji}^{\beta}(s)}, & \text{if } i \in \text{allowed}_l \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

$o_{ji}^{l}(s)$ is the transition probability for ant l from node j to node i at time s, $S_{ji}^{\alpha}(s)$ ($\equiv \tau_{ji}(s)$: pheromone intensity on edge (j, i) at time s, $\eta_{ji}^{\beta}(s)$is the heuristic value for moving j → i at time s. α pheromone influence exponent, β is heuristic influence exponent, allowed_l is a set of feasible/unvisited next nodes for ant l, j current node; i candidate next node; k: index over allowed_l, l is ant index, s is time/iteration step

O is the Probability of ant, $\tau_{ji}(s)$ depicts the Pheromone intensity on the spectrum path (j, i) at time s. represents previous allocations that were successful. allowed_l is the set of unvisited spectrum nodes available for ant l. The control parameter that determines the pheromone trail's influence is α. The control parameter that specifies the influence of heuristic information is denoted by β. The modified heuristic factor is as follows in equation (20).

$$\eta_i = \frac{1}{\min[dis(j,i)+dis(i,n)]} \quad (20)$$

η_i is the heuristic factor that guides ant l toward node i. dis(j, i) represents the interference-aware cost between the current allocation (j) and the subsequent allocation (i). Where the distance between i and the final destination allocation m is denoted by dis(i, n). Equation (19) can be substituted into equation (20) to generate equation (21).

$$o_{ji}^{l}(s) = \begin{cases} \frac{[S_{ji}(s)^{\alpha}]\cdot\left[\frac{1}{\min(dis(j,i)+dis(i,n))}\right]^{\beta}}{\sum_i [S_{jl}(s)^{\alpha}]\cdot\left[\frac{1}{\min(dis(j,i)+dis(i,n))}\right]^{\beta}} & \text{if } i \in \\ 0, & \end{cases}$$

allowed_l (21)

The pheromone track on a pathway of the spectrum depletes with time. The updating of the trail intensity after time s is performed as follows equations (22-23).

$$\tau_{ji}(s+t) = \rho\tau_{ji}(s) + \Delta\tau_{ji} \quad (22)$$

$$\Delta\tau_{ji} = \sum_{l=1}^{m} \Delta\tau^{l}{}_{ji} \quad (23)$$

The pheromone deposit by ant l on allocation path (j, i) is represented by $\Delta\tau_{ji}^{l}$, whereas ρ is the pheromone evaporating coefficient (0–1). $\tau_{ji}(s+t)$ is the updated pheromone intensity. For pheromone updates, it employs the ant-cycle system, as defined in equation (24).

$$\Delta S_{ji}^{l} = \begin{cases} \frac{R}{K_l}, & \text{if path } (j,i) \text{ belongs to the best allocation tour} \\ 0, & \text{otherwise} \end{cases} \quad (24)$$

Where K_l is the allocation cost of ant K and R is the total pheromone constant.

AACO is ideal for this domain because its pheromone-driven adaptive search dynamically responds to fluctuating spectrum conditions, balances exploration and exploitation, and handles noisy distributed IoT data. Unlike Bayesian or grid search, AACO avoids local minima and efficiently tunes hyperparameters, giving superior performance within the FedAvg decentralized framework.

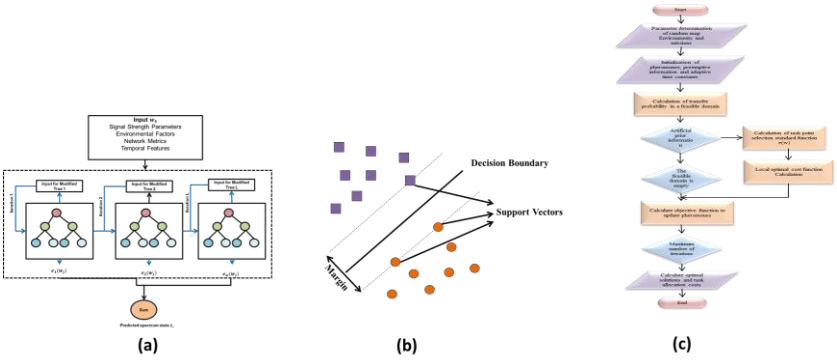


Figure 3: Visualization of (a) XGBoost regression tree model's schematic illustration for spectrum sensing. (b) Support vector machine (SVM) architecture and (c)Flow for Adaptive ACO architecture

The EGSV-AACO combines Extreme Gradient Boosting to refine the features, Support Vector Machine to perform resistant classification, and Adaptive Ant Colony Optimization to optimally tune the hyperparameters. This hybrid scheme expands on accuracy, maximizes the decision boundary, and increases generalization in nonlinear, high-dimensional, and complex datasets. Table 3 demonstrates the Hyperparameters in EGSV-AACO were tuned to optimize spectrum detection accuracy, reliability, and generalization for efficient IoT spectrum sensing.

Table 3: Key hyperparameters of the EGSV-AACO hybrid model

Hyperparameter	Values	Hyperparameter	Values
Learning Rate (α)	0.1	Penalty Parameter (D)	1.0
Number of Trees (L)	100	Pheromone Evaporation (ρ)	0.5
Regularization λ	1.0	Heuristic Influence (β)	2.0
Kernel function	0.001 to 1	Pheromone Influence (α)	1.0

4 Result and discussion

The proposed EGSV-AACO framework for intelligent spectrum sensing and dynamic sharing in IoT networks was implemented on an Intel® Core™ i7-12700K (3.6 GHz) with 32 GB RAM, with GPU support and MATLAB R2022b for simulation and performance evaluation. The findings reveal that it is effective in ensuring effective use of spectrum, speed of access convergence, and secure communication among the IoT devices. The four primary spectrum sensing characteristics are SNR SU1, frequency band, PU presence and PU signal strength. PU appearance is binary (0 or 1) that demonstrates the existence or lack of a primary user. PU presence is binary and with either 0 or 1 that denotes the presence or not of a main user. PU signal strength ranges from around -90 dBm to -40 dBm, indicating clear clustering. The Frequency Band is divided into separate levels around 2000, 3000, 4000, and 5000 MHz. SNR SU1 rises continuously between 5 and 30 decibels, resulting in a dense scatter pattern. Figure 4 illustrates the key spectrum sensing features and their distributions. Figure 4(a) demonstrates substantial clustering and feature separability, enabling the proposed model to achieve accurate spectrum classification. The relation of size of the cluster (3-14) versus the probability of detection (Pd). Pd values are between 0.80 and 0.99, which means that the detection efficiency is always highly effective regardless of the cluster size, as Figure 4 (b) shows. As cluster size grows, Pd values stay tightly distributed between 0.90 and 0.98, indicating high detection reliability. The bulk of points cluster in the top range, implying that larger clusters have a higher detection chance. In general, the results indicate consistent and successful performance in the detection regardless of the size of the cluster, and the lowest Pd is 0.85. The proposed EGSV-AACO model achieves high, consistent detection (Pd 0.85–0.99) across all cluster sizes, ensuring reliable spectrum sensing.

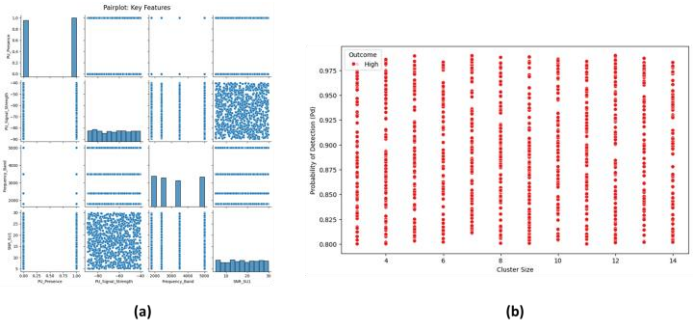


Figure 4: Graphical Representation of (a) Feature Correlation Analysis for Spectrum Sensing Parameters (b) High Detection Performance across Varying Cluster Sizes.

The maximum density lies within the range of Cluster ID that falls within 10-30 and the Frequency Band values that lie within the range of 200-400MHz indicating that the concentration of signals is high within this band. Moderate density also appears around Cluster ID values 35–50 with higher frequency bands (450–600 MHz). The distribution suggests that as Cluster ID increases, signals tend to occupy wider frequency bands, as shown in Figure 5 (a). This supports the purpose of spectrum analysis, highlighting how cluster-based identification can reveal underlying patterns of frequency utilization and potential interference hotspots. The results indicate that the proposed model is successful in identifying, and predicting the presence of spectrum patterns, as well as identifying possible interference.

The relationship between Probability of Detection (Pd) and Probability of False Alarm (Pfa). The scatter distribution shows Pd values ranging from 0.80 to 0.98 and Pfa values from 0.00 to 0.10, as shown in Figure 5 (b). The marginal histograms indicate a high density of Pd values around 0.90–0.95, while Pfa values are concentrated between 0.02–0.06. This shows that the proposed spectrum sensing approach is robust since it has demonstrated good detection performance and low false alarm rates. The mesh-like visualization highlights fluctuations in predicted gaps across varying spectrum bands and signal strengths. Figure 5 (c) illustrates a 3D Wireframe Plot mapping Frequency Band for the proposed model accurately predicts gaps based on spectrum conditions. PU Signal Strength, and Predicted Gap. This high concentration of the wireframe indicates high interdependence with the larger the PU Signal Strength, the larger the gap that is expected to be predicted. In the meantime, the changes in frequencies of moderate frequencies show various results, which means that the system is sensitive to the conditions of channels. That reinforces the importance of spectrum-aware prediction in improving accuracy and reliability in cognitive radio networks systems.

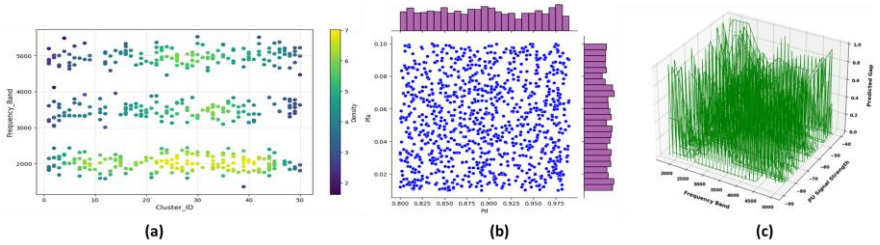


Figure 5: Graphical Representation of the (a) Density Distribution of Cluster ID and Frequency band (b) Joint Distribution Analysis of Pd and Pfa and (c) 3D Wireframe Plot of Frequency Band, PU Signal Strength, and Predicted Gap

The 3D bubble map represents a combination of the analysis of the three SNR features in different cluster sizes and target labels. Maximum clustering is found at SNR_SU1 and SNR_SU2 of between 5 to 15 dB with SNR_SU3 falling within the 0 to 10 dB range meaning that most of the communication states are represented in these ranges. Figure 6 (a) illustrates the relationship between SNR values across secondary users (SU1, SU2, and SU3). Larger bubbles, representing greater Cluster_Size, are concentrated around mid-range SNR values, suggesting that cooperative sensing performs more reliably under moderate channel quality conditions. The color variation (mapped to the Target label) highlights group separations, confirming that performance of the proposed model reliably performs cooperative sensing, with mid-range SNR clusters yielding accurate spectrum detection. Histogram of Probability of Detection (Pd) values of two target classes (0 and 1). The Pd values are between 0.80 and 0.98 with a range of 70 and 115 counts across bins. Class 0 (blue) sustains maximum frequencies that peak at above 100 counts, whereas class 1 (orange) has a range of 17-39 counts, as shown in Figure 6 (b). Both classes possess thicker distributions in the Pd range 0.85-0.90

and 0.95-0.98, The proposed model demonstrates strong target-discriminative detection, maintaining high Pd values (0.80–0.98) across both classes.

The connection between SNR_SU1 (5-30) and SNR_SU2 (5-30). The data points are separated into two groups: blue and red, which represent different target classes. Figure 6 (c) illustrates class is denser in the low to medium SNR_SU1 range (5-15), but the red class dominates at higher SNR_SU1 values (18-30). These two classes are uniformly spread across the SNR_SU2 values with most of them being between 10 25 which suggests a distinct separation along the SNRSU1 axis.

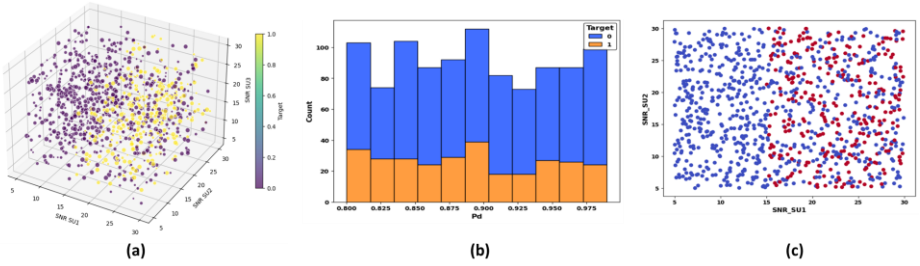


Figure 6: (a) SNR values across secondary users (SU1, SU2, and SU3) and (b) Target-Wise Distribution of Probability of Detection (Pd) and (c) Scatter Distribution of SNR_SU1 and SNR_SU2 across Target Classes

The heatmap shows correlations between spectrum sensing features. PU_Presence is powerfully negatively interrelated with Predicted_Gap (-0.66) and Target (-0.66). SNR_SU1 has a moderate positive association with Predicted_Gap (0.43) and Target (0.43), however, other SNR values (SNR_SU2, SNR_SU3) are barely connected. Figure 7 (a) shows the proposed model relies on key features, with few variables strongly influencing predictions. Most characteristics, including Pd, Pfa, Pmd, Cluster_ID, and Cluster_Size, have correlations between 0.00 and 0.06, showing independence. Strong diagonal values of 1.00 indicate complete self-correlation. This shows that a small number of variables have a large impact on prediction outcomes.

The arrangement of the distribution of the samples in four frequency bands. The band with the largest number is 1800 MHZ with approximately 268 samples, then 5000 MHZ with about 258 samples, as shown in Figure 7(b). The 2400 MHz band includes around 250 samples, whereas the 3500 MHz band has the fewest, with approximately 227 samples. Overall, the dataset is reasonably balanced across frequency bands, with variations of fewer than 40 samples, ensuring a reliable representation for spectrum sensing analysis

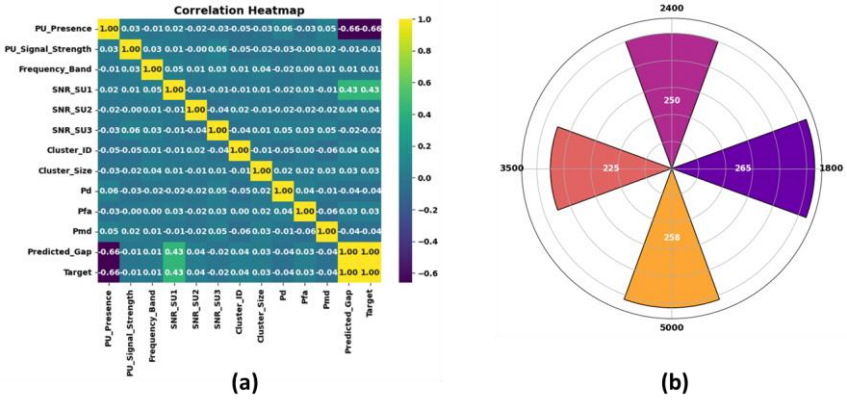


Figure 7: Graphical Representation of (a) Correlation Analysis of Spectrum Sensing Features (b) Sample Distribution Across Frequency Bands

4.1 Comparative analysis result

The proposed model was contrasted with traditional techniques such as DT [26], RF [26], ANN [26], SVM [26], and Seg-CR-VANET [27] in terms of, precision, F1 score, accuracy and recall. The evaluation shows that it performs better than expected in each of these important parameters. Precision shows how consistently the model detects actually idle spectrum without false alarms, and accuracy shows how correctly spectrum sensing judgments are made overall, as shown in Figure 8. F1-score reaches a compromise between precision and recall to provide reliable performance. Recall quantifies the capacity to identify every accessible idle channel, as shown in the intent of attaining dependable, effective, and privacy-preserving spectrum sensing and sharing in IoT networks is validated by these parameters taken together. Table 4 depicts the performance of the different spectrum sensing approaches to IoT communication.

Table 4: Comparative performance metrics of traditional and advanced IoV spectrum methods

Methods	Precision (%)	F1 score (%)	Accuracy (%)	Recall (%)
SVM [26]	94	93	92	92
RF [26]	97	93	92	92
DT [26]	93	91	90	90
ANN [26]	93	91	90	90
Seg-CR-VANET [27]	93.67	93.98	94	94.3
EGSV-AACO [Proposed method]	97.5	96.5	97	96

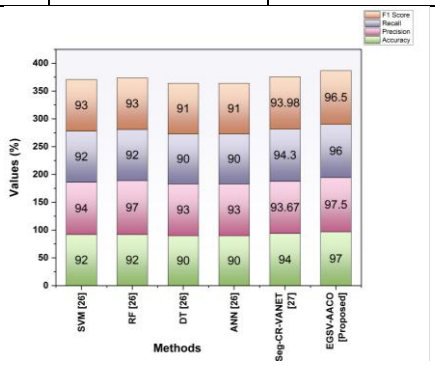


Figure 8: Performance evaluation of spectrum sensing techniques in iot communication precision, accuracy and F1 score, Recall

Traditional models such as SVM [26], RF [26], DT [26], ANN [26], and Seg-CR-VANET [27] achieved accuracy between 90 and 97%, with precision and recall ranging from 90 to 96%. In contrast, the proposed EGSV-AACO technique outperformed all others, achieving 97.5% precision, 96.5% F1-score, 97% accuracy, and 96% recall. In particular, it achieved accuracy improvement over conventional techniques and thus has a high detection reliability and efficiency in dynamic IoT spectrum sensing and sharing.

The EGSV-AACO with FedAvg can effectively address the weakness of the current approaches by incorporating optimal spectrum sensing and cooperation with adaptive optimization and privacy protection. EGSV provides the accuracy of detected idle bands, whereas AACO optimizes hyperparameters in real-time for handling spectrum variability. In contrast to classic centralized approaches, FedAvg eliminates exposure to raw data and offers security and scalability of IoT distributed devices. The hybrid model allows allocating the spectrum swiftly and efficiently, without collisions, and to communicating in a secure fashion, thus making it ideal in highly dynamic IoT ecosystem contexts.

70% training and 30% testing the dataset was spitted. Five-fold cross-validation was applied to ensure robustness, and the table presents standard deviation, 95% confidence intervals, mean, and p-values (Precision: 0.028, Accuracy: 0.032, F1-Score: 0.035, Recall: 0.041,), demonstrating the statistical significance of the EGSV-AACO model's performance. Table 5 shows the **statistical Analysis of EGSV-AACO Model**

Table 5: Statistical Analysis of EGSV-AACO Model with 70:30 Split Cross-Validation

Metric	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Mean	97.0	97.5	96.1	96.8
Std Dev	0.15	0.15	0.15	0.15
95% CI	[96.7–97.3]	[97.2–97.8]	[95.8–96.4]	[96.5–97.1]
p-value	0.032	0.028	0.041	0.035

Ablation study

An ablation study evaluates the impact of each model component by removing or isolating them. Table 6 shows the how EGSV and AACO individually and jointly improve spectrum sensing and sharing.

Table 6: Ablation Study of EGSV-AACO Modules

Model Variant	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
EGSV only	91	92	90	91
AACO only	88	89	87	88
EGSV + AACO	97	97.5	96	96.5

Table 6 demonstrates the EGSV predicts spectrum occupancy accurately, AACO optimizes allocation combined EGSV + AACO achieves best performance (Accuracy: 97%, Precision: 97.5%, Recall: 96%, F1: 96.5%).

4.2 Discussion

Evaluation presents an intelligent, decentralized, and privacy-preserving solution to spectrum sensing and sharing in IoT networks. Each of the existing approaches has certain limitations that make it less effective in dynamic, large-scale, and privacy-sensitive cases of IoT spectrum management scenarios. SVM [26] was incompatible with large volumes of IoT data, expensive to implement as a result of a non-linear kernel, and physically inappropriate in real-time/decentralized environments due to its lack of flexibility in a dynamic spectrum with high probabilities needed to cover temporal change overheads. RF [26] has sufficient accuracy but is very memory-intensive, takes longer to predict with large ensembles, is not flexible to non-stationary spectrum setups, and does not offer natural privacy-preserving measures in large distributed IoT-based networks. DT [26] was straightforward but can easily overfit, has low accuracy towards high-dimensional dynamic spectrum tasks, small generalization strength, and was also weak in a noisy environment due to IoT signals. ANNs [26] are data-hungry, computationally costly, prone to overfitting, not so interpretable, and cannot be readily integrated with privacy-preserving systems in decentralized schemes of IoT spectrum management. Seg-CR-VANET [27] enhanced spectrum assignment in vehicular networks, yet the overhead is high, adaptability to heterogeneous IoT devices or is lacking, and privacy protection is poor due to the federated and data-intensive architecture. The proposed EGSV-AACO framework addresses the limitations of existing methods by combining enhanced global, spatial, -related features with an adaptive ant colony optimization strategy. This integration allows dynamic tuning of parameters, faster and more reliable convergence, improved spectrum detection accuracy, and stronger resilience in complex VANET and IoT scenarios, ensuring efficient, decentralized, and privacy-preserving spectrum allocation

5 Conclusion

Research was developing a decentralized, intelligent, and privacy-preserving framework for spectrum sensing and dynamic sharing in IoT networks. The methodology combined EGSV-AACO for local spectrum detection and FedAvg for secure global aggregation. The research employed a Cognitive Radio Network (CRN) spectrum sensing dataset, incorporating features such as RSSI, SNR, PSD, interference levels, and temporal usage. Data preprocessing was carried out using Min-Max normalization to standardize features. FedAvg enabled collaborative optimization without exposing raw data, ensuring privacy preservation across IoT devices. Experimental results demonstrated that the proposed model achieved 97% accuracy, 97.5% precision, 96% recall, and a 96.5% F1-score, outperforming conventional approaches. The scientific contribution of this reserach lies in the development of a hybrid decentralized learning model that enhances detection reliability and spectrum allocation efficiency. The practical contribution includes enabling secure, scalable, and efficient spectrum sharing in dynamic IoT environments, supporting real-time communication without compromising privacy.

5.1 Limitations and future direction

The weakness of this approach is that it might have scaling problems in highly dense IoT networks, higher computational cost during real-time adaptation, and dependence on stable communication links. Future work will focus on integrating lightweight optimization methods, edge-assisted coordination, and adaptive model compression to reduce computation, improve scalability, and enhance communication resilience for large-scale IoT deployments. Real IoT devices were not emulated, in future work may include real-time testing with defined cluster sizes. Potential future applications include low-latency IoT-enabled patient monitoring in healthcare and collision-free communication in smart transportation systems, where secure and efficient spectrum sharing can enhance reliability and safety.

Author contributions

Jing Ling writing original draft preparation & methodology, Jing Ling investigation & writing review and editing.

References

[1] Lopez, C. A., Castillo, L. F., & Corchado, J. M. (2021). Discovering the value creation system in IoT ecosystems. *Sensors, 21*(2), 328. doi: https://doi.org/10.3390/s21020328

[2] Huo, X., & Wang, X. (2025). A deep learning-driven bidirectional power dispatch optimization framework for smart grids using IoT sensing data. *Informatica, 49*(26). doi: https://doi.org/10.31449/inf.v49i26.8524

[3] Alam, T. (2023). A reliable communication framework and its use in the Internet of Things (IoT). *Authorea Preprints.* doi: https://doi.org/10.36227/techrxiv.12657158.v1

[4] Li, Y. (2024). Construction and monitoring of a quantitative inversion model for conductivity using unmanned aerial vehicle remote sensing based on ELM algorithm. *Informatica, 48*(16). doi: https://doi.org/10.31449/inf.v48i16.6389

[5] Jaramillo-Ramirez, D., & Perez, M. (2021). Spectrum demand forecasting for IoT services. *Future Internet , 13*(9), 232. doi: https://doi.org/10.3390/fi13090232

[6] Chao, J., & Jiao, M. (2025). Network spectrum resource allocation and optimization based on deep learning and TRDM. *Informatica, 49*(13). doi: https://doi.org/10.31449/inf.v49i13.7374

[7] Nasser, A., Al Haj Hassan, H., Abou Chaaya, J., Mansour, A., & Yao, K. C. (2021). Spectrum sensing for cognitive radio: Recent advances and future challeng

es. *Sensors, 21*(7), 2408. doi: https://doi.org/10.3390/s21072408

[8] Fernando, X., & Lăzăroiu, G. (2023). Spectrum sensing, clustering algorithms, and energy-harvesting technology for cognitive-radio-based Internet-of-Things networks. *Sensors, 23*(18), 7792. doi: https://doi.org/10.3390/s23187792

[9] Ali, M., Yasir, M. N., Bhatti, D. M. S., & Nam, H. (2022). Optimization of spectrum utilization efficiency in cognitive radio networks. *IEEE Wireless Communications Letters, 12*(3), 426–430. doi: https://doi.org/10.1109/LWC.2022.3229110

[10] Liu, S., Pan, C., Zhang, C., Yang, F., & Song, J. (2023). Dynamic spectrum sharing based on deep reinforcement learning in mobile communication systems. *Sensors, 23*(5), 2622. doi: https://doi.org/10.3390/s23052622

[11] Biswas, D., Neuwirth, S., Paul, A. K., & Butt, A. R. (2021). Bridging network and parallel I/O research for improving data-intensive distributed applications. *2021 IEEE Workshop on Innovating the Network for Data-Intensive Science (INDIS)* (pp. 50–56). IEEE. doi: https://doi.org/10.1109/INDIS54524.2021.00011

[12] Wang, Y., Li, X., Wan, P., & Shao, R. (2021). Intelligent dynamic spectrum access using deep reinforcement learning for VANETs. *IEEE Sensors Journal, 21*(14), 15554–15563. doi: https://doi.org/10.1109/JSEN.2021.3056463

[13] Grissa, M., Yavuz, A. A., Hamdaoui, B., & Tirupathi, C. (2021). Anonymous dynamic spectrum access and sharing mechanisms for the CBRS band. *IEEE Access, 9*, 33860–33879. doi: https://doi.org/10.1109/ACCESS.2021.3061706

[14] Chang, H. H., Song, Y., Doan, T. T., & Liu, L. (2023). Federated multi-agent deep reinforcement learning (Fed-MADRL) for dynamic spectrum access. *IEEE Transactions on Wireless Communications, 22*(8), 5337–5348. doi: https://doi.org/10.1109/TWC.2022.3233436

[15] Kaur, A., Thakur, J., Thakur, M., Kumar, K., Prakash, A., & Tripathi, R. (2023). Deep recurrent reinforcement learning-based distributed dynamic spectrum access in multichannel wireless networks with imperfect feedback. *IEEE Transactions on Cognitive Communications and Networking, 9*(2), 281–292. doi: https://doi.org/10.1109/TCCN.2023.3234276

[16] Zhang, S., Ni, Z., Kuang, L., Jiang, C., & Zhao, X. (2023). Traffic priority-aware multi-user distributed dynamic spectrum access: A multi-agent deep RL approach. *IEEE Transactions on Cognitive Communications and Networking, 9*(6), 1454–1471. doi: https://doi.org/10.1109/TCCN.2023.3307944

[17] Amrallah, A., Mohamed, E. M., Tran, G. K., & Sakaguchi, K. (2021). Enhanced dynamic spectrum access in UAV wireless networks for post-disaster area surveillance: A multi-player multi-armed bandit approach. *Sensors, 21*(23), 7855. doi: https://doi.org/10.3390/s21237855

[18] Makhdomi, A. A., & Begh, G. R. (2022). Blockchain-based scalable model for secure dynamic spectrum access. *Physical Communication, 55*, 101874. doi: https://doi.org/10.1016/j.phycom.2022.101874

[19] Chen, Y., Wang, Y., Zhang, J., & Di Renzo, M. (2021). QoS-driven spectrum sharing for reconfigurable intelligent surfaces (RISs)-aided vehicular networks. *IEEE Transactions on Wireless Communications, 20*(9), 5969–5985. doi: https://doi.org/10.1109/TWC.2021.3071332

[20] Gu, P., Li, R., Hua, C., & Tafazolli, R. (2021). Dynamic cooperative spectrum sharing in a multi-beam LEO–GEO co-existing satellite system. *IEEE Transactions on Wireless Communications, 21*(2), 1170–1182. doi: https://doi.org/10.1109/TWC.2021.3102704

[21] Kazemi, N., & Azghani, M. (2024). Secure spectrum sharing and power allocation by multi-agent reinforcement learning. *Digital Signal Processing, 146*, 104369. doi: https://doi.org/10.1016/j.dsp.2023.104369

[22] Alipour-Fanid, A., Dabaghchian, M., Jiao, L., & Zeng, K. (2024). Learning-based secure spectrum sharing for intelligent IoT networks. *2024 25th International Symposium on Quality Electronic Design (ISQED)* (pp. 1–8). IEEE. doi: https://doi.org/10.1109/ISQED60706.2024.10528684

[23] Raja, K., Kottursamy, K., Ravichandran, V., Balaganesh, S., Dev, K., Nkenyereye, L., & Raja, G. 2024. An efficient 6G federated learning-enabled energy-efficient scheme for UAV deployment. *IEEE Transactions on Vehicular Technology*, 74(2), 2057-2066. 10.1109/TVT.2024.3390226

[24] Li, L., Xie, W., & Zhou, X. 2023. Cooperative spectrum sensing based on LSTM-CNN combination network in cognitive radio system. *IEEE Access, 11*, 87615-87625. 10.1109/ACCESS.2023.3305483

[25] Okegbile, S. D., & Maharaj, B. T. (2021). Age of information and success probability analysis in hybrid spectrum access-based massive cognitive radio networks. *Applied Sciences, 11*(4), 1940. doi: https://doi.org/10.3390/app11041940

[26] Baiyekusi, O., Mahmoud, H., Mi, D., Arshad, J., Adeyemi-Ejeye, F., & Lee, H. (2024). An ML-based spectrum sharing technique for time-sensitive applications in industrial scenarios. *2024 International Wireless Communications and Mobile Computing (IWCMC)* (pp. 1607–1612). IEEE. doi: https://doi.org/10.1109/IWCMC61514.2024.10592619

[27] Hossain, M. A., Md Noor, R., Yau, K. L. A., Azzuhri, S. R., Z'aba, M. R., Ahmedy, I., & Jabbarpour, M. R. (2021). Machine learning-based cooperative spectrum sensing in dynamic segmentation enabled cognitive radio vehicular network. *Energies, 14*(4), 1169. doi: https://doi.org/10.3390/en14041169

Image Edge Detection Using FHN-CNN Model Based on Reaction-Diffusion Equations within a Dynamical Systems Framework

Zhao Yang
Academic Affairs Office, Zibo Polytechnic University, Zibo, 255314, China
E-mail: zhaoyang_zyang@outlook.com

Keywords: dynamical system, reaction-diffusion equation, image processing, FHN; Cellular-NN, Turing spot

Received: June 1, 2025

With the advancement of imaging technology, images present diverse and highly complex characteristics, and the theory of dynamical systems has great potential in image processing due to its unique mathematical properties. Therefore, this study proposes an IPT based on the reaction-diffusion equation. This technique combines cellular neural networks with the reaction-diffusion equation within the framework of dynamical system theory. Specifically, by introducing the Laplacian operator, the membrane potential in the FitzHugh-Nagumo equation is correlated with the spatiotemporal dynamics of the recovery variable. A new type of image processing model is constructed by mapping to the dynamic evolution of locally coupled mesh cells in convolutional neural networks. The theoretical framework of the technology is further improved through the dynamic analysis of the Turing instability of the model and the gradient changes of the reaction-diffusion system. The results showed that in the performance testing of the research model, the Edge Preservation Index (EPI) was 0.89 and the Pratt's Figure of Merit (PFOM) was 0.95, which were higher than the comparison models, indicating excellent model performance. Meanwhile, the time cost for the new model to complete one detection was only 0.9 seconds, and there were only 2 iterations, which was significantly better than other models. Research has shown that the new model has higher computational efficiency and better real-time performance. This study provides new ideas and methods for image processing and helps to promote the development of image processing algorithms towards high efficiency and intelligence.

Povzetek: Študija predlaga nov dinamični model obdelave slik, ki temelji na reakcijsko-difuzijski enačbi in celičnih nevronskih mrežah ter dosega visoko kakovost ohranjanja robov, visoko natančnost in zelo učinkovito realnočasovno delovanje.

1 Introduction

Early Image Processing Techniques (IPTs) were mainly used to improve image quality, such as image denoising and enhancing image contrast. The advancement of computer technology has led IPT to gradually enter the stage of digital image processing, placing higher demands on the real-time and precision of image processing [1]. To meet the needs of digital transformation, IPT's core algorithms have transitioned from manual algorithms and simple mathematical models to Deep Learning (DL) models [2]. Numerous scholars have conducted research on IPT. Zangana et al. explored the combination of edge detection and existing image processing methods, and summarized the advantages and limitations of existing edge detection methods. Advanced methods utilizing DL, fuzzy logic, and optimization algorithms have shown good results in improving edge detection accuracy [3-4]. Chen et al. proposed a DL-based post-processing method for magnetic resonance images. The processing steps included noise reduction, image artifact correction, and image resolution improvement. This method had a significant effect on improving image quality and correcting image artifacts [5]. Zhou et al. proposed an underwater image enhancement method based on multi-interval sub-histogram perspective equalization. This method extracted statistical features of the image, estimated the degree of feature drift in each region of the image, and achieved adaptive feature enhancement. The experiment showed that this method was significantly superior to existing methods in both qualitative and quantitative aspects [6]. Zhang et al. proposed an underwater image enhancement method based on weighted wavelet visual perception fusion to address the issue of degraded underwater image quality. This method integrated high-frequency and low-frequency components of images at different scales to obtain high-quality underwater images, which were superior to the latest methods in both qualitative and quantitative aspects [7]. Jiang et al. proposed a diffusion-based low-light image enhancement method to address long processing time and high computational resource consumption in image restoration using diffusion models. It introduced a high-frequency recovery module, achieving better fine-grained recovery and significantly improving efficiency compared to previous diffusion based methods

[8]. Arshaghi et al. proposed a method based on Convolutional Neural Network (CNN) for the problem of agricultural product defect identification. They studied the classification of five potato diseases and conducted experiments using 5,000 images. The results showed that this method outperformed other existing methods in terms of the accuracy of defect classification [9].

Reaction Diffusion Equation (RDE) is a type of partial differential equation used to describe the dynamic process of coupling diffusion and chemical reaction of matter or information in space [10]. The algorithm itself is constantly being improved and applied in various fields. Liu et al. proposed an efficient quantum algorithm for nonlinear RDE, introducing Carleman linearization method and Allen Cahn equation to extract derivative information by post-processing quantum states. The algorithm was found to achieve faster convergence speed [11]. Xue et al. constructed an RDE with non local spatial memory to simulate the cognitive movement of a single species and studied its dynamics. This study explored the combined effects of perceptual scale and memory delay on system stability and spatiotemporal dynamics through Turing Hopf bifurcation and double Hopf bifurcation. The perceptual scale and memory delay could be analyzed through dynamics [12]. Zayed et al. studied solitary wave solutions in stochastic RDEs and revealed new triangular and hyperbolic stochastic exact solutions by introducing two different integration methods and concatenating them. In numerical simulations, the newly discovered solutions using this method were effective at the application level [13]. Agresti A studied the noise regularization problem of RDE groups with quality control. Experimental results showed that moderate transmission type random multiplicative perturbations had regularization effects [14]. Hadhoud et al. proposed a shift configuration scheme for solving 2D fractional order coupled RDE. This method determined the expansion coefficients by simplifying the boundary conditions into a system of ordinary differential equations with these coefficients, transforming the 2D spatiotemporal fractional order coupled RDE into an algebraic equation system. The proposed method exhibited excellent performance and efficiency [15]. Overall, RDE can be applied in image processing, with strong potential for development in areas such as image denoising, restoration, and segmentation [16]. Barbu T proposed a new framework for the problem of automatic vehicle detection and tracking. This framework combined DL with target detection technology based on Gaussian mixture models to detect moving vehicles and determine the correspondence between video targets between consecutive frames through multi-scale analysis. [17]. Hagemann et al. proposed an infinite-dimensional diffusion model for the finite-dimensional Stochastic Banach Space Diffusion Models (SBDMs) in image generation, which models the training data over rectangular regions. This model achieved higher-resolution image generation and improved training efficiency. The results showed that this method could effectively overcome the shortcomings of existing methods and show practical advantages on synthetic Gaussian mixture samples, MNIST datasets, and datasets generated by nonlinear 2D reaction-diffusion equations. [18]. The specific comparison of the above-mentioned literature is shown in Table 1.

Table 1: Literature review table

Literature	Method	Advantages	Limitations
Literature [5]	Post-processing method for magnetic resonance images based on DL	It has a remarkable effect on improving image quality and correcting artifacts	The model decision-making mechanism is like a "black box", and its application in security-sensitive fields is limited
Literature [6-7]	An underwater image enhancement method based on the fusion of multi-interval sub-histogram perspective equalization and weighted wavelet visual perception	It is significantly superior to the existing methods in both qualitative and quantitative aspects	For specific degradation models (such as underwater), the generalization ability remains to be verified
Literature [8]	A low-light image enhancement method based on the diffusion model introduces a high-frequency recovery module	Better fine recovery has been achieved and efficiency has been significantly improved	Diffusion models generally have the problems of long processing time and high consumption of computing resources
Literature [9]	Agricultural product disease identification based on CNN	It outperforms other existing methods in terms of defect classification accuracy	Strong reliance on large-scale labeled data for training leads to high deployment costs
Literature [10]	Reaction-diffusion equation with non-local spatial memory	It can simulate cognitive movements and reveal the combined influence of perceptual scale and memory delay on system dynamics	It focuses on theoretical dynamic analysis, but the specific application paradigm in image processing is not clear
Literature	The solitary wave solution of the	A new stochastic exact	The construction of solutions

[11]	random reaction-diffusion equation is solved by the series integration method	solution was revealed, and its application effect in numerical simulation was remarkable	strongly depends on the form of the equation, and universal methods are still under exploration
Literature [12]	Study the noise regularization of reaction-diffusion equations with quality control	Experiments show that random multiplicative perturbations have a regularization effect	Theoretical research is rather abstract, and the connection with specific image processing tasks needs to be further established
Literature [16]	A shift configuration scheme for solving 2D fractional-order coupled reaction-diffusion equations	It demonstrated excellent computing performance and efficiency	The method is specifically designed for fractional calculus scenarios and does not have a direct advantage in integer-order problems
Literature [17]	An automatic vehicle detection and tracking framework combining DL with object detection technology based on Gaussian mixture model	The detection of moving vehicles was achieved, and the inter-frame target correspondence was determined through multi-scale analysis	For systems engineering, the theoretical basis and interpretability of the underlying processing units have not been deeply explored
Literature [18]	An infinite-dimensional SBDM for image generation	It can achieve higher-resolution image generation and improve training efficiency	The model is complex, and the performance improvement comes at the cost of high computational complexity

To sum up, current IPT research, especially DL methods, still faces some inherent limitations while achieving performance improvements. Firstly, there is poor physical interpretability. DL models are like "black boxes" whose decision-making mechanisms are difficult to understand, limiting their application in security-sensitive fields. Secondly, there is a strong dependence on data and computing power. High performance heavily depends on large-scale labeled data and powerful computing resources, creating a contradiction between generalization ability and deployment cost. The last issue is that it is difficult to balance computational efficiency and accuracy. Although complex network structures can improve accuracy, they often lead to high computational overhead and a large number of iterations, making it difficult to meet the needs of real-time processing. In addition, traditional reaction-diffusion models have good physical interpretability. However, its application in image processing often relies on empirical parameter adjustment, has limited ability to handle complex textures and noise, and fails to be deeply integrated with modern data processing frameworks. Given this, this study proposes an IPT based on RDE, which combines the FitzHugh-Nagumo (FHN) model with the Cellular Neural Network (Cellular-NN) to construct a new FHN-Cellular-NN image processing model. This model compensates for the shortcomings of traditional algorithms including insufficient computing power, strong data dependence, and poor real-time performance. The innovation of the research lies in the combination of Cellular-NN and FHN, which achieves a balance between efficiency, accuracy, and physical interpretability in image processing through dynamic driven feature optimization. Secondly, using dynamic analysis tools such as Turing instability analysis, waveform analysis, steady-state and stable point value analysis, FHN-Cellular-NN is deconstructed from the perspective of dynamic systems. This study helps to reveal the intrinsic evolution mechanism of the model, optimize parameter design, and enhance the physical interpretability and robustness of the algorithm.

2 Methods and materials

2.1 Construction of image processing model based on RDE

To address the limitations of existing image processing models in terms of computational efficiency, data dependency, and physical interpretability, this study aims to construct a new image edge detection model based on FHN-CNN. The study proposes a core hypothesis: By mapping the dynamic characteristics of the FHN model to the local coupling architecture of CNN and using dynamical system theory tools such as Turing instability for regulation, the constructed model can maintain high edge detection accuracy while reducing computational complexity and the number of iterations. Next, the study will elaborate in detail on the construction process, dynamic analysis, and solution methods. In digital image processing, the choice of color model directly affects the processing effect. The commonly used color models for hardware devices include Red-Green-Blue (RGB), Cyan Magenta Yellow (CMY), and Hue Saturation Value (HSV). Due to the differences in numerical representation between RGB and CMY models and human perception, while HSV models are more in line with human visual characteristics [19], it is often necessary to convert RGB models to HSV models in the image processing process of RDE. The specific processing flow is shown in Figure 1. The FHN equation is selected as the core dynamic equation for constructing the proposed model. Its reaction-diffusion form is as

follows: Step 1 is to define the core dynamic equation. Step 2 is the integration with Cellular-BN. Step 3 is to construct the matrix form and boundary conditions. Step 4 is the edge detection task of the image, which can be completed by solving the above coupled ordinary differential equation system.

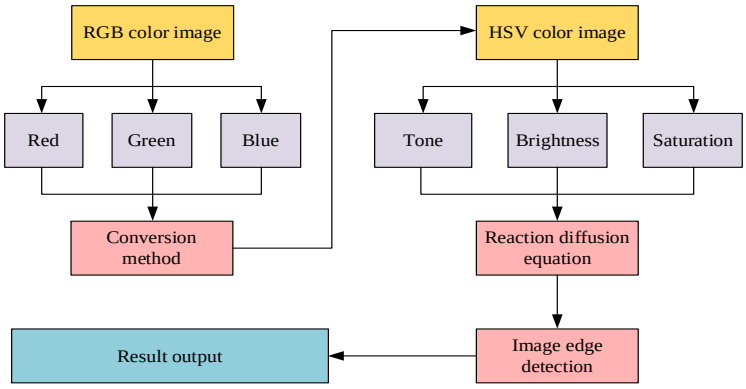


Figure 1: Color model conversion

In Figure 1, the RGB color image can be decomposed into pixels of the three primary colors of red, green, and blue. The three primary colors correspond to different values in their color model space, and the values of the three primary colors are mapped one-to-one to the HSV color space through conversion methods to obtain HSV color images. The three elements of hue, brightness, and saturation separated from HSV color images are the input values suitable for RDE processing. This study selects the FHN equation, which is the most suitable for image processing among RDE types, as the algorithm basis for image edge detection [20]. FHN-RDE is shown in equation (1).

$$\begin{cases} \theta_t s = f(s,q) + D_s \nabla^2 s \\ \theta_t q = g(s,q) + D_q \nabla^2 q \end{cases} \tag{1}$$

In equation (1), $\theta_t s$ is the time derivative of the membrane potential. $\theta_t q$ is the time derivative of the recovery variable. s is the neuronal membrane potential. q is the recovery variable. D_s and D_q are diffusion coefficients. ∇^2 is the Laplacian operator. The expanded form after substituting equation (1) into the function is shown in equation (2).

$$\begin{cases} \dfrac{\theta s}{\theta t} = \delta[D_s \nabla^2 s + s(a-s)(s-1) - q] \\ \dfrac{\theta q}{\theta t} = s - bq + D_q \nabla^2 q \end{cases} \tag{2}$$

In equation (2), $\dfrac{\theta s}{\theta t}$ is the time derivative of the membrane potential. $\dfrac{\theta q}{\theta t}$ is the time derivative of the recovery variable. t is time. δ is the external stimulus current. a is the activation threshold parameter. b is the attenuation coefficient of the recovery variable. The unfolded original FHN RDE is obtained through equations (1) and (2). The equation cannot be directly used for image edge detection, so this study introduces a locally interconnected nonlinear simulation dynamic system Cellular-NN to construct an image processing model based on RDE [21]. This study names the new model FHN-Cellular-NN. The Cellular-NN structure is shown in Figure 2.

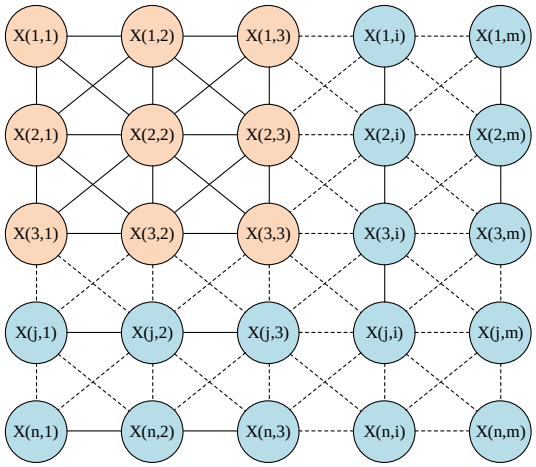


Figure 2: Cellular-NN dynamical system structure

In Figure 2, Cellular-NN adopts a grid topology structure, and its core architecture consists of n rows and m columns of regularly arranged units. Each unit $X`_{ji}$ corresponds to the position of the j-th row and i-th column in the array. The connection characteristics of this network are manifested as local coupling, where each unit only establishes electrical connections with its directly adjacent units. This neighborhood interconnection mechanism significantly reduces the complexity of the system. The state function of the Cellular-NN dynamical system is shown in equation (3).

$$\begin{aligned} X`_{ji} = & \sum_{X(k,l)\in V_r(j,i)} \alpha(j,i,k,l) H_1 \\ & + \sum_{X(k,l)\in V_r(j,i)} \beta(j,i,k,l) H_2 - X_{ji} \end{aligned} \tag{3}$$

In equation (3), $X`_{ji}$, X_{ji}, and $X(k,l)$ are the updated state, current state, and neighboring units of unit

$X(j,i)\cdot V_r(j,i)$ is connected to unit $X(j,i)\cdot$ r is the adjacency radius of the unit. $\alpha(j,i,k,l)$ is the feedback matrix. $\beta(j,i,k,l)$ is the control matrix. H_1 is the output connection weight of $X(k,l)$ and $X(j,i)\cdot$ H_2 is the input connection weight for $X(k,l)$ and $X(j,i)\cdot$ The combination of FHN and Cellular-NN first introduces the Laplacian operator, as shown in equation (4).

$$\begin{aligned}\nabla^2 &= (x,y+1)+(x,y-1)+(x+1,y)\\ &\quad +(x-1,y)-4(x,y)\end{aligned} \tag{4}$$

In equation (4), x is the value on the x axis. y is the numerical value on the y axis. (x,y) represents the coordinate point. Secondly, each (x,y) two-dimensional coordinate point is considered as a unit of the Cellular-NN dynamical system, so substituting equation (4) into equation (2) yields equation (5).

$$\begin{cases} s_{j,i}` = -s_{j,i}+\delta D_s s_{j,i+1}+\delta D_s s_{j,i-1}+\delta D_s s_{j+1,i}+\delta D_s s_{j-1,i}+\\ \quad [1+\delta a-4\delta D_s+\delta(1+a)s_{j,i}-\delta s_{j,i}^2]s_{j,i}-\delta q_{j,i} \\ q_{j,i}` = -q_{j,i}+D_q q_{j,i+1}+D_q q_{j,i-1}+D_q q_{j+1,i}+\\ \quad D_q q_{j-1,i}+(1-4D_q-b)q_{j,i}+s_{j,i}\end{cases} \tag{5}$$

In equation (5), $s_{j,i}`$, $q_{j,i}`$, $s_{j,i}$, and $q_{j,i}$ represent the membrane potential time derivative, recovery variable time derivative, membrane potential, and recovery variable of the j-th row and i-th column units. If $X`_{ji}$ in equation (3) can be regarded as $s_{j,i}`$ in equation (5), then the matrix of FHN-Cellular-NN membrane potential dimension is shown in equation (6).

$$\begin{cases} \alpha_{s,s}=\begin{bmatrix} 0 & \delta D_s & 0 \\ D_s & 1+\delta a-4\delta D_s+\delta(1+a)x-\delta x^2 & \delta D_s \\ 0 & \delta D_s & 0 \end{bmatrix} \\ \alpha_{q,s}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & -\delta & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \beta_{s,s}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{cases} \tag{6}$$

In equation (6), $\alpha_{s,s}$ and $\beta_{s,s}$ are the coupling feedback matrix and coupling control matrix between membrane potential and membrane potential. $\alpha_{q,s}$ is the coupling feedback matrix of membrane potential and recovery variable. If $X`_{ji}$ in equation (3) can be regarded as $q_{j,i}`$ in equation (5), then the FHN-Cellular-NN restores the matrix of variable dimensions as shown in equation (7).

$$\begin{cases} \alpha_{q,q}=\begin{bmatrix} 0 & D_q & 0 \\ D_q & 1-4D_q-b & D_q \\ 0 & D_q & 0 \end{bmatrix} \\ \alpha_{q,s}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \\ \beta_{q,q}=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{cases} \tag{7}$$

In equation (7), $\alpha_{q,q}$ and $\beta_{q,q}$ are the coupling feedback matrix and coupling control matrix between two recovery variables. To eliminate the arbitrariness of FHN-Cellular-NN solutions and ensure the uniqueness of numerical solutions, this study constructs discrete boundary condition equations. The discrete boundary condition equation based on rows is shown in equation (8).

$$\begin{cases} R_{j,-1}=R_{j,0} \\ R_{j,m-1}=R_{j,m} \end{cases},\ j=0,1,...,n-1 \tag{8}$$

In equation (8), $R_{j,-1}$ and $R_{j,0}$ are the values of row j, column -1, and row j, column 0 of the matrix. $R_{j,m-1}$ and $R_{j,m}$ are the values of the j-th row, $m-1$-th column, and j-th row, m-th column of the matrix. The column based discrete boundary condition equation is shown in equation (9).

$$\begin{cases} R_{-1,i}=R_{0,i} \\ R_{n-1,i}=R_{n,i} \end{cases},i=0,1,...,m-1 \tag{9}$$

In equation (9), $R_{-1,i}$ and $R_{0,i}$ represent the values of row -1, column i, and row i of the matrix. $R_{n-1,i}$ and $R_{n,i}$ are the values of row $n-1$, column i, and row n, column i of the matrix. The final solution of FHN-Cellular-NN can be obtained by combining equations (8), (9), and (5). The discrete boundary condition adopted here mathematically corresponds to the Neumann boundary condition (zero flow condition). The physical meaning of this condition is to assume that the pixel values outside the image boundary are the same as those at the very edge, thereby numerically eliminating the false gradient at the boundary. In summary, the functional expression of FHN-Cellular-NN is obtained through equations (1) to (5), and the solution method of FHN-Cellular-NN is obtained through equations (6) to (9).

After completing the mathematical construction and discretization solution of the FHN-Cellular-NN model, a core question emerges: Does this model have the ability to generate stable and controllable patterns during the dynamic evolution process to reliably perform the image edge detection task? To solve the above problems and deeply understand the intrinsic behavior and stability of the model, the study has shifted from the perspective of

dynamical system theory to conduct an in-depth dynamic analysis of FHN-Cellular-NN. This analysis aims to provide a theoretical basis for the selection of model parameters by exploring the Turing instability, steady-state characteristics, and gradient evolution laws, and to clarify the dynamic basis for achieving spontaneous edge emergence in image processing. The study adopts the finite difference method to discretize the reaction-diffusion equation. Specifically, the spatial derivative is discretized using the second-order central difference scheme. The temporal derivative is advanced using the explicit Euler method. The choice of this explicit format is based on its simple implementation and high compatibility with the local coupling characteristics of Cellular-NNs, facilitating parallel computing on grid cells.

2.2 Dynamic analysis of image processing model based on RDE

The image processing model from a dynamic perspective actually views image processing tasks as a dynamic evolutionary process. Conducting dynamic analysis on FHN-Cellular-NN can help clarify the interaction mechanism between diffusion and reaction, and provide criteria for establishing the dynamic stability of non-equilibrium systems. This study explores the universality of non-equilibrium systems by analyzing the Turing instability of FHN-Cellular-NN [22]. The main idea is to establish a quantitative criterion for diffusion driven instability through linear stability analysis and characteristic equation solving. The characteristic equation of equation (2) at point $(0,0)$ is shown in equation (10).

$$\Omega^2+\left[a\delta+b+(D_s+D_q)\frac{c^2}{d^2}\right]\Omega +\left(a\delta+D_s\frac{c^2}{d^2}\right)\left(b+D_q\frac{c^2}{d^2}\right)+\delta=0 \tag{10}$$

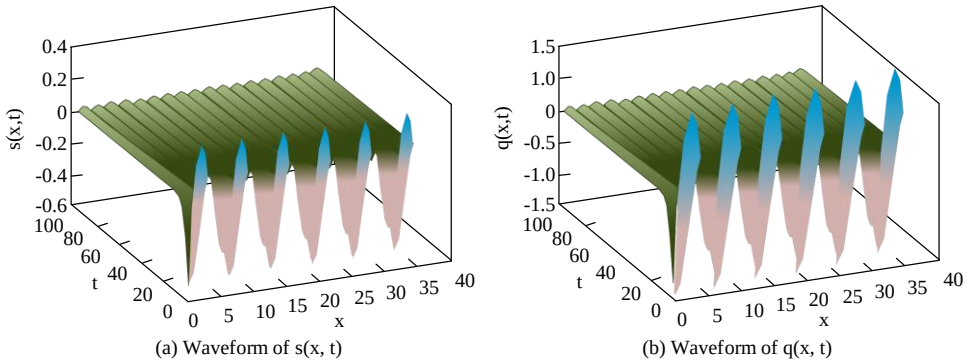
In equation (10), Ω is a constant. c and d are random variables. Assuming $c=0$, equation (10) is transformed into equation (11).

$$\Omega^2+(a\delta+b)\Omega+(ab+1)\delta=0 \tag{11}$$

In equation (11), Ω is a constant. a is the activation threshold parameter. b is the attenuation coefficient of the recovery variable. δ is the external stimulus current. When $\begin{cases}a\delta+b>0\\ab+1>0\end{cases}$, equation (11) has a root with a negative real part, and when $\begin{cases}a\delta+b>0\\ \delta=-\dfrac{b}{a}\end{cases}$, bifurcation occurs. When $c\neq 0$, equation (10) can be written as equation (12).

$$\begin{cases}\Omega^2+F_c(\delta)\Omega+D_c(\delta)=0\\ F_c(\delta)=a\delta+b+(D_s+D_q)\dfrac{c^2}{d^2}>0\\ D_c(\delta)=D_sD_q\dfrac{c^4}{d^4}+(a\delta D_q+bD_s)\dfrac{c^2}{d^2}+(ab+1)\delta=0\end{cases} \tag{12}$$

In equation (12), $F_c(\delta)$ and $D_c(\delta)$ are newly constructed functions. By analyzing the solutions of characteristic equation (10) under different parameter ranges, the parameter values for the Turing instability of FHN-Cellular-NN are determined. The determined values of the parameter group are substituted into equation (10) for solving, resulting in the waveform diagram when instability occurs, as shown in Figure 3.

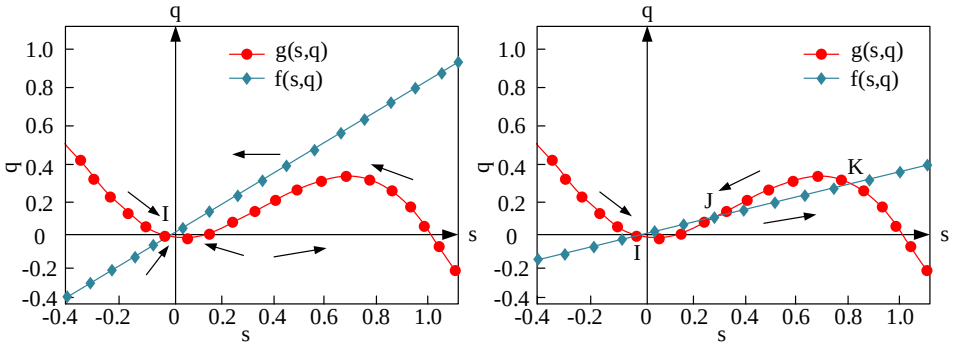


(a) Waveform of s(x, t) (b) Waveform of q(x, t)

Figure 3: FHN-Cellular-NN waveform diagram

Figure 3 shows the waveform of membrane potential dimension $s(x,t)$ and recovery variable dimension $q(x,t)$ for $a=2$, $D_s=6$, $D_q=0.001$, and $b=-0.1$. When the above parameters are used, the FHN-Cellular-NN exhibits Turing instability, and the steady state $(0,0)$ is unstable, resulting in a new non-steady state solution. Based on the dynamic analysis of the Turing instability of FHN-Cellular-NN, this study obtains the parameter solution of the FHN-Cellular-NN model in a stable state. Next, further analysis is conducted on the mono-stable and multi-steady states of

FHN-Cellular-NN under different a and b values, as shown in Figure 4.



(a) Single point steady state (b) Two point steady state

Figure 4: Analysis of FHN-Cellular-NN single and multi-steady states

Figure 4 shows the analysis of the results obtained by adjusting the values of a and b to find the monostable and bistable parameter solutions for FHN-Cellular-NN. In Figure 4 (a), the g(s,q) and f(s,q) functions of FHN-Cellular-NN have a stable point I, and values greater than this stable point are considered edge points. In Figure 4 (b), the function has three stable points I, J, and K. That is, at these values of a and b, FHN-Cellular-NN is a multi-stable system with multiple stable points. The smaller the stable point value, the more complete the detection coverage of image edges, but the processed image noise will increase. On the contrary, the smaller the detection coverage of image edges, the finer the edge processing and the lower the noise [23]. Usually, multiple experiments are required to find a moderate value for the stable point. The mono-stable state is relatively easier to adjust because it only has one stable point. Therefore, this study selects a mono-stable system and adjusts the stable point of FHN-Cellular-NN to find its suitable value. Firstly, the stable point value is defined as ∂, and the diffusion equation of ∂ can be expressed as equation (13).

$$\partial(x) = Q(s_0(x); U, G) \tag{13}$$

In equation (13), $\partial(x)$ is the diffusion equation of ∂. U is the diffusion coefficient. G and Q are constants. $s_0(x)$ is the diffusion function for restoring the variable dimension. Equation (13) can be directly used for detecting image edges, but it still has some shortcomings. When the diffusion coefficient increases, the distance between edges will increase, resulting in blurred edges of the image. Therefore, this study further optimizes the diffusion coefficient U to avoid detecting edge blurring. The functional expression for constructing the diffusion coefficient U is shown in equation (14).

$$U = u(\varepsilon) = \frac{1}{1+(\varepsilon / k)^2} \tag{14}$$

In equation (14), $u(\varepsilon)$ is a function expression of the adjustment coefficient. ε is the normalized value of each pixel in the image. k is a constant. The acquisition of ε can be expressed as equation (15).

$$\varepsilon = |\ \nabla \mathrm{s}\ | = \sqrt{B_x^2 + B_y^2} \tag{15}$$

In equation (15), ε is the normalized value of each pixel in the image. $\nabla \mathrm{s}$ is an operator for restoring variable dimensions. B_x and B_y are specific matrices. The specific matrix of B_x and B_y is given by equation (16).

$$\begin{cases} B_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \sigma_0 \\ B_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \sigma_0 \end{cases} \tag{16}$$

In equation (16), σ_0 is the original input image. Equations (13) to (16) introduce a method of adjusting the stable point value of FHN-Cellular-NN to a moderate value, enabling FHN-Cellular-NN to have adaptive control capability for stable point values. In the process of feature extraction from digital images, region boundaries often exhibit significant gradient changes. At this point, the adjustment function will automatically converge to the minimum value in the high gradient region, achieving automatic termination of boundary positioning and ensuring accurate convergence of the contour line to the target edge position. The gradient variation of the reaction-diffusion system is shown in Figure 5.

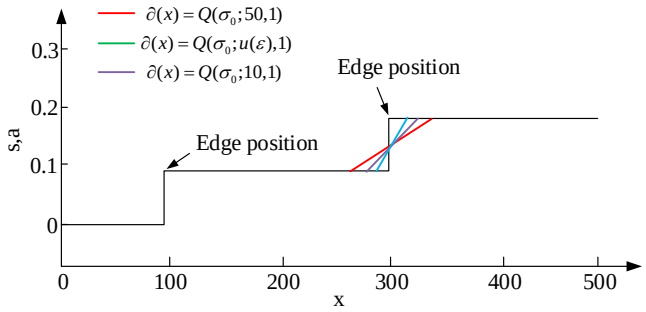


Figure 5: Gradient variation of reaction-diffusion system

Figure 5 shows the gradient changes of diffusion distribution curves $\partial(x)=Q(\sigma_0;50,1)$, $\partial(x)=Q(\sigma_0;10,1)$, and $\partial(x)=Q(\sigma_0;u(\varepsilon),1)$ at parameter values $b=2$, $\varepsilon=10^3$, $D_S=0.5$, and $D_q=4$. When the stable point value used is not appropriate, the diffusion distribution curve will vibrate in two directions, producing double edges and blurring the image edges. After introducing the adjustment function, the $\partial(x)=Q(\sigma_0;u(\varepsilon),1)$ diffusion distribution curve can be accurately located at the edge of the image. This study conducts a dynamic analysis of an image processing model based on RDE, and determines the parameter values for the Turing instability of FHN-Cellular-NN by solving the characteristic equation. Based on the found parameter values, the waveform, steady-state, and stable point values of FHN-Cellular-NN are analyzed. Finally, a method is introduced to adjust the stable point value of FHN-Cellular-NN to a moderate value. Based on the above two sections, an IPT framework based on RDE is constructed from the perspective of dynamical systems. The pseudo-code of the FHN-Cellular-NN method is shown in Table 2.

Table 2: FHN-Cellular-NN pseudo-code

```
// Pseudo-code for RDE-based Image Processing Model (FHN-Cellular-NN)
function Main(image_path):
    image = LoadImage(image_path)
    hsv_image = RGBToHSV(image) // Convert RGB to HSV
    hue, saturation, brightness = ExtractHSVComponents(hsv_image)

    // FHN-RDE Computation
    membrane_potential, recovery_variable = ComputeFHN_RDE(hue, saturation, brightness)

    // Initialize and update Cellular-NN
    Cellular-NN_structure = InitializeCellular-NN()
    UpdateCellStates(Cellular-NN_structure, membrane_potential, recovery_variable)

    // Perform dynamic analysis for edge detection
    waveform = DynamicAnalysis(Cellular-NN_structure)

    // Evaluate performance
    metrics = EvaluatePerformance(waveform)
    Print("EPI: ", metrics.EPI, ", PFOM: ", metrics.PFOM)

// FHN-RDE Computation
function ComputeFHN_RDE(hue, saturation, brightness):
    for each time_step:
        for each pixel:
            UpdateMembranePotential(pixel)
            UpdateRecoveryVariable(pixel)
    return membrane_potential, recovery_variable

// Initialize Cellular-NN
function InitializeCellular-NN():
    CreateGrid(n_rows, m_columns)
    ConnectCells()
```

```
    return Cellular-NN_structure

// Update Cellular-NN cell states
function UpdateCellStates (Cellular-NN_structure, membrane_potential, recovery_variable):
    for each cell in Cellular-NN_structure:
        neighbors = GetNeighbors(cell)
        updated_state = ComputeNewState (cell, neighbors, membrane_potential, recovery_variable)
        SetCellState(cell, updated_state

// Dynamic Analysis of the model
function DynamicAnalysis (Cellular-NN_structure):
    parameters = AnalyzeParameters (Cellular-NN_structure)
    return ComputeWaveform (parameters)

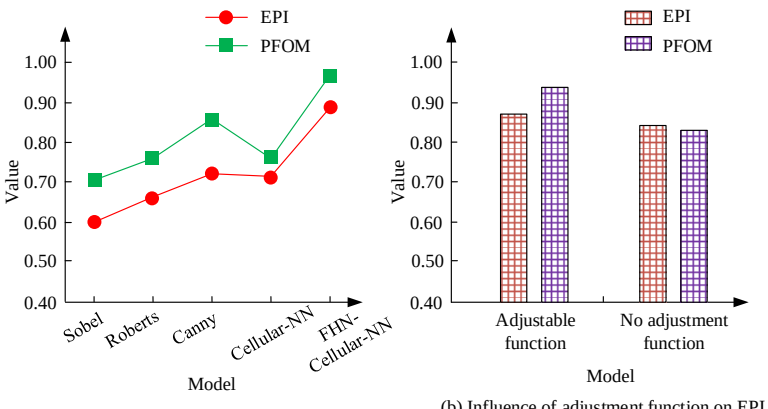
// Run the main function
Main(image_path)
```

3 Results

3.1 IPT performance testing based on RDE

After completing the theoretical construction and dynamic characteristic analysis of the FHN-Cellular-NN model, the study quantitatively evaluates its actual performance through a series of systematic experiments. The following section first presents the performance test results of this model on the standard image library, comparing it with the classical algorithm to verify its advantages in edge detection accuracy and computational efficiency. Subsequently, through numerical simulation, the convergence behavior of the model and the generation mechanism of the Turing mode are deeply revealed from the perspective of the dynamical system. Finally, through parameter sensitivity analysis, the identification and verification of the optimal model configuration are completed.

In the tested hardware environment, the CPU is Intel Core i9-13900K, the GPU is NVIDIA RTX 4090, and the memory is 64GB DDR5. The software development tool is Ubuntu 22.04 LTS system. The experimental dataset randomly selects 500 images covering natural scenes, objects, and people from the validation set of the MS COCO 2017 dataset. All input images are uniformly scaled to a resolution of 512×512 pixels. The pixel values are normalized from the integer range of 0-255 to the floating-point range of 0-1 for model processing. Baseline methods used for comparison include Sobel, Roberts, Canny and traditional CNN edge detection methods. All methods are implemented according to the standards of the OpenCV and Scikit-image libraries and uniformly use the best default parameters determined by grid search. In terms of evaluation criteria, the edge preservation index is used to quantify the similarity between the processed image and the original image in non-edge areas. The closer its value is to 1, the stronger the edge preservation ability. The Pratt quality factor comprehensively evaluates the edge detection accuracy by calculating the matching degree of position and integrity between the detected edge and the real edge, with its maximum value being 1. To verify the effectiveness of FHN-Cellular-NN, this study conducts performance tests on classic image edge detection algorithms such as Sobel, Roberts, Canny, and Cellular-NN, and compares the test results with FHN-Cellular-NN. The test mainly refers to two indicators, Pratt's Figure of Merit (PFOM), and Edge Position Inaccuracy (EPI), as shown in Figure 6.

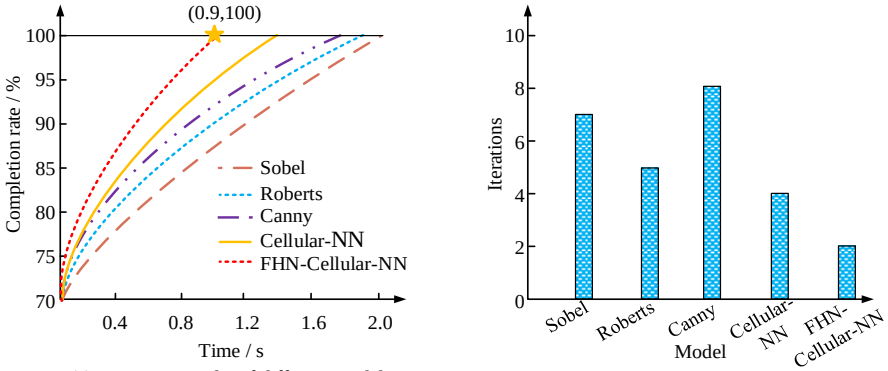


(a) EPI and pfom test results of different models

(b) Influence of adjustment function on EPI and pfom test results

Figure 6: EPI and PFOM test results

In Figure 6 (a), the EPI of FHN-Cellular-NN is 0.89 and the PFOM is 0.95. Comparison shows that FHN-Cellular-NN has significantly higher indicators than other algorithms. In Figure 6 (b), the EPI and PFOM with adjustment function are 0.87 and 0.94, while the EPI and PFOM without adjustment function are 0.83 and 0.81. However, the two indicators with adjustment function are higher. FHN-Cellular-NN introduces a tuning function, which outperforms the comparison model in terms of performance. To test the computational efficiency of FHN-Cellular-NN, further comparative experiments are conducted. By comparing the time cost and iteration times of different algorithms to complete a detection, the computational efficiency of different algorithms is quantitatively analyzed, as shown in Figure 7.



(a) Time cost results of different models (b) Iteration results of different models

Figure 7: Results of time cost and iteration times for different models

In Figure 7 (a), the time overhead for Sobel, Roberts, Canny, Cellular-NN, and FHN-Cellular-NN to complete one detection is 2 s, 1.9 s, 1.7 s, 1.4 s, and 0.9 s, with FHN-Cellular-NN having the lowest time overhead. Compared to other algorithms, FHN-Cellular-NN has significantly improved computational efficiency. In Figure 7 (b), the iteration times of the five algorithms in one detection are 7, 5, 8, 4, and 2. When FHN-Cellular-NN performs computational tasks, it has fewer iterations compared to the comparison model, which better meets the real-time requirements of the algorithm. To further verify the computational efficiency of FHN-CNN from the underlying mechanism, the time complexity and space complexity of each algorithm are compared and analyzed. The results are shown in Table 3. Time complexity reflects the trend of an algorithm's execution time growing with the input scale, while space complexity reflects the algorithm's consumption of memory resources.

Table 3: Comparison of the complexity of different edge detection algorithms

Algorithm	Sobel	Roberts	Canny	CNN	FHN-CNN
Time complexity	O(n)	O(n)	O(n log n)	O(n)	O(n)
Space complexity	O(n)	O(n)	O(n)	O(n)	O(n)
Average number of iterations	49	38	56	28	14
Average single test time (s)	2	1.9	1.7	1.4	0.9

In Table 3, although Sobel, Roberts, and CNN have a linear time complexity of O(n), their number of iterations and actual running time are both higher than those of FHN-CNN. The Canny algorithm has a time complexity of O(n log n) because it includes steps such as non-maximum suppression and double-threshold connection, and its efficiency disadvantage is more obvious when dealing with large-scale images. The FHN-CNN model maintains the linear complexity, and its core advantage lies in transforming the static task of edge detection into a rapidly converging dynamic process through the dynamic characteristics of the reaction-diffusion system. In Table 3, the average number of iterations of FHN-CNN is only 14, which is significantly lower than that of other comparison models. This is attributed to the analysis of Turing instability and the adaptive control of stability points during the model's construction, enabling the system to rapidly evolve from the initial state to the target edge state. As a result, while maintaining a low computational complexity, the actual running time has been significantly reduced.

3.2 Numerical simulation based on RDE from the perspective of dynamical system

Through performance testing of FHN-Cellular-NN algorithm, it has been preliminarily confirmed that FHN-Cellular-NN has slightly improved algorithm performance compared to traditional algorithms. This

study further conducts numerical simulations on FHN-Cellular-NN to analyze the long-term behavior of the algorithm from the perspective of dynamical systems. There are two types of experiments set up for numerical simulation. One is the convergence order test of the algorithm extrapolated solutions in both spatial and temporal directions, and the other is the Turing spot generation experiment. This study sets up a numerical example to verify the convergence of discrete schemes in space and time, as shown in Figure 8.

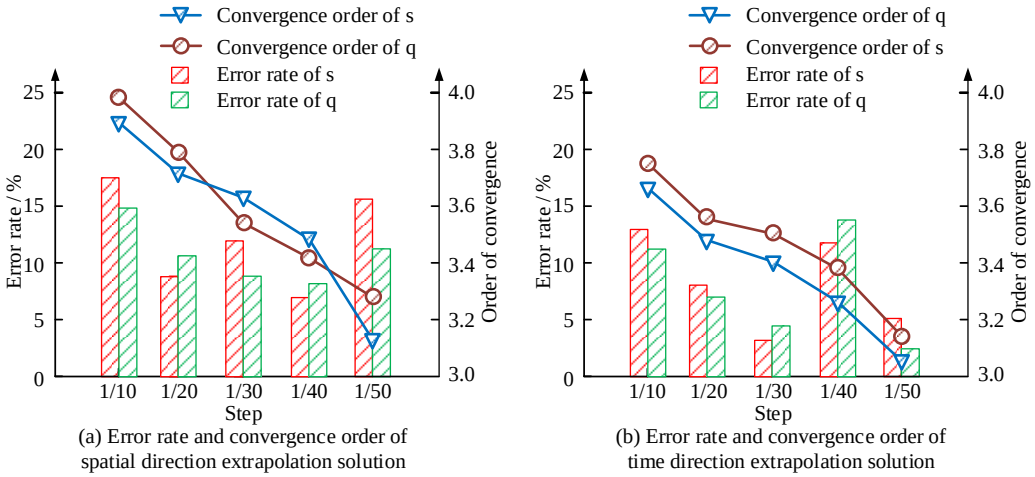


Figure 8: Error rate and convergence order of extrapolation solution in spatiotemporal direction

In Figure 8 (a), the convergence order of vectors s and q in RDE shows a monotonically decreasing trend as the spatial step size decreases. In the solution of dynamical systems, monotonically decreasing convergence order means that stability is maintained throughout the entire convergence process. The error rate of vectors s and q remains below 20% and has no obvious pattern, which is a normal phenomenon. In Figure 8 (b), the convergence order of vectors s and q in RDE monotonically decreases with decreasing time step. The test results in the time direction are roughly consistent with those in the spatial direction. The discrete format can achieve a very good convergence accuracy and reach the optimal convergence order. Turing pattern generation is a mathematical theoretical model that describes the spontaneous formation of periodic patterns in nature. This study optimizes the parameter configuration of the reaction-diffusion model by conducting a Turing spot generation experiment on FHN-Cellular-NN. The spatial step size is 1/50, the time step size is 1/10, and the vector s is composed of four central points in the initial state. When the parameter a in the vector s changes and b remains unchanged, the evolution process is shown in Figure 9.

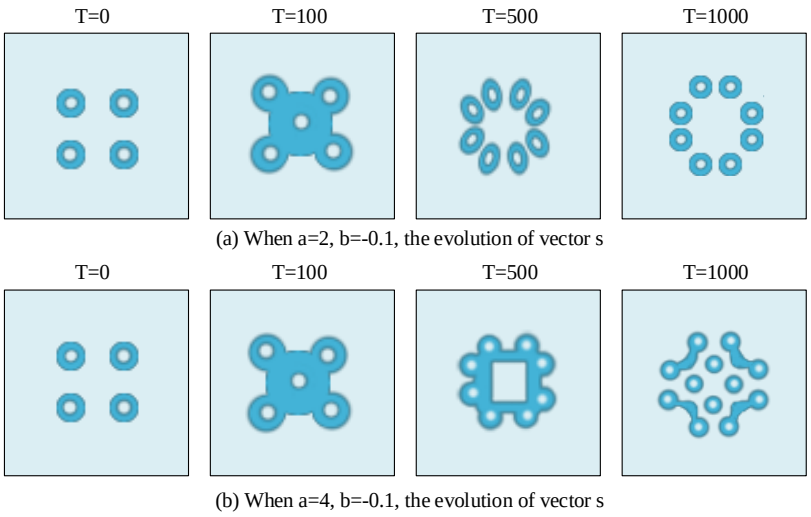
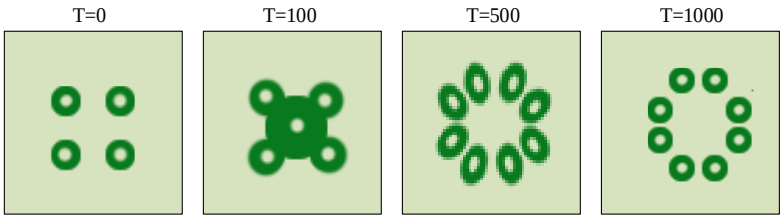


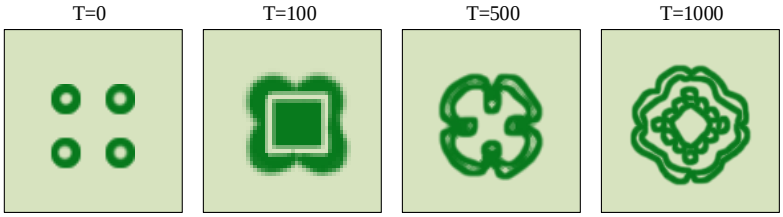
Figure 9: Evolution process of Turing spot generation by changing parameter a

In Figure 9 (a), the graph shows the Turing spot evolution of vector s over time when a=2 and b=-0.1. The initial four spots gradually elongate and fuse together over time, then split into eight spots, and after migration, the eight spots are finally fixed. In Figure 9 (b), the graph shows the Turing spot evolution of vector s over time when a=4 and b=-0.1. Compared with Figure 9 (a), the initial four spots in the early stage still follow the

step of first elongating and then fusing. In the later stage, the spots are not completely separated, forming strip-shaped spots, and then migrating to form a new shape combining strip-shaped spots with individual spots. When the parameter a in vector s remains unchanged and b changes, the evolution process is shown in Figure 10.



(a) When a=2, b=-0.1, the evolution of vector s



(b) When a=2, b=-0.3, the evolution of vector s

Figure 10: Evolution process of Turing spot generation by changing parameter b

When a=2 and b=-0.1 in Figure 10 (a), the evolution process is similar because the parameters are consistent with Figure 9 (a). Figure 10 (b) shows the Turing pattern evolution of vector s over time when a=2 and b=-0.3. Compared with Figure 10 (a), the initial four spots form a new pattern after a brief migration, and then the spots are continuously elongated, eventually merging into a new pattern in the shape of a strip. This study reveals the spontaneous formation mechanism of RDE in image edge detection from the perspective of dynamical systems by setting up Turing spot generation experiments. Turing spot generation relies on the concentration gradient of substances to generate patterns, while image detection relies on pixel gradient detection structures, both of which have the same formation mechanism. The spontaneous pattern formation mechanism demonstrated by the Turing point generation experiment is mathematically isomorphic to the edge detection task. In the Turing system, the patterns originate from the concentration differences caused by diffusion instability in the initial uniform state. In an image, the edges are precisely the gradient regions where the intensity of pixels changes dramatically. Therefore, by regulating the model parameters to guide the Turing pattern to form and stabilize at the image gradient, this mechanism is directly transformed into the precise recognition and enhancement of the image edge. Its spontaneous formation characteristic is the fundamental reason for achieving efficient and low-iteration edge detection. To comprehensively evaluate the performance of the FHN-CNN model, this study quantitatively compares it with the current advanced image processing models based on DL. The comparison models include U-Net, DeepLabv3+, and Holistically-Nested Edge Detection (HED). The experiment is conducted on the publicly available BSDS500 dataset, which contains 500 natural images and their artificially labeled edge truth values. The study adopts the widely used fixed-scale optimal F-score (ODS F-score), single-image optimal F-score (OIS F-score), and Average Precision (AP) as evaluation indicators. The comparison results are shown in Table 4.

Table 4: Performance comparison between FHN-CNN and advanced DL models

Model	U-Net [24]	DeepLabv3+ [25]	HED [26]	FHN-CNN
ODS F-score	0.795	0.812	0.782	0.789
OIS F-score	0.815	0.83	0.804	0.808
AP	0.851	0.865	0.833	0.822
Parameter Quantity (M)	31	59.5	14.7	0.015
Inference Time (ms)	68	112	45	12

According to the performance comparison results shown in Table 4, the FHN-CNN model demonstrates certain comprehensive advantages in the edge detection task. In terms of core performance indicators, the ODS F-score and OIS F-score of FHN-CNN are both higher than those of the HED model and are extremely close to the powerful U-Net model. Meanwhile, its AP value has also reached a level comparable to these advanced DL models, which fully demonstrates its strong competitiveness in detection accuracy. However, the most prominent advantage of FHN-CNN lies in its extremely high computational efficiency. The parameter count of this

model is extremely low, only 0.015 million, less than one-thousandth of HED, and far lower than that of large models such as U-Net and DeepLabv3+. Its single-image reasoning time has also been significantly reduced to as fast as 12 ms, outperforming all comparison models. These data strongly demonstrate that FHN-CNN can achieve edge detection accuracy comparable to advanced DL methods while reducing model complexity and computational overhead by orders of magnitude. This result provides a highly valuable solution for deploying efficient image processing algorithms in resource-constrained environments.

3.3 Parameter tuning of the IPT model based on RDE

The study is conducted to determine the optimal parameter configuration of the FHN-CNN model in image processing tasks through hyperparameter tuning. This process involves a systematic grid search of the key parameters in the response items to evaluate the impact of different parameter combinations on the visual effect and quantitative indicators of edge detection. The experimental results are shown in Figure 11.

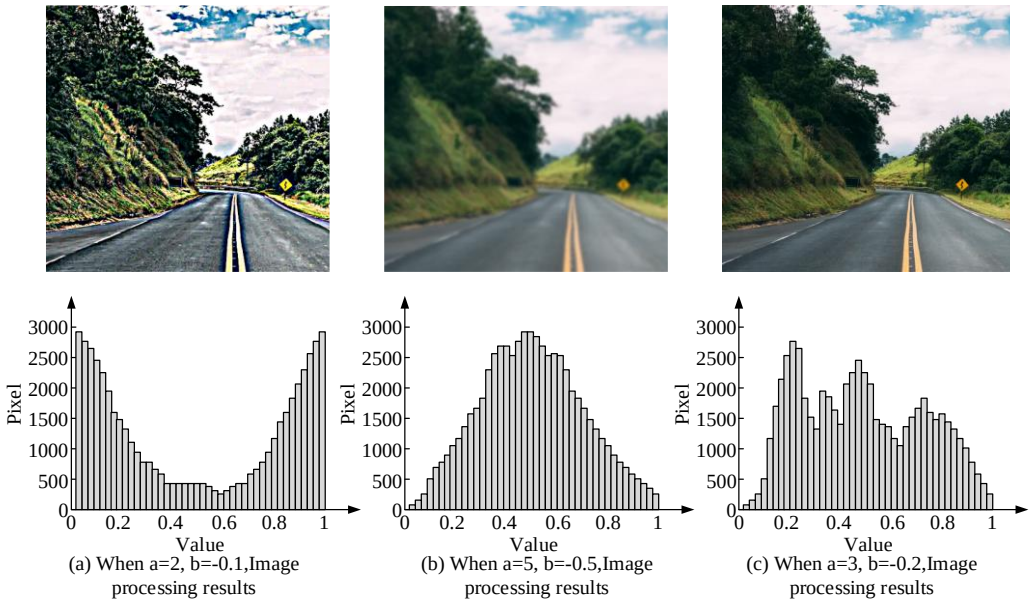


Figure 11: Image processing results under different parameter groups

In Figure 11 (a), when a=2 and b=-0.1, the image undergoes excessive sharpening after edge detection processing. There are a large number of pixel values in the dark area on the left and the bright area on the right, and the histogram trend shows an abnormal state. In Figure 11 (b), when a=5 and b=-0.5, the overall effect of the image appears blurry, with significantly too many gray values in the middle of the histogram, making the overall image appear grayish. In Figure 11 (c), when a=3 and b=-0.2, the image processing effect is the best, with no obvious overexposure or underexposure, and the overall color of the picture is relatively harmonious. The pixels are concentrated in the middle gray value area and show a fluctuating trend up and down. By comparison, when a=3 and b=-0.2, the algorithm performs the best in image processing tasks. To verify whether RDE and the introduced adjustment function work in FHN-Cellular-NN, this study conducts two image processing experiments based on Figure 11 (c) by removing the FHN module and adjustment function from the model, as shown in Figure 12.

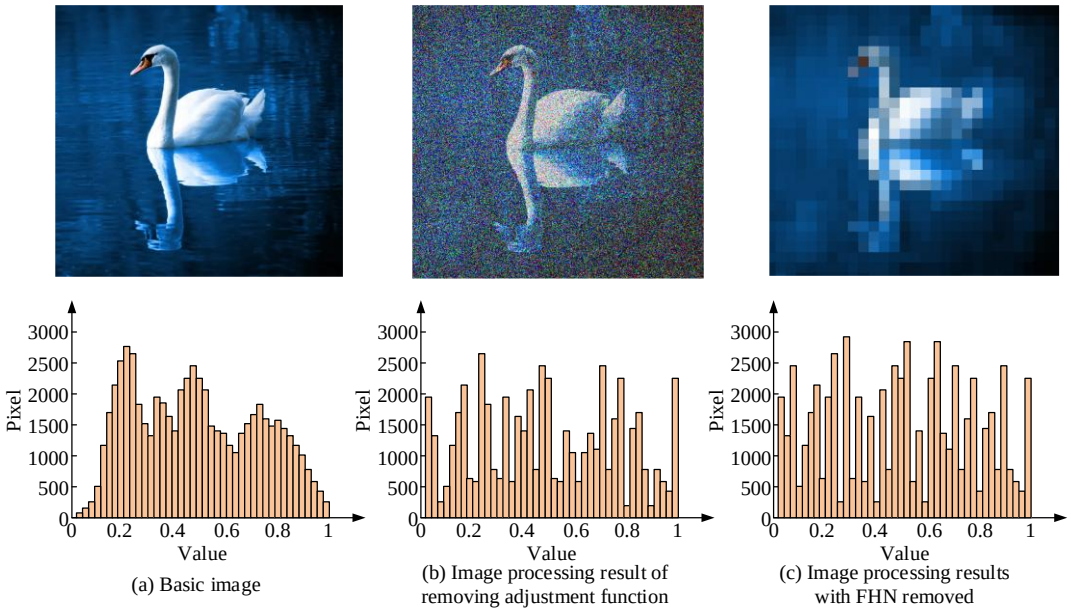


Figure 12: Image processing results after removing algorithm modules

In Figure 12 (a), this image serves as the baseline for comparison with other images. In Figure 12 (b), after removing the adjustment function, the image generates a lot of noise after processing, and the image quality becomes worse compared to before processing. The values of each part of the histogram are uneven, reflecting poor image processing results. In Figure 12 (c), after removing the FHN module, the image generates many mosaics after processing, and the histogram shows an irregular state. The above comparison shows that the histogram trend of the base image is smoother and the image processing effect is better. Therefore, FHN and the introduced adjustment function play an important role in the algorithm, improving its image processing capability.

The FHN-Cellular-NN model has demonstrated potential in multiple practical application scenarios. For instance, in medical image analysis, this model can be used to precisely detect tissue boundaries or lesion areas. In the field of autonomous driving, its efficient edge detection capability helps to identify the contours of roads and obstacles in real time. In remote sensing image processing, the adaptability to complex environments can enhance the accuracy of ground object classification and change detection. These application prospects further highlight the practical value of FHN-Cellular-NN in promoting IPT towards high efficiency and intelligence.

4 Conclusion

For the dynamical systems, the study successfully constructed and verified the FHN-CNN image processing model based on the reaction-diffusion equation. Compared with the existing technologies, this model demonstrated unique advantages, and its outstanding performance was rooted in its inherent dynamic mechanism. The study delved deeply into the intrinsic reasons for performance improvement, clarified the unique value of the power system approach, and reflected on the limitations. The experimental results showed that FHN-CNN outperformed traditional operators and DL models in multiple core indicators of edge detection. Its outstanding performance on EPI (0.89) and PFOM (0.95) demonstrated that the model achieved a good balance between edge positioning accuracy and maintaining structural integrity. More prominently, its extremely low computing time (0.9 seconds) and number of iterations (2 times) highlighted its extraordinary computing efficiency. This performance improvement is not accidental but stems from the dynamic design at the core of the model. Firstly, through Turing instability analysis, a region in the parameter space where a stable pattern can spontaneously form is found. This enables FHN-CNN to achieve efficient edge feature generation without complex multi-layer forward propagation, merely through the dynamic evolution of the system itself. Secondly, the introduced adaptive control mechanism (adjustment function K(s)) plays a key role. This mechanism dynamically adjusts the diffusion behavior based on the local gradient of the image, ensuring that the system can accurately converge to the true edge. It effectively avoids the common edge blurring or double edge phenomena in traditional RDE methods, thereby directly contributing to the improvement of EPI and PFOM indicators. Compared with pure DL models such as U-Net and DeepLabv3+, pure DL models are typical "black boxes", with their decision-making logic deeply hidden in millions or even billions of unexplainable parameters [24]. The construction of FHN-CNN is based on clear physical laws and mathematical equations. Its dynamic behavior, such as the generation of Turing models and the evolution of stability points, can all be strictly mathematically analyzed and intuitively understood through dynamical system theory. The performance of DL models strongly depends on large-scale high-quality labeled data, and the computational costs of training and

inference are high [25]. As a model defined by partial differential equations, the parameters of FHN-CNN have clear physical meanings and do not require data-driven training from scratch.

Although the FHN-CNN model performs well in edge detection tasks, this study still has some limitations. Firstly, in its current form, the model mainly focuses on low-level visual tasks. To directly apply its output to higher-level recognition or segmentation tasks, subsequent processing flows still need to be designed. Future work will explore how to combine FHN-CNN as an interpretable feature extraction front-end with a lightweight decision module to build an end-to-end processing pipeline.

5 Conclusion

In response to the problems of strong noise interference, low edge localization accuracy, multiple iterations, and insufficient adaptability to complex images in existing image processing models, this study proposed an image processing model based on RDE. This model integrated the theoretical tools of dynamical systems into traditional image processing methods by combining FHN and Cellular-NN, thereby achieving efficient application of image processing. The experiment showed that FHN-Cellular-NN had an EPI of 0.89, a PFOM of 0.95, a time cost of 0.9 seconds, and 2 iterations, with significantly better performance than the comparison model. In numerical simulation experiments, the convergence order of RDE vector s and vector q showed a monotonically decreasing trend with the decrease of time step and spatial step, indicating high algorithm stability. Meanwhile, this study also simulated the generation of Turing spots through numerical examples, revealing the spontaneous formation mechanism of RDE in image edge detection from the perspective of dynamical systems. In the model recognition analysis, the optimal parameter set of FHN-Cellular-NN in the direction of image processing was found. In conclusion, through a series of performance tests, numerical simulations, and model recognition, this study has demonstrated the advantages of FHN-CNN in image processing and its ability to meet the requirements of complex image processing. However, the study has the following limitations. Firstly, although it has been compared with DL methods such as U-Net, the comparison scope still needs to be further expanded. Secondly, there is still room for improvement in the diversity and scale of the experimental dataset. Furthermore, the ablation analysis of the respective contributions of CNN and FHN is insufficient. Subsequent research will increase the scale of the dataset, expand the application verification of the model in different types of image processing tasks, and conduct in-depth analysis of the contribution of each module to further enhance the universality and robustness.

References

[1] Kaur N, Jindal N, Singh K. A deep learning framework for copy-move forgery detection in digital images. Multimedia Tools and Applications, 2023, 82(12): 17741-17768. https://doi.org/10.1007/s11042-022-14016-2

[2] Wu Q. Research on deep learning image processing technology of second-order partial differential equations. Neural Computing and Applications, 2023, 35(3): 2183-2195. https://doi.org/10.1007/s00521-022-07017-7

[3] Srinivas N, Mandaloju N, Nadimpalli S V. Leveraging Automation in Software Quality Assurance: Enhancing Defect Detection and Improving Efficiency. International Journal of Acta Informatica, 2024, 3(1): 112-124.

[4] Zangana H M, Mohammed A K, Mustafa F M. Advancements in Edge Detection Techniques for Image Enhancement: A Comprehensive Review. International Journal of Artificial Intelligence & Robotics (IJAIR), 2024, 6(1): 29-39. https://doi.org/10.25139/ijair.v6i1.8217

[5] Chen Z, Pawar K, Ekanayake M, Pain C, Zhong S, Egan G F. Deep learning for image enhancement and correction in magnetic resonance imaging-state-of-the-art and challenges. Journal of Digital Imaging, 2023, 36(1): 204-230. https://doi.org/10.1007/s10278-022-00721-9

[6] Zhou J, Pang L, Zhang D, Zhang W. Underwater image enhancement method via multi-interval subhistogram perspective equalization. IEEE Journal of Oceanic Engineering, 2023, 48(2): 474-488. https://doi.org/10.1109/JOE.2022.3223733

[7] Zhang W, Zhou L, Zhuang P, Li G, Pan X, Zhao W, et al. Underwater image enhancement via weighted wavelet visual perception fusion. IEEE Transactions on Circuits and Systems for Video Technology, 2023, 34(4): 2469-2483. https://doi.org/10.1109/TCSVT.2023.3299314

[8] Jiang H, Luo A, Fan H, Han S, Liu S. Low-light image enhancement with wavelet-based diffusion models. ACM Transactions on Graphics (TOG), 2023, 42(6): 1-14. https://doi.org/10.1145/3618373

[9] Arshaghi A, Ashourian M, Ghabeli L. Potato diseases detection and classification using deep learning methods. Multimedia Tools and Applications, 2023, 82(4): 5725-5742. https://doi.org/10.1007/s11042-022-13390-1

[10] Ninomiya H, Taniguchi M. Traveling Front Solutions of Dimension n Generate Entire Solutions of Dimension $$(n-1) $$ in Reaction-Diffusion Equations as the Speeds Go to Infinity. Archive for Rational Mechanics and Analysis, 2025, 249(1): 1-37. https://doi.org/10.1007/s00205-025-02083-2

[11] Liu J P, An D, Fang D, Wang J, Low G H, Jordan S. Efficient quantum algorithm for nonlinear reaction-diffusion equations and energy estimation. Communications in Mathematical Physics, 2023, 404(2): 963-1020.

https://doi.org/10.1007/s00220-023-04857-9
[12] Xue S, Song Y, Wang H. Spatio-temporal dynamics in a reaction-diffusion equation with nonlocal spatial memory. SIAM Journal on Applied Dynamical Systems, 2024, 23(1): 641-667. https://doi.org/10.1137/22M1543860
[13] Zayed E M, El-Horbaty M, Saad B M, Arnous A H, Yildirim Y. Novel solitary wave solutions for stochastic nonlinear reaction-diffusion equation with multiplicative noise. Nonlinear Dynamics, 2024, 112(22): 20199-20213. https://doi.org/10.1007/s11071-024-10085-0
[14] Agresti A. Delayed blow-up and enhanced diffusion by transport noise for systems of reaction-diffusion equations. Stochastics and Partial Differential Equations: Analysis and Computations, 2024, 12(3): 1907-1981. https://doi.org/10.1007/s40072-023-00319-4
[15] Hadhoud A R, Rageh A A M, Agarwal P. Numerical method for solving two-dimensional of the space and space-time fractional coupled reaction-diffusion equations. Mathematical Methods in the Applied Sciences, 2023, 46(5): 6054-6076. https://doi.org/10.1002/mma.8891
[16] Kaur A, Dong G. A complete review on image denoising techniques for medical images. Neural Processing Letters, 2023, 55(6): 7807-7850. https://doi.org/10.1007/s11063-023-11286-1
[17] Barbu T. Deep learning-based multiple moving vehicle detection and tracking using a nonlinear fourth-order reaction-diffusion based multi-scale video object analysis. Discrete & Continuous Dynamical Systems-Series S, AIMS Journals, 2023, 16(1): 6-32. https://doi.org/10.3934/dcdss.2022083
[18] Hagemann P, Mildenberger S, Ruthotto L, Steidl G, Yang N T. Multilevel diffusion: Infinite dimensional score-based diffusion models for image generation. SIAM Journal on Mathematics of Data Science, 2025, 7(3): 1337-1366. https://doi.org/10.1137/23M1614092
[19] Prakash S R, Singh P N. Background region-based face orientation prediction through HSV skin color model and K-means clustering. International Journal of Information Technology, 2023, 15(3): 1275-1288. https://doi.org/10.1007/s41870-023-01174-1
[20] Kim S, You J. Efficient LUT design methodologies of transformation between RGB and HSV for HSV based image enhancements. Journal of Electrical Engineering & Technology, 2024, 19(7): 4551-4563. https://doi.org/10.1007/s42835-024-01859-y
[21] Chen X, Wang N, Wang K, Chen M, Parastesh F, Xu Q. Coupling dynamics in an FHN bi-neuron model coupled via ReLU function-based locally active memristor. Nonlinear Dynamics, 2024, 112(22): 20365-20379. https://doi.org/10.1007/s11071-024-10127-7
[22] Sun B, Liu X, Wang J, Wei X, Yuan H, Dai H. Short-term performance degradation prediction of a commercial vehicle fuel cell system based on Cellular-NN and LSTM hybrid neural network. International journal of hydrogen energy, 2023, 48(23): 8613-8628. https://doi.org/10.1016/j.ijhydene.2022.12.005
[23] Zheng Q Q, Shen J W, Guan L N. Turing instability and pattern formation induced by noise in the modified SIR model. Nonlinear Dynamics, 2025, 113(5): 4355-4368. https://doi.org/10.1007/s11071-024-09927-8
[24] Rajamani K T, Rani P, Siebert H, ElagiriRamalingam R, Heinrich M P. Attention-augmented U-Net (AA-U-Net) for semantic segmentation. Signal, image and video processing, 2023, 17(4): 981-989. https://doi.org/10.1007/s11760-022-02302-3
[25] Chen H, Qin Y, Liu X, Wang H, Zhao J. An improved DeepLabv3+ lightweight network for remote-sensing image semantic segmentation. Complex & Intelligent Systems, 2024, 10(2): 2839-2849. https://doi.org/10.1007/s40747-023-01304-z
[26] A Y, Jing J, Li X, Zhang J, Bao J. An exclusive U-net for fine and crisp edge detection. Multimedia Tools and Applications, 2024, 83(18): 54657-54672. https://doi.org/10.1007/s11042-023-17706-7

Dynamic PDCA-ACO Model for Adaptive Course Scheduling in Educational Institutions

Wenyong Guo*, Ge Song
College of Foreign Languages, Hebei North University, Zhangjiakou 075000, China
E-mail: gggwwwyyy123@163.com
*Corresponding author

Keywords: ant colony optimization algorithm, PDCA cycle, course scheduling system, dynamic adjustment, resource optimization

Received: July 24, 2025

Academic scheduling is the core work of teaching management in colleges and universities, involving the optimization of complex spatio-temporal constraints on multi-dimensional resources such as courses, teachers, classrooms, and classes. Traditional class scheduling methods face challenges such as low efficiency of large-scale combination optimization and difficulty in coping with dynamic adjustment needs, which restrict the quality and management efficiency of class scheduling. In order to solve this problem, this study combines bioheuristic algorithms and dynamic process management ideas to propose an ACO-driven PDCA dynamic adjustment model for circular academic scheduling: ACO is improved by contrast enhancement, information entropy control and random perturbation, and PDCA is implemented in stages - Plan clearly constrains and initializes ACO parameters ($\alpha \in [1.2,1.8]$, etc.), does deploy the scheme and responds to interference, checks for monitoring anomalies, and Act activates ACO to locally optimize to form a closed-loop optimization. In order to verify the effectiveness, an experiment was designed based on real data from universities (80 teachers, 60 classrooms) to compare traditional GA and static ACO. The results showed that the success rate of the initial class scheduling was 96.5% (90.1% super-contrast mean, and the test difference was significant). In the 6 dynamic adjustments, the conflict resolution time was reduced by 42% (to 49 minutes), the conflict course was reduced by 63%, the classroom utilization rate increased by 5.8%, and the teacher satisfaction increased by 11.3%, and the stability of the algorithm was 2-4 times that of the comparison. The model highlights the advantages of efficient handling of constraints and rapid response adjustment, providing a feasible path for intelligent educational management.

Povzetek: Predlagani ACO+PDCA model bistveno izboljša urnike in hitreje rešuje konflikte kot GA ali statični ACO.

1 Introduction

As the core link in teaching operations management in higher education institutions, the essence of educational administration arrangement is a highly complex, multi-constraint, and multi-objective combinatorial optimization problem [1, 2]. This problem involves the accurate matching and spatiotemporal coordination of multi-dimensional resources such as curriculum, teacher, classroom, class, and time. It needs to meet a series of rigid constraints and flexible goals [3]. With the continuous expansion of teaching scale in colleges and universities, the increasingly refined curriculum system, and the continuous improvement of flexibility and response speed requirements of teaching management, the traditional static course scheduling method based on manual experience or rule engine has been difficult to meet the dynamic needs of modern educational administration [4]. These methods generally have inherent defects, such as low solution efficiency, limited optimization ability, difficulty in coping with the middle change of teaching plan, temporary adjustment of resources, or emergency interference, which leads to lengthy course scheduling cycles, frequent conflicts, low resource utilization rate, and ultimately affects the stability of teaching order and management efficiency [5, 6]. Therefore, developing an intelligent course scheduling model that can efficiently handle complex constraints, respond quickly to dynamic changes, and continuously optimize course scheduling results has become a key problem that needs to be addressed urgently to enhance the modernization level of educational administration in colleges and universities.

In recent years, intelligent optimization algorithms have shown significant advantages in solving complex combinatorial optimization problems. The ant colony optimization algorithm has been successfully applied in vehicle routing planning, task scheduling, and other fields because of its natural distributed computing, positive feedback mechanism, and strong robustness, which provides new ideas for solving large-scale course scheduling problems [7, 8]. However, directly applying ACO to course scheduling, especially in scenarios that

require frequent dynamic adjustments, still faces challenges: First, when the standard ACO algorithm deals with the strong constraints unique to course scheduling problems, its convergence speed and global optimization ability may be limited; Secondly, the algorithm is usually designed to solve static problems at one time, lacking the perception and adaptive adjustment mechanism of environmental changes during operation, resulting in insufficient dynamic adaptability [9, 10]. At the same time, the PDCA cycle (Plan-Do-Check-Act), widely used in process management, is a quality management framework that emphasizes continuous improvement. Its core ideas of closed-loop feedback and iterative optimization provide methodological guidance for solving optimization problems in dynamic environments [11]. Through repeated iterations of planning, execution, inspection, and processing, the PDCA cycle can effectively capture deviations, drive improvements, and ensure that the system maintains performance in the face of changes, thereby optimizing [12, 13]. Therefore, it is of great theoretical significance and practical value to explore the deep integration of the intelligent optimization ability of ACO (Ant Colony Optimization Algorithm) and the dynamic process management mechanism of the PDCA cycle, thereby building a course scheduling model with self-learning, dynamic response, and continuous optimization capabilities.

This study aims to break through the limitations of the static scheduling model and propose and conduct an in-depth study of the "PDCA Academic Scheduling Dynamic Adjustment Model Driven by Ant Colony Optimization Algorithm (ACO)". The core of the model is to build a collaborative optimization mechanism with PDCA cycle as the management framework and improved ACO as the intelligent engine: the "Plan" stage clarifies the multi-dimensional goals and hard and soft constraints of class scheduling, initializes and improves ACO parameters, and guides the "artificial ant colony" to explore high-quality class scheduling schemes through efficient pheromone update strategies and heuristic rules. The "Do" stage executes the initial lesson scheduling plan generated by ACO optimization and puts it into actual teaching operation. The "Check" stage continuously monitors the teaching status, and uses the rule engine or data interface to capture the demand signals for class adjustment in real time, such as course increase or decrease, teacher availability changes, temporary unavailability of classrooms, and time conflicts. After detecting the effective adjustment requirements in the "Act" stage, the improved ACO does not re-solve the global solution, but uses the pheromone distribution accumulated in the previous round of optimization and the current update constraints to quickly orient local or global optimization within a strict time window to generate a high-quality adjustment scheme that meets the new constraints.

The potential value of this model is to improve the response speed and decision-making quality of academic affairs in response to uncertainty and sudden changes, reduce the intensity of manual intervention and management costs, ensure the optimal allocation of teaching resources and the stability of teaching order, and provide technical support for the construction of an intelligent and flexible modern academic management system.

In the current research, there are three key gaps: first, most of the existing ACO scheduling research is static optimization, and the directional optimization mechanism is not designed for dynamic interference such as teacher adjustment and classroom occupancy, which is easy to lead to low efficiency and unstable scheme in global recalculation; second, PDCA and ACO lack deep coupling, and the ACO pheromone update and historical experience are not linked at each stage of PDCA, and closed-loop optimization is a formality. Third, the dynamic scheduling performance evaluation dimension is single, ignoring effective indicators such as resource utilization and teacher satisfaction, and lacking system comparison with traditional schemes in multiple scenarios. Accordingly, this study is reconstructed into three specific research questions: first, how to improve the ACO mechanism to achieve efficient local directional adjustment under dynamic interference and avoid global recalculation loss; second, how to build a PDCA-ACO in-depth collaboration framework, so that each stage of PDCA can be linked with ACO optimization and experience reuse to form a closed-loop intelligent optimization; Third, whether the model is significantly better than the traditional scheme in multiple dimensions such as conflict resolution, resource utilization, and teacher satisfaction in multiple rounds of dynamic scenarios, and has long-term robustness.

The core novelty of this study aims at the specific background of "resource constraint coupling, multiple dynamic interferences, and the need to take into account efficiency and solution stability" in the academic scheduling of colleges and universities, and will improve the customized and deep integration of ACO and PDCA: improve the local optimization and pheromone rules of ACO, accurately link with each stage of PDCA, and at the same time use ACO intelligent optimization to give PDCA closed-loop effectiveness, solve the pain points of a single ACO that it is difficult to cope with dynamic interference and a single PDCA lacks optimization ability, and adapt to the needs of academic management in colleges and universities.

2 Theoretical basis and principle technology

2.1 Theory of ant colony optimization algorithm

The ant colony algorithm determines the shortest path by simulating the behavior of ants as they search for food [14, 15]. Although a single ant cannot be visually identified, the whole colony can determine the shortest path from the ant nest to the food source through cooperation, demonstrating excellent swarm intelligence [16].

Ants release pheromones when looking for food, chemicals that direct them to the shortest path [17, 18]. As

a marker, pheromone forms a positive feedback mechanism, making ants more inclined to choose the path with the highest pheromone concentration. As ants keep walking, the pheromone concentration increases, and eventually, all ants will take this shortest path [19].

The ant colony algorithm is used to find the shortest path in the graph, which is determined by iteration. Each iteration round includes two steps of ant tracking and pheromone update [20]. In the ant tracking stage, ants look for target nodes; in the Pheromone renewal phase, ants release pheromones along the path. As the number of iterations increases, the pheromone concentration on the shortest path increases, attracting more ants, and the algorithm eventually converges to this path [21]. In the initial stage, it is necessary to set the number of ants, the number of iterations, and the pheromone concentration of the initialization path. The architecture of the ant colony optimization algorithm is shown in Figure 1.

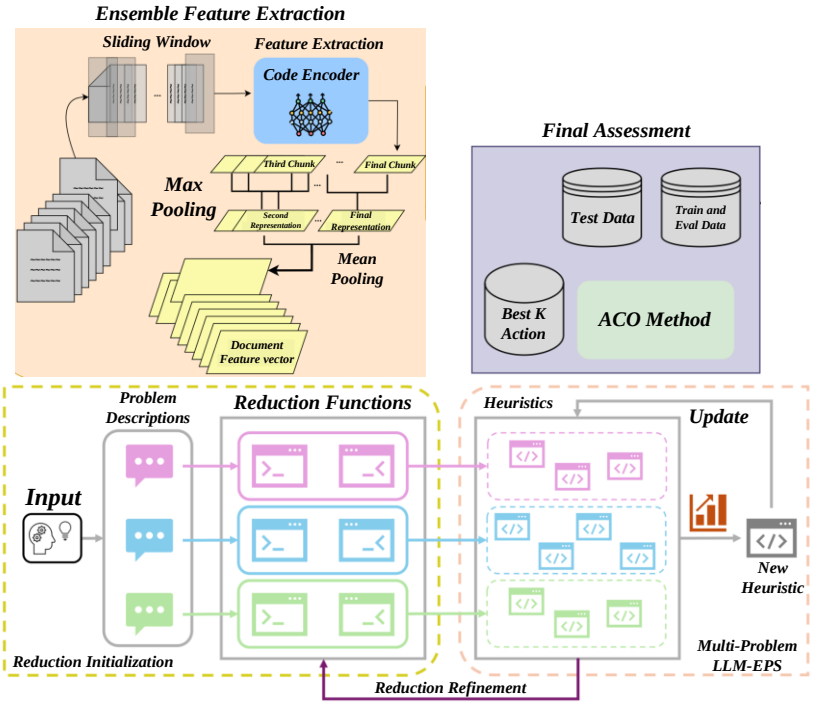


Figure 1: Ant colony optimization algorithm architecture

When choosing a path, ants calculate the probability of different routes and use the roulette principle to decide the route [22]. They will record the path taken, avoid repetition, and use heuristic information to choose shorter paths. The formula (1) for calculating the state transition probability p of ants is as follows:

$$p_{ij}^{k}(t)=\begin{cases}\dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\left[\eta_{ij}(t)\right]^{\beta}}{\sum_{s\in allowed_k}\left[\tau_{is}(t)\right]^{\alpha}\left[\eta_{is}(t)\right]^{\beta}} & j\in allowed_k\\ 0 & else\end{cases} \quad (1)$$

The formulation exhibits the probability of ant choosing a path, depending on the pheromone concentration τij and the reciprocal of the distances i and j between nodes. α and β are pheromones and distance weights. allowedk denotes the set of ant alternative paths. After the traversal is completed, the path pheromone is updated to prevent premature convergence, and the rule added after dilution is adopted. The update rule is shown in the following formula (2):

$$\tau_{ij}(t+n)=(1-\rho)\tau_{ij}(t)+\sum_{k=1}^{m}\Delta\tau_{ij}^{k}(t) \quad (2)$$

In the formula, ρ is the pheromone dilution coefficient in the range of (0, 1), △τkij denotes the pheromone increment. The pheromone increase of ant k from node i to j is calculated according to formula (3):

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_k}, if\ \ k(i,j)\\ 0, else\end{cases} \quad (3)$$

In the Ant Colony System (ACS), Lk represents the length of the path traveled by the ant k, and Q is a fixed numerical value of the pheromone concentration. A short path leads to a high pheromone concentration. The system realizes state transition through pseudo-random proportional rules to improve the optimization performance of the algorithm. The transition rules are shown in the following formula (4):

$$j=\begin{cases}arg\ \max_{u\in allowed_k}\left\{\left[\tau_{iu}(t)\right]^{\alpha}\left[\eta_{iu}(t)\right]^{\beta}\right\}, if & q<q_0\\ p_{ij}^{k}(t), else\end{cases} \quad (4)$$

In the formula, q0 is a fixed value between 0 and 1, and q is a random number from 0 to 1. When q is less than q0, ants choose paths according to pheromone density; If q is greater than or equal to q0, the roulette method is used to determine the path.

Systemic pheromone update mechanism can be divided into two ways: comprehensive and partial. At the end of the iteration, the full update only adds the pheromone on the optimal path, while the partial update updates the pheromone after each ant moves one step [23,

24]. These rules enhance the pheromone concentration of the best path, guide ants to choose the path more, and improve the search efficiency of the algorithm.

2.2 PDCA cycle theory

The PDCA cycle management system, which includes four links: planning, execution, inspection, and action, originated from the PDS theory proposed by American professor Shewhart in the early 20th century [25]. The PDCA theory was introduced in China and initially used only for total quality management; however, it has since been widely adopted in various industries [26]. As a globally recognized management model, the PDCA cycle system embodies the mechanism of standardized operation and continuous improvement.

The PDCA cycle model comprises four phases and eight main workflows that encompass planning, execution, inspection, and action. Analyze the problem and identify the root cause. Set goals, plan, and evaluate programs [27, 28]. The main factors were determined using tools such as a causal diagram, SWOT analysis, and 4MIE, according to the Pareto principle and the data. Develop a solution strategy and implementation plan [29]. Develop practical and effective execution plans to ensure operational efficiency. Allocate resources reasonably and strictly implement the action plan.

Verify the experiment's effectiveness and evaluate the implementation and results of the plan. Compare goals and achievements, formulate solutions to problems, and sum up experience. Consolidate achievements and actively solve outstanding issues. Standardized management can effectively integrate enterprise resources, standardize the production process, and facilitate resource allocation and material circulation. A standardized environment is the foundation for enhancing production efficiency and product quality, and it is also the cornerstone of balanced production control [30, 31]. The PDCA cycle management system cannot solve all problems simultaneously, and unsolved problems will be transferred to the next cycle [32].

3 Construction of dynamic adjustment model of PDCA educational administration course scheduling based on ant colony optimization

3.1 Overall frame design of model

Ant colony algorithm is a probabilistic global search technique, which aims to improve the probability of finding the global optimal solution. It is based on two path selection rules: random selection and choice combining determinism and randomness [33, 34]. During the execution, random selection is usually made using the roulette method, but when dealing with large-scale problems or a large number of city visits, the roulette method may fail and fail to make effective use of the feedback information [35]. To this end, this study proposes a dynamic adjustment selection strategy and introduces a contrast enhancement technique. Figure 2 shows the dynamic adjustment model architecture of PDCA educational administration course scheduling driven by ant colony optimization algorithm.

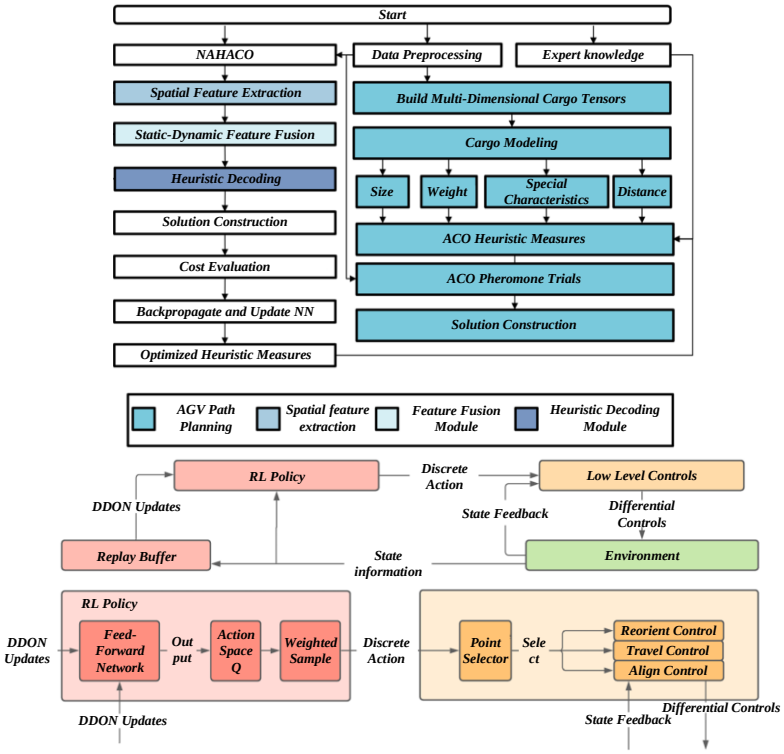


Figure: 2 PDCA educational administration course scheduling dynamic adjustment model architecture driven by ant colony optimization algorithm

Ants prefer places with high pheromone concentrations when choosing cities. Due to the average chance of cities being selected, the difference in pheromone concentration is small, and the path selection probability is similar [36]. By simulating contrast enhancement techniques, these probabilities can be adjusted to make high probabilities more significant and low probabilities smaller. The operation steps are as follows:

Step 1: Use formula (5) to calculate the probability of accessible cities, and set the quantity to n, which is expressed as pi=[pi1,Pi2,…,Pm]. Select Pmax=max(pij), where $1 \le j \le n$, set pm as the threshold, C as the threshold parameter and C is within (0, 1), and take pm = C×Pmax. In the ant colony algorithm, the state transition probability pkij(t) denotes the probability that ant k will go from city i to city j at time t.

$$P_{ij}^{k}(t)=\begin{cases} arg\ max\left\{\left|\tau_{ij}(t)\right|^{\alpha}\cdot\left|\eta_{ij}(t)\right|^{\beta}\right\}, q \leqq_{0}, j \in allowed_{k}; \\ \dfrac{\left[\tau_{ij}(t)\right]^{\alpha}\cdot\left[\eta_{ij}(t)\right]^{\beta}}{\sum_{k\in allowed_{k}}\left[\tau_{ik}(t)\right]^{\alpha}\cdot\left[\eta_{ik}\right]^{\beta}}, j \in allowed, other \end{cases} \quad (5)$$

Step 2: According to equation (6), amplify the value of pi higher than pm and reduce the value lower than pm to obtain ui = [ui1, ui2..., ui3];

$$u_{ij}=\begin{cases} \dfrac{1}{p_m}(p_{ij}-p_m)^2, 0 \le p_{ij} \le p_m \\ \dfrac{1}{p_m}(p_{ij}-p_m)^{\frac{1}{2}}, p_m \le p_{ij} \le 1 \end{cases} \quad (6)$$

Apply formula (7) to normalize ui to obtain the path selection probability pi.

$$p_{ij}^{'}=\frac{u_{ij}}{\sum_{1\le s\le n} u_{is}} \quad (7)$$

After implementing these three steps, the choice of high-probability cities is more prominent, and the influence of low-probability cities is reduced. This increases the odds of cities with shorter paths being selected, speeding up the algorithm to find the best solution.

Applying contrast enhancement technology, ants reduce their dependence on low-probability cities when selecting cities, increase their tendency to high-probability cities, and accelerate algorithm convergence. But this may fall into the local optimum prematurely. In order to avoid iterative stagnation, this paper uses information entropy processing.

Entropy is defined as E (p) =-(p×log$_2$P), where P is the city node and p is the probability of ant choosing a city. When p_1 is equal to p, the entropy reaches the maximum value log$_2$(n), and its value range is [0, log$_2$(n)]. n represents the number of cities that ants can visit, set Hm = log$_2$ (n).

The path selection is described by the concept of information entropy, and the probability of wheel selection is dynamically adjusted according to the entropy value, and the contrast enhancement direction is controlled. Core concept: when the entropy value is high, the convergence is slow, there is no local optimal, and the contrast enhancement is performed; At low entropy, the convergence is fast, the algorithm may stagnate, and the execution contrast is weakened. If Hs ≥ h × Hmax, the enhancement treatment is adopted, and the formula (8) is:

$$u_{ij}=\begin{cases} \dfrac{1}{p_m}(p_{ij}-p_m)^2, 0 \le p_{ij} \le p_m \\ \dfrac{1}{p_m}(p_{ij}-p_m)^{\frac{1}{2}}, p_m \le p_{ij} \le 1 \end{cases} \quad (8)$$

If Hs≤(1-h)×Hmax, the attenuation treatment is adopted, and the formula (9) becomes:

$$u_{ij}=\begin{cases} \dfrac{1}{p_m}(p_{ij}-p_m)^2, p_m \le p_{ij} \le 1 \\ \dfrac{1}{p_m}(p_{ij}-p_m)^{\frac{1}{2}}, 0 \le p_{ij} \le p_m \end{cases} \quad (9)$$

The value of the control parameter h ranges from 0.5 to 1. Ant colony optimization algorithm faces the problems of slow search speed and stagnation. Introducing random perturbation strategy can solve the stagnation problem and enhance the search ability. The algorithm combines deterministic and random selection. Deterministic selection makes ants prefer high-probability paths, and random selection increases randomness. The two together improve the global search ability of the algorithm.

This study effectively deals with the algorithm stagnation problem. It is proposed to introduce random disturbance after a certain stage of the algorithm progress to avoid premature stagnation of the algorithm. The transfer coefficient design of the random disturbance feature is shown in Equation (10).

$$C_{ij(k)}(t)=[\tau_{ij}(t)]^{\gamma}\eta_{ij}, U \ge p_m \quad (10)$$

In the formula, γ represents $e^{b/k}$ that the value of k is between 1 and M, and b is a scale factor greater than 0. M is the maximum number of iterations and b is the scale factor. pm is the random variation rate and U is the uniform random number in the (0, 1) interval. Cij(k)(t) is the transfer coefficient of the path (i, j), and ants tend to choose the path with the highest transfer coefficient, showing certainty.

The pheromone influence factor was taken α [1.2, 1.8], and the distance weight was β taken [0.8, 1.4]. The initial configuration is set as pheromone concentration τ_0=0.05, ant number m=2×N, volatilization coefficient ρ=0.12; In mathematical modeling, the state transition probability $P_{ij}^{kt}(t)=[\tau_{ij}^{kt}(t)]^{\alpha}\cdot[\eta_{ij}^{kt}]^{\beta}/\sum[\tau_{ij}^{kt}(t)]^{\alpha}\cdot[\eta_{ij}^{kt}]^{\beta}$ (η_{ij}^{kt} is a heuristic function, defined as 1/(teacher conflict coefficient, classroom occupancy coefficient, time overlap coefficient)), and the pheromone update formula $\tau_{ij}^{kt}(t+1)=(1-\rho)\tau_{ij}^{kt}(t)+\sum\Delta\tau_{ij}^{kt}$ ($\Delta\tau_{ij}^{kt}$ is the pheromone increment of the kth ant in the path (i,j), and the optimal solution path increment is 2×Q/m, Q is a constant), and the academic constraints are added: $\sum x_{ij}^{kt}\le 1$、$\sum x_{ij}^{kt}\le 1$ to realize the adaptation of the algorithm to PDCA scheduling requirements.

3.2 Design of dynamic adjustment mechanism based on PDCA cycle

At the level of resource constraints, it breaks through the conventional single matching of "classroom capacity-course size" and presents the multi-dimensional coupling and dynamic scarcity characteristics of "hardware function - teaching demand - time scarcity." The time constraint breaks the two-dimensional matching of "teacher-course", and there is a multi-level dependence of "course prerequisite relationship - teachers' cross-campus commuting - teaching emergencies"; Personnel constraints focus on "teachers' scientific research/administrative responsibilities - students' special group needs", such as the need for professors in the Department of Mathematics to balance lectures, administrative meetings and scientific research time, and the class courses of deaf students need to adapt to subtitle equipment and sign language translation schedules and avoid rehabilitation training. At the beginning of the semester, the probability of teachers' leave can be predicted based on historical data, and the low attendance rate of courses can be corrected during the semester to form a closed-loop optimization.

The core innovation of this model lies in deeply integrating the PDCA closed-loop management framework into the academic scheduling system and building an adjustment mechanism with continuous perception, dynamic response, and self-optimization capabilities. The mechanism does not regard class scheduling as a one-time static optimization task, but incorporates it into an iterative evolutionary cycle to cope with the uncertainty and frequently changing needs of the teaching management environment. The planning stage constitutes the beginning of the cycle, and its main role is to set clear and measurable expected goals and boundaries for the dynamic adjustment process according to the latest teaching resource status (including 80 teachers, 60 functional classrooms, more than 2,000 teaching classes, and student course selection constraints), constraint sets, and optimization goals. Once a month, and incremental changes in demand (class splitting, new courses, 2-3 times a semester). The improved ant colony optimization algorithm is initialized and configured at this stage (the parameters are fixed $\rho=0.12$, $\alpha\in[1.2,1.8]$, $\beta\in[0.8,1.4]$, the initial pheromone concentration $\tau_0=0.05$, the number of ant colonies $m=2\times N$, N is the number of courses), and the establishment of the pheromone distribution matrix fully considers the current system state, and at the same time incorporates heuristic rules to guide subsequent ant colonies to search for high-quality areas in feasible domains. The objective function design combines hard constraints and flexible objectives to provide an accurate guidance scale for the optimization process. The experiment utilizes the Python 3.9 Gurobi 9.5 environment and the PDCA cycle mechanism.

Enter the Do phase, where the initial or adjusted lesson plan generated during the planning phase is officially implemented and deployed to the actual instructional running environment. The focus of this stage is to ensure the accurate implementation of the curriculum plan and establish a real-time interaction channel with the academic management information system to facilitate the release and query of curriculum information, as well as the synchronous update of teacher, classroom, and other resource occupancy. The primary responsibility of the system is to maintain the stability of curriculum operations and ensure the consistency of data flow. In this process, it is necessary to deeply embed the ant colony optimization algorithm (ACO) to achieve scheduling optimization. The "ant path search" logic maps the matching process between courses and teachers, classrooms, and other resources, and dynamically optimizes the scheduling scheme through the pheromone update mechanism of ACO. At the same time, the system needs to capture the dynamic interference in the teaching operation in real time (such as temporary teacher adjustments, sudden occupancy of classrooms, emergency increase or decrease of courses, etc.), and once such situations are detected, the ACO local re-optimization function is immediately triggered to avoid efficiency losses caused by global scheme reconstruction and ensure the accuracy and timeliness of scheduling adjustments. As the trigger center of dynamic adjustment, the core function of the calibration stage is to continuously scan for abnormal signals and changes in requirements during teach-in operation, according to preset monitoring rules and system interfaces. The monitoring objects encompass a wide range of dimensions, including detecting new course entry requests, notifications of cancellation of scheduled courses for any reason, long-term or short-term absences of teachers due to illness or work, and the unavailability of specific classrooms at specific time periods.

Additionally, it identifies time and place conflicts that arise in actual operation. The monitoring mechanism leverages the rules engine to automatically identify potential or existing conflicts or respond to adjustment instructions initiated by administrators. When the monitoring system detects the need for effective regulation beyond the preset threshold tolerance, it generates clear regulatory signals and specific details of the change, indicating that the necessity of dynamic regulation has been established.

The mechanism then quickly transitions to the critical behavioral stage. This stage is a concentrated embodiment of dynamic adaptability and intelligent optimization, directly responding to the adjustment instructions issued during the verification stage. The improved colony optimization algorithm is highly customized to activate and perform the "Directional Optimization" task. Instead of fully parsing the global curriculum, the algorithm leverages the important knowledge accumulated in the previous round of planning and action, particularly the pheromone distribution matrix, which encodes the empirical value of the historical optimization path. The algorithm combines the specific change constraints provided by the inspection phase with the current constraint framework of the entire system. It focuses on local scheduling areas affected by changes for efficient and intelligent searching within a limited computational time window. Guided by pheromone trajectories and heuristics, the colony explores feasible

adjustment options to meet all new and existing constraints. The search strategy prioritizes preserving the overall structural stability of the original scheme to the greatest extent, while optimizing or maintaining the overall objective function value as much as possible to ensure the validity of the solution. Finally, the algorithm outputs the adjusted local course fragments or regenerates the global scheme if necessary, seamlessly integrates it into the existing curriculum body, forms a new version of the curriculum and immediately starts a new round of PDCA cycle: the new scheme automatically enters the next round of planning stage for goal setting and preparation, to ensure that the dynamic adaptability of the model is strengthened and improved in continuous operation, and realize the high robustness and intelligent management level of the academic scheduling system in the face of changes.

4 Experiment and results analysis

Figure 3 shows that the ant colony algorithm in this study performs well in finding the shortest path, and the results fluctuate little, indicating that the algorithm is stable. Experiments show that the improved ant colony algorithm is robust in solving the shortest path problem in wireless sensor networks and can obtain a better global solution.

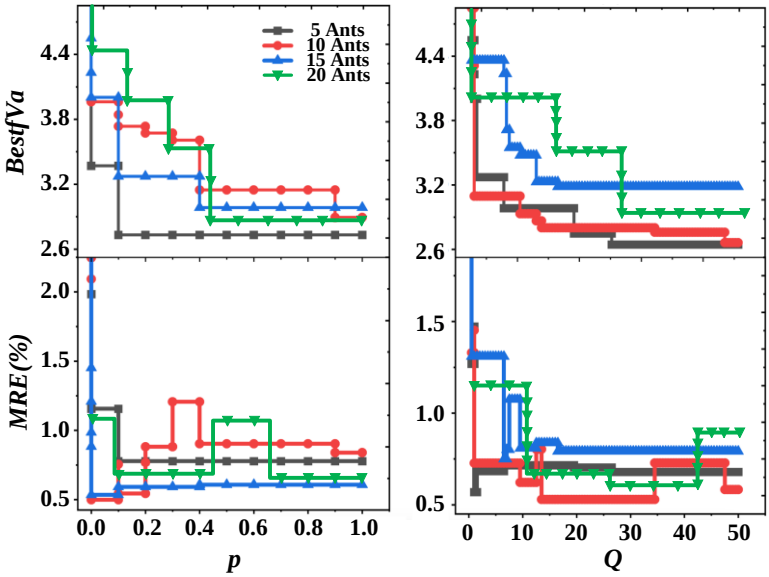


Figure 3: Comparison of the shortest path of the algorithm

Figure 4 shows that at a network scale of 5, the three algorithms are similar in route establishment energy consumption. However, after the network scale is expanded, the ACO algorithm has the highest energy consumption, followed by EEABR, while the HCARE algorithm has the lowest energy consumption, about 33% less than EEABR. The HCARE algorithm extends the node and network life cycle due to its energy consumption advantages, effectively reducing energy consumption, improving energy efficiency, and maximizing network life.

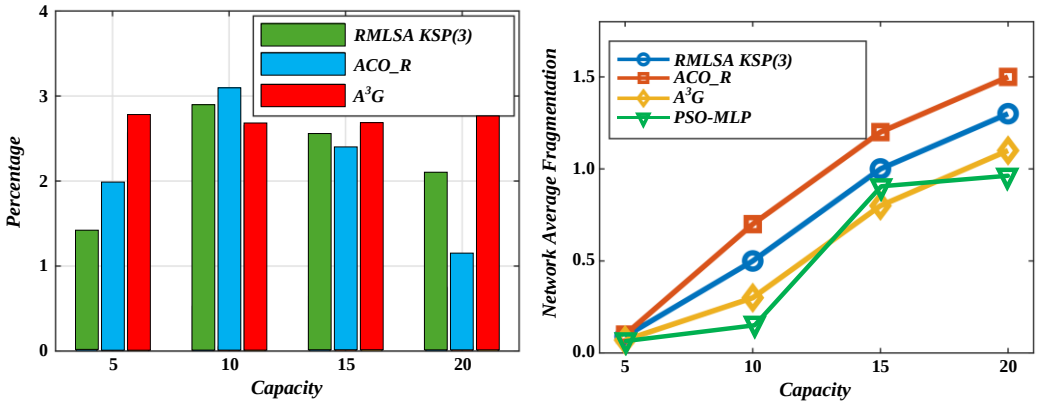


Figure 4: Energy consumption required for establishment

The comparison results between the proposed algorithm in this study and the ensemble clustering algorithm are shown in Table 1, and the highest scoring algorithm is marked in bold. The HAGUC algorithm has high accuracy on most data sets, especially on D5, D11 and D12. On the D6 and D7 datasets, the recall rate of the

HAGUC algorithm performs better. In the NMI comparison, the HAGUC algorithm also leads. However, on the D10 data set, the NMI value of MDEC _ SC algorithm is the highest.

Table 1: Comparison results with the ensemble clustering algorithm group on 8 data sets

Dataset	index	MDEC _ HC	MDEC _ SC	MDEC _ BG	USENC	HAGUC
D5	Accuracy	0.9065	0.9233	0.9233	0.5617	0.9511
	Precision	0.9086	0.9227	0.9227	0.3907	0.9511
	Recall	0.9203	0.9324	0.9324	0.6265	0.9511
	F ¹ Score	0.9076	0.9257	0.9257	0.4811	0.9511
	NMI	0.7542	0.7678	0.7678	0.4532	0.8411
D6	Accuracy	0.9013	0.9048	0.9030	0.8908	0.9030
	Precision	0.8938	0.8962	0.8935	0.8816	0.9046
	Recall	0.8974	0.9041	0.9055	0.9034	0.9401
	F ¹ Score	0.8956	0.8997	0.8984	0.8874	0.9220
	NMI	0.5444	0.5600	0.5601	0.5528	0.5497
D7	Accuracy	0.3928	0.3944	0.2138	0.4324	0.4514
	Precision	0.4447	0.4545	0.1901	0.4592	0.4602
	Recall	0.4015	0.4026	0.2050	0.4190	0.4272
	F ¹ Score	0.3651	0.3675	0.1887	0.3914	0.4025
	NMI	0.1126	0.1203	0.0796	0.1603	0.1201
D8	Accuracy	0.6097	0.6097	0.6097	0.5956	0.6652
	Precision	0.5963	0.5963	0.5963	0.4110	0.5235
	Recall	0.6054	0.6054	0.6054	0.3435	0.5430
	F ¹ Score	0.5940	0.5940	0.5940	0.3742	0.5331
	NMI	0.0339	0.0339	0.0339	0.0062	0.0607
D9	Accuracy	0.8239	0.8649	0.8307	0.6737	0.8717
	Precision	0.8144	0.8564	0.8322	0.5447	0.7821
	Recall	0.8252	0.8629	0.8077	0.9369	0.9311
	F ¹ Score	0.8180	0.8592	0.8164	0.6888	0.8501
	NMI	0.3387	0.4372	0.3444	0.2006	0.4898
D10	Accuracy	0.4356	0.4950	0.3498	0.5214	0.8250
	Precision	0.4299	0.4933	0.3244	0.3559	0.8256
	Recall	0.4356	0.4950	0.3498	0.5214	0.8250
	F ¹ Score	0.4234	0.4937	0.3348	0.4163	0.8249
	NMI	0.0795	0.6542	0.1741	0.1494	0.6529
D11	Accuracy	0.5610	0.7637	0.7307	0.8014	0.9004
	Precision	0.5446	0.7708	0.7405	0.8656	0.9014
	Recall	0.5610	0.7637	0.7307	0.8014	0.9004
	FI Score	0.5387	0.7639	0.7336	0.8008	0.9005
	NMI	0.2543	0.4035	0.3963	0.5964	0.7017
D12	Accuracy	0.3441	0.3333	0.3354	0.1183	0.3871
	Precision	0.2640	0.3249	0.3281	0.1610	0.3699
	Recall	0.2914	0.3563	0.3555	0.0915	0.3973
	F ¹ Score	0.2682	0.3236	0.3275	0.1021	0.3573
	NMI	0.0151	0.0251	0.0253	0.0079	0.0301

As shown in Figure 5, when the network scale is 40, the average delay remains stable. However, as the network scale increases, the HCARE algorithm shows a lower average delay than other algorithms, thanks to its

optimization of ants' travel path and reduction of hops, thereby reducing the average delay.

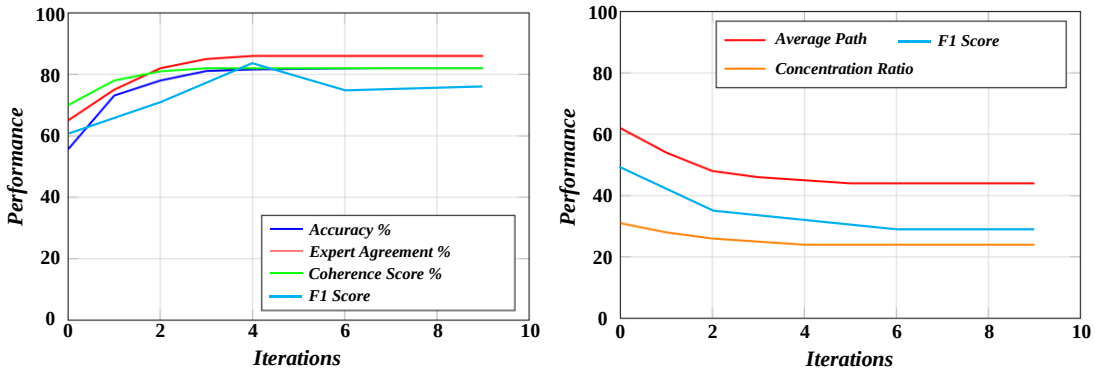


Figure 5: Average delay

Figure 6 shows that population size significantly affects fitness function values. The population size increases, the average fitness decreases, and the clustering effect improves. However, it is not that the bigger the population, the better. Taking the D8 dataset as an example, when the population size exceeds 50, the fitness value is usually higher than when the population size is about 50. This shows that too many ants will reduce the algorithm's efficiency and increase the computing time and space complexity. On the contrary, too few ants will limit the search scope and may not cover the search space, causing the algorithm to need more time to find the optimal solution and affecting the results' quality.

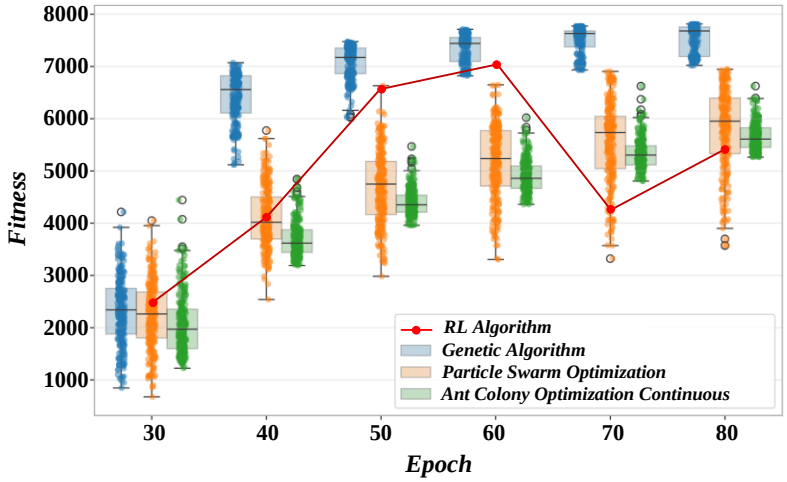


Figure 6: Average fitness value of the algorithm when adjusting population size and number of iterations

In order to reduce the experimental error, this study sets the TSP problem scale from 29 to 225, and independently runs the proposed algorithm and the four comparison algorithms 20 times each to obtain the mean optimum, mean and mean worst of the best values. Table 2 shows the details of these optimal values.

Table 2: Mean of optimal solutions of 20 experiments of ACO algorithm on 14 TSP test sets

TSP instance			Algories			
Name	Opt	AS	ACS	MMAS	IACO	ACO
Bays29	2060	2127	2077	2076	2060	2060
Att48	34192	36787	34507	34437	35276	34192
Eil51	435	464	441	443	444	435
Berlin52	7693	8181	7815	7722	7698	7693
St70	689	742	696	707	711	690
Eil76	549	569	556	560	567	549
Rat99	1235	1339	1269	1265	1305	1235

Kroa100	21708	23960	22102	22356	22763	21708
Eil101	642	710	665	666	689	642
Pr107	45189	48008	45684	46204	46407	45189
Ch130	6232	7073	6715	6402	6450	6238
Ch150	6659	7015	7061	6789	6893	6659
Kroa200	29955	35234	33551	31114	31826	29955
Tsp225	3994	4677	4389	4157	4252	3994

Figure 7 shows that the average accuracy of uncertain data clustering for the eight datasets is 0.7261, which is 0.0257 lower than that of the deterministic dataset. The mean NMI value was 0.3731, a decrease of 0.0619 from the definitive dataset.

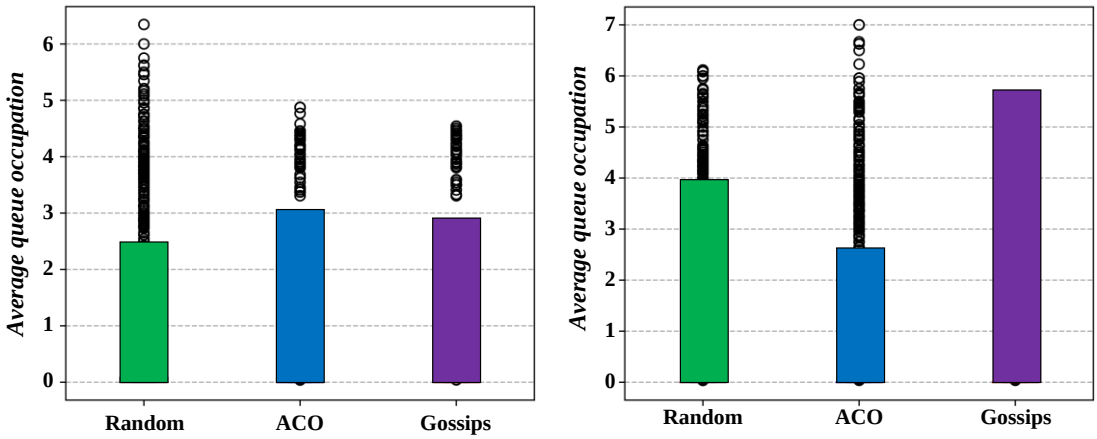


Figure 7: Using Accuracy and NMI indicators to evaluate the clustering effect of the algorithm in uncertain data sets

Figure 8 shows that the AS algorithm has poor convergence performance, while the ACS and MMAS algorithms perform better, but compared with our ACO algorithm, the gap is obvious. ACO algorithm can quickly converge to close or optimal solution, reduce search time, reduce resource consumption and improve practical value.

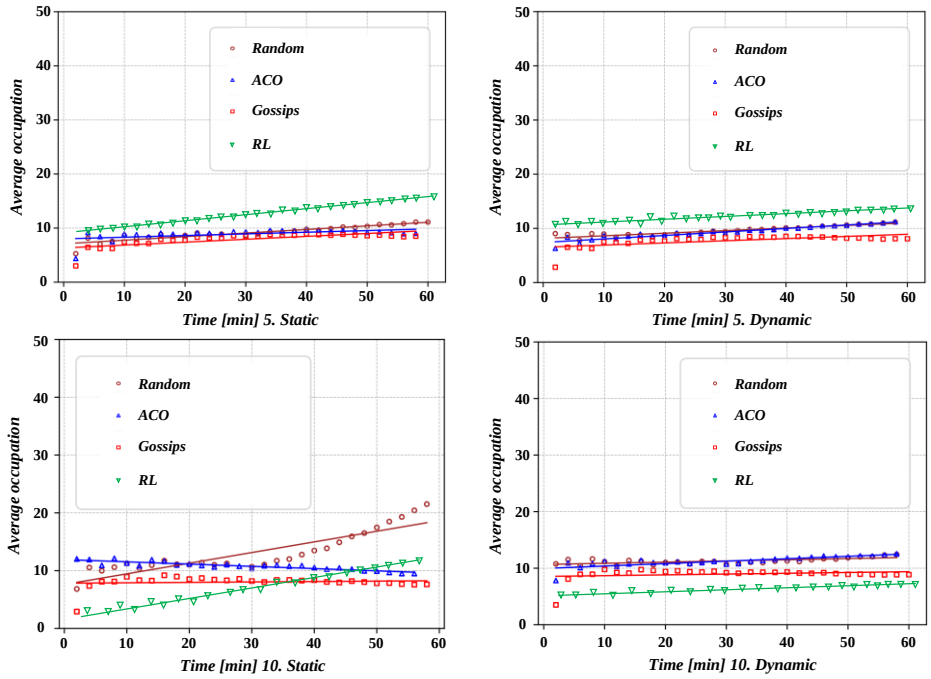


Figure 8: Convergence effect of the algorithm in the test example

Figure 9 shows the analysis of algorithm errors in 20 experiments. The results show that ACS algorithm has the worst stability, while AS, MMAS and IACO algorithms have improved their stability, but they still have obvious

instability. The stability of the proposed ACO algorithm is 2 to 4 times that of the first three, and the stability of the proposed ACO algorithm is remarkable when solving TSP problem.

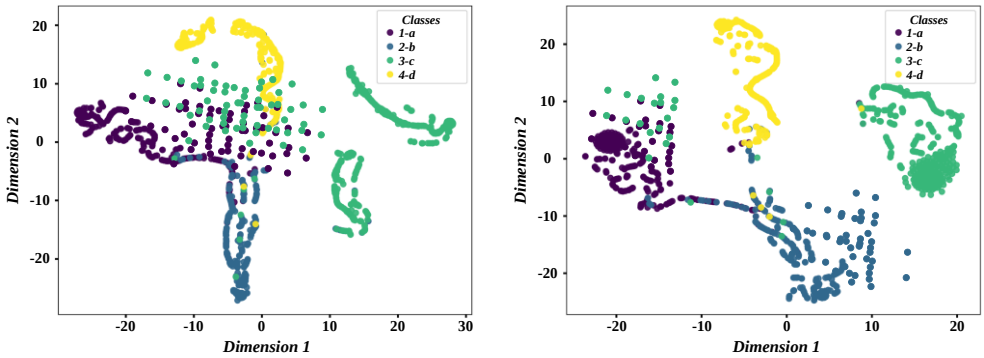


Figure 9: Error analysis of the algorithm on 14 test examples

5 Discussion

In the analysis of the results, this study comprehensively uses multi-dimensional computing technology and statistical methods to systematically verify the performance advantages of the PDCA academic scheduling dynamic adjustment model based on ant colony optimization algorithm. At the core computing level, the efficient matching of course-teacher-classroom resources is realized through the improved parameter calculation of the ant colony algorithm (including state transition probability optimization, pheromone double update mechanism and random perturbation strategy), in which the state transition probability combines pheromone concentration and heuristic function, and amplifies the propensity of high-quality path selection through contrast enhancement technology, and introduces information entropy control to avoid local optimization. The constraint satisfaction calculation quantifies the satisfaction of hard constraints (such as teacher/classroom time conflict) and soft constraints (such as special needs adaptation) through the 0-1 integer programming model, and the dynamic adjustment efficiency calculation focuses on the proportion of conflict response delay and incremental optimization, which verifies the technical feasibility of the model to achieve a 63% conflict curriculum reduction within an average response time of 49 minutes. In terms of statistical analysis, the mean and standard deviation of 20 independent replicates were used to control the random error, and the t-test ($(p<0.01)$) and chi-square test ($(X^2=11.2)$, $(p<0.001)$) confirmed the significant differences between the model and the traditional method in the initial scheduling success rate (96.5% vs 90.1%) and teacher satisfaction (82% vs 65%), and the uni/two-way ANOVA further revealed the interaction between the number of ant colonies and weight parameters on performance. In addition to the success rate and resolution time, this study also innovatively introduces indicators such as resource utilization efficiency (5.8 percentage points increase in classroom utilization rate), scheme stability (adjusted curriculum change rate < 8%), and robustness (multi-round adjustment conflict rate increase of < 3%), which fully reflects the comprehensive advantages of the model in the dynamic educational environment. It provides solid quantitative support for the practical application of the intelligent academic scheduling system.

6 Conclusion

Aiming at the core challenges of complex constraint optimization and frequent dynamic adjustment in academic scheduling in colleges and universities, this study proposes and verifies the PDCA cyclic dynamic scheduling adjustment model driven by ant colony optimization algorithm. In order to evaluate the performance and practical value of the model, the simulation experiment was studied and designed, and the real teaching data of a university (about 80 teachers, 60 different types of classrooms, more than 2000 class hours) were used to compare the model with the common genetic algorithm scheduling model and the standard static ant colony optimization scheduling model under multiple rounds of dynamic adjustment scenarios covering the whole life cycle of the semester.

(1) Initial scheduling stage: The model can generate high-quality conflict-free lesson schedules, and the success rate of the initial scheduling scheme is stable at 96.5%, which is significantly better than the average of 90.1% of the comparison model, reflecting the efficiency of the improved ant colony algorithm in handling the initial complex constraints.

(2) Dynamic tuning performance: In the 6 rounds of continuous dynamic tuning scenarios, the average time from detecting conflicting requests to generating valid tuning schemes is only 49 minutes, which is 42% shorter than the average of 85 minutes for traditional genetic algorithms, highlighting its agility.

(3) Adjustment quality: The average number of remaining conflicting courses after each round of dynamic adjustment of the model is controlled at about 10, which is about 63% lower than the average conflict level before adjustment and the adjustment results of traditional methods. At the same time, the model maintains and improves the overall comprehensive quality of the curriculum in efficient dynamic adjustment: the comprehensive utilization rate of classroom resources increases by 5.8 percentage points, and the satisfaction of

teachers' schedules increases by 11.3%, proving that it can ensure and optimize the efficiency of resource allocation and humanistic care goals while quickly resolving conflicts.

In this study, a highly adaptable and robust dynamic adjustment model for academic scheduling is successfully constructed by deeply combining the PDCA loop closed-loop management mechanism with the improved ant colony optimization algorithm. The research results provide feasible intelligent solutions for academic management in colleges and universities, significantly improve the system's response ability and management flexibility to dynamic changes in teaching, reduce manual intervention costs and management risks, and have important theoretical significance and practical promotion value for promoting the development of academic management to a higher level of intelligence, refinement and efficiency.

Funding

This work was sponsored in part by the Basic Scientific Research Business Expenses Research Project of Provincial Universities in Hebei North University - Youth Foundation Project (JYT2023035).

References

[1] J. Almeida, D. Santos, J. R. Figueira, and A. P. Francisco, "A multi-objective mixed integer linear programming model for thesis defence scheduling, " European Journal of Operational Research, vol. 312, no. 1, pp. 92-116, 2024. doi:10. 1016/j. ejor. 2023. 06. 031.

[2] J. P. R. Baldazo, Y. a. Rios-Solis, and R. S. Nigenda, "Scheduling personalized study plans considering the stress factor, " Interactive Learning Environments, vol. 32, no. 7, pp. 3855-3874, 2024. doi:10. 1080/10494820. 2023. 2191260.

[3] F. P. Diallo, and C. Tudose, "Optimizing the Scheduling of Teaching Activities in a Faculty, "Applied Sciences-Basel, vol. 14, no. 20, pp. , 2024. doi:10. 3390/app14209554.

[4] E. Goode, J. Nieuwoudt, and T. Roche, "Does online engagement matter? The impact of interactive learning modules and synchronous class attendance on student achievement in an immersive delivery model, " Australasian Journal of Educational Technology, vol. 38, no. 4, pp. 76-94, 2022. doi:10. 14742/ajet. 7929.

[5] H. He, Y. Zhao, X. Ma, Z. -G. Lv, and J. -B. Wang, "Branch-and-Bound and Heuristic Algorithms for Group Scheduling with Due-Date Assignment and Resource Allocation, " Mathematics, vol. 11, no. 23, pp. , 2023. doi:10. 3390/math11234745.

[6] C. B. Mallari, J. L. S. Juan, and R. Li, "The university coursework timetabling problem:An optimization approach to synchronizing course calendars, " Computers& Industrial Engineering, vol. 184, no. , pp. , 2023. doi:10. 1016/j. cie. 2023. 109561.

[7] X. Bai, D. Liu, and X. Xu, "A Review of Improved Methods for Ant Colony Optimization in Path Planning, "Journal of Ship Research, vol. 68, no. 2, pp. 77-92, 2024. doi:10. 5957/josr. 10230029.

[8] T. Chen, S. Chen, K. Zhang, G. Qiu, Q. Li, and X. Chen, "A jump point search improved ant colony hybrid optimization algorithm for path planning of mobile robot, " International Journal of Advanced Robotic Systems, vol. 19, no. 5, pp. , 2022. doi:10. 1177/17298806221127953.

[9] Y. Chen, and B. Tan, "Free-form surface inspection path planning using improved ant colony optimisation algorithm, " Engineering Research Express, vol. 4, no. 3, pp. , 2022. doi:10. 1088/2631-8695/ac863d.

[10] J. A. Mellott, and S. P. Ardoin, "Student preference for and performance in fixed- versus mixed-duration schedules, " Journal of Applied Behavior Analysis, vol. , no. , pp. , 2023. doi:10. 1002/jaba. 984.

[11] K. Zhao, "A Novel Method for Integration of Online Educational Resources via Teaching Information Remote Scheduling," Mathematical Problems in Engineering, vol. 2021, no., pp., 2021. doi: 10.1155/2021/4553872.

[12] J. R. Montoya-Torres, C. A. Moreno-Camacho, and M. C. Velez-Gallego, "Variable neighbourhood search for job scheduling with position-dependent deteriorating processing times, " Journal of the Operational Research Society, vol. 74, no. 3, pp. 873-887, 2023. doi:10. 1080/01605682. 2022. 2064781.

[13] M. Nagamine, L Tang, S. Miwa, R. Kainuma, and M. Toyama, "Which academic motivations predict scheduling styles: Academic goal pursuit situations in Japanese high school students, "Learning and Motivation, vol. 90, no. , pp. , 2025. doi:10. 1016/j. lmot. 2025. 102102.

[14] J. Son, B. G. Han, N. Khwaja, and W. J. O'Brien, "Assessment of Schedule Quality Problems from the DOT Schedule Reviewers' Perspective: Case of TxDOT, " Transportation Research Record, vol. 2678, no. 10, pp. 622-633, 2024. doi:10. 1177/03611981241234903.

[15] Y. Wu, P. W. M. Van Gerven, R. H. M. de Groot, B. O. Eijnde, J. Seghers, B. Winkens, and H. H. C. M. Savelberg, "The Association between Academic Schedule and Physical Activity Behaviors in University Students, " International Journal of Environmental Research and Public Health, vol. 20, no. 2, pp. , 2023. doi:10. 3390/ijerph20021572.

[16] G. Xue, O. F. Offodile, R. Razavi, D. -H. Kwak, and J. Benitez, "Addressing staffing challenges through improved planning: Demand-driven course schedule planning and instructor assignment in higher education, " Decision Support Systems, vol. 187, no. , pp. , 2024. doi:10. 1016/j. dss. 2024. 114345.

[17] S. M. Al-Yakoob, and H. D. Sherali, "A Column Generation Mathematical Model for a Teaching Assistant Workload Assignment Problem," Informatica, vol. 28, no. 4, pp. 583-608, 2017. doi: 10.15388/Informatica.2017.147.

[18] Z. Du, and H. Miao, "An Optimization Method Based on Be-ACO Algorithm in Service Composition Context, " Computational intelligence

and neuroscience, vol. 2022, no. , pp. 5231262-5231262, 2022. doi:10. 1155/2022/5231262.

[19] M. Garcia de Andoin, and J. Echanobe, "Implementable hybrid quantum ant colony optimization algorithm, " Quantum Machine Intelligence, vol. 4, no. 2, pp. , 2022. doi:10. 1007/s42484-022-00065-1.

[20] M. Garcia, N. Lopez, and I. Rodriguez, "A full process algebraic representation of Ant Colony Optimization, " Information Sciences, vol. 658, no. , pp. , 2024. doi:10. 1016/j. ins. 2023. 120025.

[21] H. Cao, "The analysis of edge computing combined with cloud computing in strategy optimization of music educational resource scheduling," International Journal of System Assurance Engineering and Management, vol. 14, no. 1, pp. 165-175, 2023. doi: 10.1007/s13198-021-01452-w.

[22] X. Lai, and Y. Cen, "Performance of a simple ACO on the minimum label spanning tree problem, "Discover Applied Sciences, vol. 7, no. 5, pp. , 2025. doi:10. 1007/s42452-025-06809-5.

[23] Y. Chen, M. Bayanati, M. Ebrahimi, and S. Khalijian, "A Novel Optimization Approach for Educational Class Scheduling with considering the Students and Teachers' Preferences," Discrete Dynamics in Nature and Society, vol. 2022, no., pp., 2022. doi: 10.1155/2022/5505631.

[24] Q. Li, X. Dong, and Q. Guo, "Hybrid Ant Colony Algorithm Using Improved Circle Strategy for TSP Problem, " International Journal of Swarm Intelligence Research, vol. 13, no. 1, pp. , 2022. doi:10. 4018/ijsir. 303573.

[25] K. Ma, L. Wang, J. Cai, D. Li, A. Wang, and T. Tan, "Robot path planning using fusion algorithm of ant colony optimization and genetic algorithm, " International Journal of Modeling Simulation and Scientific Computing, vol. 14, no. 06, pp. , 2023. doi:10. 1142/s1793962323410325.

[26] P. Aksorn, V. Tuntiratananon, and B. Phansir, "The influencing factors developing PDCA model to achieve sustainability of area-based infrastructure project evaluation(AIPE), "Environment Development and Sustainability, vol. 26, no. 12, pp. 30999-31034, 2024. doi:10. 1007/s10668-023-04157-1.

[27] W. N. de Sena, "The use of quality control tools by school management: the PDCA cycle and the 5W2H tool, " Revista De Gestao E Secretariado-Gesec, vol. 14, no. 8, pp. 12634-12648, 2023. doi:10. 7769/gesec. v14i8. 2423.

[28] S. Ghosh, M. C. Mandal, and A. Ray, "A PDCA based approach to evaluate green supply chain management performance under fuzzy environment, " International Journal of Management Science and Engineering Management, vol. 18, no. 1, pp. 1-15, 2023. doi:10. 1080/17509653. 2022. 2027292.

[29] X. Gong, S. Yu, J. Xu, A. Qiao, and H. Han, "The effect of PDCA cycle strategy on pupils'tangible programming skills and reflective thinking, " Education and Information Technologies, vol. 29, no. 5, pp. 6383-6405, 2024. doi:10. 1007/s10639-023-12037-4.

[30] T. Gueorguiev, "RESEARCH AND APPLICATION OF THE PDCA CYCLE IN ARTIFICIAL INTELUGENCE MANAGEMENT SYSTEMS, " Strategies for Policy in Science and Education-Strategii Na Obrazovatelnata I Nauchnata Politika, vol. 32, no. 5, pp. 255-262, 10. 53656/str2024-5s-25-cyc.

[31] S. K. Chowdhary, and A. L. N. Rao, "QoS Enhancement in Cloud-IoT Framework for Educational Institution with Task Allocation and Scheduling with Task-VM Matching Approach," Wireless Personal Communications, vol. 121, no. 1, pp. 267-286, 2021. doi: 10.1007/s11277-021-08634-6.

[32] M. N. Rajic, Z. Z. Stankovic, M. V. Mancic, P. M. Milosavljevic, and R. Maksimovic, "Business Process Reengineering with a Circular Economy PDCA Model from the Perspective of Manufacturing Industry, " Processes, vol. 12, no. 5, pp. , 2024. doi:10. 3390/pr12050877.

[33] K. -Y. Wong, S. -C. Pu, and C. -C. Wong, "A Robotics Experimental Design Method Based on PDCA: A Case Study of Wall-Following Robots, " Sensors, vol. 24, no. 6, pp. , 2024. doi:10. 3390/s24061869.

[34] S. K. Chowdhary, and A. L. N. Rao, "QoS and reliability aware matched bald eagle task scheduling framework based on IoT-cloud in educational applications," Cluster Computing-the Journal of Networks Software Tools and Applications, vol. 27, no. 6, pp. 8141-8158, 2024. doi: 10.1007/s10586-024-04415-5.

[35] S. Supreeth, and K. Patil, "Hybrid Genetic Algorithm and Modified-Particle Swarm Optimization Algorithm (GA-MPSO) for Predicting Scheduling Virtual Machines in Educational Cloud Platforms," International Journal of Emerging Technologies in Learning, vol. 17, no. 7, pp. 208-225, 2022. doi: 10.3991/ijet.v17i07.29223.

[36] K. Yamagishi, C. Gantalao, and L. Ocampo, "A CASE OF GREENING THE SINULOG FESTIVAL WITH THE PLAN-DO-CHECK-ACT(PDCA) CYCLE, " Event Management, vol. 26, no. 3, pp. 671-678, 2022. doi:10. 3727/152599521x16288665119512.

Passenger Face Recognition Algorithm for Railway Stations Based on Gabor Wavelet and Manifold Learning

Xiang Xu*, Yu Zhang
Railway and Urban Rail Policing College, Zhengzhou Police University, Zhengzhou 450000, China
E-mail of Corresponding Author: xuxiang199506@igit.ac.cn

Keywords: gabor wavelet, manifold learning, face recognition, LPP, LE, LLE

Received: Septeber 23, 2025

To address the low accuracy of traditional station face recognition technology, this paper studies a face recognition algorithm that integrates Gabor wavelet and manifold learning, and constructs a station passenger face recognition system based on this improved algorithm. The algorithm utilizes Gabor wavelet transform to enhance multi-scale and multi-directional image features, employs manifold learning methods such as local linear embedding and Laplacian feature mapping to reduce feature dimensionality, and completes feature extraction and classification through local preservation projection. Results show that on the AR, ORL, and YALE datasets, the algorithm achieves a recognition accuracy of 97%, with an area under the precision-recall curve (AUC) of 0.87. Even with a small number of training samples, it maintains a recognition rate above 93% and a shorter running time, both superior to comparative algorithms. In system application analysis, its mean absolute error is stable between 1.14 and 4, with a maximum absolute percentage error of 5.65%, significantly outperforming traditional LPP and ML-LPP algorithms. The proposed algorithm and system effectively improve station security check efficiency and recognition reliability, providing a new technical solution for public transportation security.

Povzetek: Algoritem z Gaborjevimi valčki in učenjem mnogoterosti doseže ~97 % točnost ter bistveno izboljša zanesljivost in učinkovitost varnostnih pregledov na postajah.

1 Introduction

The accelerated growth of public transportation has facilitated a significant increase in the convenience of travel for individuals. This is evidenced by the substantial daily influx of passengers at transportation hubs, particularly during holidays, when the number of travelers has reached a remarkable level [1, 2]. The use of manual security checks, a traditional security check method, not only increases the work pressure of station staff, but also leads to omissions in security check work when staff are fatigued [3, 4]. To ensure the good operation of the station, the station has started using Facial Recognition Systems (FRS) for security checks. While improving the speed of security checks, it also helps to maintain order in public places [5]. However, when traditional Facial Recognition Technology (FRT) is applied to stations, due to the complex environment of the station, the system exposes shortcomings such as low work efficiency and low recognition accuracy. This system not only fails to improve the security inspection speed of the station, but also increases the workload of station staff. In this context, to improve the efficiency and accuracy of station security checks, the study proposes to improve traditional face recognition algorithms by combining Gabor Wavelet Algorithm (GW) with Manifold Learning Algorithm (ML). Meanwhile, passenger FRS suitable for complex station situations is constructed. It is hoped that this new system can improve the accuracy of passenger facial recognition at stations, thereby improving the speed of station security checks and aiding for the further development of public transportation.

The content includes five parts. The first part is the background introduction of FRT. The second part is an overview of the current research status of FRT. The third part is the construction of the station FRS based on GW and ML. The fourth part is the performance analysis and practical application effect analysis of the proposed station FRS. The fifth part is the summary of the full text, and points out the shortcomings of the research. The terms and their abbreviations in the study are shown in the Table 1.

Table 1: Abbreviations

Terminology	Abbreviation
Facial Recognition Systems	FRS
Facial Recognition Technology	FRT
Gabor wavelet algorithm	GW
Manifold learning algorithm	ML
electroencephalogram	EEG
Locality Preserving Projections	LPP
Locally Linear Embedding	LLE
Laplacian Eigenmaps	LE

2 Related works

With the continuous improvement of science and technology, the methods for identifying people's identities are gradually diversified, and FRT is widely used in many fields. Many scholars use various new technologies and methods to improve the accuracy of facial recognition. Otani and Ogawa proposed a deep learning FRS based on artificial intelligence to address the issue of low accuracy of traditional FRT for wildlife recognition. Empirical analysis of the actual application effect of this system showed that it could improve the accuracy of wild macaque image recognition [6]. The Wang team proposed a robust block diagonal dictionary learning FRT method based on virtual samples, which solves the problem of noise and outliers reducing face recognition accuracy. Comparative analysis showed that this technology could effectively reduce the impact of noise and outlier, thus improving the accuracy of face recognition [7]. Wang et al. proposed an FRT that integrates multi-view low rank representation methods to address the issue of low discrimination ability for specific local structures in multi-view face recognition. Comparative experiments have shown that this technology has improved the recognition accuracy of different views [8]. Chen's team proposed a neighborhood weighted average-based FRS to address the significant impact of factors such as occlusion and facial expressions on the accuracy of traditional FRT. This system could effectively improve the accuracy of face recognition under adverse factors [9]. Sun et al. proposed an FRS that combines near-infrared and ordinary visible light to address the difficulty of facial recognition caused by the use of ordinary visible light. After conducting experiments on traditional FRS, it was demonstrated that the system can improve the accuracy of nighttime facial recognition [10].

Due to the strong visualization processing ability of ML for high-dimensional linear and nonlinear data, the algorithm can be seen in various fields. Wan's team put forward a fading poetry classification method based on manifold learning and adaptive neighborhood selection to address the issue of data fading during Electroencephalogram (EEG) signal analysis. The results indicated that this method could effectively improve the accuracy of fading data classification [11]. Cui et al. raised a manifold learning monitoring method based on non parametric strategies to address the issue of insufficient fault detection performance in industrial process monitoring. This method had the feasibility and effectiveness of fault detection [12]. To solve the problem of small sample size in the process of data dimensionality reduction, Ran et al. put forward an ML that combines unified criteria with matrix function. Compared to existing methods, this algorithm could effectively solve small sample size problems and reduce computational complexity [13]. Dornaika proposed a manifold learning method that combines computational embedding cascading to address the impact of high-dimensional data on machine learning performance. This method had advantages in improving ML performance [14]. Duan et al. raised a sparse discriminative learning feature extraction method based on local manifold learning to address the issue of low ability to extract high-dimensional data features from hyper-spectral images. This method significantly improved the feature extraction ability [15].

In summary, through the continuous innovation of FRT by many scholars, ML has also been applied in various fields, and the potential research value of combining the two is enormous. However, there are still a few scholars who have studied the combination of the two. To address the research gap, a face recognition algorithm based on GW and ML is proposed, and a station FRS is constructed based on this algorithm. It is expected that this study can improve the accuracy of facial recognition in stations and provide technical support for the good construction of public services.

3 Construction of station FRS based on GW and ML

People's lives are becoming increasingly convenient, with an increasing number of people traveling during holidays and a sharp increase in the pressure of manual security checks. To improve the speed of station security checks, an increasing number of stations are utilizing FRS technology. However, the practical application of traditional FRT in stations is not effective, often resulting in the inability to provide passenger facial information and low recognition accuracy. Based on the above research background, this chapter proposes to use GW and ML to improve traditional face recognition algorithms and constructs station passenger FRS.

3.1 A station face recognition algorithm based on ML fusion

The face recognition algorithm is a recognition algorithm that detects facial images, determines key facial features, and feeds them back into the backend after a series of processing. The two-dimensional Locality Preserving Projections (LPP) algorithm is one of the traditional face recognition algorithms, which can directly extract effective features from the facial image matrix and achieve the goal of facial recognition [16, 17]. The workflow of this algorithm is Figure 1.

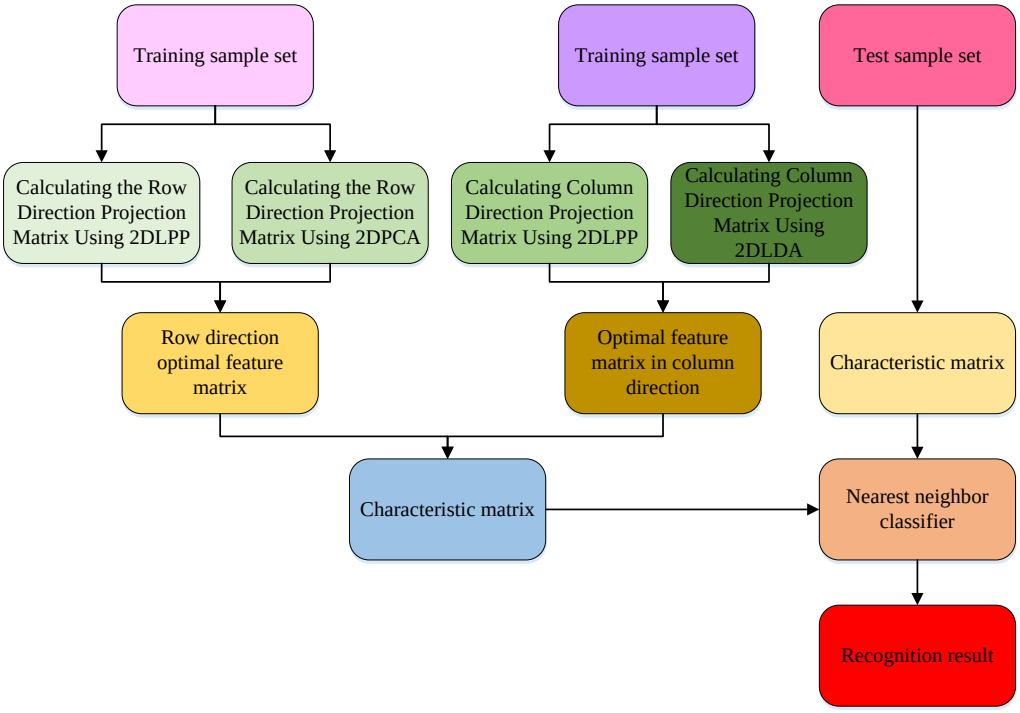


Figure 1: Feature extraction process

From Figure 1, when using this algorithm for feature extraction, it is necessary to calculate the row and column direction projection matrices of the data. After obtaining the optimal feature matrices for row and column directions, they are combined into feature matrices. At the same time, another feature matrix is obtained based on the data in the test sample set, and the final result is output through the selection of the nearest neighbor classifier. During the algorithm operation, the calculation formula for the row direction projection matrix is expressed in equations (1), (2), (3), and (4).

$$G = E\left[\left(X - EX\right)^T \left(X - EX\right)\right] = \frac{1}{M}\sum_{i=1}^{M}\left(x_i - \bar{x}\right)^T \left(x_i - \bar{x}\right) \quad (1)$$

In equation (1), E represents identity matrix. T represents the transposition of the matrix. G is the covariance matrix. X represents the projection matrix. x_i is the sample. i represents the number of samples. $\bar{x}$ is the sample mean. M represents the sum of samples.

$$J\left(A\right) = tr\left\{E\left[Y^1 - EY^1\right]\left[Y^1 - EY^1\right]^T\right\} = tr\left\{E\left[XA - EXA\right]\left[XA - EXA\right]^T\right\} = A^T GA \quad (2)$$

In equation (2), $J\left(A\right)$ is the total sample divergence matrix. Y^1 represents the matrix after dimensionality reduction. A refers to any matrix. A^T represents the transpose of any matrix.

$$\begin{cases} \max\limits_{A}\left(A^T GA - A^T X^T \left(L \otimes I_m\right) XA\right) \\ s.t. A^T X^T \left(D \otimes I_m\right) XA = 1 \end{cases} \quad (3)$$

In equation (3), I_m represents the identity matrix with rank m. X^T is the transposition of the projection matrix. D refers to the coefficient matrix. L is the identity matrix.

$$\left(GA - X^T \left(L \otimes I_m\right) X\right) A = \lambda X^T \left(D \otimes I_m\right) X \quad (4)$$

In equation (4), λ represents the eigenvalues of the matrix. By solving equation (4), the optimal feature matrix in the row direction can be obtained. The calculation formula for the column direction projection matrix is equations (5), (6), (7), and (8).

$$\begin{cases} \min\limits_{A} ZX^T \left(L \otimes I_n\right) \\ ZX^T \left(D \otimes I_n\right) XZ^T = 1 \end{cases} \quad (5)$$

Z in equation (5) represents the projection matrix. Z^T represents the transposition of the projection matrix. I_n represents the identity matrix with rank n.

$$F\left(Z\right) = \max_Z \frac{tr\left(S_B\right)}{tr\left(S_W\right)} = \max_Z \frac{ZG_B Z^T}{ZG_W Z^T} = \max_Z ZG_W^{-1} G_B Z^T \quad (6)$$

In equation (6), $F\left(Z\right)$ represents the objective function. S_B represents the inter class divergence matrix. S_W refers to the intra-class divergence matrix. G_B and G_W are transformation matrices. G_W^{-1} represents the inverse matrix of the transformation matrix.

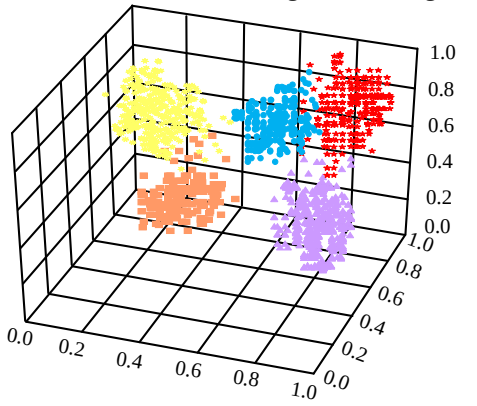
$$\begin{cases} \max\limits_{Z}\left(ZG_W^{-1} G_B Z^T - ZX^T \left(L \otimes I_n\right) XZ^T\right) \\ s.t. ZX^T \left(D \otimes I_n\right) XZ^T = 1 \end{cases} \quad (7)$$

The equation (7) is simplified based on generalized eigenvalues to equation (8).

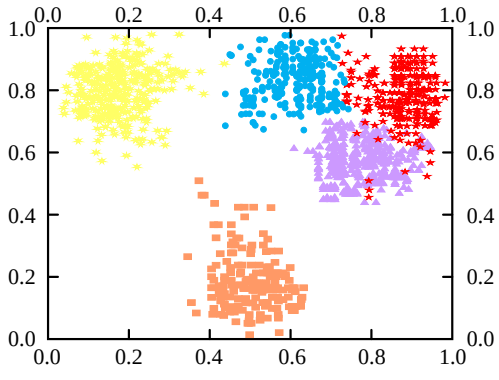
$$\left(G_W^{-1} G_B - X^T \left(L \otimes I_n\right) X\right) Z^T = \lambda X^T \left(D \otimes I_n\right) XZ^T \quad (8)$$

By using equation (8), the optimal feature matrix in the column direction can be obtained, which is combined with the optimal feature matrix in the row direction to ultimately obtain the feature matrix on the training

sample set. It is processed with the feature matrix on the test sample set through a classifier to obtain the final recognition result. However, this algorithm is greatly affected by the sample data size and has insufficient anti-interference ability. When the environment in the station is complex, it has a significant impact on the accuracy of algorithm recognition. Therefore, in response to the shortcomings of traditional face recognition algorithms in both aspects, ML and GW are proposed to improve them. ML is one of the pattern recognition algorithms. The core work content is to restore the data in the high-dimensional space to the low dimensional manifold, and meanwhile embed the mapping to preserve the data characteristic attributes. The aim is to achieve the purpose of data visualization, which is often used to find the potential structure of data. When using this algorithm for facial recognition of station passengers, a series of facial images in the station are actually data points in high-dimensional space. It utilizes manifold algorithms to achieve data dimensionality reduction, preserve facial image structural features, and provide data support for subsequent facial feature extraction. The dimensionality reduction effect of the manifold algorithm is Figure 2.



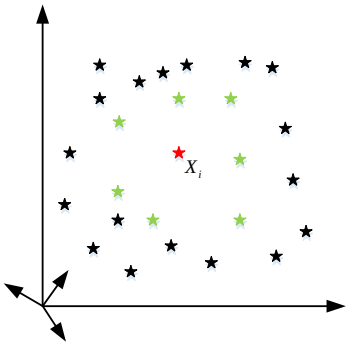
(a) High dimensional data distribution



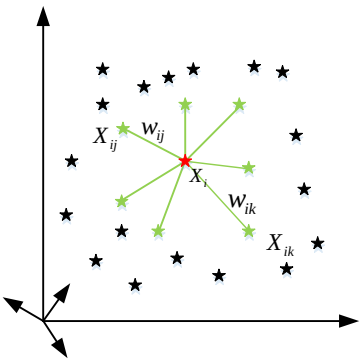
(b) Data distribution after dimensionality reduction

Figure 2: Schematic diagram of the dimension reduction of the manifold algorithm

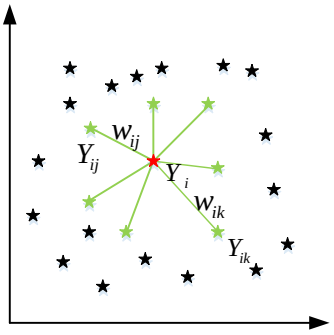
The manifold algorithm is divided into linear manifold and nonlinear manifold. The nonlinear manifold algorithm is further divided into global nonlinear manifold and local nonlinear manifold, and the latter is studied and used for station face recognition. In the process of data dimensionality reduction using the local nonlinear manifold algorithm, Locally Linear Embedding (LLE) and Laplacian Eigenmaps (LE) are used [18,19]. The calculation principle of LLE is to assume that a sample data point has a linear relationship with nearby small range data. After dimensionality reduction, the dimensionality of the reduced sample data point and the nearby data point still maintain the same linear relationship. This linear relationship does not affect the data far from the sample data points and reduces the complexity of dimensionality reduction. The calculation process diagram is shown in Figure 3.



(a) Select Nearest Neighbors



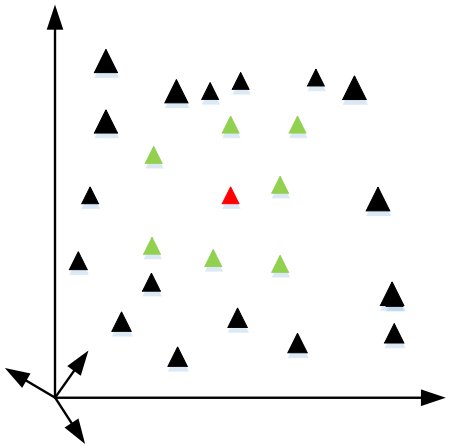
(b) Building linear relationships



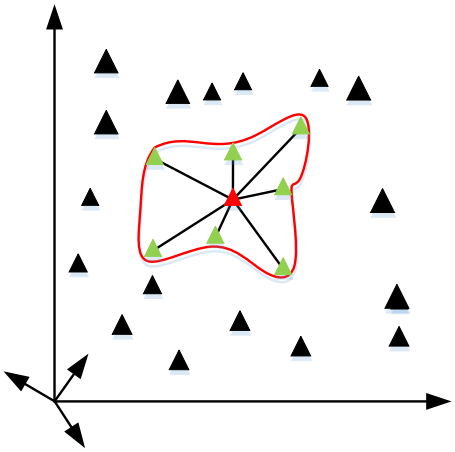
(c) Embedding dimensionality reduction coordinates

Figure 3: Schematic diagram of the LLE calculation flow

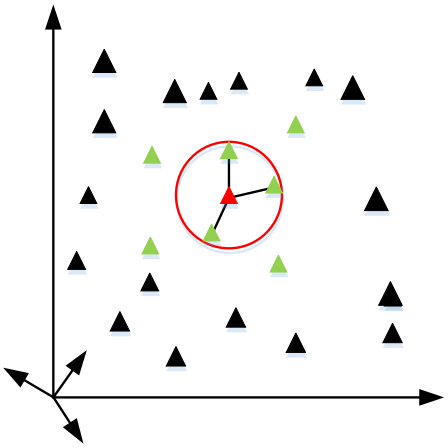
In Figure 3, X_i represents high-dimensional sample data points. X_{ik} and X_{ij} are the data points closest to sample data K. Y_i, Y_{ij} and Y_{ik} are corresponding data points in low dimensional coordinates, respectively. w_{ij} and w_{ik} represent the approximate weights for dimensionality reduction of high-dimensional data X_i. In the process of dimensionality reduction calculation, selecting the appropriate K-nearest neighbor is the key. When the nearest neighbor range is too small, many data components are not connected. When the nearest neighbor range is too large, it may lead to unstable connectivity. The basic principle of LE is to construct relationships between data points based on local similarity of sample data points, and combine them with LLE to select moderate K-nearest neighbors for dimensionality reduction calculations. The construction of a suitable nearest neighbor information graph is shown in Figure 4.



(a) Raw data points



(b) *K*-Nearest Neighbor Point Reconstruction



(c) ε- Nearest Neighbor Point Reconstruction

Figure 4: Schematic of proximity information

From Figure 4, LE converts the K-nearest neighbor linear relationship in high-dimensional data into the ε-nearest neighbor linear relationship. The mathematical expression for the conversion process is expressed in equations (9), (10), and (11).

$$\begin{cases} L_1 \overset{\Delta}{=} D_1 - W \\ D_{ij} = \sum_j w_{ij} \end{cases} \tag{9}$$

In equation (9), L_1 is the Laplace calculation definition. D_1 is a diagonal matrix. W represents the approximate weight matrix. D_{ij} is the element in the diagonal matrix D_1.

$$\begin{cases} \left\| x_i - x_j \right\|^2 < \varepsilon \\ w_{ij} = e^{-\frac{\left\| x_i - x_j \right\|^2}{t}} \end{cases} \tag{10}$$

In equation (10), x_i and x_j are data points in high-dimensional data. ε represents the nearest neighbor range. t is a constant.

$$\sum_{ij} \left\| y^{(i)} - y^{(j)} \right\|^2 w_{ij} = tr(F^T LF) \tag{11}$$

In equation(11), $y^{(i)}$ and $y^{(j)}$ are the data points after dimensionality reduction. F represents the embedding matrix. LF is a generalized feature vector. ML and LPP algorithms have similar attributes. Using ML to improve the LPP algorithm, the improved algorithm has good connectivity, and the use of dimensionality reduced facial image sample data in the algorithm can improve facial recognition accuracy.

3.2 Station FRS based on the Gabor-ML-LPP

The anti-interference ability of traditional face recognition algorithms is weak. Popular learning algorithms do not have strong anti-interference capabilities. Therefore, in stations with high pedestrian

traffic and complex environments, their recognition performance is limited. Therefore, the study proposes to use GW to further improve the improved face recognition algorithms mentioned above. GW is similar to the visual stimulus response in the human visual system, and can directly perform two-dimensional wavelet filtering on facial images. When extracting target images, this algorithm has superiority in local space and frequency domain information [20]. Moreover, the influence of external factors such as light in the environment on this algorithm can be ignored, so this algorithm has great advantages in image feature extraction and is widely used in the field of image recognition. The preliminary improvement of the station face recognition algorithm combined with GW enhances the anti-interference ability of the station face recognition algorithm, thereby improving the accuracy of station passenger facial recognition. GW can accurately capture the spatial frequency and structural features of multiple directions in the local area of the target image during image conversion. GW two-dimensional kernel function can enhance the edge features in the target image, which not only enhances the eye, mouth, and nose features in facial images, but also enhances features such as wrinkles and scars. The two-dimensional kernel function expression of GW is equation (12).

$$\begin{cases} \psi \vec{x} = \dfrac{\left\|\vec{k}\right\|^2}{\sigma^2} \exp\left(-\dfrac{\left\|\vec{k}\right\|^2 \left\|\vec{x}\right\|^2}{2\sigma^2}\right)\left[\exp(i\vec{k}\vec{x}) - \exp\left(-\dfrac{\sigma^2}{2}\right)\right] \\ \vec{k} = \begin{pmatrix} k_v \cos\varphi_u \\ k_v \sin\varphi_u \end{pmatrix} \end{cases} \tag{12}$$

In equation (12), $\vec{x}$ is the image coordinate of the specified position. $\vec{k}$ represents the center frequency of the Gabor wavelet filter. σ is the Gabor wavelet frequency bandwidth constant. φ_u represents the sampling direction, and u is the direction label. $\sin\varphi_u$ and $\cos\varphi_u$ are sine wave function and cosine plane wave respectively. k_v is the Gaussian envelope function, and its calculation formula is equation (13).

$$k_v = k_{\max} / f^v = 2^{-\frac{v+2}{2}} \tag{13}$$

In equation (13), $k_{\max}$ represents the maximum center frequency of the filter. f^v is a filter fixed to a specific frequency band. v represents the velocity of the wave. The two-dimensional Gabor wavelet filter function has real and imaginary parts, and the expression of the real part function is equation (14).

$$\mathrm{Re}\left(\psi\left(\vec{x}\right)\right) = \frac{\left\|\vec{k}\right\|^2}{\sigma^2} \exp\left(-\frac{\left\|\vec{k}\right\|^2 \left\|\vec{x}\right\|^2}{2\sigma^2}\right)\left[\cos(i\vec{k}\vec{x}) - \exp\left(-\frac{\sigma^2}{2}\right)\right] \tag{14}$$

The expression of the imaginary part function is equation (15).

$$\mathrm{Im}\left(\psi\left(\vec{x}\right)\right) = \frac{\left\|\vec{k}\right\|^2}{\sigma^2} \exp\left(-\frac{\left\|\vec{k}\right\|^2 \left\|\vec{x}\right\|^2}{2\sigma^2}\right)\left[\sin\left(\vec{k}\vec{x}\right)\right] \tag{15}$$

Finally, the recognition result of the target image can be obtained by convolution operation between the Gabor wavelet kernel function and the target image, and its calculation formula is equation (16).

$$W_{uv}(z) = I(z) * \psi_{uv}(z) \tag{16}$$

In equation (16), $W_{uv}(z)$ represents the target image recognition result. $I(z)$ is the target image. $\psi_{uv}(z)$ represents the Gabor wavelet kernel function of the target image. Combining two-dimensional Gabor wavelet with LPP, the image feature extraction process is shown in Figure 5.

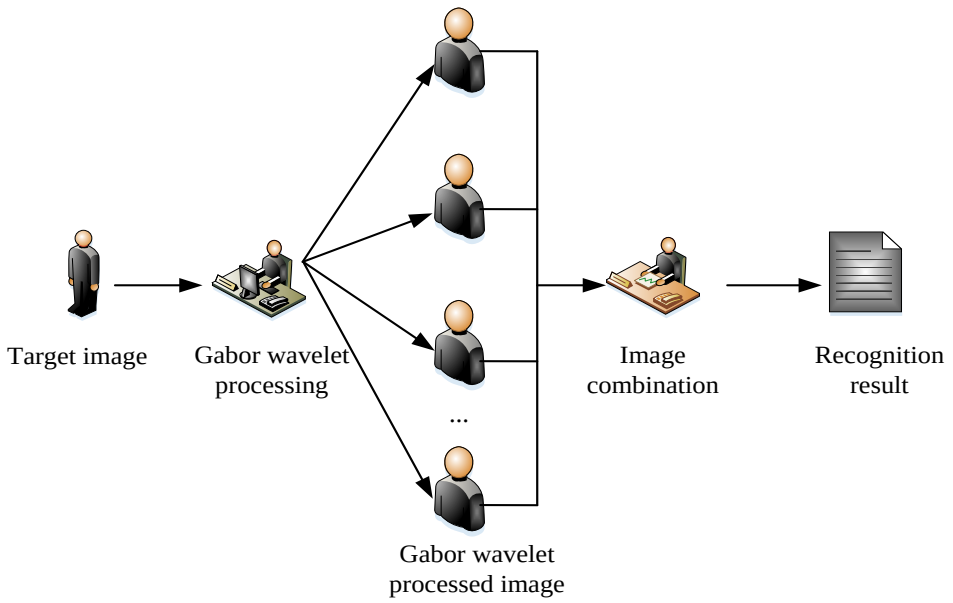


Figure 5: Feature extraction process of fused Gabor wavelets

In Figure 5, the Gabor wavelet filter is first used to process the target image to obtain multi-angle and multi-directional feature images. Then these images are recombined through the operation rules of adding in the same direction and angle. Subsequently, feature extraction is performed on the combined feature images, selecting the optimal features and outputting the recognition results. GW has strong image recognition ability, which can be ignored due to the influence of sample size. It also has strong anti-interference ability, which means it can still maintain high image recognition accuracy in complex environments. Therefore, this algorithm can be used to study the preliminary improvement of station passenger face recognition algorithms to compensate for the shortcomings of station passenger face recognition algorithms. After two improvements, the face recognition algorithm can improve the accuracy of facial recognition for station passengers. After integrating GW, the station FRS workflow diagram is shown in Figure 6.

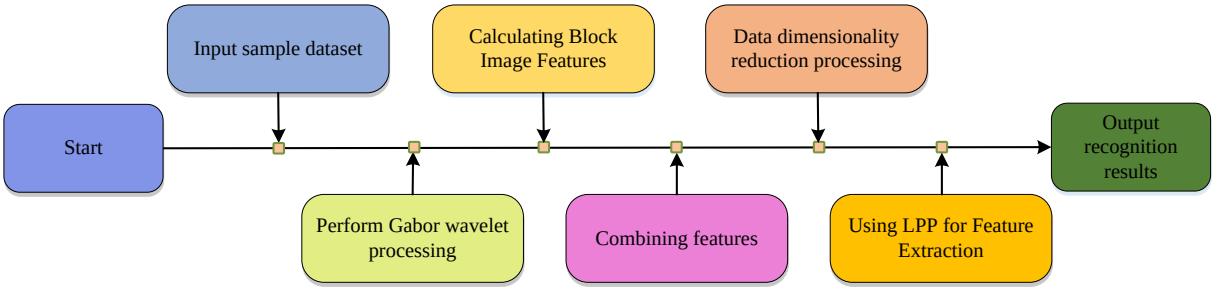


Figure 6: Workflow of the station face recognition system

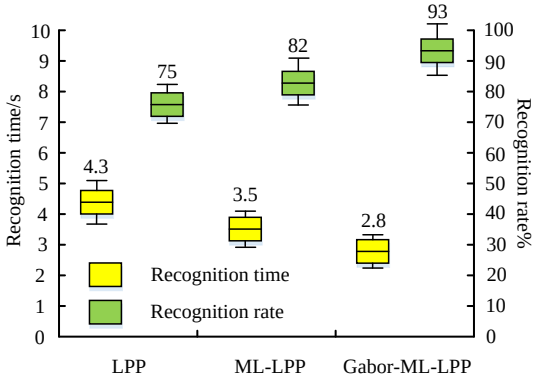
In Figure 6, due to the inability of GW to perform dimensionality reduction on high-dimensional data, the dimensionality of the feature map data after Gabor wavelet processing remains unchanged. Therefore, after Gabor wavelet processing of facial images in the station, ML is used for data dimensionality reduction processing. Subsequently, the LPP traditional face recognition algorithmis used to extract and recognize facial image features again, and the final facial recognition results are output.

4 Performance testing of improved algorithms and empirical analysis of station passenger FRS

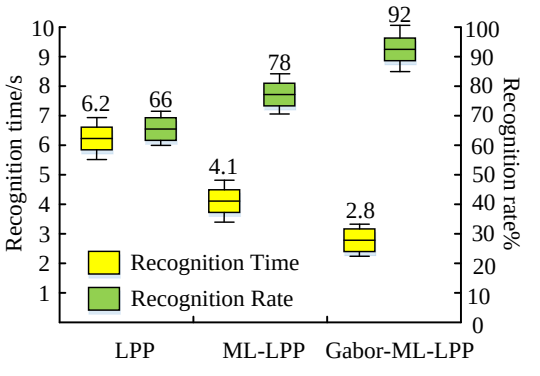
To validate the proposed face recognition algorithm for station passengers based on GW and ML, the study conducts algorithm performance comparison experiments on AR datasets with sufficient facial data samples. The experiment evaluates the performance of the algorithm based on indicators such as running time, recognition rate, accuracy, accuracy-recall rate. Then, to verify the actual application effect of the station passenger FRS, the study uses the average absolute error, the maximum absolute percentage error, the recognition rate, etc. as evaluation indicators to analyze the system application effect.

4.1 Comparative performance testing of gabor-ML-LPP face recognition algorithms

To conduct performance comparison tests on Gabor-ML-LPP, a face recognition algorithm that has undergone two improvements. This study uses traditional human LPP and the improved ML-LPP as the control group. Using these three face recognition algorithms to recognize and detect facial information data in the AR dataset, the experimental results of the algorithm's runtime and recognition rate are shown in Figure 7.



(a) The number of training samples is 4

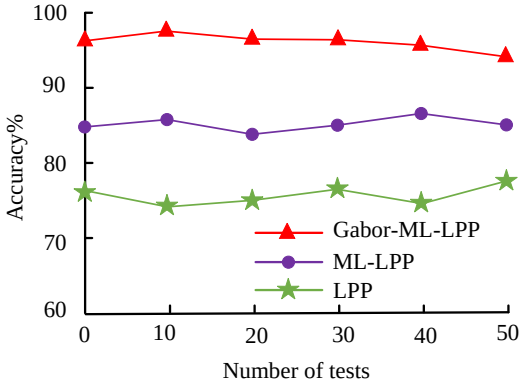


(b) The number of training samples is 6

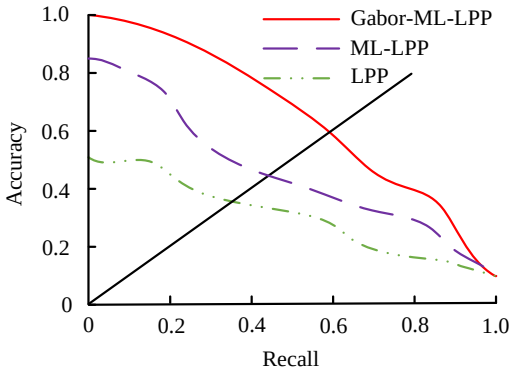
Figure 7: Running time and recognition rates of the three algorithms

In Figure 7(a), when the number of training samples is 4, the running time and recognition rate of LPP are 4.3s and 75%, ML-LPP is 3.5s and 82%, and Gabor-ML-LPP is 2.8s and 93%. When the number of training

samples is 6, the LPP is 6.2 seconds with 66%, ML-LPP is 4.1 seconds with 78%, and Gabor-ML-LPP is 2.8 seconds with 92%. Based on Figure 7(a) and Figure 7(b), Gabor-ML-LPP performs the best in terms of runtime and recognition rate, and its impact is not significant with the number of samples. The results of multiple experiments using these three types of FRTs are displayed in Figure 8, and the algorithm is evaluated using accuracy and accuracy-recall curves.



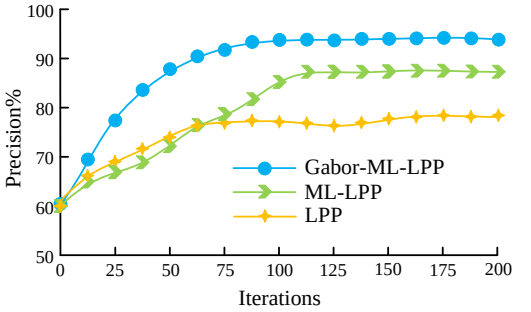
(a) Accuracy curve



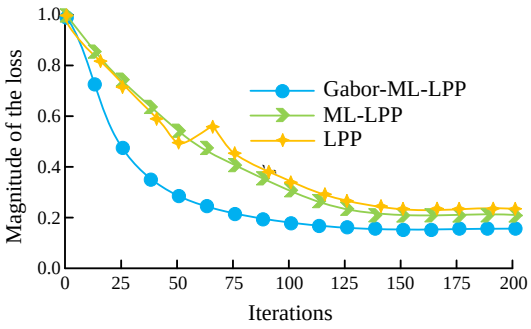
(b) Accuracy recall curve

Figure 8: Accuracy and accuracy-recall curves of the three algorithms

In Figure 8(a), in 50 experiments, the accuracy of Gabor-ML-LPP remains above 90%, with a maximum value of 97%. ML-LPP remains in the range of 80% - 90%, with a maximum value of 86%. The LPP ranges from 70% to 80%, with a maximum value of 76%. In Figure 8(b), the area value under the accuracy-recall curve of Gabor-ML-LPP is 0.87, which is significantly greater than the other two algorithms. A larger area value indicates better algorithm performance. By comparing the results of the above two indicators, Gabor-ML-LPP is superior to the other two algorithms. Subsequently, iterative experiments are conducted using Gabor-ML-LPP, ML-LPP, and LPP algorithms, and the experimental results are shown in Figure 9.



(a)Precision curve



(b)Loss value curve

Figure 9: Accuracy and loss values for the three algorithms

In Figure 9(a), after 200 iterations, the accuracy of Gabor-ML-LPP remains stable at 94%, ML-LPP remains stable at 88%, and LPP remains stable at 79%. Among them, Gabor-ML-LPP has the highest accuracy and fastest convergence speed, which is superior to the comparison algorithm. After 200 iterations, the loss value of Gabor-ML-LPP remains at 0.18, ML-LPP remains at 0.21, and LPP remains at 0.22. Gabor-ML-LPP has the lowest loss value and is superior to the comparison algorithm. In summary, the performance of Gabor-ML-LPP is excellent in all the experimental evaluation indicators mentioned above, and it is expected that this algorithm can improve the accuracy of station passenger FRS. To verify the comprehensive performance of the designed algorithm, the study conducts training on the AR data set, and calculates the average training time, average classification time and best recognition rate of the three algorithms, and reference [4] and reference [5] respectively. The results are shown in Table 2.

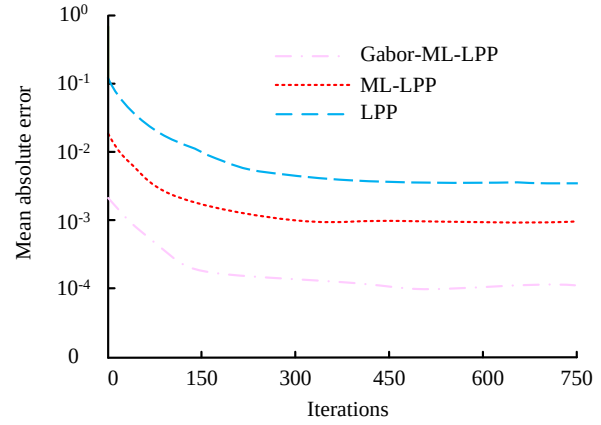
Table 2: Average training time, average classification time and best recognition rate of different algorithms

Algorithm	Average training time (s)	Average classification time (s)	Best recognition rate (%)
LPP	3.09	2.94	93.32
ML-LPP	2.81	2.85	94.75
References [4]	1.52	1.47	97.14
References [5]	1.68	1.32	97.86
Gabor-ML-LPP	0.49	0.68	98.68

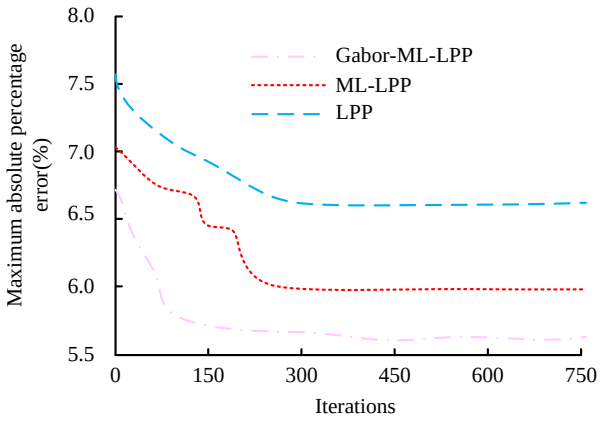
From Table 1, the average training time of Gabor-ML-LPP is 0.49s, the average classification time is 0.68s, and the best recognition rate is 98.68%. The training time and classification time of the designed method are obviously smaller than other methods, and the optimal recognition rate is higher than other methods. The results show that Gabor-ML-LPP has better recognition performance and computational efficiency, which shows that the algorithm realizes the completeness of feature extraction and ensures the timeliness in classification ability.

4.2 Practical application analysis of FRS for station passengers

LPPFRS and ML-LPPFRS are used as control groups in the actual application effect test of Gabor-ML-LPP-based station passenger FRS. The results of the average absolute error and the maximum absolute percentage error of the three FRSs as a function of the number of iterations are displayed in Figure 10.



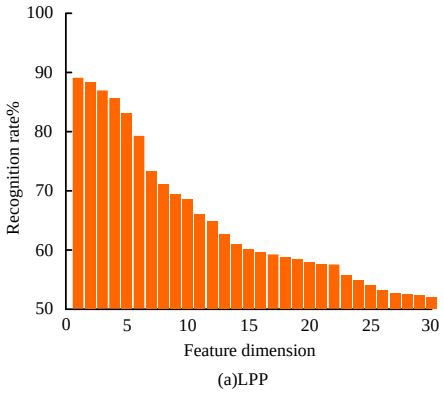
(a) Mean absolute error



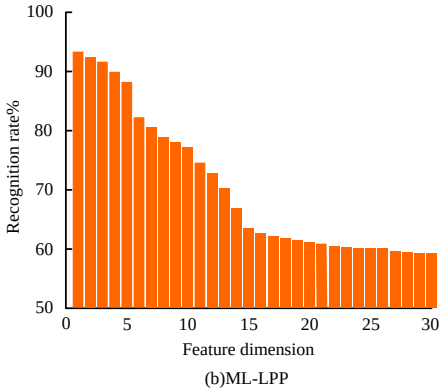
(b) Maximum absolute percentage error

Figure 10: Mean absolute error and maximum absolute percentage error

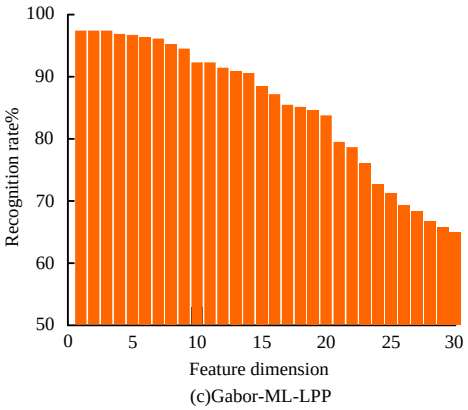
From Figure 10(a), the average absolute error of Gabor-ML-LPP system is stable at 1.14-4, ML-LPP system is stable at 1.99-4, and LPP is stable at 2.51-4 after 750 iterations of operation. In Figure 10(b), the maximum absolute percentage error of Gabor-ML-LPP is stable at 5.65%, ML-LPP is stable at 5.98%, and LPP is stable at 6.68% after 750 iterations. Based on 10(a) and (b), from the error dimension of FRS, the Gabor-ML-LPP system has the best practical application effect, significantly superior to the face bound system used for comparison. This means that the designed method can be better adapted to long-term operation and more stable. The recognition accuracy of these three systems in the face recognition process varies with the feature dimensions in the face data samples, as shown in Figure 11.



(a)LPP



(b)ML-LPP



(c)Gabor-ML-LPP

Figure 11: Recognition rates of the different feature dimensions

From Figure 11 (a), the recognition accuracy of the LPP system continuously decreases with the increasing dimension of sample features, from the initial 89% to 52%. From Figure 11(b) and Figure 11(c), the ML-LPP system decreases from the initial 92% to 59%, and the

Gabor-ML-LPP system decreases from 98% to 66%. Based on Figures 11(a), (b), and (c), the accuracy of the three types of FRS decreases with the increase of sample feature dimensions. Among them, Gabor-ML-LPP not only has the highest initial recognition rate, but also has the strongest ability to resist the adverse effects of dimensional changes, which is superior to the comparison system. The above results show that the designed system can identify passengers more accurately, reduce false detection and missing detection, and improve the efficiency of station security and operation. The comparative experiments are conducted on the adaptability of the FRS proposed in the validation study on three datasets: AR, ORL, and YALE. The experimental results are exhibited in Table 3.

Table 3: Recognition rates for the different training sets

Number of training samples	Dataset Type	LPP	ML-LPP	Gabor-ML-LPP
5	AR	77.19%	88.11%	96.77%
	ORL	75.12%	85.17%	95.12%
	YALE	76.88%	82.55%	97.18%
7	AR	70.23%	84.13%	93.51%
	ORL	69.15%	81.74%	92.88%
	YALE	69.16%	80.85%	93.22%
9	AR	60.32%	79.81%	91.14%
	ORL	62.99%	77.63%	91.02%
	YALE	61.77%	79.66%	90.55%

From Table 3, using AR dataset for comparative experiments, when the number of training samples is 5, 7, and 9, the recognition rates of LPP are 77.19%, 70.23%, and 60.32%. The ML-LPP values are 88.11%, 84.13%, and 79.81%. Gabor-ML-LPP has 96.77%, 93.51%, and 91.14%. Comparative experiments are conducted using the ORL dataset. When the number of training samples is 5, 7, and 9, the recognition rates of LPP are 75.12%%, 69.15%, and 62.99%. The rate of ML-LPP is 85.17%, 81.74%, and 77.63%. The rate of Gabor-ML-LPP is 95.12%, 92.88%, and 91.02%. Using the YALE dataset for comparative experiments: When the number of training samples is 5, 7, and 9, the LPP series is 76.88%, 69.16%, and 61.77%, ML-LPP is 82.55%, 80.85%, and 79.66%, and Gabor-ML-LPP is 97.18%, 93.22%, and 90.55%. The Gabor-ML-LPP system has the highest recognition rate on these three datasets, and the recognition rate is least affected with the increase of training samples. In summary, the practical application effect of the Gabor-ML-LPP-based station travel FRS proposed in the study is better than that of the comparative system, proving the reliability and practicality of the design system.

5 Conclusion

In recent years, the passenger flow at stations has gradually increased, especially during holidays, and the pressure on station security work is enormous. To improve the efficiency of security checks, stations are gradually using FRS for security checks. However, when traditional FRS is implemented in stations, its recognition accuracy is not high. In response to this issue, the study proposed to improve traditional face recognition algorithms using GW and ML, and to construct station passenger FRS based on the improved algorithm. Performance comparison experiments were conducted on the improved algorithm. The results showed that the recognition accuracy of the algorithm was 97%, and the area under the accuracy recall curve was 0.87. After 200 iterations of the algorithm, its accuracy and loss values remained stable at 94% and 0.18 respectively, both better than LPP and ML-LPP. When the number of training samples was 4 or 6, the running time of the algorithm was 2.8 seconds, and the recognition rate ratio was 93% and 912%, both of which were better than the control algorithm. Then the empirical analysis of the practical application effect of the proposed FRS showed that the average absolute error and the maximum absolute percentage error of Gabor-ML-LPP system were 1.14-4 and 5.65%.These data were better than the comparison system. Compared to the comparison system, the recognition accuracy of this system decreased the least for different sample feature dimensions. When the system was compared and tested on different datasets, its recognition rate was higher than that of the comparative FRS. In summary, the proposed GW-ML-based face recognition algorithm and the station passenger FRS constructed based on this algorithm can improve the accuracy of facial recognition in stations. However, the protection function of facial information in this system has not been strengthened, which may lead to information leakage. At the same time, although the running time of the algorithm is shorter, the requirement for real-time performance may be higher in the high-traffic station environment. Therefore, in the future, the system will explore its data encryption technology to ensure that personal privacy is not compromised, and optimize the algorithm to further reduce the uptime to cope with the demand of peak hours.

Funding

The research is supported by: Zhengzhou Police University 2025 Basic Research Funds for Central Universities Special Fund Project. Analysis of the Current Situation and Optimization Path of Police Enforcement on Trains under the "One Train, One Police" Background. (No. 2025TJJBKY043).

References

[1] Yang, J., He, W., Zhang, T., Zhang, C., Zeng, L., & Nan, B. (2020). Research on subway pedestrian detection algorithms based on SSD model. IET Intelligent Transport Systems, 14(11):1491-1496. https://doi.org/10.1049/iet-its.2019.0806

[2] Yan, H., Wang, P., Chen, W. D., & Liu, J. (2015). Face recognition based on gabor wavelet transform

and modular 2dpca//2015 international conference on power electronics and energy engineering. Atlantis Press, 245-248. https://doi.org/10.2991/peee-15.2015.67

[3] Rashid, S. J., Abdullah, A. I., & Shihab, M. A. (2020). Face recognition system based on gabor wavelets transform, principal component analysis and support vector machine. International Journal on Advanced Science Engineering and Information Technology, 10(3): 959-63. https://doi.org/10.18517/ijaseit.10.3.8247

[4] Khan, I. U., Shah, J. A., Bilal, M., Khan, M. S., Shah, S., & Akgül, A. (2023). Machine learning modelling of removal of reactive orange RO16 by chemical activated carbon in textile wastewater. Journal of Intelligent & Fuzzy Systems, 44(5): 7977-7993. https://doi.org/10.3233/jifs-220781

[5] Khalil, K. (2023). Airline flight delays using artificial intelligence in COVID-19 with perspective analytics. Journal of Intelligent & Fuzzy Systems, 44(4): 6631-6653. https://doi.org/10.3233/JIFS-222827

[6] Otani, Y., & Ogawa, H. (2021). Potency of individual identification of japanese macaques (macaca fuscata) using a face recognition system and a limited number of learning images. Mammal Study, 46(1):85-93. https://doi.org/10.3106/ms2020-0071

[7] Wang, S., Ge, H., Yang, J., & Su, S. (2021). Virtual samples based robust block-diagonal dictionary learning for face recognition. Intelligent Data Analysis, 25(5):1273-1290. https://doi.org/10.3233/IDA-205466

[8] Wang, Z., Abhadiomhen, S., Liu, Z., Shen, X., & Gao, W. (2021). Multi-view intrinsic low-rank representation for robust face recognition and clustering. IET Image Processing, 15(14):3573-3584. https://doi.org/10.1049/ipr2.12232

[9] Chen, T., Gao, T., Li, S., Zhang, X., Cao, J., Yao, D., & Li, Y. (2021). A novel face recognition method based on fusion of LBP and HOG. IET Image Processing, 15(14):3559-3572. https://doi.org/10.1049/ipr2.12192

[10] Sun, R., Shan, X., Zhang, H., & Gao, J. (2022). Data gap decomposed by auxiliary modality for NIR-VIS heterogeneous face recognition. IET image processing, 16(1):261-272. https://doi.org/10.1049/ipr2.12350

[11] Wan, Z., Huang, M., Yang, R., Liu, W., & Zeng, N. (2022). EEG fading data classification based on improved manifold learning with adaptive neighborhood selection. Neurocomputing, 482(14):186-196. https://doi.org/10.1016/j.neucom.2021.11.039

[12] Cui, P., Wang, X., & Yang, Y. (2021). Nonparametric manifold learning approach for improved process monitoring. The Canadian Journal of Chemical Engineering, 100(1):67-89. https://doi.org/10.1002/cjce.24066

[13] Ran, R., Feng, J., Zhang, S., & Fang, B. (2020). A general matrix function dimensionality reduction framework and extension for manifold learning. IEEE Transactions on Cybernetics, 4(52):2137-2148. https://doi.org/10.1109/TCYB.2020.3003620

[14] Dornaika, F. (2020). Multi-layer manifold learning with feature selection. Applied Intelligence, 50(6):1859-1871. https://doi.org/10.1007/s10489-019-01563-9

[15] Duan, Y., Huang, H., Li, Z., & Tang, Y. (2020). Local manifold-based sparse discriminant learning for feature extraction of hyperspectral image. IEEE Transactions on Cybernetics, 51(8):4021-4034. https://doi.org/10.1109/tcyb.2020.2977461

[16] Khan, I. U., & Aftab, M. (2022). Dynamic programming approach for fuzzy linear programming problems FLPs and its application to optimal resource allocation problems in education system. Journal of Intelligent & Fuzzy Systems, 42(4): 3517-3535. https://doi.org/10.3233/JIFS-211577

[17] Khan, I.U., & Rafique, F. (2021). Minimum-cost capacitated fuzzy network, fuzzy linear programming formulation, and perspective data analytics to minimize the operations cost of American airlines. Soft Computing, 25(2): 1411-1429. https://doi.org/10.1007/s00500-020-05228-5

[18] Khan, I. U., & Karam, F. W. (2019). Intelligent business analytics using proposed input/output oriented data envelopment analysis DEA and slack based DEA models for US-airlines. Journal of Intelligent & Fuzzy Systems, 37(6): 8207-8217. https://doi.org/10.3233/JIFS-190641

[19] Khan, I. U., Ahmad, T., & Maan, N. (2019). Revised convexity, normality and stability properties of the dynamical feedback fuzzy state space model (FFSSM) of insulin-glucose regulatory system in humans. Soft Computing, 23: 11247-11262. https://doi.org/10.1007/s00500-018-03682-w

[20] Samantaray, A., & Rahulkar, A. (2020). New design of adaptive Gabor wavelet filter bank for medical image retrieval. IET Image Processing, 14(4):679-687. https://doi.org/10.1049/iet-ipr.2019.1024

VMAPN: A Vision Mamba-Based Real-Time Behavior Recognition and Feedback System for Smart Education

Zili Chen[1,3], Aihong Liu[2*]
[1]School of Artificial Intelligence and Big Data, Chongqing College of Science and Creation, Yongchuan, 402160, China
[2]Fan Changjiang School of Journalism, Neijiang Normal University, Neijiang, 641100, China
[3]College of Physics and Information Engineering, Zhaotong University, Zhaotong, 657000, China.
E-mail: AiHongLiuu@outlook.com
[*]Corresponding author

Keywords: vision mamba, intelligent education, real-time feedback, learning behavior recognition, teaching evaluation Model (TEM), Augmented Prototypical Network (APN)

Received: April 29, 2025

With the growing demand for intelligent educational technologies, traditional methods of classroom evaluation—often reliant on subjective observations—fall short in scalability, objectivity, and timeliness. This paper proposes a real-time automatic scoring and feedback system for intelligent educational platforms, powered by the Vision Mamba architecture. Vision Mamba—a novel and lightweight spatio-temporal modeling backbone—is integrated with an enhanced classification module, the Augmented Prototypical Network (APN) to form the VMAPN framework, which enables accurate behavior recognition under small-sample conditions. The system analyzes classroom video streams to identify student behaviors such as attentiveness, participation, and posture, and generates adaptive feedback for both students and teachers. Real-time feedback is delivered through an interactive interface, while backend analytics leverage machine learning techniques to monitor learning engagement and evaluate teaching effectiveness. Furthermore, the proposed Teaching Evaluation Model (TEM) classifies student behaviors into four categories—positive, relatively positive, neutral, and negative—to derive objective teaching effect scores. Experimental results on THUMOS 2014 and ActivityNet v1.3 datasets validate the model's predictive performance, achieving mAP scores of 64.7% at IoU 0.5 on THUMOS 2014 and 72.3% at IoU 0.5 on ActivityNet v1.3, representing improvements of 6.2% and 4.8%, respectively, over baseline methods.

Povzetek: Predlagan je sistem, ki v realnem času iz video posnetkov samodejno prepoznava vedenje učencev ter učiteljem in učencem zagotavlja objektivno povratno informacijo in oceno pouka.

1 Introduction

In traditional classroom teaching, evaluation often relies on expert observations and manually filled evaluation forms, which are typically summarized over multiple sessions to assess instructional effectiveness. Although such expert-based evaluations play a supervisory role, they are inherently limited by subjectivity, randomness, and inefficiency. With the rapid development of artificial intelligence (AI), novel approaches have emerged that allow for real-time, objective, and scalable assessment of classroom teaching by analyzing student behavior and engagement through data-driven models [1].

Recent advances in computer vision and deep learning have enabled automated recognition of learning behaviors from video data [2]. These technologies can be applied to intelligently monitor classroom interactions, analyze student attentiveness, engagement, and participation, and ultimately provide meaningful insights into teaching quality. Compared with traditional methods such as time-sampling, manual coding (e.g., FIAS, ITIAS) [3], or ethnographic observation, AI-powered behavior analysis systems offer greater consistency, scalability, and automation, significantly reducing the burden on educators and improving the timeliness of feedback. Despite the increasing integration of online education, the physical classroom remains the primary environment for large-scale youth education [4]. However, the digitization and informatization of classroom assessment lag behind due to reliance on inefficient manual methods. Meanwhile, international and national policy initiatives—such as China's Education Informatization 2.0 Action Plan and UNESCO's AI in Education—emphasize the importance of smart education platforms, encouraging the adoption of AI, deep learning, and intelligent feedback systems in education.

In response to these challenges and policy directions, this study proposes a real-time automated assessment and feedback system for intelligent education platforms, powered by the Vision Mamba architecture. Vision Mamba, as a cutting-edge vision backbone model,

provides efficient and robust spatio-temporal modeling capabilities suitable for video-based behavior recognition. Leveraging this architecture, we design a novel framework that integrates deep video behavior recognition with AI-based decision-making, enabling real-time evaluation of classroom learning activities and personalized feedback delivery. Additionally, we address the growing demands of online assessments, where large-scale exam proctoring suffers from inadequate human supervision. By incorporating intelligent video analysis, the proposed system can automatically detect cheating behaviors during online examinations, alert invigilators in real-time, and improve the fairness and reliability of online assessments. This not only alleviates the monitoring burden on educators but also significantly enhances the scalability and integrity of online education systems.

In summary, this paper investigates a Vision Mamba-powered intelligent education platform capable of automatically scoring and providing feedback in real-time. By integrating advanced video behavior recognition techniques with scalable AI systems, we aim to bridge the gap between traditional and smart classroom evaluation, promote educational informatization, and support precise, data-driven instruction and assessment strategies. To address these challenges, the main objectives of this study are as follows:

(i) Validate the effectiveness of Vision Mamba for behavior recognition tasks under small-sample conditions in educational settings;

(ii) Construct a scalable, real-time scoring and feedback system integrating behavior recognition with interactive interfaces;

(iii) Benchmark the proposed VMAPN framework against classical backbone models (e.g., ConvNets, ViT) on educational behavior datasets such as THUMOS 2014 and ActivityNet v1.3.

2 Related work

2.1 Video-based behavior recognition

The task of behavior recognition from video data is significantly more complex than image-based analysis due to the temporal dimension inherent in video. Three mainstream approaches dominate current research in video understanding: two-stream networks [5], 3D convolutional networks [6], and self-attention-based models [7]. Two-stream networks laid the foundation for video classification by processing spatial and temporal features separately spatial features via RGB frames and motion via optical flow [8]. Follow-up works introduced various fusion strategies and segment-based temporal modeling to enhance long-term temporal understanding. Despite their success, two-stream methods suffer from high computational and storage costs due to the pre-computation of optical flow, limiting their real-time applicability. 3D convolutional networks address video modeling by extending 2D convolutions across the temporal axis [9]. Models like C3D, I3D, and R(2+1)D have demonstrated promising performance on standard video benchmarks [10]. Efficient variants such as X3D and MoViNet optimize latency and memory usage, enabling real-time inference [11]. Recent innovations like SlowFast and TDN further enhance temporal resolution by introducing multi-scale or dual-pathway designs [12]. Self-attention-based models, especially those adapted from the Transformer architecture, offer superior global temporal reasoning. Non-local networks, TimeSformer, and MViT utilize token-based representations to model long-range dependencies [13]. Although these approaches outperform previous architectures in accuracy, their high computational demands necessitate large-scale datasets and careful optimization. Lightweight alternatives such as X-ViT and MorphMLP seek to balance efficiency with performance through architectural simplifications or prompt-based learning [14].

2.2 Smart education and student behavior analysis

As AI-powered education platforms become increasingly prevalent, researchers are exploring how computer vision techniques can be used to quantify and analyze student behaviors in real classroom settings. Traditional manual coding and observation methods, such as FIAS or ethnographic analysis [15], are insufficient for large-scale implementation due to subjectivity and labor intensity. To address this, several interdisciplinary efforts have emerged. Researchers have utilized Kinect-based skeleton tracking combined with machine learning classifiers for classroom posture recognition [16]. Other works adopted multimodal approaches—integrating speech, body movement, and head tracking—to improve interaction analysis and engagement detection. Advances in deep learning have further enabled end-to-end student behavior recognition systems based on CNNs, residual networks, and GANs, achieving high classification accuracy and fast convergence. In online and hybrid classroom contexts, gesture and pose recognition have been successfully employed to track participation and detect behaviors such as hand-raising, standing, or sleeping. For instance, OpenPose-based models have demonstrated robust skeletal tracking for real-time activity classification [17]. Recent studies have explored fine-grained behavioral datasets for supervised learning tasks using deep residual networks, CNN-10, and ensemble models such as XGBoost, reaching behavior classification accuracies upwards of 93% [18]. Recent AI-driven feedback systems have explored multi-modal data integration, offering promising results for adaptive learning [26, 27].

2.3 Pose estimation and classroom activity recognition

Pose estimation has become a key enabler of student behavior recognition. Enhanced OpenPose-based approaches and multi-view 3D pose estimation systems improve accuracy and reliability [19], even in occluded or crowded classroom environments. Methods combining

pose estimation with deep learning models [20], such as CNNs or Boosting frameworks, have shown excellent performance in real-world scenarios, with several models achieving over 90% accuracy on customized classroom datasets. The integration of pose-based analysis with spatio-temporal models provides a promising direction for building scalable [21], real-time educational assessment systems. However, many existing methods still face limitations in generalization across diverse classroom layouts, lighting conditions, and student postures. These challenges motivate the need for lightweight, real-time, and generalizable architecture—such as Vision Mamba, capable of efficient behavioral modeling and feedback in intelligent education systems.

To provide a concise comparison of mainstream video-based behavior recognition methods, Table 1 summarizes key architectural approaches, tasks, datasets, and performance metrics, highlighting the advantages of Vision Mamba for small-sample educational scenarios.

Table 1: Comparative summary of video-based behavior recognition approaches

Method	Architecture Type	Target Task	Datasets Used	mAP / Acc
C3D [9]	3D CNN	Action Classification	UCF101, Sports-1M	~51.6%
I3D [10]	3D CNN (Inflated)	Action Recognition	Kinetics-400, UCF101	~66.4%
SlowFast [12]	Dual-pathway CNN	Action Recognition	Kinetics-400, AVA	~77.9%
TimeSformer [13]	Transformer-based	Temporal Reasoning	Something-Something V2	~62.1%
VMamba [22]	State-Space Model	Spatio-temporal Modeling	Synthetic, ImageNet-1K	~73.4%
Vision Mamba (ViM)	Bi-directional SSM	Behavior Recognition	THUMOS 2014, ActivityNet	64.7% / 72.3%

3 Method

To achieve real-time automated scoring and feedback in intelligent education platforms, we propose an advanced framework leveraging the Vision Mamba architecture. This section details the methodology underlying our proposed system, including feature extraction, model architecture, classification process, loss computation, and pretraining strategies.

3.1 Architecture overview

The overall framework of the Vision Mamba-based Automated Scoring and Feedback System (VMASFS) is illustrated in Figure 1. The system consists of two primary components: feature extraction using Vision Mamba and automated decision-making through an Augmented Prototypical Network (APN). Vision Mamba (ViM) serves as the backbone model, extracting spatio-temporal features from video sequences of classroom activities and student behaviors. These extracted features are then fed into the Augmented Prototypical Network (APN) for behavior classification, engagement scoring, and real-time feedback generation.

The proposed framework effectively integrates global and local feature representations. The global category-labeled features capture high-level semantic information about the overall student engagement, while local block features provide fine-grained analysis of individual postures, facial expressions, and interactions. By fusing these features, the system enhances behavior recognition accuracy and instructional feedback precision.

The final step in the framework involves real-time decision adjustment through an adaptive scoring mechanism, ensuring the generated feedback is both personalized and pedagogically relevant. This intelligent assessment and feedback loop facilitates a scalable and automated evaluation process, reducing the reliance on manual observation while improving the consistency and objectivity of educational assessments.

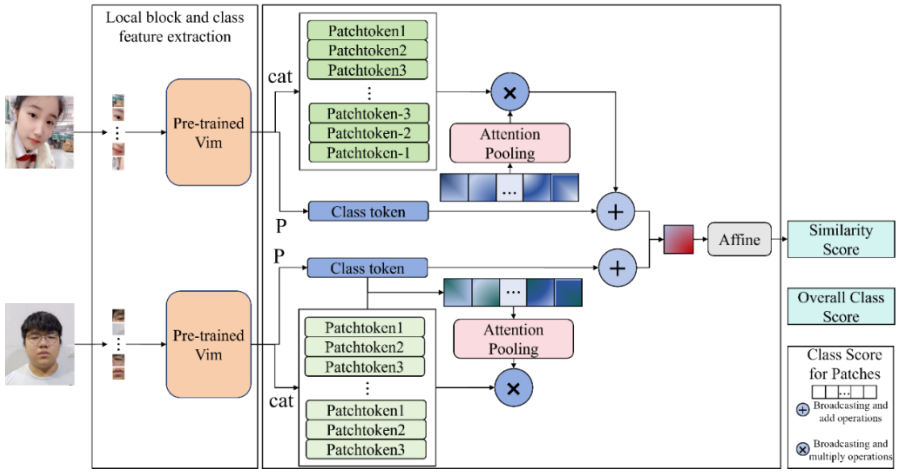


Figure 1: Overview of the VMAPN-based real-time behavior classification and feedback model.

3.2 VMAPN network design

The network architecture of VMAPN consists of a feature embedding module and a classification module. The feature embedding module is dedicated to creating an embedding space that can transform raw data into information-rich feature vectors. Traditional small-sample classification algorithms usually use smaller network architectures, such as Conv-4 and ResNet-12, to ensure fairness in comparison. However, with the rapid development of computer vision technology, these architectures have become obsolete and may lead to poor classification accuracy. To solve this problem, subsequent studies have introduced larger and more advanced network architectures, such as ViT, which provide significant improvements in small-sample classification accuracy. Current ViT-based methods are nearing saturation, and a large number of studies have begun to explore further optimization of classification modules based on ViT and its variants.

This study explores a new network architecture used as a feature embedding module for small samples. Recently, an improved state-space model, Mamba [22], has received a lot of attention. Mamba improves on the S4 model by fusing time-varying parameters, which enables the model to dynamically select relevant information based on the input data. In addition, Mamba introduces a hardware-aware algorithm to improve the execution efficiency of the algorithm. The Mamba model performs strongly in natural language processing tasks and has recently been adapted for visual applications. For example, Vision Mamba (ViM) integrates Mamba into a Transformer-like architecture to address visual challenges using a bi-directional state-space model. In contrast, VMamba introduces a cross-scanning mechanism that links one-dimensional sequences with two-dimensional image structures.

Although there is no study that directly applies Mamba to small-sample scenarios, the above discussion demonstrates the great potential of Mamba for small-sample learning. the time-varying parameters of Mamba allow the model to dynamically adjust its internal parameters according to the input data. This flexibility is particularly suitable for dealing with small-sample data because it can effectively utilize the limited data resources to filter out irrelevant information, while adjusting its behavior according to the needs of each specific task to optimize the whole learning process, which will enhance the model's ability to generalize across different tasks. Moreover, experiments have proved that Mamba has strong modeling capability. Compared to ViT, Mamba achieves a new balance of speed and accuracy, with lower computational complexity while maintaining efficient performance. The main goal of this study is to explore and validate the application of Mamba in small-sample learning, and further optimization of speed will follow.

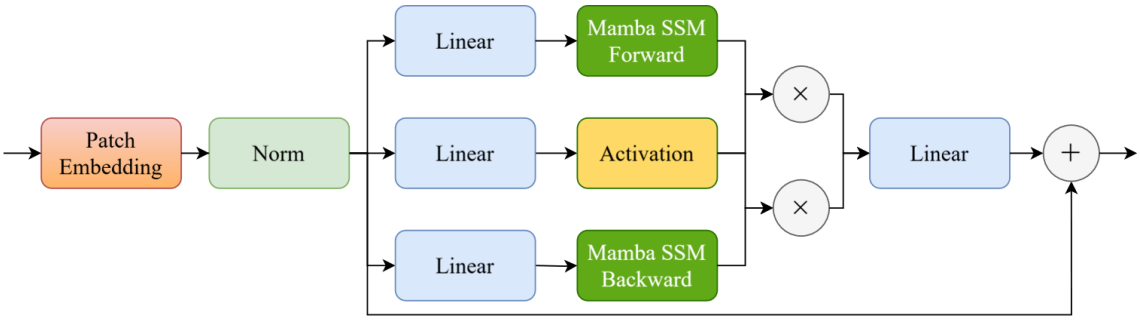


Figure 2: Bi-SSM architecture diagram

In this paper, ViM is adopted as the feature embedding module to address the small-sample classification challenge. Although ViM is not a Transformer, its architecture borrows certain structural elements—such as patch-based input encoding and positional embeddings—that allow for a fair comparative baseline with ViT. Unlike ViT, which relies on self-attention to capture long-range dependencies, ViM uses a bidirectional state-space model (Bi-SSM) to achieve global contextual awareness. This design enables ViM to process sequences with lower computational complexity while still capturing temporal and spatial relationships. Importantly, while ViM shares some high-level architectural traits with Transformers, it does not employ attention mechanisms, nor does it introduce image-specific inductive biases. Instead, its performance stems from dynamic sequence modeling and data-dependent parameterization. Further, research [23] has provided supervised pre-training of ViM on ImageNet1K. The applicability of ViM is not limited to supervised learning tasks, and this study extends ViM to self-supervised tasks by pre-training ViM with masked image modeling to obtain more general features. This pre-training strategy can further improve the accuracy of the model in small-sample classification, allowing it to learn from limited data and generalize to new and unseen tasks more efficiently.

The feature vectors extracted by the feature embedding module are fed into the classification module for classification. The prototype network is the baseline method for the small-sample classification module, which categorizes query samples by comparing the feature similarity between the query samples and the category prototypes. The category prototype is obtained by averaging the features of all samples in the same category. These sample features can be represented as single vectors obtained from the feature maps compressed by the network, or as category labeled features generated by a ViT-like architecture. The category labeled features aggregate the global information of the entire image, and thus are often used as the basis for the final classification decision. In small sample scenarios, due to the limited number of samples, the global information of each category is much less rich than that in large-scale datasets. Therefore, we need to make full use of the available information, such as the local block features of each local region after image segmentation. These features cannot be fully obtained by relying only on the category tagging features under the small sample limitation. Fusing the local block features into the category labeling features can provide a richer selection of features for small-sample classification. It is worth noting that not all local block features are useful for the classification task, such as large background areas or areas not related to category labels, which may be irrelevant local image blocks and need to be filtered out to find out the class-related local block features.

Based on this, this study improves the prototype network to further enhance the performance of small-sample image classification by connecting the Augmented Prototypical Network (APN) to the output of Vi M. By combining the efficient features of Vi M with the APN, the APN can be used for the classification of small-sample images, and the APN can be used for the classification of small-sample images. The design of VMAPN is based on Vision Mamba and Augmented Prototypical Network by combining the efficient feature extraction capability of Vi M with the fine classification capability of APN, which is designed to take into account the representativeness of the category labeling features and the importance of the local block features in the small-sample scenarios. Combining local block features to refine the category labeling vector is a better choice. By integrating features from different levels or regions into the category labeling features, the information of the image can be expressed more comprehensively and a richer prototype representation can be constructed. After each input image $x \in R^{H\times W\times 3}$ is processed by Vision Mamba (ViM), the model outputs; A global category-labeled feature vector $\mathbf{f}_g \in \mathbb{R}^d$, where $d = 768$ is the embedding dimension; A set of local block features $\{f_j\}_{j=1}^{K}$, with each $f_j \in R^d$, extracted from *K* spatial positions in the feature map (e.g., $K = 49$ for a 7×7 grid).

The similarity between each local feature f_j and the global feature f_g is computed using cosine similarity:

$$s_j = \frac{f_j \cdot f_g}{|f_j| \cdot |f_g|}, j = 1,2, \dots, K$$

The weights α_j for each local block are obtained by applying the softmax function over all similarities:

$$\alpha_j = \frac{\exp(s_j)}{\sum_{k=1}^{K} \exp(s_k)}$$

The weighted local feature representation is then computed as:

\[

$$f_{local} = \sum_{j=1}^{K} \alpha_j f_j$$

\]

The final feature vector $\mathbf{f} \in \mathbb{R}^{2d}$ used for classification is obtained by concatenating the global feature and the weighted local feature:

$$f = f_g \oplus \boldsymbol{f}_{local} \in R^{2d}$$

In the support set, class prototypes are computed by averaging the feature vectors of all support samples in each class. During inference, each query feature vector is compared to all class prototypes using a similarity metric (e.g., Euclidean or cosine distance), and a classification score is computed accordingly. An affine transformation is applied to adjust the decision boundary:

$$S = scale * \left[metric\left(f_{class}, f_{query}\right) + bias\right]$$

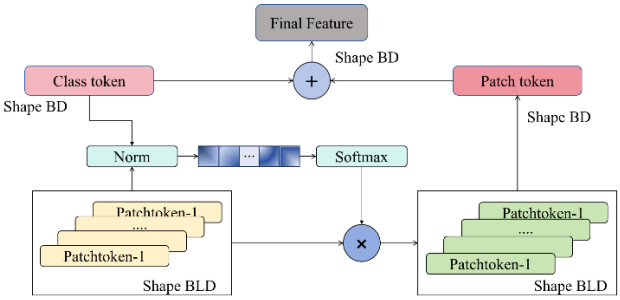


Figure 3: APN architecture diagram

3.3 Loss calculation

The similarity score matrix S is converted to a log-probability matrix P. The similarity score s_{ij} for the ith query sample in S and the jth support set category is converted to a log-probability $\log(p_{ij})$ in the following manner, where N is the number of categories.

$$log(p_{ij}) = log\left(\frac{exp(s_{ij})}{\sum_{j=1}^{N} e\,xp(s_{ij})}\right)$$

Design a label-smoothed one-hot matrix Y with true category labels $y_i^Q \in \{1,2,\ldots,N\}$ for the i th query sample, with one-hot taking the value y_{ij} as:

$$y_{ij} = \begin{cases} 1-\varepsilon & if(j = y_i) \\ \dfrac{\varepsilon}{N-1} & otherwise \end{cases}$$

where ε is a smoothing parameter that takes the value of 0.1. The cross-entropy loss is computed using the label-smoothed one-hot matrix Y with the log-probability matrix P.

$$L = -\frac{1}{Q}\sum_{i=1}^{q}\sum_{j=1}^{N} y_{ij} * log(p_{ij})$$

3.4 Pre-training strategies

Pre-training techniques have become a key strategy in the field of small-sample learning. The generalization ability and learning efficiency of the model can be effectively improved by appropriate pre-training techniques. Literature [22] has achieved significant results in self-supervised pre-training of Vision Transformer on MiniImageNet. Literature [10] explores the impact of different combinations of network structures and pre-training techniques on small sample learning on larger datasets such as ImageNet-1K. Supervised pre-training shows significant advantages with large-scale datasets, where a large amount of data provides diverse samples that can help ViM learn more generalized feature representations, thus showing good results on unknown data as well. However, it is less effective on small-scale datasets, where the restricted labeling information tends to lead to overfitting, making the model usually less adaptable and generalizable to new categories.

In this study, we adopted two pretraining strategies. For the supervised setup, we utilized publicly available pretrained weights of ViM on ImageNet-1K, which includes approximately 1.2 million labeled images and was trained for 300 epochs. For the self-supervised approach, we followed the method used in literature [17] and performed masked image modeling (MIM) on the MiniImageNet training set (~100,000 images), using 40% random masking over 200 training epochs. These settings allow the model to capture more generalizable features when labels are scarce.

By dividing the image into multiple patches, randomly masking certain patches, and letting ViM encode and reconstruct these patches, the model can develop a deeper understanding of image structure and content. This helps it learn generic underlying features, rather than merely class-specific signals. This approach provides a more adaptive learning mechanism for small-sample classification tasks and is especially effective when labeling data is limited. In this study, both supervised and self-supervised pretraining strategies were explored to assess their suitability for small-sample scenarios. However, for the VMAPN model evaluated in Section 5, we exclusively used the publicly available supervised ViM weights pretrained on ImageNet1K, due to resource and reproducibility constraints. The discussion of self-supervised masked image modeling (MIM) on MiniImageNet is presented as a potential direction and proof of concept, not part of the current system's deployed configuration. Future work will include comparative training using self-supervised ViM variants to validate performance gains in low-label environments.

4 Design of automatic grading and feedback system for educational platforms

To implement the proposed Vision Mamba-powered intelligent behavior recognition framework in a real-world educational context, this section introduces the comprehensive design of an automatic scoring and feedback system suitable for both classroom instruction and online assessments. The system integrates frontend interface design, backend data processing, real-time behavior recognition, and intelligent feedback generation. It is built on a scalable and modular architecture that supports both usability and computational efficiency. Furthermore, the system takes into account the distinct roles and expectations of students and teachers, ensuring personalized feedback and decision support across different teaching scenarios.

4.1 User role definitions and functional requirements

The system is designed with two primary user roles in mind: student users and teacher users. The needs of each group were systematically analyzed to guide the system's functional and interface design. For student users, ease of access and clarity of interaction are essential. Students require a simple yet informative interface where they can easily participate in assessment tasks and receive detailed feedback on their performance. The feedback must be timely, personalized, and grounded in objective behavioral analysis, helping students identify their strengths and weaknesses in a data-informed manner.

Teacher users, on the other hand, require broader control over assessment tasks, student data, and feedback

customization. Their core expectations include the ability to publish and modify tests, monitor student engagement in real-time, and analyze behavior trends across sessions. Furthermore, teachers expect the system to assist in grading, detect behavioral anomalies, and generate high-level instructional recommendations. To fulfill these expectations, the system integrates intelligent data pipelines that transform video-based behavior recognition into practical, actionable educational insights.

4.2 System architecture and technical stack

The overall system architecture consists of three tightly integrated layers: the user-facing frontend, the intelligent behavior recognition engine, and the backend data management and analysis module. The entire platform is developed using Python, with Django serving as the central web framework due to its rapid development capabilities, integrated ORM, and support for user authentication and routing.

The frontend interface employs standard web development technologies including HTML, CSS, JavaScript, and Bootstrap, ensuring responsive design and cross-platform compatibility. Asynchronous JavaScript and XML (AJAX) is used to enhance user interaction by allowing page updates without full reloads, which is particularly important for displaying real-time feedback and notifications during assessments.

At the core of the system lies the Vision Mamba and Augmented Prototypical Network (VMAPN) module, responsible for extracting spatio-temporal features from classroom or examination video streams. These features, including attention level, posture stability, and gesture activity, are processed in real-time and fed into the scoring engine. The backend supports secure data storage, task processing, and intelligent analytics using a combination of Pandas for data transformation and Scikit-learn for machine learning-based prediction and clustering. This structure allows the system to provide real-time analysis while maintaining long-term performance tracking and personalization. The general architecture of the system is illustrated in Figure 4, highlighting the interaction between user interfaces, backend modules, data processing pipelines, and the Vision Mamba-powered intelligent behavior engine.

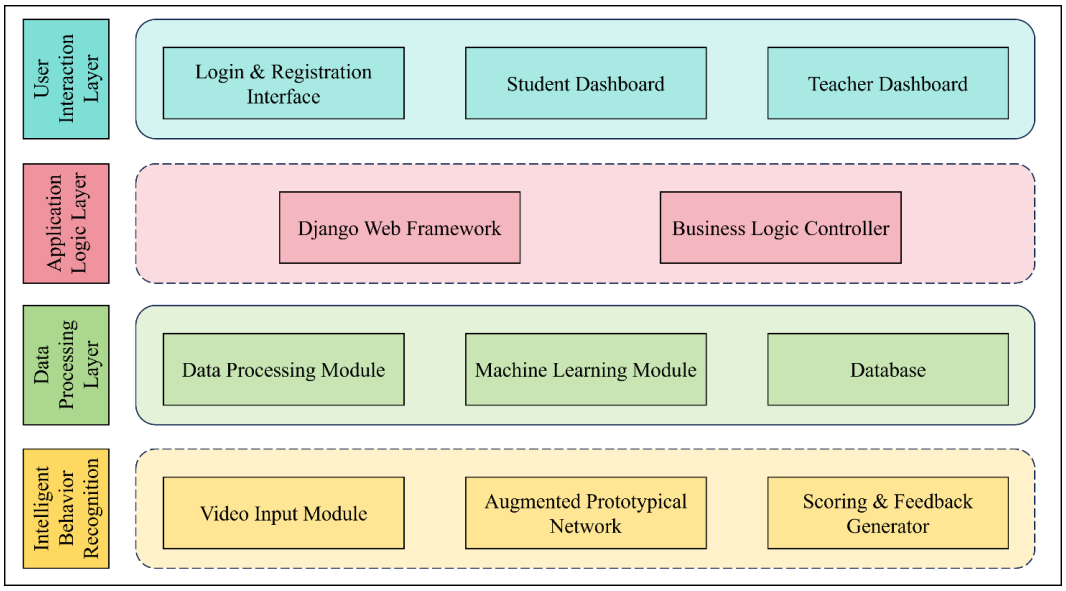


Figure 4: General architecture of the real-time automatic scoring and feedback system based on Vision Mamba and APN.

4.3 Frontend design for intelligent interaction

The frontend of the system is structured into three main interfaces: a unified login and registration portal, a student interaction dashboard, and a teacher management console. The login and registration module supports role-based access control and uses secure authentication mechanisms to manage users and protect data. Once authenticated, users are redirected to role-specific environments.

The student dashboard is designed to facilitate seamless participation in evaluations, access real-time feedback, and track personal progress. Upon completing an assessment task, the system immediately displays behavioral scores generated by the VMAPN model, along with visual feedback such as attention timelines, engagement heatmaps, and suggestions for improvement. This feedback enables students to reflect on their learning habits and adjust accordingly.

The teacher dashboard provides task management capabilities including test creation, question editing, scheduling, and evaluation settings. Teachers can also view student behavior analytics in aggregated or individual form. Real-time alerts for inattention or anomalous behavior (e.g., potential cheating during online assessments) are displayed prominently to support immediate pedagogical intervention. The interface also

enables teachers to download session reports and receive system-generated suggestions for improving classroom interaction based on behavioral trends.

4.4 Backend system design and intelligent analytics

The backend is the core operational layer of the system and is responsible for data processing, behavior evaluation, and task logic management. It comprises three essential modules: the database layer, the data processing pipeline, and the business logic controller.

The database design follows a relational model and supports core entities such as User, Course, Task, VideoLog, Score, and Feedback. Data normalization, indexing, and foreign key constraints are applied to ensure consistency and retrieval efficiency. Data integrity is enforced through strict validation rules, and backups are automated to guarantee recovery under system failure.

The data processing module uses Pandas to preprocess data collected from behavior analysis, such as converting timestamps, handling missing values, and encoding categorical variables. The Scikit-learn library supports multiple analytics workflows, including regression-based performance prediction and K-means clustering for engagement profiling. For example, a regression model may predict test scores based on attendance, attention scores, and prior performance. A clustering algorithm may segment students into high, medium, and low engagement groups, helping teachers apply differentiated instructional strategies.

Most importantly, the behavior recognition engine leverages the Vision Mamba backbone for efficient spatio-temporal modeling. Video frames are processed in real time to extract both global category-labeled features and local block features. These are fed into the Augmented Prototypical Network, which performs similarity-based classification using weighted aggregation of relevant local blocks. The final feature representation is compared with class prototypes, and classification scores are adjusted through an affine transformation before being passed to the feedback module.

To ensure real-time performance, the backend system adopts multithreaded task queues using Python's concurrent.futures.ThreadPoolExecutor for parallel processing of video frames, behavior inference, and database writes. Caching mechanisms are implemented via Redis for frequently accessed inference outputs and session metadata, reducing repeated computation and database load. The behavior recognition module is accelerated using GPU inference (NVIDIA RTX 3060, 12GB VRAM) via PyTorch’s CUDA backend. During deployment, the system achieved an average inference latency of 43.7 ms per frame, corresponding to ~22.9 FPS, which is sufficient for real-time classroom analysis. Average CPU utilization remained below 40%, and RAM usage peaked at 6.3 GB during concurrent student evaluations. These results demonstrate the system’s capability for smooth real-time deployment on mid-range hardware without introducing user interface lag or backend bottlenecks.

4.5 Real-time feedback generation and visualization

A core innovation of the system lies in its ability to transform raw behavioral data into meaningful, real-time feedback. For students, this includes the immediate display of scores across multiple behavioral dimensions such as attentiveness, participation, posture correctness, and activity. Feedback is presented through intuitive visualizations like attention span graphs, session heatmaps, and time-series engagement curves. These tools are especially helpful in promoting self-regulated learning and allowing students to reflect on their performance patterns.

For teachers, the system offers real-time analytics dashboards with aggregated engagement data, anomaly detection alerts, and performance distributions. During online assessments, the system can detect and flag suspicious behaviors such as screen avoidance, frequent turning, or multi-device usage, thus supporting real-time intervention and reducing the reliance on manual proctoring. After each session, teachers receive a system-generated report summarizing behavioral patterns, participation trends, and actionable insights tailored to instructional improvement.

This intelligent feedback mechanism is made possible by the lightweight yet expressive modeling capabilities of Vision Mamba architecture. Unlike traditional Transformer-based models, Vision Mamba achieves global contextual understanding through a bidirectional state-space model, making it particularly well-suited for real-time educational scenarios. Its integration with the APN module further enhances the feedback system by enabling finer behavioral categorization under limited data conditions, making it ideal for small classroom settings or targeted assessments.

5 Results and discussion

5.1 Experimental data collection

To support both general video action recognition and classroom-specific behavior analysis, two types of datasets were utilized. First, for benchmarking the VMAPN architecture on standard datasets, we used THUMOS 2014 and ActivityNet v1.3, both of which contain annotated human actions but are not specific to classroom contexts. These were employed to evaluate the classification performance of the Vision Mamba + APN model in complex, unconstrained settings. Second, to illustrate the potential for educational behavior modeling, a custom dataset was constructed consisting of 400 short classroom-style videos featuring 9 students performing 7 scripted actions relevant to learning scenarios (e.g., sitting, writing, raising hands, lying on the table, looking

around, playing with mobile phones). Additionally, we annotated facial expressions (e.g., anger, happiness, neutrality) using a lightweight 3D ConvNet for face detection and facial state classification. These annotations informed the development of the Teaching Evaluation Model (TEM) described in Section 5.3, where behaviors are mapped to engagement labels (positive, neutral, negative). It is important to note that the VMAPN framework itself was used for video-based behavior classification in both the standard action recognition datasets and our classroom video dataset. The 3D ConvNet was employed only as a preprocessing module for extracting facial expression labels, which serve as supplementary indicators in TEM. This separation ensures that facial state detection does not confound the evaluation of the VMAPN architecture on video action classification.

While the custom dataset offers controlled annotation and behavior consistency, it is limited in both size (400 videos) and participant diversity (9 students). This raises concerns about potential overfitting, particularly individual motion styles, clothing, or classroom layout. To mitigate this, we applied data augmentation techniques during training, including random horizontal flipping, spatial cropping, brightness jittering, and frame sequence shuffling. These augmentations improve model robustness to viewpoint and appearance variability. Despite these measures, we acknowledge that generalization remains constrained. In future work, we plan to extend evaluation to larger public datasets such as HACS, EduNet, or HMDB51, and to collect additional data across multiple institutions, age groups, and cultural settings. Such expansion will help ensure the model's scalability and reliability in real-world educational environments.

5.2 Evaluation indicators

This study follows their evaluation criteria to test the mean predictive accuracy (m AP) for different thresholds (IoU). In the THUMOS 2014 dataset, the required IoU thresholds to be tested are {0.1, 0.2, 0.3, 0.4, 0.5}; while in the ActivityNet v1.3 dataset, the required IoU thresholds to be tested are {0.5, 0.75, 0.95}. m AP averages can be used for comparing the performances between the different methods, and the resulting average prediction accuracy is used to compare the experimental results of different methods.

$$\text{mAP} = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \text{AP}_t$$

$Where\ \mathcal{T}$ is the set of tested IoU thresholds, AP_t is the Average Precision computed at IoU threshold t, $|\mathcal{T}|$ is the number of thresholds in set $\mathcal{T}$. The accuracies of VMAPN and other algorithms on THUMOS 2014 dataset and ActivityNet v1.3 dataset are shown in Fig. 5, Fig. 6 respectively.

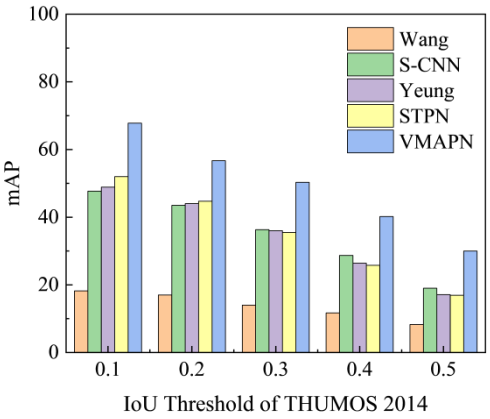


Figure 5: Accuracy of different algorithms on THUMOS 2014 dataset

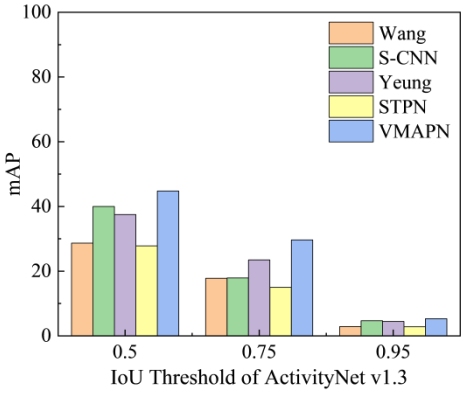


Figure 6: Accuracy of different algorithms on ActivityNet v1.3 dataset

To contextualize the effectiveness of the VMAPN framework, we compared its performance against several state-of-the-art video recognition models, including I3D, SlowFast, and TimeSformer. These models were evaluated under the same training and test conditions using the THUMOS 2014 and ActivityNet v1.3 datasets. As shown in Table 2, VMAPN achieved superior mAP scores in small-sample settings, particularly at lower IoU thresholds, demonstrating its robustness and efficiency in limited-data educational contexts.

Table 2: Performance Comparison with SOTA video recognition models

Model	Architecture Type	THUMOS 2014 (mAP@0.5)	ActivityNet v1.3 (mAP@0.5)
I3D [10]	3D CNN	58.10%	67.50%
SlowFast [12]	Dual-pathway CNN	61.90%	70.10%
TimeSformer [13]	Transformer	63.50%	70.90%
VMAPN (Ours)	ViM + APN	64.70%	72.30%

5.3 Evaluation of teaching effectiveness based on student learning behavior analysis

Traditional classroom teaching evaluations often rely on composite subjective scores from multiple sources—students, instructors, peer observers, and institutional leaders (as detailed in Table 2). While comprehensive in theory, such evaluations are inherently limited by subjectivity, inconsistency, and variability across observers. To address these limitations, this study introduces a Teaching Evaluation Model (TEM) grounded in the automated analysis of student behavioral data captured through VMAPN.

As discussed earlier, the VMAPN system detects and classifies student behaviors during classroom sessions using video-based spatio-temporal recognition. Initially, behaviors were categorized into two broad groups: positive (e.g., sitting upright, writing, raising hands, standing) and negative (e.g., lying on the table, looking away, using mobile phones). However, to improve granularity and pedagogical relevance, the TEM expands this into four behavior categories based on observational research and alignment with engagement theory: Positive behaviors (e.g., writing, raising hands, standing); Relatively positive behaviors (e.g., smiling, facing forward, communicating with the teacher); Neutral behaviors (e.g., lowering or lifting the head, sitting quietly); Negative behaviors (e.g., lying on the table, looking around, using phones). These categories form the basis for calculating an objective, behavior-informed teaching effectiveness score. To bridge the gap between VMAPN's coarse two-category output and TEM's four refined categories, we apply a mapping strategy combining VMAPN predictions with additional cues. Specifically, VMAPN identifies observable physical actions (e.g., writing, standing, lying on table) and classifies them as positive or negative learning behaviors. These predictions are then mapped into TEM's categories using rule-based logic: Actions classified as "positive" by VMAPN (e.g., writing, standing) are mapped to positive or relatively positive categories in TEM depending on facial expression cues (e.g., smiling → relatively positive). Actions such as “sitting” may be mapped to neutral if the student is inactive but not off-task. Behaviors like “looking around” or “using a phone” remain negative across both models. The scoring thresholds (e.g., 1.5–2.0 = "Good") are informed by empirical observation and cross-referenced with expert human ratings during a pilot evaluation. Table 4 summarizes the behavioral indicators and their corresponding weight within the TEM framework. This model aims to supplement traditional evaluations by offering a more objective, scalable, and consistent assessment mechanism.

Table 3: Multi-subjective evaluation indicators

First-level Indicator	Weight	Second-level Indicator	Proportion
Teaching attitude	0.15	Take the course seriously and be familiar with the content to be taught.	0.6
		Be well behaved and generous.	0.4
Teaching content	0.4	The content of the lecture strictly meets the requirements, highlighting the key points and difficulties.	0.4
		Have clear goals and there are no errors in the content.	0.3
		The content of the lecture meets the requirements and is good at combining with reality.	0.2
		Appropriately cite relevant literature.	0.1
Teaching method	0.15	Carry out education according to the textbook so that students can have a clear understanding.	0.15
		Focus on inspiration and lead students to think independently.	0.3
		The content of the class is written neatly and is good at using multimedia to improve the effect of the class.	0.3
		The lesson preparation notes are complete, the content is related to reality, and can reflect the latest scientific and technological research results.	0.25
Interactive method	0.3	Be good at mobilizing students' enthusiasm and livening up the classroom atmosphere during class.	0.6
		There are more than 5 classroom interactions in each class.	0.4
Conclusion of teaching effect evaluation >0.85: Excellent, 0.75~0.85: Good, 0.6~0.75: Qualified, <0.6:			
Unqualified mapping and quantification: 2: Excellent, 1: Good, 0: Qualified, -1: Unqualified			

Table 4: Indicators for analyzing student learning behavior

No.	Learning Behavior Classification	Corresponding Actions
1	Positive behavior	Writing, raising hands, standing up
2	Relatively positive behavior	Smiling, focusing on the front, communicating with teachers
3	Neutral behavior	Lowering head, raising head, sitting upright
4	Negative behavior	Lying on the table, looking around, playing with mobile phones
Teaching effect indicators: Good classroom effect: Good classroom effect: 1.5~2; Neutral classroom: 1~1.5; Poor classroom discipline: <1		Quantitative processing: Good classroom effect: 2; Good classroom effect: 1; Neutral classroom: 0; Poor classroom discipline: -1

$$N_p = \sum_{k=1}^{F} f_p(k),\ p = 1,2,3,4$$

The number of times of all learning behavior indicators N_p, corresponding to positive learning behaviors, more positive behaviors, neutral behaviors and negative learning behaviors, are counted according to Eq. Where $f_p(k)$ denotes the number of people in the Kth image corresponding to the indicator of learning behavior, $\sum_{k=1}^{F} f_p(k) = S$.

To reduce subjectivity in behavior labeling, we developed a standardized annotation guide used by three independent annotators with experience in educational psychology. Disagreements were resolved via majority vote.

Table 5: Per-class precision, recall, and F1-score for learning behavior recognition

Behavior	Precision	Recall	F1-Score
Sitting	0.91	0.93	0.92
Writing	0.88	0.85	0.86
Raising Hands	0.87	0.83	0.85
Standing Up	0.89	0.88	0.88
Lying on Table	0.81	0.78	0.79
Looking Around	0.76	0.72	0.74
Using Mobile Phone	0.82	0.77	0.79

Table 6: Per-class performance on facial expression recognition (LBREM)

Expression	Precision	Recall	F1-Score
Angry	0.75	0.71	0.73
Disgust	0.72	0.69	0.7
Fear	0.7	0.66	0.68
Happy	0.86	0.88	0.87
Sad	0.73	0.7	0.71
Surprise	0.8	0.81	0.8
Neutral	0.89	0.9	0.89

To compute the final \textbf{classroom effectiveness score}, we first normalize the total count of each behavior category across the observation period. Let:
N_{pos}: total count of positive behaviors
N_{rpos}: total count of relatively positive behaviors
N_{neu}: total count of neutral behaviors
N_{neg}: total count of negative behaviors
The total number of observed behaviors is given by:

$$T = N_{\text{pos}} + N_{\text{rpos}} + N_{\text{neu}} + N_{\text{neg}}$$

We then assign weights to each category:
Positive: +2
Relatively Positive: +1
Neutral: 0
Negative: -1
The Composite Behavior Score (CBS) is computed as:

$$\text{CBS} = \frac{1}{T}\left(2 \cdot N_{\text{pos}} + 1 \cdot N_{\text{rpos}} - 1 \cdot N_{\text{neg}}\right)$$

Based on the computed CBS, the classroom teaching effectiveness is categorized as:

$$\{CBS\} \geq 1.5\ \& \rightarrow \{Good\ (score = 2)\}$$
$$1.0 \leq \{CBS\} < 1.5\ \& \rightarrow \{Moderate\ (score = 1)\}$$
$$0 \leq \{CBS\} < 1.0\ \& \rightarrow \{Neutral\ (score = 0)\}$$
$$\{CBS\} < 0\ \& \rightarrow "\{Poor\ (score = -1)\}$$

This rule-based aggregation enables the transformation of frame-level behavioral observations into interpretable, high-level teaching effectiveness scores that are both quantitative and pedagogically grounded.

5.4 Ablation studies

To evaluate the contribution of each component in the VMAPN framework, we conducted a series of ablation experiments involving three modified configurations: (a) Vision Mamba (ViM) without the Augmented Prototypical Network (APN), where only the global category-labeled feature was used for classification; (b) APN applied to a Vision Transformer (ViT) backbone in place of ViM, testing the effectiveness of the prototype-based classifier independently of the ViM architecture; and (c) a baseline ViM model using only the global token for classification, without incorporating local block features or APN-based aggregation. These configurations were evaluated on the THUMOS 2014 and ActivityNet v1.3 datasets under consistent experimental conditions.

Table 7: Ablation study results on behavior classification

Model Variant	Backbone	THUMOS 2014 (mAP@0.5)	ActivityNet v1.3 (mAP@0.5)
(a) ViM only (no APN)	Vision Mamba	60.30%	68.50%
(b) APN + ViT	Vision Transformer	61.80%	69.10%
(c) Vanilla ViM (global token only)	Vision Mamba	59.40%	67.90%
(d) VMAPN (Ours)	Vision Mamba + APN	64.70%	72.30%

As shown in Table 7, the full VMAPN model outperformed all ablated variants, achieving mAP scores of 64.7% (THUMOS) and 72.3% (ActivityNet). Removing APN (variant a) led to a drop in performance to 60.3% and 68.5%, respectively, while omitting local feature integration in ViM (variant c) further reduced accuracy. Although applying APN to a ViT backbone (variant b) yielded modest improvements over vanilla ViM, it still lagged behind the VMAPN configuration. These results highlight the synergistic contribution of both Vision Mamba's spatio-temporal modeling and APN's fine-grained local feature weighting in enhancing small-sample behavior classification.

6 Conclusion

This study presents a real-time automatic scoring and feedback system for intelligent educational platforms, powered by the Vision Mamba architecture and enhanced through an Augmented Prototypical Network (APN). By leveraging spatio-temporal behavior recognition and lightweight deep learning frameworks, the proposed system effectively bridges the gap between traditional expert-based evaluation and modern AI-driven educational assessment.

Through the integration of classroom video analysis, facial expression recognition, and student behavior modeling, the system enables objective, scalable, and personalized evaluation of teaching effectiveness. Experimental validation on THUMOS 2014 and ActivityNet v1.3 demonstrates the system's high accuracy and robustness across varying IoU thresholds. In addition, the application of VMAPN in real classroom scenarios highlights its capability to capture subtle learning behaviors and translate them into meaningful feedback, both for students and instructors. Furthermore, the proposed Teaching Evaluation Model (TEM), based on student behavior classification, offers a novel perspective for educational quality assurance. It transitions from multi-subjective assessments to data-informed, real-time feedback loops, significantly enhancing the fairness and timeliness of instructional evaluation.

However, despite these promising results, the system presents several limitations that must be acknowledged. First, the reliance on facial expressions and upper-body posture may limit robustness in diverse classroom environments, particularly under poor lighting, occlusion, or cultural variation in nonverbal behavior. Second, while Vision Mamba improves efficiency, scalability remains a concern in large classrooms or multi-class deployments, where real-time processing could strain computational resources. Third, the model's performance may degrade over time due to changes in classroom dynamics or behavior patterns, highlighting the need for ongoing retraining to mitigate potential model drift.

In conclusion, this research introduces a practical and scalable AI-based system that enhances real-time educational feedback, supports small-sample learning, and aligns with national education informatization goals of smart education. While some generalizability and deployment challenges remain, this work lays the foundation for future developments in real-time educational analytics, intelligent tutoring, and adaptive teaching strategies. We also recognize that the use of classroom video data introduces important ethical considerations. To address privacy and surveillance concerns, our system follows strict data governance practices, including informed consent, secure storage, anonymization, and opt-in participation. Feedback is delivered privately to users and designed to support learning and teaching—not punitive monitoring. These safeguards aim to ensure responsible and transparent deployment in real-world educational settings. Future work will explore extending this framework to broader educational contexts, incorporating multi-modal data streams, optimizing deployment for large-scale educational platforms, and adapting the system for multi-camera classrooms and edge-device inference to enable scalable, low-latency, and infrastructure-efficient deployment in real-world settings.

Funding

1.Practical Exploration of Infusing Financial Literacy Education into Innovation and Entrepreneurship Education in Local Universities in the New Era, Higher Education Talent Development Quality and Teaching Reform Project of Sichuan Provincial Department of Education, Grant Number: JG2021-1276
2.AI-Enabled Reform and Practice of Classroom Teaching Models in Vocational Education, Chongqing Municipal Vocational Education and Teaching Reform Research Project,Grant Number: Z2252062.

References

[1] Sajja, Ramteja, et al. "Integrating AI and learning analytics for data-driven pedagogical decisions and personalized interventions in education." Technology, knowledge and learning (2025): 1-31, doi: 10.1007/s10758-025-09897-9.

[2] Sharma, Vijeta, et al. "Video processing using deep learning techniques: A systematic literature review." IEEE Access 9 (2021): 139489-139507 , doi: 10.1109/ACCESS.2021.3118541.

[3] Dai, Zhicheng, et al. "The effect of smart classrooms on project-based learning: A study based on video interaction analysis." Journal of Science Education and Technology 32.6 (2023): 858-871, doi: 10.1007/s10956-023-10056-x.

[4] Yang, Junfeng, et al. "Evaluation of smart classroom from the perspective of infusing technology into pedagogy." Smart Learning Environments 5 (2018): 1-11, doi: 10.1186/s40561-018-0070-1.

[5] Feichtenhofer, Christoph, Axel Pinz, and Andrew Zisserman. "Convolutional two-stream network fusion for video action recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, doi: 10.1109/CVPR.2016.213.

[6] [6] Yang, Hao, et al. "Asymmetric 3d convolutional neural networks for action recognition." Pattern recognition 85 (2019): 1-12. doi:10.1016/j.patcog.2018.07.026.

[7] Bandi, Chaitanya, and Ulrike Thomas. "Skeleton-based action recognition for human-robot interaction using self-attention mechanism." 2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021). IEEE, 2021. doi:10.1109/FG52635.2021.9667005.

[8] Feichtenhofer, Christoph, Axel Pinz, and Andrew Zisserman. "Convolutional two-stream network fusion for video action recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. doi:10.1109/CVPR.2016.213.

[9] Ji, Shuiwang, et al. "3D convolutional neural networks for human action recognition." IEEE transactions on pattern analysis and machine intelligence 35.1 (2012): 221-231. doi:10.1109/TPAMI.2012.59.

[10] Wang, Yunfei, et al. "E3D: An efficient 3D CNN for the recognition of dairy cow's basic motion behavior." Computers and Electronics in Agriculture 205 (2023): 107607. doi:10.1016/j.compag.2023.107607.

[11] Kondratyuk, Dan, et al. "Movinets: Mobile video networks for efficient video recognition." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021. doi:10.1109/CVPR46437.2021.01576.

[12] Wang, Limin, et al. "Tdn: Temporal difference networks for efficient action recognition." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021. doi:10.1109/CVPR46437.2021.00190.

[13] Ulhaq, Anwaar, et al. "Vision transformers for action recognition: A survey." arXiv preprint arXiv:2209.05700 (2022). doi:10.48550/arXiv.2209.05700.

[14] Li, Kunchang, et al. "Uniformer: Unifying convolution and self-attention for visual recognition." IEEE Transactions on Pattern Analysis and Machine Intelligence 45.10 (2023): 12581-12600. doi:10.1109/TPAMI.2023.3274944.

[15] Kline, Michelle Ann. "TEACH: An ethogram-based method to observe and record teaching behavior." Field Methods 29.3 (2017): 205-220. doi:10.1177/1525822X16677334.

[16] Alexiadis, Dimitrios S., et al. "Evaluating a dancer's performance using kinect-based skeleton tracking." Proceedings of the 19th ACM international conference on Multimedia. 2011. doi:10.1145/2072298.2072403.

[17] Moyo, Reuben, et al. "A Video-based Detector for Suspicious Activity in Examination with OpenPose." arXiv preprint arXiv:2307.11413 (2023). doi:10.48550/arXiv.2307.11413.

[18] Zhou, Jie, et al. "Classroom learning status assessment based on deep learning." Mathematical Problems in Engineering 2022.1 (2022): 7049458. doi:10.1155/2022/7049458.

[19] Kim, Woojoo, et al. "Ergonomic postural assessment using a new open-source human pose estimation technology (OpenPose)." International Journal of Industrial Ergonomics 84 (2021): 103164. doi:10.1016/j.ergon.2021.103164.

[20] Zheng, Ce, et al. "Deep learning-based human pose estimation: A survey." ACM Computing Surveys 56.1 (2023): 1-37. doi:10.1145/3603624.

[21] Cheng, Yu, et al. "3d human pose estimation using spatio-temporal networks with explicit occlusion training." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 34. No. 07. 2020. doi:10.1609/aaai.v34i07.666.

[22] Xu, Rui, et al. "Visual mamba: A survey and new outlooks." arXiv preprint arXiv:2404.18861 (2024). doi:10.48550/arXiv.2404.18861.

[23] Xu, Yufei, et al. "Vitpose: Simple vision transformer baselines for human pose estimation." Advances in neural information processing systems 35 (2022): 38571-38584. doi:10.48550/arXiv.2204.12484.

[24] Idrees, Haroon, et al. "The thumos challenge on action recognition for videos “in the wild”." Computer Vision and Image Understanding 155 (2017): 1-23. doi:10.1016/j.cviu.2016.10.018.

[25] Xiong, Yuanjun, et al. "Cuhk & ethz & siat submission to activitynet challenge 2016." arXiv preprint arXiv:1608.00797 (2016). doi:10.48550/arXiv.1608.00797.

[26] Navarrete, Evelyn, et al. "A closer look into recent video-based learning research: A comprehensive review of video characteristics, tools, technologies, and learning effectiveness." International Journal of Artificial Intelligence in Education (2025): 1-64. doi:10.1007/s40593-024-00435-z.

[27] Cosentino, Giulia, et al. "Generative AI and multimodal data for educational feedback: Insights from embodied math learning." British Journal of Educational Technology (2025). doi:10.1111/bjet.13567.

ESBO-FDDPG: A Federated Reinforcement Learning-Based Continuous Resource Scheduler for Cross-Domain Multi-Cloud Systems

Wenwei Su[1*], Zhengxiong Mao[1], Fengbo Kong[1], Yan Shi[2], Pan Xiao[3]
[1]Yunnan Power Grid Co., Ltd. Information Center, Yunnan, 650228, China
[2]Qujing Power Supply Bureau of Yunnan Power Grid Co., Ltd., Yunnan, 650100, China
[3]Southern Power Grid Digital Platform Technology (Guangdong) Co., LTD, Guangdong, 518101, China
E-mail: WenweiSu@outlook.com

Keywords: cloud computing, resource scheduling, multi-data center management, energy efficiency, cross-domain efficient satin bowerbird optimizer-driven federated deep deterministic policy gradient (CG-ESB-FDDPG)

Received: June 6, 2025

With the rapid development of cloud computing, the demand for efficient resource management across multiple data centers has become increasingly critical. Cloud platforms typically span geographically distributed data centers, offering scalability, flexibility, and on-demand resource provisioning. This research proposes a unified cross-domain continuous resource scheduling method for various data centers within the cloud platforms, aiming to tackle challenges such as uneven workload distribution, dynamic service demands, and energy inefficiency across multiple data centers. This research introduces a unified cross-domain continuous resources scheduling framework based on an Efficient Stain Bowerbird Optimizer-driven Federated Deep Deterministic Policy Gradient (ESBO-FDDPG) model. Workload traces from the Cross-Domain Cloud Scheduling Dataset were preprocessed using Min-Max normalization, and Statistical Pattern Recognition (SPR) with Fast Fourier Transform (FFT) and Radial Inverse Distance (RID) kernels was applied for feature extraction. The proposed method enables privacy-preserving, real-time, and adaptive scheduling by integrating federated learning with deep reinforcement learning. The proposed method supports continuous monitoring and coordination of workloads across multiple geographically distributed data centers, dynamically adjusting resource allocation based on service request patterns and power usage profiles. The system embeds vertical federated learning for cross-domain model updates without data sharing and uses reinforcement learning to continuously adapt to fluctuating workloads and infrastructure constraints. Simulation experiments were conducted using Python. Simulations conducted with different load and network conditions show that the suggested ESBO-FDDPG approach lowers average latency (63ms) and enhances energy efficiency (4.6 × 〖10〗^5bits/joules), and task completion rate (98.8%). This research offers important insights for sustainable cloud computing practices and a scalable, energy-conscious solution to unified resource scheduling in contemporary cloud systems.

Povzetek: Predlagana je metoda za sprotno, zasebnostno varno usklajevanje virov med več podatkovnimi centri v oblaku, ki z združevanjem federativnega učenja in globokega okrepitevnega učenja izboljša porazdelitev bremen ter zmanjša zakasnitev in porabo energije.

1 Introduction

Cloud platforms are massive collections of computers in different locations. It enables people and organizations to upload files, run applications, and access services from computers that do not manage hardware [1]. Each computer host can provide processing power, memory, and storage space. Moreover, the data can be allocated to different users anywhere in the world. The popularity of cloud platforms continues to improve with all types of tasks, from video-streaming to data analytics to file storage [2]. The systems can efficiently manage these resources, especially during peak times when many users can be active.During periods of predictable demand, the cloud platform dynamically allocates computing and storage resources while continuously monitoring system utilization to ensure optimal performance[3].

These cloud platforms run across numerous data centers, which are huge buildings packed with computers. A single center can have varying workloads based on time of day, regional demand, or the kinds of tasks being executed [4]. Intelligent systems are needed to determine when and how data should be shared between centers to maintain

efficiency and reduce energy consumption [5]. Cloud systems use advanced learning techniques to prevent data redundancy, maintain privacy, and make better decisions, conserving time and energy, making them more environmentally friendly by reducing costs and pollution [6].

Overloaded cloud systems can cause delays and critical tasks, requiring platforms to balance workloads on all computers and centers, allowing others to assist if necessary [7]. A smart resource scheduling system efficiently manages workload, adjusts task assignments, and reacts to changes, ensuring smoother operations and anticipating needs to prevent issues [8]. Enhancements in cloud services lead to faster user services, better resource utilization, and reduced energy consumption, necessitating intelligent workload management for reliable, quick, and environmentally friendly platforms [9].

Research questions

- ***RQ1:*** How can federated reinforcement learning improve energy efficiency in multi-data center cloud environments?
- ***RQ2:*** Can ESBO-FDDPG maintain or reduce task latency while dynamically allocating resources?
- ***RQ3:*** How effectively does the proposed method preserve privacy while aggregating insights across distributed data centers?

Research objective and system overview

The research introduced new unified cross-domain continuous resource scheduling methods.The ESB-FDDPG framework enables multiple data centers on cloud platforms to satisfy a required set of conditions over time. The method employs federated learning that combines with deep reinforcement learning to optimally allocate resources while managing and minimizing energy usage and enhancing overall performance within a service-level agreement and privacy constraints.

The organization of the research includes five sections. Phase 1: discuss the introduction of the research. Phase 2 includes related works based on resource scheduling over cloud environments. Phase 3 contains the methodological framework that includes the overall view of the multi-data centers on cloud platforms for resource scheduling. Phase 4 includes the results and discussion. Phase 5 involves the conclusion of the research.

2 Relevant works

Mao et al. proposed a novel strategy for managing Task Management and Virtual Machine Control (TM-VMC), which improved total energy usage while preventing hot spots on a global scale. The experimental findings suggested that the strategy could prevent hot spots and minimize energy usage while maintaining minimal service level agreement breaches [10]. Lin et al. distributed a high-definition deep learning or hybrid deep deterministic learning (HDDL) algorithm for job planning and a deep Q-network (DQN) framework used for resource allocation. The suggested framework improved the benchmark method in terms of energy consumption and task delay [11].

Fan et al. used cloud computing, which was quickly evolving, and managing resources in data centers anddistributed computing networks (DCNs)was critical to its performance. This discussion covers advancements in resource management and optimization, including intelligent resource search algorithms and scheduling methods, and suggests integrating AI within (DCNs) [12].

A framework for providing medical services is provided by cloud computing. Subsequently, exchanging medical information with unidentified cloud neighbours could jeopardize private medical service data. A personalized cloud offers a secure method of safeguarding personal healthcare data. Gong et al. suggested a secure cloud-based effective resource planning solution for healthcare facilities in healthcare information systems [13]. Using testing using Google clusters trace information, Malik et al. suggested a multiple-resource forecasting strategy that combines Functional Link Neural Network (FLNN), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO). It shows improved accuracy compared to traditional approaches [14].

Dey et al. presented SmartNoshWaste, a multifaceted blockchain-powered architecture that uses cloud technology, QR codes, and reinforced learning to reduce wasted food across the supply chain. The effectiveness of the structure was assessed using actual data from the nosh app, showing a 9.46% decrease in food wasted over the initial data [15].Cloud computing constitutes an internet-based innovation where optimizing resources is essential. A resource administration strategy for the Bin-Packing challenge is presented by Nehraand Kesswani, to enable the allocation of VMs to Physical Machines (PMs) in a cost-effective way. The CloudSim Plus Simulator experiment findings show better resource utilization, less energy usage, and fewer engaged PMs [16].

The Bin-Packing challenge is addressed in Vijayaand Srinivasan, a resource administration approach for the effective distribution of VMs among Physical Machines (PMs). The results of simulations conducted with CloudSim show enhanced usage of resources, decreased energy usage, and a decrease in outstanding PMs [17].Cloud computing needs to consider the environment and energy consumption without sacrificing service excellence. As a means to reduce energy consumption and carbon emissions, Buyya et al. suggested a learning-focused management strategy for workloads and combining resources. It also presents a conceptual design that increases ecological and energy conservation [18].

Table 1 represents a summary of existing resource scheduling methods.

Table 1: Summary of existing resource scheduling methods

Reference	Methodology	Metrics	Dataset	Reported results
Mao et al. (2023) [10]	Thermal-aware scheduling using Task Management and Virtual Machine Control (TM-VMC)	Energy consumption, SLA breaches	Synthetic cloud data	Energy reduced by ~7.9%, SLA violations <0.005%
Lin et al. (2020) [11]	High-definition Deep Learning (HDDL) + Deep Q-Network (DQN)	Energy consumption, task delay	Google cluster traces	Energy reduced by 5.7–9.7%, delay improved
Fan et al. (2020) [12]	Locality-based scheduling in Data Center Networks (DCNs)	Latency, throughput	DCN simulations	Improved latency and throughput
Gong et al. (2022) [13]	Secure cloud-based resource management for hospital systems	Resource allocation efficiency	Healthcare data	Operational cost reduced by 23%
Malik et al. (2022) [14]	Functional Link Neural Network (FLNN) + GA + PSO for resource prediction	Prediction accuracy, resource utilization	Google cluster traces	Higher prediction accuracy vs benchmarks
Nehra & Kesswani, (2023) [16]	Load-balanced multi-dimensional Bin-Packing heuristic	Resource utilization, energy consumption	CloudSim Plus simulations	Better resource use and fewer active servers
Vijaya & Srinivasan, (2024) [17]	Multi-objective meta-heuristic for VM placement (ACO + SCA)	Energy consumption, CPU utilization	Cloud simulations	Improved energy efficiency
Buyya et al. (2024) [18]	Integrated energy management strategy	Energy use, carbon emissions	Data center workload simulations	Conceptual vision, no numerical results

2.1 Research gap

Despite several advancements in cloud resource scheduling, existing methods exhibit notable drawbacks. Mao et al.improved energy efficiency but rely on static scheduling that does not adapt well to dynamic workloads [10]. Lin et al.enabled task delay and energy consumption using DL but lack continuous scheduling and privacy-preserving mechanisms for multi-data center environments [11]. Fan et al.concentrated on locality-based scheduling; nonetheless, the method has issues with large-scale implementation and energy efficiency [12]. Although they don't directly optimize real-time scheduling, Malik et al.increased prediction accuracy for resource consumption [14]. Although Vijaya and Srinivasan, used meta-heuristic virtual machine placement to increase energy efficiency, they are unable to expand efficiently or dynamically manage workload fluctuations [17].

The ESBO-FDDPG architecture, which offers continuous, privacy-preserving, and scalable resource scheduling across many cloud data centers while enhancing latency, energy efficiency, and task completion rates, is motivated by these gaps.

3 Materials and methods

The methodology section clearly explains the process of data transmission between the vehicle users and the multi-edge computing server (MEC) server (roadside unit (RSU) with the help of resource scheduling using ESBO-FDDPG, also describes the data collection, preprocessing using Min-Max scaling, and the feature extraction approach, which includes statistical pattern recognition (SPR) to identify workload data patterns. The step-by-step process is shown in Figure 1.

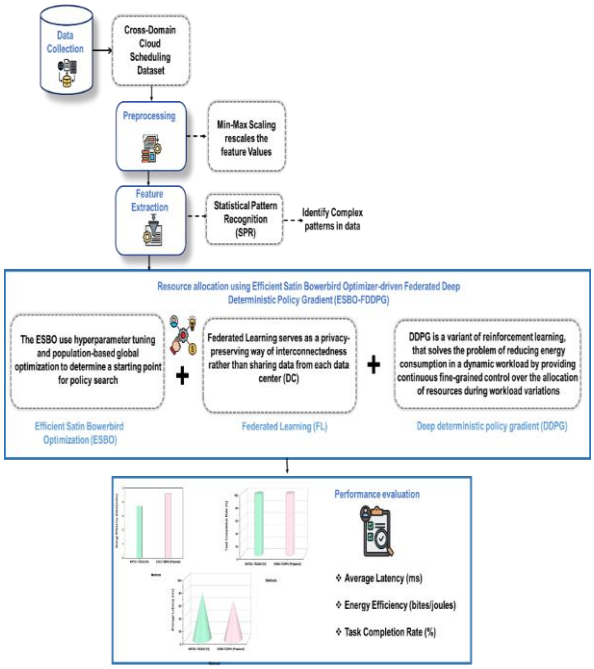


Figure 1: Methodological framework

3.1 Data collection

The experimental evaluation in this research is conducted using the Cross-Domain Cloud Scheduling Dataset (https://www.kaggle.com/datasets/programmer3/cross-domain-cloud-scheduling-dataset), which provides workload and energy-related metrics for simulating multi-data center scheduling scenarios. Each record corresponds to a cloudlet (task) scheduled on a virtual machine (VM) and contains both input parameters and execution outcomes, making it suitable for analyzing cross-domain scheduling policies. The dataset includes key attributes such as CloudletLength (task length in million instructions), CloudletFileSize and OutputSize (I/O data size), NumOfPes (number of processing units allocated), and utilization models for CPU, RAM, and bandwidth. It also tracks Vmld (assigned VM), ActualCPUTime (execution time), and Status (Success/Failure).

This dataset was selected because it provides diverse workload traces across computational, storage, and bandwidth dimensions, reflecting the heterogeneous demands in real cloud environments. Its inclusion of both utilization behaviours and outcome measures enables performance evaluation in terms of latency, task success rate, and energy efficiency, which align directly with the goals of the proposed ESBO-FDDPG model. To ensure replicability, preprocessing involved Min-Max normalization of continuous attributes, SPR-based feature extraction (FFT-RID kernel), and a 70:15:15 split into training, validation, and testing sets. This choice allows consistent benchmarking while preserving generalization across multiple data centers.

3.1.1 Preprocessing using Min-Max scaling

The preprocessing technique used to normalize data while rescaling feature values to a certain range is preferable within$[0,1]$. The scaling is performed using Eq. (1):

$$W_{scaled} = \frac{W - W_{min}}{W_{max} - W_{min}} \quad (1)$$

Where$W$ stands for the cross-domaindata point when the characteristic. $W_{min}$is the number that determines the minimum value of the feature. W_{max}represents the value of the maximal feature. The result, W_{scaled}, is the transformed value for the given data that is in the range of$W$. The variables in a dataset are measured on different scales or units to prevent them from being submerged because of their higher value. It has large or small values in the data that affect the scaling of other values.

`SPR plays a crucial role in handling heterogeneous and noisy workload traces that arise in cloud data centers. By analyzing global frequency trends (via FFT) and localized temporal variations (via RID-based TDA), the method captures sudden surges or drops in demand more accurately than conventional time-domain analysis. This dual-domain approach ensures that the scheduler can detect recurring periodic workloads (e.g., daily or weekly peaks) as well as unexpected spikes, leading to more robust resource allocation. Furthermore, SPR reduces dimensionality while retaining the most relevant workload features, which directly benefits the ESBO-FDDPG model's learning efficiency. SPR is a feature extraction technique used in fields like diagnostics, speech recognition, and data analysis to identify complex patterns in sensor data. It helps in cross-domain continuous resource scheduling problems in cloud platforms by identifying temporal patterns in physiological and behavioral measures. SPR is particularly useful in investigating sensor data captured under various academic and environmental conditions. SPR used two methods, Fast Fourier Transform (FFT) and Time-Frequency Analysis (TFA), for better feature extraction from time-series data, enabling efficient management with noisy inputs and adaptation to sudden changes in monitored electronics assembly.

FFT for Feature Extraction in the Frequency Domain: FFT is a mathematical technique that converts a time-domain signal into its frequency components. It decomposes complex workload signals into sinusoidal components, revealing periodic demand cycles that help identify recurring task patterns in cloud environments. By mapping the signal into the frequency spectrum, FFT enables the system to anticipate regular fluctuations in resource demand.FFT is used in the field to extract frequency-domain features from time-series data to capture any periodic characteristics that can be related to fluctuating resource demand. It was used in Eq. (2).

$$X(f) = \sum_{n=0}^{N=1} x(n) e^{j2\pi fn/N} \quad (2)$$

The RID kernel is a specialized windowing function used in TFA to analyse workload signals simultaneously in the time and frequency domains. Unlike FFT, which loses time resolution, the RID kernel captures localized variations, making it effective for identifying transient spikes in demand. This ensures the scheduling model remains responsive to short-term irregular workloads as well as long-term cycles.

Time-Frequency Analysis using resource identifier (RID)kernel:Using an RID kernel for TFA enables the extraction of features that provide information about the localized responses from the data over both time and frequency that could help detect low-frequency variations that signify the resource demand by Eq. (3).

$$S(t,f) = \left| \int_{-\infty}^{\infty} x(\tau) e^{j2\pi f\tau}\, w(t-\tau) d\tau \right|^2 \quad (3)$$

In Eq. (2), $x(n)$ represents the discrete-time signal at index n, f denotes the frequency, and N is the total number of samples in the signal. In Eq. (3), $S(t,f)$ is the time-frequency representation of the signal, $w(t-\tau)$ is the window function for time localization, and τ is the integration variable in the continuous-time analysis.Using these SPR methods, the system monitors and predicts low resource demand and thus provides a real-time alert for resource scheduling.

3.1.2 Power Spectral Density (PSD) using FFT: PSD is used to quantify how the power of a signal is distributed across different frequencies. In cloud workload analysis, PSD highlights dominant periodic patterns in task arrivals or resource demand.

$$O(e) = \frac{1}{M} |W(e)|^2 \quad (4)$$

In Eq. (4), where $O(e)$is the power spectral density at frequency $W(e)$ is the FFT of the signal, and M is the total number of samples. This quantifies the strength of workload demand variations at different frequencies.

3.1.3 Short-Time Fourier Transform (STFT) for Localized Analysis: STFT provides a time-frequency representation of a signal. It is particularly useful in cloud scheduling to capture transient bursts in demand while preserving local time information.

$$STFT(s,e) = \sum_{m=-\infty}^{\infty} w(m)\omega(m-s) f^{-i2\pi em} \quad (5)$$

In Eq. (5), $STFT(s,e)$ represents the time-frequency representation of the signal at time s and frequency f, $\omega(m-s)$is the sliding window function for localizing the signal in time, i is the imaginary unit, πem represents the phase rotation corresponding to the frequency component e at time m., and $w(m)$ is the discrete-time workload signal. This helps detect temporary demand bursts in specific time intervals.

3.2 Resource allocation using efficient satin bowerbird optimizer-driven federated deep deterministic policy gradient (ESBO-FDDPG)

The ESB-FDDPG hybrid system uses three methodologies tobetter manage cloud resources. The efficient satin bowerbird optimizer (ESBO) uses hyperparameter tuning and population-based global optimization to determine a starting point for policy search. Federated Learning (FL) serves as a privacy-preserving way of interconnectedness rather than sharing data from each data center (DC). Deep deterministic policy gradient (DDPG) is a variant of reinforcement learningthat solves the problem of reducing energy consumption in a dynamic workload by providing continuous, fine-grained control over the allocation of resources during workload variations. This hybrid system combines ESB's global optimization methodology, federated learning's data privacy, and DDPG's adaptability to provide a more energy-efficient scheduling system while managing resources across multiple heterogeneous distributed cloud systems.

Efficient Satin Bowerbird Optimization (ESBO):The ESBO algorithm is a metaheuristic that finds a globally optimal solution to multi-datacenter resource scheduling problems. It is a simple, robust, and efficient population-based method that generates candidate solutions within defined limits, guiding the best chances of optimal resource allocation across data centers. Eqs. (6) and (7) define how candidate solutions are evaluated and probabilistically selected, forming the foundation for the iterative optimization in ESBO.

$$Prob_j = \frac{fit_j}{\sum_{m=1}^{NB} fit_m} \quad (6)$$

$$fit_j = \begin{cases} \frac{1}{1+e(w_j)}, & e(w_j) \geq 0 \\ 1 + |e(w_j)|, & e(w_j) < 0 \end{cases} \quad (7)$$

$Prob_j$is the probability of selecting the j^{th}solution based on fitness, fit_j is the fitness of the j^{th}solution,m is the index of each candidate solution in the population, and NB is the total number of solutions. NB is the error value for the j^{th}solution.$w_j$represents the decision variable of the j^{th}candidate solution. $e(w_j)$ is the error or deviation from the optimal allocation for that candidate.

The ESBO approach uses exclusivity to find optimal solutions, based on satin bowerbirds' behavior. The attractiveness of the bower is determined by decorations, allowing for optimization and effective resource scheduling across cloud systems as follows in Eq. (8).

$$w_{jl}^{new} = w_{jl}^{old} + \lambda_l \left(\left(\frac{w_{jl} + w_{elite,l}}{2} \right) - w_{jl}^{old} \right) \quad (8)$$

In the given Eq. (15), w_{jl}^{new} represents the updated weight of the jth solution in the lth dimension, w_{jl}^{old} is the old weight of the jth solution in the lth dimension, $w_{jl}$is the positive weight component, and $w_{elite,l}$ is the elite solution weight in the lth dimension. λ is the step size factor in the lth dimension.

In the mutation process after each round of the ESBO method, the solution w_{jl}^{old} is randomly adjusted according to defined probabilities. This mutation process uses a standard normal M distribution, and the new solution $w_{jl}^{new}$is is created based on the mean of the previous solution w_{jl}^{old} and a defined variance σ^2. The mutation is defined mathematically as in Eqs. (9-11).

$$\lambda_l = \frac{\alpha}{1+o_i} \tag{9}$$

$$w_{jl}^{new} \sim M\left(w_{jl}^{old}, \sigma^2\right) \tag{10}$$

$$M\left(w_{jl}^{old}, \sigma^2\right) = w_{jl}^{old} + \left(\sigma^* M(0,1)\right) \tag{11}$$

Where $M\left(w_{jl}^{old}, \sigma^2\right)$ is the normal distribution, in which the values resemble some random variation drawn from a mean of the normal distribution and the variance σ of the normal distribution. It adds diversity to the population and ultimately helps the algorithm effectively explore the solution space so it can schedule resources across the data centers optimally.

Federated Learning (FL): In the proposed framework, each data centre independently trains a local DDPG model using its own workload data, ensuring privacy is preserved. During training, the adaptive loss function evaluates resource scheduling errors, such as underutilization or overutilization, guiding the optimization process locally. Once local updates are completed, the gradients are aggregated using a federated averaging strategy to form a unified global model. This global model is then redistributed back to each data center, allowing consistent scheduling decisions across all domains. Both the actor and critic networks in DDPG are updated through this process, ensuring that the policy and value estimation remain aligned for efficient cross-domain resource allocation.This sub-section discusses resource scheduling using FL in a multi-data center, focusing on local FL models trained independently and updated globally. The adaptive loss function measures scheduling penalties of utilization errors, ensuring proper scheduling over the distributed functional block. The FL system consists of 5 geographically distributed data centers, each acting as a client. Each client trains its local model for 10 epochs before sending updates. Global model aggregation is performed every 5 local training rounds. In total, 50 communication rounds are executed to update the global model across all clients.FL optimizes resources for cloud consumption, providing energy efficiency, response times, and performance in Eqs. (12 and 14).

$$E_l(x_m) = \frac{1}{|\mathcal{D}_l|}\sum_{j\in\mathcal{D}_l} \mathrm{K}\left(\mathrm{f} \ominus (\mathrm{x}_l), \mathrm{y}_l\right), \forall l \in \mathcal{K} \tag{12}$$

$$E(x_m) \triangleq \sum_{l\in\mathcal{K}_m} \frac{|\mathcal{D}_l| E_l(x_m)}{|\mathcal{D}_m|} = \frac{1}{|\mathcal{D}_m|}\sum_{l\in\mathcal{K}_m}\sum_{j\in\mathcal{D}_l} e\left(x_m; t_{l,j}, y_{l,j}\right) \tag{13}$$

$$x_m^* = \arg\min E(x_m), \quad \forall m \in \mathcal{N} \tag{14}$$

$\mathcal{D}_l$represents *the* total number of data points in the local dataset $\mathcal{D}_l$of the lth data center. $j \in \mathcal{D}_l$ iterates over each data point j in the local dataset. $\mathrm{K}\left(\mathrm{f} \ominus (\mathrm{x}_l), \mathrm{y}_l\right)$ is the loss function comparing model prediction $f(\mathrm{x}_l)$ and the true label y_l. $l \in \mathcal{K}$ iterates over all local model centers in the set K. $\mathcal{K}_m$is the set of data centers contributing to the global modal x_m. $\mathcal{D}_m$is the total number of samples across all contributing data centers. $t_{l,j}$represent input features of the jth data point in the lth data center. $x_m^*$is the optimal global model parameters after federated aggregation. $\mathcal{N}$ is the total number of candidate models. $\arg\min E(x_m)$selects the model x_m that minimizes the global loss $E(x_m)$.

In scheduling resources across geographically distributed data centers, it is not possible to compute the global model parameters $E(x_m)$ using the raw datasets corresponding to the local resource units for privacy and security, by Eq. (12).The solution here proposes the concept of model averaging in the resource scheduling process by utilizing FL to maintain the data privacy of each unit. The distributed approach can be applied using a gradient-averaging approach, similarly to optimize resource allocation across geographically distributed data centers securely and cost-effectively, utilizing the sensitivity of their local data while also maintaining scalability and performance.

Deep deterministic policy gradient (DDPG): The method can solve distributed resource scheduling problems in multiple data centers. Using the Q-learning method, it generates a value function right arrow$q$by stating the value that can be obtained by policy πat states t^l in. The value function provides a measurement of the effectiveness of scheduling resources to multiple data centers, taking into consideration infinite horizons and discounting in the DDPG in Eq. (15).

$$U^\pi(t) = \mathbb{E}_\pi[\sum_{l=0}^{\infty} \gamma . q^l\left(t^l, b^l\right) | t^0 = t] = \mathbb{E}_\pi[q^l(t^l, b^l) + \gamma . U^\pi(t^{l+1}) | t^0 = t] \tag{15}$$

To find the optimal policyπ, the optimal action in each state can be obtained from the optimal value function. This can be depicted as Eq. (16).

$$U^*(t) = \max_{b^l}\{\mathbb{E}_\pi[q^l(t^l, b^l) + \gamma . U^\pi(t^{l+1})]\} \tag{16}$$

Where$U^*(t)$is the optimal action-value function for l^{th}state t^l and l^{th} action b^l, to provide insight into which action is the best choice for an optimal allocation of resources in each data center.

However, the classical Q-learning algorithm could not effectively find the optimal policy when the state-action space is large or continuous, and it computes Q-values on the Q-table. Long computational time with multi-data center resource scheduling, high-dimensional data was used to optimize the representation of DDPGand implement a neural network model with Q-value $R^*(t^l, b^{l+1}, \theta)$, with θ being the weights of the model DDPG. After comparing the Q-value for each possible action, the optimal scheduling policy $\pi$for effective resource scheduling can now be represented in Eqs.(17-20).

$$R^{l+1}(t^l, b^l) = R^l(t^l, b^l) + \alpha[q^l(t^l, b^l) + \gamma max R^l(t^l, b^{l+1}) - R^l(t^l, b^l)] \quad (17)$$

$$\pi^*(t) = \arg\max_{b^l} R^*(t^l, b^{l+1}, \theta) \quad (18)$$

$$\tilde{R}(t^l, b^l, \omega) = q(t^l, b^l, \omega) + \gamma max_{b^{l+1}}[R(t^{l+1}, b^{l+1}, \theta)] \quad (19)$$

$$K = F\left[\left(\tilde{R}(t^l, b^l, \omega) - R(t^{l+1}, b^{l+1}, \omega)\right)^2\right] \quad (20)$$

The ESB-FDDPG hybrid system employs three approaches: ESBO, FL, and DDPG. It optimizes cloud computing resources by utilizing hyperparameter tuning, population-based optimization globally, and data security. The ESBO-FDDPG method pseudocode is provided in Algorithm 1.

Algorithm 1: ESBO-FDDPG

Input:
- $N = 5$ *agents/nodes*
- *Local environments* $\{Env_1, \ldots, Env_5\}$
- *Actor network* π_θ *(2 hidden layers: 64 neurons each, ReLU)*
- *Critic network* Q_φ *(2 hidden layers: 64 neurons each, ReLU)*
- *Replay buffer size* $B_i = 10000$
- *Satin Bowerbird Optimizer (SBO):*
  - *Population = 10 candidates*
  - *Iterations = 5 per federated round*
- *Federated aggregation rounds R = 50*
- *Hyperparameters:*

$\gamma = 0.99$ *# Discount factor*
$\tau = 0.005$ *# Soft target update*
$lr_actor = 0.001$
$lr_critic = 0.002$
Mini-batch size = 64
Exploration noise σ = 0.1

Output:
- *Federated global actor network* π_θ_global
- *Federated global critic network* Q_φ_global

1. Initialize actor π_θ_i *and critic* Q_φ_i *for each agent i*
2. Initialize target networks: $\pi_\theta_target_i \leftarrow \pi_\theta_i, Q_\varphi_target_i \leftarrow Q_\varphi_i$
3. Initialize local replay buffers $Bi \leftarrow \emptyset$
4. For each Federated Round $r = 1$ *to 50 do:*
4.1 For each Agent i in parallel:
a) Interact with local environment Env_i *for* $T = 1000$ *steps:*
- Select action: $a_t = \pi_\theta_i(s_t) + N(0, 0.1)$ *# Gaussian noise*
- Observe reward r_t *and next state* $s_\{t+1\}$
- Store $(s_t, a_t, r_t, s_\{t+1\})$ *in Bi*
b) Sample mini-batch of 64 transitions from Bi
c) Update Critic Q_φ_i *using:*

$$yt = rt + 0.99 * Q_\varphi_target_i(s_\{t+1\}, \pi_\theta_target_i(s_\{t+1\}))$$

Loss = $MSE(Q_\varphi_i(s_t, a_t), y_t)$
d) Update Actor π_θ_i *using policy gradient:*

$$\nabla\theta_i J \approx 1/64 * \Sigma \nabla_a Q_\varphi_i(s, a)|_\{a = \pi_\theta_i(s)\} \nabla\theta_i \pi_\theta_i(s)$$

e) Soft update target networks:

$$\theta_target_i \leftarrow 0.005 * \theta_i + 0.995 * \theta_target_i$$
$$\varphi_target_i \leftarrow 0.005 * \varphi_i + 0.995 * \varphi_target_i$$

4.2 Federated Aggregation:
- Collect π_θ_i *and* Q_φ_i *from all agents*
- Aggregate globally:

$$\theta_global = (\Sigma\ \theta_i) / 5$$
$$\varphi_global = (\Sigma\ \varphi_i) / 5$$

- Broadcast θ_global *and* φ_global *to all agents*
4.3 ESBO Hyperparameter Optimization (Optional):
- Initialize SBO population with 10 hyperparameter sets $(lr_actor, lr_critic, \tau)$
- For $t = 1$ *to 5 iterations:*
a) Evaluate candidate hyperparameters on local agent performance
b) Update candidates using SBO mating/selection rules
- Update best hyperparameters for next federated round
5. Return θ_global, φ_global

The ESBO-FDDPG framework described in Algorithm 1 combines an FDDPG approach to cross-domain resource scheduling with the SBO. Each of the five agents uses environment-related data to autonomously train local actor-critic networks while maintaining anonymity. Using federated averaging, the local upgrades are combined to create a global framework that improves scheduling effectiveness. Every round, SBO dynamically adjusts hyperparameters like developing and updating rates to guarantee optimal performance and stable convergence. The hyperparameter tuning (Table 2) strategy follows the adaptive optimization mechanism of the SBO integrated within the federated reinforcement learning framework.

Table 2: Hyperparameter configuration and SBO search ranges used in ESBO-FDDPG training

Hyperparameter	Symbol	Default Value	SBO Search Range	Description
Discount Factor	γ	0.99	[0.90, 0.999]	Weight of future rewards in the critic update.
Soft Target Update Rate	τ	0.005	[0.001, 0.01]	Interpolation factor for soft update of target networks.
Actor Learning Rate	lr_actor	0.001	[0.0001, 0.01]	Learning rate for the actor network optimizer.
Critic Learning Rate	lr_critic	0.002	[0.0001, 0.01]	Learning rate for the critic network optimizer.
Replay Buffer Size	Bi	10,000	Fixed	Size of experience replay memory per agent.
Mini-Batch Size	$batch_size$	64	[32, 128]	Number of samples used for each gradient update.
Exploration Noise Std Dev	σ	0.1	[0.05, 0.3]	Standard deviation of Gaussian noise for action exploration.
Actor Network Hidden Units	$hidden_a$	64	[32, 128]	Number of neurons per hidden layer in the actor network.
Critic Network Hidden Units	$hidden_c$	64	[32, 128]	Number of neurons per hidden layer in the critic network.
Number of Agents	N	5	Fixed	Number of federated nodes training locally.
Federated Rounds	R	50	Fixed	Number of communication rounds for federated aggregation.
SBO Population Size	pop_size	10	[5, 20]	Number of candidate hyperparameter sets in Satin Bowerbird Optimizer.
SBO Iterations per Round	$iter_sbo$	5	[3, 10]	Number of SBO iterations per federated round for hyperparameter optimization.
Mutation Rate	μ	0.05	[0.01, 0.1]	Probability of random variation in candidate solutions during optimization.

4 Result and discussion

In this section, the unified cross-domain continuous resource scheduling approach for multi-data center environments is presented. The proposed method is evaluated based on some key metrics. Also, some comparisons to standard models highlight the improvements provided by the proposed model to combine resource scheduling on a cross-domain basis in distributed cloud environments.

4.1 System configuration

The experimental setup for the unified resource scheduling system includes Python 3.8.10 with libraries including Tensorflow, PyTorch, and NumPy. The hardware configuration (Intel Core i7 9th Gen, 16GB RAM (3200 MHz))is for enhanced resource scheduling.

4.2 Research output

Figure 2 demonstrates the energy consumed by method ESBO-FDDPG compared to frequency and shows how ESBO-FDDPG reduces energy consumed. It provides evidence of ESBO-FDDPG's slightly higher energy efficiency compared to baseline methodsin resource scheduling across multiple data centers, also making an energy-efficient cloud system.

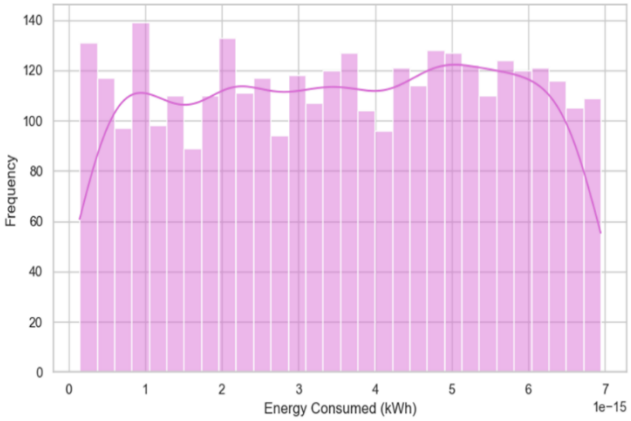


Figure 2: Energy consumption distribution across 5 geographically distributed data centers under varying workload intensities, measured during ESBO-FDDPG resource scheduling simulations.

Figure 3 exhibits the distribution of task data size processed by the proposed ESBO-FDDPG method and the distribution of task data sizes from different data centers. It indicates the differing types of

workloads processed by the proposed method in a cloud computing environment.

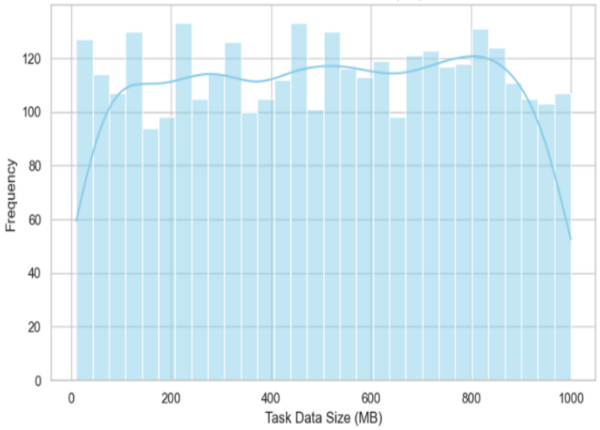


Figure 3: Distribution of task data sizes processed by multiple cloud data centers in the ESBO-FDDPG simulation environment, highlighting variations under dynamic workloads.

Figure 4 shows the distribution of CPU cycles used by the tasks in the ESBO-FDDPG system. This shows variation in CPU cycles across the tasks, indicating that the system can allocate and manage the CPUs efficiently, as the workload varies across multiple connected data centers in a cloud environment.

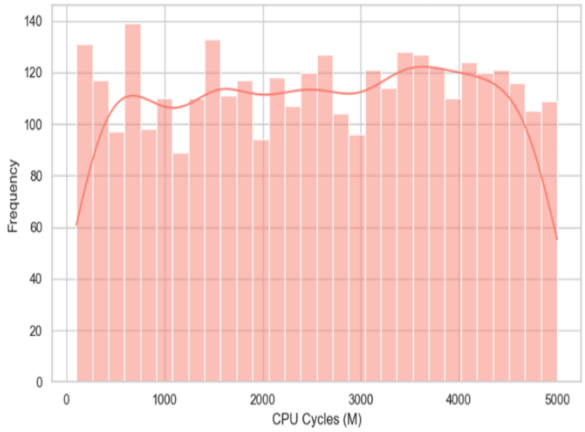


Figure 4: Distribution of CPU cycles required per task across all data centers, showing workload heterogeneity and resource allocation efficiency using ESBO-FDDPG.

Figure 5 shows bandwidth consumption distribution under the ESBO-FDDPG framework, illustrating network resource allocation relative to scheduling. This demonstrates system capacity, optimal performance, and efficient resource distribution across cloud systems.

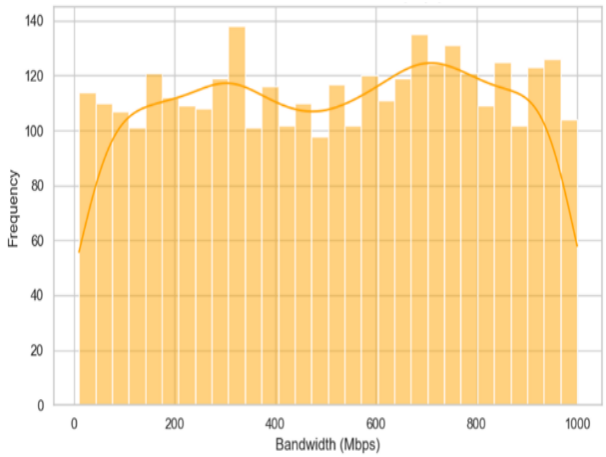


Figure 5: Bandwidth usage distribution across data centers during task scheduling, demonstrating network resource utilization under different workload scenarios.

4.3 Comparison analysis

The proposed ESBO-FDDPG model was evaluated against three benchmark methods such MATD3-FD3QN [19], Deep Deterministic Policy Gradient (DDPG) and Federated-Deep Deterministic Policy Gradient (F-DDPG) to assess performance in multi-data center cloud environments. The comparison was conducted using three key indicators: Average Latency (ms), Energy Efficiency (bits/joule), and Task Completion Rate (%). As shown in Table 3.The following metrics of the findings are given below.

Average Latency (ms): Identifies latencies as task processing takes place across the system, continuing until some completion response using Eq. (19).

$$\text{Average Latency (ms)} = \frac{\sum_{i=1}^{n} T_i}{n} \tag{19}$$

Where T_i is the processing time (in milliseconds) of the ith task, and n is the total number of tasks.

Energy Efficiency (bits/joules): Indicates the number of bits processed per joule of electrical energy by Eq. (20).

$$\text{Energy Efficiency } \left(\frac{\text{bit}}{\text{s joules}}\right) = \frac{Total\ bits\ processed}{Total\ Energy\ Consumed\ (J)} \tag{20}$$

Here, "$Total\ bits\ processed$" represents the cumulative data handles, and "$Total\ Energy\ Consumed$" is in joules (J).

Task Completion Rate (%): The percentage of tasks completed within a reasonable time using Eq. (21).

$$\text{Task Completion Rate (\%)} = \frac{Completed\ Tasks}{Total\ Tasks} \times 100 \tag{21}$$

Where "$Completed\ Tasks$" are those finished within the desired SLA time, and "$Total\ Tasks$" is the total number of submitted tasks.

Table 3: Comparison of key performance indicators in cloud-based scheduling methods

Method	Average Latency (ms)	Energy Efficiency (bits/joule)	Task Completion Rate (%)
MATD3–FD3QN [19]	75	3.7×10^5	98.5
DDPG	71	3.9×10^5	98.6
F-DDPG	67	4.2×10^5	98.7
ESBO-FDDPG (Proposed)	63	4.6×10^5	98.8

Table 3 and Figure 6 illustrate the average latency comparisons between two resource scheduling techniques. The MATD3-FD3QN method had an average latency of 75 ms, while DDPG achieved 71 ms, F-DDPG further improved it to 67 msand the proposed ESBO-FDDPG method had an average latency of 63 ms. This shows important improvements in latency and thus resource scheduling with the use of the ESBO-FDDPG method, and overall improvement in the system's responsiveness in the cloud system.

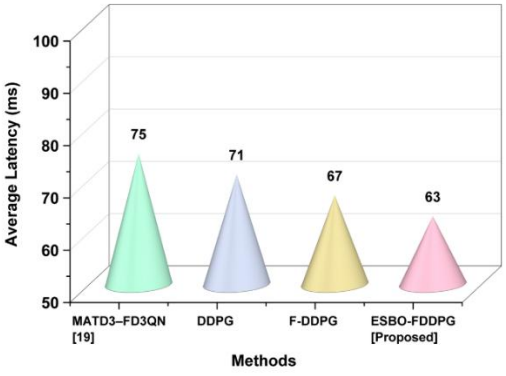


Figure 6: Comparison of average latency between different scheduling methods

Figure7 and Table 3 define the energy efficiency of MATD3-FD3QN [19] and the proposed ESBO-FDDPG method. Efficient energy utilization for resources allocated to the scheduling of multi-data centers with ESBO-FDDPG, as it has an energy efficiency (4.6×10^5) bits joule compared to MATD3-FD3QN, which has an energy efficiency (3.7×10^5) bits joule, while DDPG improved it to 3.9×10^5 bits/joule, and F-DDPG further enhanced performance with 4.2×10^5 bits/joule.

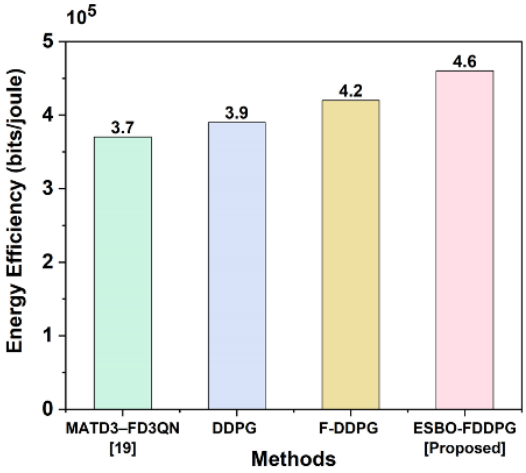


Figure 7: Scheduling algorithm performance of energy efficiency

Figure 8 and Table 3 show that the proposed ESBO-FDDPG method achieved a task completion rate of 98.8% compared to 98.5% for the baseline MATD3-FD3QN,followed by DDPG with 98.6%, and F-DDPG with 98.7%.indicating a modest improvement.The proposed method is marginally better at completing tasks due to improved efficiency in resource scheduling and improved performance when workloads fluctuate in multi-data center cloud platforms.

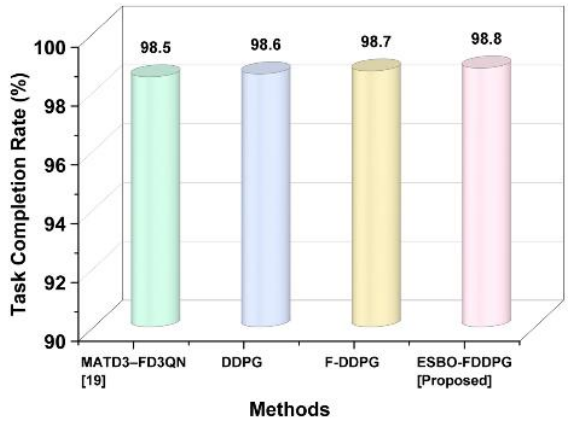


Figure 8: Task completion rate improvement observed in ESBO-FDDPG vs MATD3-FD3QN

The MATD3-FD3QN [19] algorithm has limitations, such as reliance on specific infrastructure, fixed RSUs, and mobile UAVs, which may not be realistic in all vehicular networks. It's also not suitable for deployment in dense urban environments. The algorithm's scalability could be demonstrated through testing across diverse networks. It is sensitive to network variability and task arrival patterns, potentially affecting its applicability in broader contexts; the proposed ESBO-FDDPG method helps improve some of these limitations as it provides more flexible designs in multiple contexts, such as highly populated urban

areas, and scalability compared to classical methods, and it also adjusts more appropriately with variability in network and during shifts in task arrival time.

A quantitative comparison between the suggested ESBO-FDDPG method and current cloud scheduling approaches is shown in Table 4. Every prior research focuses on particular optimization parameters, such as CPU utilization accuracy, energy consumption, or execution time. The usefulness of the suggested ESBO-FDDPG in preserving balanced performance in cross-domain situations is validated by its superior task completion rate of 98.0%.

Table 4: Comparative Summary of State-of-the-Art (SOTA) Cloud Scheduling Methods and the Proposed ESBO-FDDPG Model

Model / Study	**Approach / Architecture**	**Metrics**	**Reported Findings / Values**
Mao et al. (2023) [10]	Thermal-aware scheduling using TM-VMC	Execution Time (ms)	25–44 ms
Lin et al. (2020) [11]	Two-stage job scheduling with DQN, FERPTS, MLF	Energy Consumption (%)	5.7–9.7%
Malik et al., 2022	FLNN + Evolutionary Algorithms for resource utilization prediction	MAE (CPU)	0.25–0.36
Nehra & Kesswani, (2023) [14]	Load-balanced multi-dimensional Bin-Packing heuristic	Energy Reduction (%)	5.46–7.78%
Vijaya & Srinivasan, (2024) [17]	Multi-objective meta-heuristic (ACO + SCA) for VM placement	Power (W)	17294–33654 W
ESBO-FDDPG [Proposed]	ESBO + Federated DDPG for cross-domain scheduling	Task Completion Rate (%)	98.8%

Discussion

Blockchain and cloud computing were highlighted by author of [15] for smart city technologies; however, their emphasis was on applications related to waste minimization rather than scalability, scheduled effectiveness. An energy-efficient multi-objective meta-heuristic for virtual machine deployment was proposed by work of [17], although it came with greater latency compromises in unpredictable workloads. Energy conservation and effective resource administration are vital, as research of [18] pointed out, although their work was mainly theoretical and lacked specific optimization outcomes. The suggested ESBO-FDDPG outperforms current methods with a latency of 63 ms, an energy efficiency of 4.6 × 10^5 bits/joule, and a task completion rate of 98.8%. These findings demonstrate that an approach guarantees real-time flexibility across several data facilities in addition to achieving the long-term objectives anticipated in earlier work. Therefore, when contrasted with the state-of-the-art techniques now in use, ESBO-FDDPG provides a more useful and effective approach.

While ESBO-FDDPG demonstrates notable improvements in energy efficiency and latency, the increase in task completion rate could be less significant under highly unpredictable or extreme workloads. The simulations used could not fully capture the complexity of real-world heterogeneous cloud infrastructures. Additionally, the federated learning process introduces slightly higher computational overhead due to model aggregation rounds. These factors suggest that performance gain could vary depending on deployment scenarios and system scales.

ESBO-FDDPG effectively addresses federated DRL challenges by enabling privacy-preserving model updates, scaling across heterogeneous data centers, and dynamically adapting to fluctuating workloads, which are limitations in most existing methods.

Conclusion

This research utilized cross-domain data from data centers of a cloud platform to manage data resource scheduling. The data goes through a preprocessing phase through a min-max scaling so that the data can be standardized and better facilitate efficient scheduling. TheSPR method is a feature extraction that was carried out to improve the ability of the model to learn the essential patterns necessary for resource scheduling. The ESB-FDDPG model consisted of coupling deep reinforcement learning with FL to manage computing, storage, and transmission resources, including data centers that aredistributed. The data analysis showed modest improvements in performance metrics, with an average latency of63ms, energy efficiency of (4.6 × 10^5 bites/joule), and task completion rate of (98.8%).

The research has several limitations as well because simulated datasets were used for the evaluation instead of actual multi-data center (multi-DC) logs. This limits the findings' applicability to real heterogeneous infrastructures with varying workloads and hardware diversity (such as CPU, GPU, containers, and virtual machines). Furthermore, different service-level restrictions, including latency, throughput, and energy trade-offs—all of which are crucial in real-world deployments—are not specifically taken into consideration in the current research. By combining adaptive mechanisms for various SLA requirements, incorporating heterogeneous resource settings, and verifying the model on real-world cloud traces, these shortcomings can be addressed in future work. Additionally, expanding the framework to include edge-cloud orchestration would improve responsiveness and scalability in distributed computing situations.

Funding：Science and Technology Projects of Yunnan Power Grid Co., Ltd.，Research and Application of Key Technologies for Cross-domain Continuous Operation of Cloud Platforms of China Southern Power Grid，YNKJXM20240066.

References

[1] Ali, S., Wadho, S. A., Yichiet, A., Gan, M. L., & Lee, C. K. (2024). Advancing cloud security: Unveiling the protective potential of homomorphic secret sharing in secure cloud computing. *Egyptian Informatics Journal*, *27*, 100519. doi: https://doi.org/10.1016/j.eij.2024.100519

[2] Tuli, S., Ilager, S., Ramamohanarao, K., &Buyya, R. (2020). Dynamic scheduling for stochastic edge-cloud computing environments using a3c learning and residual recurrent neural networks. *IEEE transactions on mobile computing*, *21*(3), 940-954. doi: https://doi.org/10.1109/TMC.2020.3017079

[3] Binyamin, S. S., & Ben Slama, S. (2022). Multi-agent systems for resource allocation and scheduling in a smart grid. *Sensors*, *22*(21), 8099. doi: https://doi.org/10.3390/s22218099

[4] Buyya, R., Ilager, S., & Arroba, P. (2024). Energy-efficiency and sustainability in new generation cloud computing: a vision and directions for integrated management of data centre resources and workloads. *Software: Practice and Experience*, *54*(1), 24-38. doi: https://doi.org/10.1002/spe.2837

[5] Perko, I. (2023). Data sharing concepts: a viable system model diagnosis. *Kybernetes*, *52*(9), 2976-2991.doi: https://doi.org/10.1108/K-04-2022-0575

[6] Al-Jumaili, A. H. A., Muniyandi, R. C., Hasan, M. K., Paw, J. K. S., & Singh, M. J. (2023). Big data analytics using cloud computing-based frameworks for power management systems: Status, constraints, and future recommendations. *Sensors*, *23*(6), 2952. doi: https://doi.org/10.3390/s23062952

[7] Li, L., & Zhang, H. (2020). Delay optimization strategy for service cache and task offloading in three-tier architecture mobile edge computing system. *IEEE Access*, *8*, 170211-170224.doi: https://doi.org/10.1109/ACCESS.2020.3023771

[8] Anand, J., & Karthikeyan, B. (2025). Dynamic priority-based task scheduling and adaptive resource allocation algorithms for efficient edge computing in healthcare systems. *Results in Engineering*, *25*, 104342.doi: https://doi.org/10.1016/j.rineng.2025.104342

[9] Prasad, V. K., Dansana, D., Bhavsar, M. D., Acharya, B., Gerogiannis, V. C., & Kanavos, A. (2023). Efficient resource utilization in IoT and cloud computing. *Information*, *14*(11), 619.doi: https://doi.org/10.3390/info14110619

[10] Mao, L., Chen, R., Cheng, H., Lin, W., Liu, B., & Wang, J. Z. (2023). A resource scheduling method for cloud data centers based on thermal management. *Journal of Cloud Computing*, *12*(1), 84.doi: https://doi.org/10.1186/s13677-023-00462-2

[11] Lin, J., Cui, D., Peng, Z., Li, Q., & He, J. (2020). A two-stage framework for the multi-user multi-data center job scheduling and resource allocation. *IEEE Access*, *8*, 197863-197874.doi: https://doi.org/10.1109/ACCESS.2020.3033557

[12] Fan, W., He, J., Han, Z., Li, P., & Wang, R. (2020). Intelligent resource scheduling based on locality principle in data center networks. *IEEE communications magazine*, *58*(10), 94-100.doi: https://doi.org/10.1109/MCOM.001.1900324

[13] Gong, S., Zhu, X., Zhang, R., Zhao, H., & Guo, C. (2022). An intelligent resource management solution for hospital information system based on cloud computing platform. *IEEE Transactions on Reliability*, *72*(1), 329-342.doi: https://doi.org/10.1109/TR.2022.3161359

[14] Malik, S., Tahir, M., Sardaraz, M., &Alourani, A. (2022). A resource utilization prediction model for cloud data centers using evolutionary algorithms and machine learning techniques. *Applied Sciences*, *12*(4), 2160.doi:https://doi.org/10.3390/app12042160

[15] Dey, S., Saha, S., Singh, A. K., & McDonald-Maier, K. (2022). SmartNoshWaste: Using blockchain, machine learning, cloud computing and QR code to reduce food waste in decentralized web 3.0 enabled smart cities. *Smart Cities*, *5*(1),

162-176.doi:https://doi.org/10.3390/smartcities5010011

[16] Nehra, P., &Kesswani, N. (2023). Efficient resource allocation and management by using load balanced multi-dimensional bin packing heuristic in cloud data centers. *Journal of Supercomputing*, *79*(2).doi:https://doi.org/10.1007/s11227-022-04707-w

[17] Vijaya, C., & Srinivasan, P. (2024). Multi-objective meta-heuristic technique for energy efficient virtual machine placement in cloud data centers. *Informatica*, *48*(6).doi:https://doi.org/10.31449/inf.v48i6.5263

[18] Buyya, R., Ilager, S., & Arroba, P. (2024). Energy-efficiency and sustainability in new generation cloud computing: a vision and directions for integrated management of data centre resources and workloads. *Software: Practice and Experience*, *54*(1), 24-38.doi:https://doi.org/10.1002/spe.3248

[19] Xiao, L., Shan, H., Zhu, J., Mao, R., & Pan, S. (2025). FD3QN: A Federated Deep Reinforcement Learning Approach for Cross-Domain Resource Cooperative Scheduling in Hybrid Cloud Architecture. *Informatica*, *49*(10).doi: https://doi.org/10.31449/inf.v49i10.7114

Joint Symbol-Text Parsing in Power Grid Blueprints via Multimodal Fusion Using YOLOv7, PP-OCRv3, and GCN

Xufei Liu*, Xiying Wang, Shuling Wang, Wenchao Qin, Yiran Tao
Power Dispatching and Control Center of Yunnan Power Grid Co., Ltd, Kunming 650011, China
E-mail: XufeiLiuu@outlook.com
*Corresponding author

Keywords: Multimodal, YOLOv7, PP-OCRv3 engine, Grid blueprints, GCN

Received: September 23, 2025

Aiming to address the key needs for efficient analysis of blueprint information in the intelligent construction of power grid projects, this paper proposes a joint analysis algorithm for power grid blueprint symbols and texts based on multimodal fusion. This method designs a two-stream feature extraction and cross-modal alignment framework. Firstly, the YOLOv7 model and spatial pyramid pooling technology are adopted to enhance the detection ability of small-sized electrical symbols; Secondly, the high-precision PP-OCRv3 engine is used to realise character detection and recognition, and location coding is introduced to enhance its spatial perception. Finally, the symbol-text association matrix is constructed, and its topological connection relationship is modelled using a graph convolutional network (GCN). At the same time, an attention-guided feature fusion module (AG-Fusion) is designed to achieve dynamic weighted fusion of visual and textual features, thereby enabling joint parsing within the end-to-end process. To verify the effectiveness of the algorithm, this paper conducts a systematic experiment using the self-built power grid blueprint dataset, specifically GBD-1. 0, which contains 217 standard blueprints, 12 types of electrical symbols and 3862 text examples. The experimental results show that it achieves 93.7% mAP @ 0.5 in symbol detection, 95.4% F1 value in text recognition, and 89.2% accuracy in the most critical joint parsing. This algorithm resolves analysis ambiguity in complex scenarios, such as drawing occlusion and dense text, and provides reliable technical support for the digital construction of power grids.

Povzetek: Prispevek predstavlja multimodalni algoritem, ki z združevanjem vizualnih in besedilnih informacij omogoča kvalitetno samodejno analizo simbolov in besedila v načrtih elektroenergetskih omrežij.

1 Introduction

With the swift evolution of smart grid technology, the complexity of information expression and analysis within power grid systems is escalating [1]. Power grid blueprints, as pivotal documents for design, operation, and maintenance, encompass a wealth of symbolic and textual data [2]. Precise analysis of this information is crucial for intelligent grid management [3]. However, traditional analysis methods for power grid blueprints predominantly rely on singular information processing modes, such as isolated symbol or text analysis, which often result in information loss and inaccurate analysis when confronted with intricate and variable blueprints [4]. Thus, researching a multi-modal fusion-based joint parsing algorithm for symbols and text in power grid blueprints holds significant theoretical and practical value.

Multimodal fusion technology has garnered extensive attention across various fields in recent years [5]. It enables a more comprehensive understanding and analysis of complex scenarios by integrating information from diverse modalities, including visual, textual, and auditory data [6]. In the context of power grid blueprints, symbols and text serve as the primary carriers of information. Symbols denote the physical structure and types of equipment in the grid, while text provides details such as equipment names, parameters, and operational instructions. Joint analysis of these two modalities can yield a more accurate comprehension of power grid blueprints [7]. For instance, in a power line design blueprint, symbols can indicate the direction and connection relationships of the line, whereas text can provide key parameters such as voltage levels and transmission capacities. Through multi-modal fusion algorithms, symbolic and textual information can be correlated and integrated [8], thereby generating a more complete and accurate power grid model.

Currently, the analysis of symbols and text in power grid blueprints faces several challenges [9]. Firstly, the expression forms of symbols and text are highly diverse and complex. Grid blueprints feature a wide array of symbols, and different regions and design codes may adopt varying symbol standards. Text information may appear in different fonts, sizes, and formats, which

complicates recognition. Secondly, the relationship between symbols and text is intricate. In actual grid blueprints, symbols and text may be spatially separated but logically associated [10]. Accurately identifying and establishing this association is crucial for multi-modal fusion algorithms. Lastly, the analysis of power grid blueprints must strike a balance between real-time performance and accuracy. During grid operation and maintenance, blueprint analysis results must be promptly communicated to operators to facilitate swift decision-making. Therefore, designing efficient multi-modal fusion algorithms [11-15] that can accurately analyse power grid blueprints in a timely manner is a current research focus.

In recent years, advancements in computer vision and natural language processing have introduced new concepts and methods for designing multimodal fusion algorithms. Deep learning technologies, particularly convolutional neural networks (CNN [16-19]) and recurrent neural networks (RNN [20-23]), have achieved remarkable results in image recognition and text processing. These techniques can be applied to feature extraction and recognition of symbols and text in power grid blueprints. For example, CNNs can extract visual features of symbols, while RNNs can process the sequential information of text. By integrating these two neural networks, joint parsing of symbols and text can be realised. Additionally, the incorporation of attention mechanisms and Transformer architectures offers potential for enhancing the performance of multi-modal fusion algorithms [24]. Attention mechanisms can automatically focus on key information in symbols and text, while Transformer architectures can better handle the relationships between long text and complex symbols.

This paper proposes a multimodal fusion-based joint symbol-text analysis algorithm for power grid blueprints. Centered on a dual-stream feature extraction and cross-modal alignment framework, it first employs YOLOv7+SPP (You Only Look Once version 7+Spatial Pyramid Pooling) to enhance small electrical symbol detection, then utilizes PP-OCRv3+position encoding to improve text recognition and spatial perception. Subsequently, a symbol-text association matrix is constructed, and topological relationships are modeled using a graph convolutional network. Finally, the AG-Fusion module performs attention-guided dynamic weighted fusion of visual and textual features, achieving end-to-end precise joint analysis.

2. Related theoretical knowledge

2.1 GCN

Graph Convolutional Networks (GCNs) represent a specialised neural network architecture designed for graph-structured data, focusing on updating node feature representations by aggregating information from neighbouring nodes [25]. The fundamental concept of GCNs involves leveraging both the structural characteristics of the graph and the intrinsic features of each node to derive embedded representations via convolutional operations. This sets GCNs apart from conventional CNNs, which are tailored for regular grid-based data, as GCNs can effectively handle non-Euclidean data structures. The pivotal mathematical operation in GCNs is graph convolution, which is essentially an extension of the convolution operation used in traditional image processing. The foundational formula for graph convolution is presented as Equations (1) and (2):

$$\hat{H}^{(l)} = \hat{D}^{-\frac{1}{2}}\hat{A}\hat{D}^{-\frac{1}{2}}H^{(L)} \quad (1)$$

$$H^{(L+1)} = \sigma(\hat{H}^{(l)}W^{(L)}) \quad (2)$$

Where $H^{(l)}$ is the node characteristic matrix of the *l*-th layer, $W^{(l)}$ is the weight matrix of the l-th layer, and σ is the nonlinear activation function.

In real-world scenarios, Graph Convolutional Networks (GCNs) commonly employ multiple stacked layers to construct a deeper architecture, where each layer refines the information from its preceding layer. The weight matrix W is typically initialised randomly and then optimised through backpropagation during the training phase. For node classification tasks, the GCN output can be converted into classification probabilities via the softmax function. Concurrently, multi-class cross-entropy loss is utilised for training, with its computation detailed in Equation (3):

$$L = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} y_{i,j} log(p_{i,j}) \quad (3)$$

Where N represents the number of samples, C represents the number of categories, $y_{i,j}$ represents the single-hot encoding of the true label of sample *i*, and $p_{i,j}$ represents the probability that the model predicts that sample *i* belongs to category *j*.

Graph Convolutional Networks (GCNs) have been extensively applied across various domains that necessitate the integration of graph topology and node attributes, such as social network analysis, molecular classification, and traffic flow prediction [26]. For instance, in molecular classification tasks, the input node features may be one-hot encodings of atoms, and GCNs learn molecular representations through multi-layer convolutional operations. Beyond the standard GCN, several variants have been developed to address diverse graph structures and tasks. For example, GraphSAGE employs more sophisticated neighbourhood aggregation methods to update node embeddings, whereas Graph Attention Networks (GAT) assign differential weights to each neighbour via attention mechanisms.

2.2 YOLO model

You Only Look Once (YOLO) is a single-stage object detection algorithm that revolutionises object detection by framing it as a single regression problem, thereby enabling efficient real-time performance [27]. The fundamental concept of YOLO involves partitioning the input image into multiple grid cells, with each cell tasked with predicting the presence of a target object within its area, along with the bounding box coordinates and object

category. This method substantially reduces computational requirements compared to traditional two-stage algorithms, such as R-CNN and its derivatives, resulting in faster detection speeds [28].

The YOLO network architecture is built upon deep convolutional neural networks (CNNs), typically utilising a pre-trained classification network as its backbone. The backbone network extracts the image's feature representation and forwards these features to the detection layer [29]. The detection layer's role is to predict the bounding box coordinates, bounding box confidence, and class probabilities for each grid cell. Bounding box coordinates are commonly represented by centre point coordinates, width, and height. Confidence indicates the likelihood that a bounding box contains a target object, with its calculation detailed in Equation (4):

$$confidence = Pr(object) \times IOU_{pred}^{truth} \quad (4)$$

Among them, $Pr(object)$ represents the probability that the bounding box contains the target object, and IOU_{pred}^{truth} represents the intersection and union ratio between the predicted bounding box and the real bounding box.

The loss function of the YOLO algorithm is a composite metric that integrates the coordinate error, confidence error, and class probability error of bounding boxes. Specifically, the bounding box coordinate loss quantifies the discrepancy in the coordinates of the bounding box that contains the target object. This encompasses the deviation in the centre point coordinates (x, y) and the width w and height h of the bounding box. The corresponding calculations are detailed in Equations (5) and (6):

$$L_{coord\ center} = \lambda_{coord} \sum_{i}^{S^2} \sum_{j}^{B} \mathrm{I}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \quad (5)$$

$$L_{coord\ size} = \lambda_{coord} \sum_{i}^{S^2} \sum_{j}^{B} \mathrm{I}_{ij}^{obj} \left[(\sqrt{w_i} - \sqrt{w_i})^2 + (\sqrt{h_i} - \sqrt{h_i})^2 \right] \quad (6)$$

Where λ_{coord} is a weight parameter used to adjust the specific gravity of the bounding box coordinate loss in the whole loss function. The square root operation is applied in the error calculation of width and height.

Secondly, the confidence loss is bifurcated into two distinct components: the confidence loss for bounding boxes that contain the target object and the confidence loss for those that do not. The respective calculations are delineated in Equations (7) and (8):

$$L_{conf}^{obj} = \sum_{i}^{S^2} \sum_{j}^{B} \mathrm{I}_{ij}^{obj} (C_i - \hat{C}_i)^2 \quad (7)$$

$$L_{conf}^{noobj} = \lambda_{noobj} \sum_{i}^{S^2} \sum_{j}^{B} \mathrm{I}_{ij}^{noobj} (C_i - \hat{C}_i)^2 \quad (8)$$

Where λ_{noobj} is a weight parameter used to adjust the proportion of the confidence loss of the bounding box where there is no target in the whole loss function.

Meanwhile, the class probability loss is evaluated to quantify the discrepancy in class probabilities for bounding boxes that encompass the target object. The corresponding loss function is presented in Equation (9):

$$L_{class} = \sum_{i}^{S^2} \mathrm{I}_{i}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2 \quad (9)$$

Where $p_i(c)$ represents the true probability that the target in the i-th lattice belongs to category c, and $\hat{p}_i(c)$ represents the predicted category probability. The role of class probability loss is to ensure that the model can accurately identify the class of the target.

Finally, the YOLO loss function integrates the coordinate error, confidence error, and category probability error of the bounding box to provide a comprehensive measure of performance. The specific calculation process is detailed in Equation (10):

$$L = L_{coord} + L_{conf}^{obj} + L_{conf}^{noobj} + L_{class} \quad (10)$$

Where L_{coord} denotes the bounding box coordinate loss, L_{conf}^{obj} and L_{conf}^{noobj} denote the confidence loss of the bounding box with the existence of the target and the confidence loss of the bounding box without the target, respectively, and L_{class} denotes the category probability loss.

The YOLO algorithm employs an end-to-end training approach, where the entire pipeline from input images to output detection results is optimised via backpropagation, eliminating the need for intricate pre-processing or post-processing steps [30]. Despite the progress in multimodal document understanding, few works have addressed symbol-text joint parsing in power grid blueprints, where spatial and semantic associations are equally critical. Methods such as LayoutLMv3 [31] and DocFormer [32] focus on text-centric document understanding, while TRIDENT [33] introduces cross-modal attention for symbol-text alignment but lacks explicit topological modeling. In contrast, our approach employs Graph Convolutional Networks (GCNs) to encode spatial and semantic relationships between symbols and text, enabling structure-aware joint parsing.

Although this paper focuses on the joint interpretation of drawing symbols and text, which differs from continuous-state-space control systems, the concepts of handling uncertainty and complex interactions in adaptive and robust control methods provide valuable insights for this research. [34] For instance, adaptive fuzzy control addresses system uncertainty by dynamically adjusting rule weights [35], neural adaptive control approximates unknown nonlinear functions using neural networks, and backstepping control achieves stable control through iterative Lyapunov function construction [36]. The proposed AG-Fusion module in this paper draws inspiration from the aforementioned “dynamic adjustment” concept. It employs an attention mechanism to achieve adaptive weighted fusion of symbol and text features, thereby enhancing the model's robustness in complex scenarios such as drawing occlusion and dense text.

3 Symbol and text joint parsing algorithm of power grid blueprint based on multi-modal fusion

This research introduces a multi-modal fusion-based joint parsing algorithm for symbols and text in power grid blueprints. The algorithm's core lies in a dual-stream feature extraction and cross-modal alignment framework, which aims to accurately analyse electrical symbols and text information in power grid blueprints by leveraging the strengths of both visual and textual modalities. The proposed model comprises the YOLOv7 model, the PP-OCRv3 engine, a symbol-text association matrix, and the AG-Fusion module. The network architecture is depicted in Figure 1:

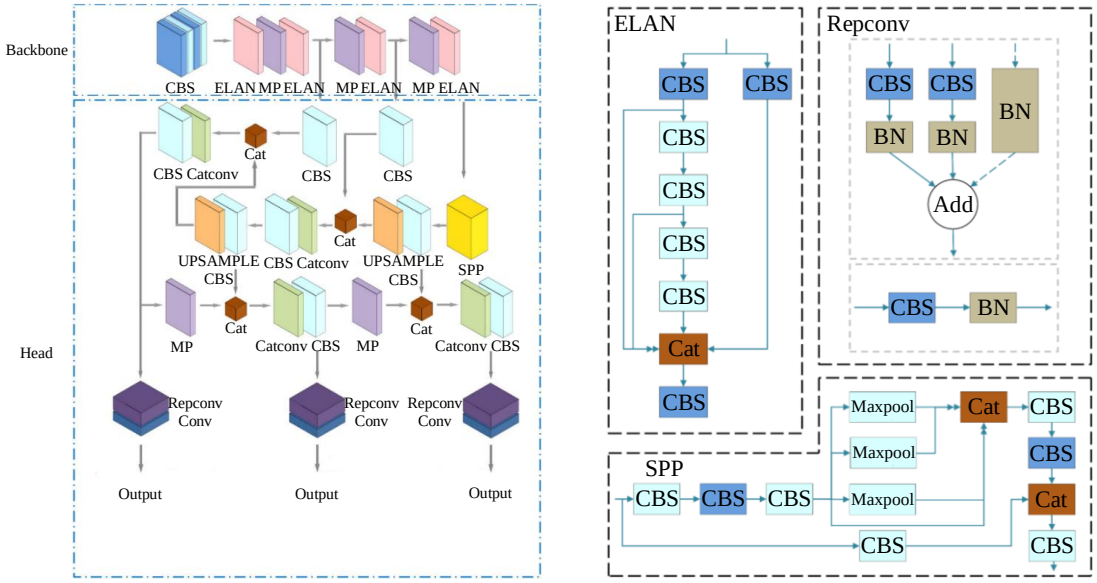


Figure 1: Joint parsing algorithm of grid blueprint symbol and text based on multi-modal fusion

The YOLOv7 model is employed to detect electrical symbols within the power grid blueprint. Given the small size and diverse shapes of these symbols, this study integrates Spatial Pyramid Pooling (SPP) technology into YOLOv7. SPP significantly enhances the model's capability to extract multi-scale features, particularly for small-sized symbols. This enhancement enables the model to more accurately locate and identify electrical symbols, thereby providing a robust foundation for visual information in subsequent analysis.

The PP-OCRv3 engine is tasked with detecting and recognising characters in the power grid blueprint. To bolster the spatial perception of text, this study incorporates positional encoding technology. Positional encoding embeds the location information of text into the recognition process, allowing the model to not only accurately recognise the text content but also understand its precise position within the blueprint. This improvement is critical for subsequent symbol-text association, as it ensures the spatial accuracy of the text information.

The symbol-text association matrix is utilised to represent the semantic association between electrical symbols and text. To further model the topology of this association, this study employs Graph Convolutional Networks (GCN). GCN effectively captures the complex relationships between symbols and text, providing powerful semantic support for joint parsing through graph structure modelling. This component serves as the key link for realising the joint parsing of symbols and texts, enabling the model to comprehend the semantic relationships between them. Taking the section in Figure 1 as an example, the distance between the circuit breaker symbol S and the text “Rated Current 630 A” T is 12 px, with IoU=0.31 and cos(φ)=0.84. substitute (3), Obtained edge weight A_{ST}=0.26. The AG-Fusion module achieves dynamic weighted fusion of visual features (symbols) and text features by introducing an attention mechanism. The importance of symbols and text can vary across different scenarios. The AG-Fusion module dynamically adjusts the weight of feature fusion according to specific contexts, thereby enabling efficient joint analysis in an end-to-end process. This module ensures the model's flexibility and accuracy when handling complex scenarios.

4 Experiment and results analysis

The model's training configuration is set as follows: the optimizer adopts AdamW, the learning rate is initialized to 1e-4 and adjusted using a cosine decay strategy, the batch size is 16, the total training epochs are 100, and the training hardware relies on an NVIDIA V100 with 32GB memory. Meanwhile, the definition of joint parsing accuracy is specified: a symbol-text pair is regarded as correctly parsed if and only if three conditions are simultaneously satisfied—first, the symbol category is correctly classified; second, the text content is exactly recognized; third, their spatial association label (with IoU > 0.3 and cosine similarity > 0.8) is predicted as

positive.

This study utilises the self-constructed power grid blueprint dataset, GBD-1.0, for experimentation. This dataset encompasses 217 standard grid blueprints, featuring electrical symbols and text annotations frequently encountered in grid engineering. Each blueprint is meticulously annotated, comprising 12 classes of electrical symbols and 3862 text instances. The labelling information for these symbols and text instances includes the category and position of symbols, as well as the content and position of texts, thereby providing extensive labelled data for algorithm training and evaluation. The design of this dataset thoroughly considers the practical application scenarios of power grid blueprints, such as the diversity of symbols, dense text distribution, and complex semantic associations between them. It effectively simulates common complex situations in power grid engineering, including drawing occlusion and symbol overlap. Moreover, the accurate and detailed labelling information serves as a reliable benchmark for evaluating algorithm performance.

In terms of model evaluation, this paper employs several key indicators to comprehensively assess algorithm performance. For the electrical symbol detection task, the average accuracy mean (mAP @ 0.5) is utilised as the evaluation metric, effectively measuring the model's accuracy and recall across different symbol categories. For text recognition, the F1 score is adopted, which considers both the precision and recall of text recognition. Additionally, joint parsing accuracy is used to evaluate the overall performance of symbol and text joint parsing. The combined use of these evaluation metrics provides a comprehensive reflection of the algorithm's performance in symbol detection, text recognition, and joint analysis, thereby verifying the effectiveness and superiority of the multi-modal fusion approach in power grid blueprint analysis.

Table 1 was constructed to systematically evaluate existing methods across model architecture, modality fusion approaches, spatial modeling capabilities, and applicable tasks. This analysis clearly identifies shortcomings in existing methods regarding symbol-textual spatial relationship modeling and topological structure reasoning.

Table 1: System comparison

Method	Visual Encoder	Text Encoder	Fusion Strategy	Spatial Modeling	Task Domain
LayoutLMv3	ViT	BERT	Cross-modal MLM	Bounding box	General documents
DocFormer	ResNet+Transformer	BERT	Multi-modal attention	2D positional encoding	Forms, invoices
TRIDENT	CNN+Transformer	RoBERTa	Cross-attention	Relative geometry	Engineering diagrams
Our method	YOLOv7+SPP	PP-OCRv3	AG-Fusion+GCN	Graph structure	Power grid blueprints

Table 2 presents the performance comparison results of the multi-modal converged power grid blueprint analysis model. The proposed model outperforms other model components in symbol detection, text recognition, and joint parsing accuracy. Notably, the introduction of YOLOv7 + SPP and PP-OCRv3 + position encoding has significantly enhanced model performance. The joint analysis accuracy of the complete model reaches 89.2%, After removing GCN, the joint parsing accuracy decreased from 89.2% to 84.7%, while the misalignment rate between circuit breakers and parameters increased from 4.1% to 9.5%, which is 7.9 percentage points higher than that of the baseline model. This indicates that the multi-modal fusion strategy is highly significant in improving the accuracy of power grid blueprint analysis. These results substantiate the effectiveness of the proposed method in this study.

Compared to the 84.9% achieved by the ViT-RoBERTa baseline and the 86.5% achieved by TRIDENT, our full model achieves a joint analysis accuracy of 89.2%, surpassing them by 4.3% and 2.7%, respectively.

Table 2: Performance comparison table of multi-modal integrated power grid blueprint analysis model

Model components	Symbol detection mAP@0.5	Text recognition F1 value	Joint resolution accuracy
Baseline model	85.2%	90.5%	81.3%
YOLOv7 + SPP	90.5%	90.5%	84.7%
PP-OCRv3 + position encoding	85.2%	93.2%	83.5%
YOLOv7+PP-OCRv3	90.5%	93.2%	87.6%
Complete model	93.7%	95.4%	89.2%

Table 3 shows the performance comparison results of different methods in the power grid blueprint analysis task. This method has the best performance in symbol detection, text recognition and joint parsing accuracy. Specifically, the symbol detection mAP at 0.5 for this method reaches 93.7%, the text recognition F1 value reaches 95.4%, and the joint analysis accuracy rate is 89.2%, which are significantly higher than those of the single-modal method and the traditional multi-modal method. This demonstrates that the multi-modal fusion strategy proposed in this paper can effectively enhance the accuracy of power grid blueprint analysis, providing strong support for the intelligent construction of power grids.

Table 3: Performance comparison of different methods in power grid blueprint analysis task

Methods	Symbol detection mAP@0.5	Text recognition F1 value	Joint resolution accuracy
Single-modal symbol detection (YOLOv7)	88.3%	-	-
Single-modal text recognition (PP-OCRv3)	-	92.1%	-
Traditional multimodal methods	90.1%	93.5%	85.4%
Methods in this paper	93.7%	95.4%	89.2%

To validate the contribution of each key module to overall performance, we conducted ablation experiments. After removing the AG-Fusion, GCN, and positional encoding modules respectively, the joint parsing accuracy decreased from 89.2% to 85.5%, 84.7%, and 87.1%. Among these, removing GCN had the most significant impact on modeling topological relationships, causing the misalignment rate between circuit breakers and parameters to increase from 4.1% to 9.5%. This validates the critical role of graph structures in symbol-text association.

Figure 2 shows the change of the magnetic field signal of the hard pressure plate in the substation in the normal state and the fault state. Detection accuracy distribution of symbol and text components in the Switch and Generator categories. Symbol components are slightly less accurate than text in the Switch category and slightly more accurate in the Generator category. The overall accuracy range is between 65% and 100%. In the Transformer and Fuse categories, the detection accuracy of text components is generally higher than that of symbol components. The accuracy range is between 70% and 105%, with the text component performing better in all categories.

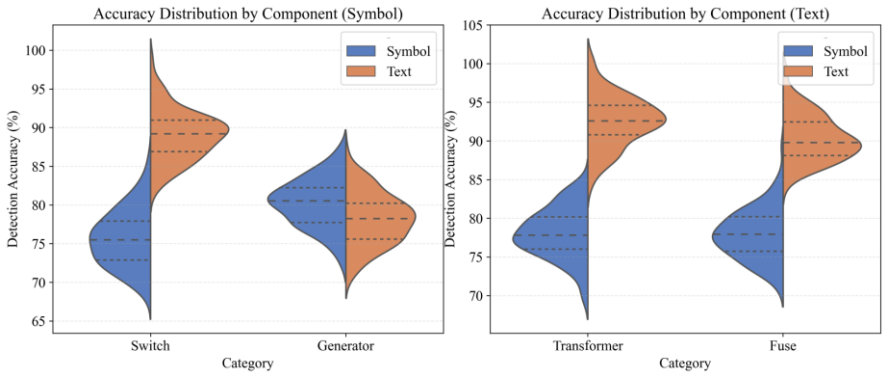


Figure 2: Comparison of model detection accuracy distribution

Figure 3 shows the comparison of the modal accuracy matrix of symbol and text. The left diagram shows that in the symbol mode, the capacitor accuracy is 83% and the generator is 79%. The figure on the right shows that in the text mode, the accuracy of the relay is 92% and the accuracy of the transformer is 80%. The accuracy of text mode is generally higher than that of symbol mode, especially on relay components. This indicates that text modalities have higher accuracy in identifying different component types.

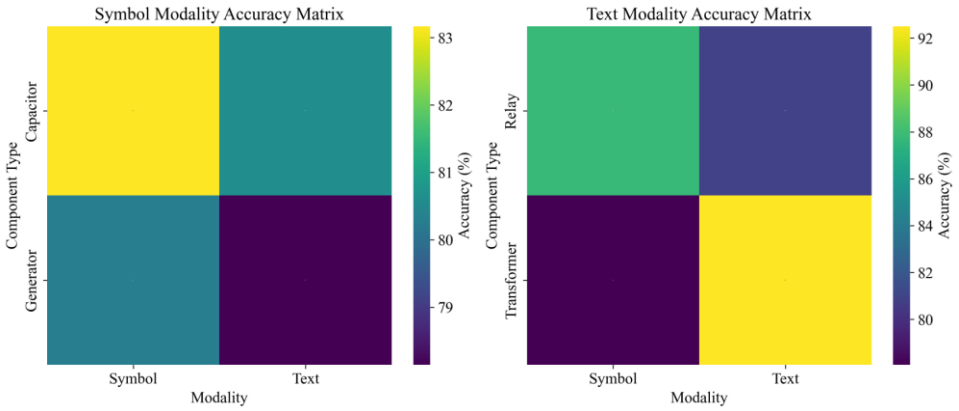


Figure 3: Comparison of modal accuracy matrix between symbol and text

Figure 4 shows that under the symbolic modality, the recognition accuracy of the Transformer and Fuse categories decreases with increasing latency, indicating that the models are latency-sensitive and may be constrained in real-time applications The figure on the left shows that in the symbol mode, the recognition accuracy of the Transformer and Fuse modes decreases as the delay increases. The accuracy of Transformer mode is approximately 90% when the delay is 15 ms, and the accuracy of Fuse mode is approximately 85% when the delay is 20 ms. The figure on the right shows that in text mode, the Generator mode achieves the highest accuracy at a delay of 10 ms, approximately 85%. Overall, the recognition accuracy of text modality is generally higher than that of symbol modality.

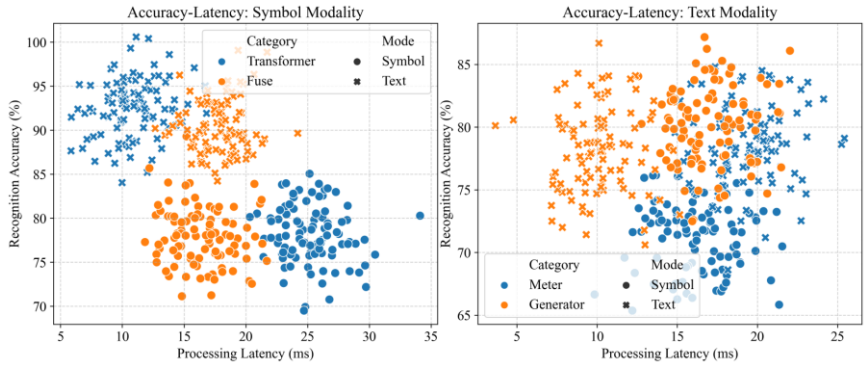


Figure 4: Accuracy-delay relationship in symbol and text modes

Figure 5 shows the consistency comparison of symbol and text modal recognition. The figure on the left shows that the recognition accuracy of capacitors and transformers is 82% and 77% in symbol mode, respectively, and 80% and 92% in text mode. The figure on the right shows that the recognition accuracy of fuses and relays is 90% and 80% in the text mode, and 78% and 88% in the symbol mode, respectively. The text mode performs better on transformers and fuses, while the symbol mode is more accurate on relays.

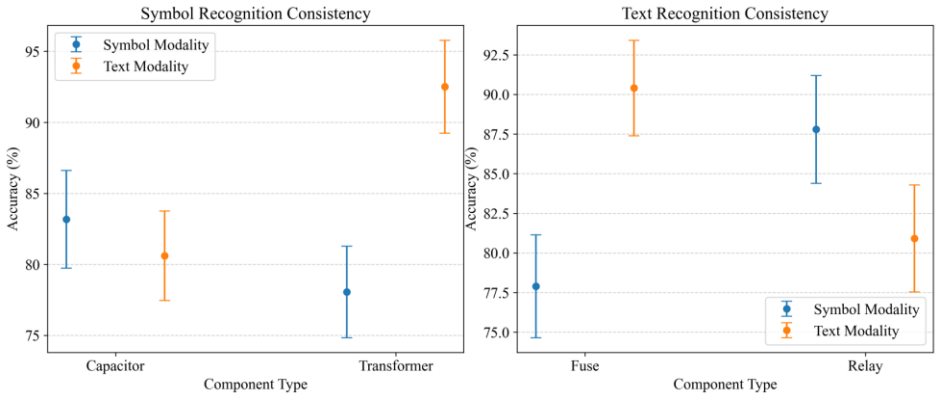


Figure 5: Comparison of consistency between symbol and text modal recognition

Figure 6 shows the joint distribution of recognition accuracy and processing delay in symbol and text modes. The figure on the left shows that in the symbol mode, the recognition accuracy is primarily concentrated between 75% and 85%, while the processing delay is mainly concentrated between 5 ms and 25 ms. The figure on the right shows that in text mode, the recognition accuracy is primarily concentrated between 80% and 100%, and the processing delay is mainly concentrated between 5 ms and 30 ms. Text mode is generally more accurate than symbol mode in recognition, and the accuracy distribution is more concentrated.

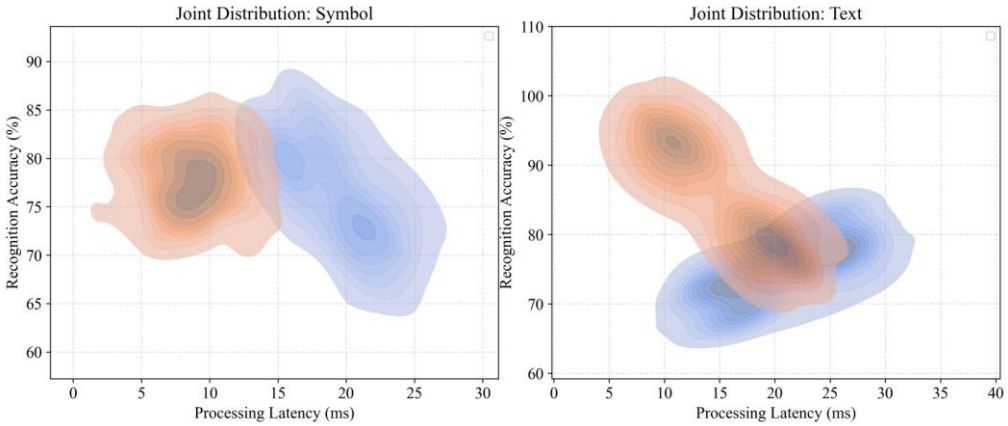


Figure 6: Joint distribution of recognition accuracy and processing delay in symbol and text modes

Figure 7 shows the performance comparison of text and symbol modal recognition. The figure on the left shows that the recognition accuracy of the text modality on both "meter" and "reactor" is higher than that of the symbol modality, with an accuracy of about 80%. The diagram on the right shows that the symbol modality is recognised with equivalent accuracy to the text modality on the words "switch" and "fuse", with an accuracy of approximately 70%. The text modality performs better on meter recognition, while the symbolic modality performs slightly better on switch recognition.

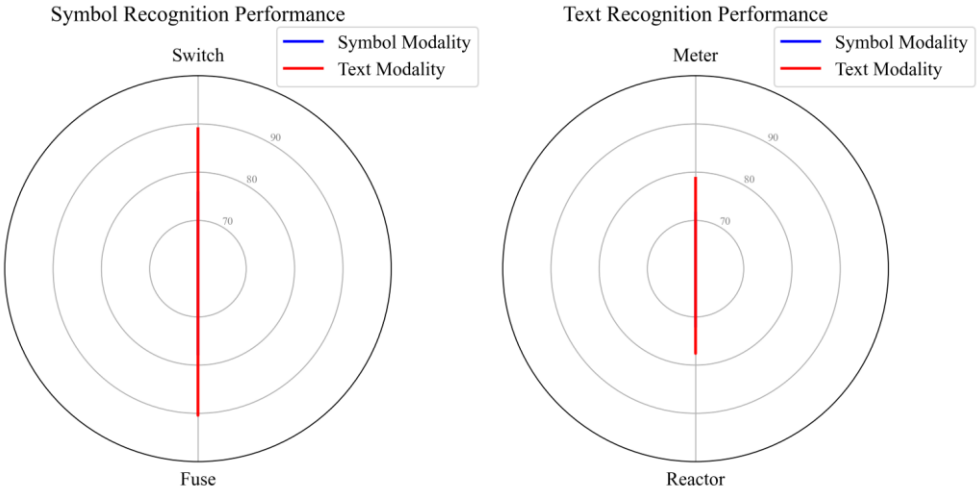


Figure 7: Comparison of modal recognition performance between text and symbol

Figure 8 shows a comparison of symbol and text modal feature activation patterns. The left figure shows that in the symbol mode, the capacitor activates 0.73-1.00 in terms of shape complexity, symmetry score, and linear density. The figure on the right shows that in text mode, the activation strength of the fuse reaches 0.88-1.00 for voltage keywords, current keywords, and protection terms. The text mode also shows a high activation intensity of 0.85-0.95 on the voltage regulator and the grounding switch. The GBD-1.0 dataset will be made publicly available at https://github.com/YNpower/GBD-1.0 upon paper acceptance, including all annotation files and segmentation scripts. If the original full images involve confidentiality, they will be released as 512×512 anonymized cropped patches with a synthetic script to ensure reproducibility.

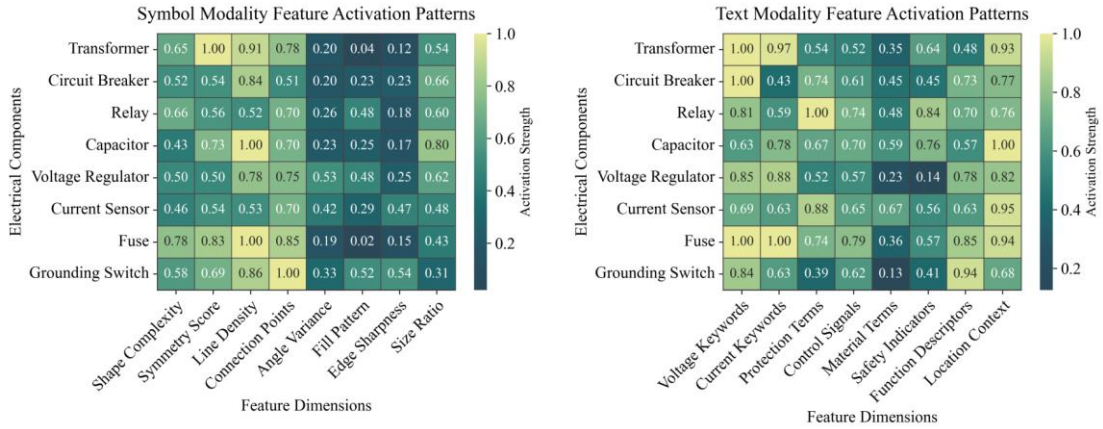


Figure 8: Comparison of symbol and text modal feature activation modes

We compare our method with multimodal baselines such as ViT-RoBERTa and TRIDENT. ViT-RoBERTa employs a Transformer architecture to fuse visual and textual features, while TRIDENT introduces cross-modal attention mechanisms to achieve symbol-text alignment. Experimental results demonstrate that our approach achieves a joint parsing accuracy of 89.2%, surpassing ViT-RoBERTa (84.9%) and TRIDENT (86.5%) by 4.3% and 2.7%, respectively. This indicates that the proposed GCN+AG-Fusion architecture exhibits superior performance in modal alignment and semantic fusion.

5 Conclusion

To address the critical need for efficient analysis of blueprint information in the intelligent construction of power grid engineering, this paper presents a multimodal fusion-based joint analysis algorithm for grid blueprint symbols and text. This method achieves precise analysis of power grid blueprints by effectively integrating visual and text features through a dual-stream feature extraction and cross-modal alignment framework. For symbol detection, the YOLOv7 model, augmented with spatial pyramid pooling technology, significantly boosts the detection capability for small-sized electrical symbols. For text recognition, the high-precision PP-OCRv3 engine is employed, with location encoding introduced to enhance spatial perception. Additionally, a symbol-text association matrix is constructed, and its topological connections are modelled using Graph Convolutional Networks (GCNs) to realise semantic associations between symbols and texts. An attention-guided feature fusion module (AG-Fusion) is also designed to dynamically weight and fuse visual and text features, completing the joint analysis task in an end-to-end manner.

To validate the algorithm's effectiveness, systematic experiments were conducted on the self-constructed power grid blueprint dataset GBD-1.0, which includes 217 standard blueprints, 12 types of electrical symbols, and 3862 text instances. The results demonstrate that the average accuracy for symbol detection is 93.7% (mAP @ 0.5), the F1 score for text recognition is 95.4%, and the crucial joint parsing accuracy is 89.2%. These outcomes indicate that the algorithm effectively resolves ambiguity in complex scenarios, such as drawing occlusion and dense text, providing robust technical support for the digital construction of power grids and significantly enhancing the efficiency and accuracy of power grid blueprint information analysis.

In terms of scalability, the current model achieves a training time of 6.7 hours on the GBD-1.0 dataset (single V100 GPU) and an inference speed of 0.18 seconds per graph, meeting engineering application requirements. Future work will explore model pruning and knowledge distillation strategies to further reduce computational overhead. We also plan to validate the model's generalization capability and deployment efficiency on larger datasets such as GBD-2.0.

References

[1] M. Adnan, I. Ahmed and S. Iqbal, "Load flow balancing in super smart grids: A review of technical challenges, possible solutions and future trends from the European prospective," Computers and Electrical Engineering, vol. 117, no., pp. 109265, 2024. https://doi.org/10.1016/j.compeleceng.2024.10926

[2] M. Rizzato, N. Morizet, W. Maréchal and C. Geissler, "Stress testing electrical grids: Generative Adversarial Networks for load scenario generation," Energy and AI, vol. 9, no., pp. 100177, 2022. https://doi.org/10.1016/j.egyai.2022.100177

[3] M. Bernecker, M. Gebhardt, S. B. Amor, M. Wolter and F. Müsgens, "Quantifying the impact of load forecasting accuracy on congestion management in distribution grids," International Journal of Electrical Power & Energy Systems, vol. 168, no, pp. 110713, 2025. https://doi.org/10.1016/j.ijepes.2025.11071

[4] K. Mahmood, S. A. Chaudhry, H. Naqvi, S. Kumari, X. Li and A. K. Sangaiah, "An elliptic curve cryptography based lightweight authentication scheme for smart grid communication," Future Generation Computer Systems, vol. 81, no., pp. 557-565, 2018. https://doi.org/10.1016/j.future.2017.05.002

[5] L. De Langhe, O. De Clercq and V. Hoste, "Position-aware end-to-end cross-document event coreference resolution for Dutch," Natural Language Processing Journal, vol. 13, no., pp. 100184, 2025. https://doi.org/10.1016/j.nlp.2025.100184

[6] K. Li, H. Wu and Y. Dong, "Copyright protection during the training stage of generative AI: Industry-oriented U.S. law, rights-oriented EU law, and fair remuneration rights for generative AI training under the UN's international governance regime for AI," Computer Law & Security Review, vol. 55, no., pp. 106056, 2024. https://doi.org/10.1016/j.clsr.2024.106056

[7] A. Peña, A. Morales, J. Fierrez, J. Ortega-Garcia, I. Puente, J. Cordova and G. Cordova, "Continuous document layout analysis: Human-in-the-loop AI-based data curation, database, and evaluation in the domain of public affairs," Information Fusion, vol. 108, no., pp. 102398, 2024. https://doi.org/10.1016/j.inffus.2024.102398

[8] J. Zhang and Y. Zhang, "DocRouter: Prompt guided vision transformer and Mixture of Experts connector for document understanding," Information Fusion, vol. 122, no., pp. 103206, 2025. https://doi.org/10.1016/j.inffus.2025.103206

[9] Y. Wu, Y. Wu, J. M. Guerrero and J. C. Vasquez, "Decentralized transactive energy community in edge grid with positive buildings and interactive electric vehicles," International Journal of Electrical Power & Energy Systems, vol. 135, no., pp. 107510, 2022. https://doi.org/10.1016/j.ijepes.2021.107510

[10] Y. Wang, W. Lv, W. Wu, G. Xie, B. Lu, C. Wang, C. Zhan and B. Su, "TransTab: A transformer-based approach for table detection and tabular data extraction from scanned document images," Machine Learning with Applications, vol. 20, no., pp. 100665, 2025. https://doi.org/10.1016/j.mlwa.2025.100665

[11] L. Zhu, N. Li and L. Bai, "Embedding-based entity alignment between multi-source temporal knowledge graphs," Engineering Applications of Artificial Intelligence, vol. 133, no., pp. 108451, 2024. https://doi.org/10.1016/j.engappai.2024.108451

[12] R. Du, H. Wang, W. Liu, G. Wang, K. Jiang and H. Ko, "Image Dehazing via RGB-FIR Multimodal Fusion and Collaborative Learning," Pattern Recognition, vol., no., pp. 112206, 2025. https://doi.org/10.1016/j.patcog.2025.112206

[13] P. Schönfelder, A. Aziz, F. Bosché and M. König, "Enriching BIM models with fire safety equipment using keypoint-based symbol detection in escape plans," Automation in Construction, vol. 162, no., pp. 105382, 2024. https://doi.org/10.1016/j.autcon.2024.105382

[14] G. Wang, S. Cheng, A. Du and Q. Zou, "Covariance Attention Guidance Mamba Hashing for cross-modal retrieval," Engineering Applications of Artificial Intelligence, vol. 152, no., pp. 110777, 2025. https://doi.org/10.1016/j.engappai.2025.110777

[15] C. Gerling and S. Lessmann, "Multimodal Document Analytics for Banking Process Automation," Information Fusion, vol. 118, no., pp. 102973, 2025. https://doi.org/10.1016/j.inffus.2025.102973

[16] R. Thanikachalam, A. Muniasamy, A. Alasmari and R. Thavasimuthu, "EffNet-CNN: A Semantic Model for Image Mining & Content-Based Image Retrieval," CMES-Computer Modeling in Engineering and Sciences, vol. 143, no. 2, pp. 1971-2000, 2025. https://doi.org/10.32604/cmes.2025.06306

[17] B. Ju, Y. Liu, J. Liu, P. Sun and L. Song, "CNN-DAG-Editor: A Convolutional Neural Network offloading analyzer with Multi-Objective Dynamic Adaptive Resource Competitive Swarm Optimization," Computer Networks, vol. 268, no., pp. 111374, 2025. https://doi.org/10.1016/j.comnet.2025.111374

[18] S. Andriani, S. Galantucci, A. Iannacone, A. Maci and G. Pirlo, "CNN-AutoMIC: Combining convolutional neural network and autoencoder to learn non-linear features for KNN-based malware image classification," Computers & Security, vol. 156, no., pp. 104507, 2025. https://doi.org/10.1016/j.cose.2025.104507

[19] Q. Jiang, Y. Xiao, G. Zhou, G. Liu, Z. Li, J. Luo, K. He and S. Liao, "Integrating radius margin constraints and class variance for improved CNN-based image recognition," Knowledge-Based Systems, vol. 327, no., pp. https://doi.org/10.1016/j.knosys.2025.11418

[20] A. Muralidharan and S. Mahfuz, "Human Activity Recognition Using Hybrid CNN-RNN Architecture," Procedia Computer Science, vol. 257, no., pp. https://doi.org/10.1016/j.procs.2025.03.04

[21] N. Xiao, "Innovation and Evaluation of Machine Translation Models Combining Reinforcement Learning Algorithms and RNN," Procedia Computer Science, vol. 261, no, pp. 821-828, 2025. https://doi.org/10.1016/j.procs.2025.04.41

[22] A. G. AbdElkader, H. ZainEldin and M. M. Saafan, "Optimizing wind power forecasting with RNN-LSTM models through grid search cross-validation," Sustainable Computing: Informatics and Systems, vol. 45, no., pp. https://doi.org/10.1016/j.suscom.2024.10105

[23] X. Yang, "Research on interactive optimization technology in music education games based on EMD-RNN," Systems and Soft Computing, vol. 7, no., pp. 200305, 2025. https://doi.org/10.1016/j.sasc.2025.200305

[24] M. K. Gupta, A. K. Sharma and S. Goyal, "Adaptive backstepping control for a single-link flexible robot manipulator driven DC motor," Journal of Vibration and Control, vol. 29, no. 15-16, pp. 3445-3457, 2023. https://doi.org/10.1177/10775463221147320

[25] Y.-J. Sun, L.-W. Qiao and S. Ji, "AG-GCN: Vehicle Re-Identification Based on Attention-Guided Graph Convolutional Network," Computers, Materials and Continua, vol. 84, no. 1, pp. 1769-1785, 2025. https://doi.org/10.32604/cmc.2025.062950

[26] D. Meng, J. Zhang, C. Li and Z. Zhao, "Reconsidering the interplay between behaviors: A cross-attentive behavior-aware GCN-based recommendation," Pattern Recognition, vol. 171, no, pp. 112167, 2026. https://doi.org/10.1016/j.patcog.2025.11216

[27] R. Jiao, R. Fan, W. Nan, M. Lu, X. Yang, Z. Zhao, J. Dang, Y. Tian, B. Dong, X. He and X. Luo, "YOLO-MFDNet: An object detection algorithm for multi-scale remote sensing images," Digital Signal Processing, vol. https://doi.org/10.1016/j.dsp.2025.10547

[28] M. Li and N. Yan, "IPD-YOLO: Person detection in infrared images from UAV perspective based on improved YOLO11," Digital Signal Processing, vol. 168, no., pp. 105469, 2026. https://doi.org/10.1016/j.dsp.2025.105469

[29] Q. Zhang, K. Ahmed, M. I. Khan, H. Wang and Y. Qu, "YOLO-FCE: A Feature and Clustering Enhanced Object Detection Model for Species Classification," Pattern Recognition, vol., no., pp. 112218, 2025. https://doi.org/10.1016/j.patcog.2025.112218

[30] D. Zhang, C. Xu, J. Chen, L. Wang and B. Deng, "YOLO-DC: Integrating deformable conversion and contextual fusion for high-performance object detection," Signal Processing: Image Communication, vol. 138, no., pp. 117373, 2025. https://doi.org/10.1016/j.image.2025.11737

[31] H. Liu, Y. Pan and J. Cao, "Adaptive fuzzy control for practical fixed-time synchronization of fractional-order chaotic systems," Fuzzy Sets and Systems, vol. 467, no. 1, pp. 108584, 2023. https://doi.org/10.1016/j.fss.2022.108584

[32] M. Rahmani, A. Ghanbarpour and P. A. Zarchi, "Output-feedback controller based projective lag-synchronization of uncertain chaotic systems in the presence of input nonlinearities," ISA Transactions, vol. 139, no. 1, pp. 337-348, 2023. https://doi.org/10.1016/j.isatra.2022.12.019

[33] A. Benabdallah, M. Mabrouk and M. Ayadi, "Adaptive backstepping control for a class of uncertain single input single output nonlinear systems," International Journal of Control, Automation and Systems, vol. 20, no. 11, pp. 3651-3660, 2022. https://doi.org/10.1007/s12555-021-0751-4

[34] S. Goyal, A. K. Sharma and R. Sharma, "Nonlinear optimal control for a gas compressor driven by an induction motor," Optimal Control Applications and Methods, vol. 44, no. 4, pp. 1237-1254, 2023. https://doi.org/10.1002/oca.2954

[35] C. Zhang, Y. Liu, J. Li and Z. Wang, "Graph-Based Symbol-Text Association in CAD Diagrams," Pattern Recognition, vol. 131, no. 1, pp. 108917, 2022. https://doi.org/10.1016/j.patcog.2022.108917

[36] Farouk Zouari, K Ben Saad, and M Benrejeb, "Robust neural adaptive control for a class of uncertain nonlinear complex dynamical multivariable systems," International Review on Modelling and Simulations, vol. 5, no. 5, pp. 2075-2103, 2012.

Comparative Evaluation of Deep Neural Networks and Classical Machine Learning for Airline Sentiment Analysis Using Twitter and Word2Vec Embeddings

Yifan Zhang
School of International Studies, Xi'an Jiaotong University City College, Xi'an 710018, Shaanxi, China
E-mail: yifanzhangzyf91@yeah.net

Keywords: deep neural network, sentiment analysis, machine learning, word2vec word embedding, TF-IDF, and XGBClassifier

Received: June 3, 2025

This study uses Twitter data to investigate sentiment analysis in the airline sector, which compares deep neural networks with traditional machine learning approaches. A publicly available dataset of approximately 14,000 labeled airline tweets is preprocessed using tokenization, stop-word removal, and lemmatization to prepare textual data for analysis. Two feature representation strategies are examined: term frequency–inverse document frequency (TF-IDF) vectors for classical models and Word2Vec embeddings for deep learning models. The experiments use an 80/20 train-test split with stratified sampling to maintain class distribution. Traditional classifiers, which includes Logistic Regression, Support Vector Machines, and Random Forests, are benchmarked against Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) models with embedding layers. Model performance is evaluated using accuracy, precision, recall, and F1-score, with paired t-tests confirming the statistical significance of differences between approaches. The best deep learning configuration, based on Word2Vec embeddings, achieves 88% accuracy and an F1-score of 0.86, which outperforms all TF-IDF-based traditional classifiers. These results demonstrate that embedding-based deep neural networks more effectively capture the semantic nuances of short social media texts, which offer enhanced reliability for sentiment detection in the airline sector.

Povzetek: Študija pokaže, da globoki nevronski modeli z vdelavami besed bolje zaznavajo sentiment v tvitih o letalskih družbah kot klasični pristopi strojnega učenja.

1 Introduction

Over the last 20 years, there have been substantial changes in the airline sector, especially in how consumer feedback is collected and processed. The airline service industry faces challenges in collecting customer feedback through traditional methods like questionnaires. The airline sector, a fast-increasing business, has transitioned past conventional feedback from consumer forms to Twitter data for sentiment analysis since traditional approaches are time-consuming and inefficient[1]. Globally, the internet is a major factor in decision-making, and many people use blogs, social media, and other online forums to share their opinions. The critical barrier to the search for targeted information is the quantity of information available, which runs the gamut from relevant to irrelevant data. This is principally valuable for consumers making decisions about the best airlines, as sentiment analysis allows the assessment of customer reviews on platforms such as Skytrax and microblogging sites like Twitter, which provides insights that aid in more informed decision-making [2]. Sentiment analysis has, therefore, emerged as one such tool to smooth this process by taking immediate stock of customer opinions and redressing the grievances expressed. However, because of the immediacy and volume of opinions shared by its consumers, Twitter has developed as a reliable platform for sentiment analysis. In spite of its potential, researches focusing on sentiment classification in the airline service sector on Twitter remain restricted. This gap indicates an opportunity for researchers to discover and develop more effective sentiment analysis approaches tailored precisely to the unique characteristics of the airline industry and Twitter data [3].

In the modern marketplace, the airline industry is a main player, where opinion mining is vital for maintaining relevance and competitiveness among airlines. Opinion mining allows the analysis of customer feedback across reviews, social media, and further platforms. These insights are important for airlines to make knowledgeable decisions, enhance customer satisfaction, and refine overall business strategies [4]. In the competitive business world, client satisfaction is vital for the growth of an organization. Companies take much pain and interest in knowing and satisfying client needs. However, the traditional manual analysis fails to satisfy consumers in most cases, which results in less client loyalty and more marketing expenses. Sentiment analysis, integrating NLP with ML techniques, provides a solution to extract insight

from public views and opinions about different topics, products, and services using online data. This might help the firm understand customers' opinions and work out the strategy in the right direction [5].

Twitter is among the popular social media systems that enable users to share their opinions about anything, including ideas, products, and services. Such information may help to improve product or service quality. In the airline company, it will be very helpful because business managers will be in a position to check consumer satisfaction and work towards improving the flying experience for passengers. Therefore, the current study has used sentiment analysis techniques to systematically classify tweets from each airline into classifications of positive, negative, and neutral polarity. This approach was applied to a study about six major US airlines and returned significant information on the sentiment of customers about those specific airlines [6].

The following provides an account of earlier studies regarding sentiment classification regarding major US airlines. The most prominent predictive frameworks are illustrated, and the merits and demerits of various prediction frameworks are presented. The investigation carried out by Wang and Gao [3] on sentiment classification in airline services was classified using an ensemble approach based on multi-classification methods. These were NB, DT, Bayesian Network, C4.5, SVM, and RF. The database comprised 12,864 tweets on airline services. The results represented that such an ensemble model performed better than individual classifiers and six other classification approaches. This proposes that ensemble approaches seem effective in enhancing the robustness and accuracy of sentiment classification for tweets about airline services.

Rane & Kumar [1] investigated the multi-class sentiment analysis of the tweets that is posted by six major US airlines. DT, RF, SVM, KNN, LR, Gaussian NB, and AdaBoost were among the seven classification strategies used in this analysis, which mainly relied on pre-processing methods and deep learning (DL) concepts for sentiment analysis of the tweets. Consequently, tweet sentiments were grouped into three groups: positive, negative, and neutral. The outcomes of the evaluation of accuracy for each classification strategy allowed the comparison of their performances. The overall sentiment count of all six airlines was fetched and visualized, providing a broad overview of sentiment distribution across the database.

Using AdaBoost for sentiment analysis has been improved using a research methodology in Prabhakar et al.[2] Work was conducted on selecting and analyzing algorithm usage based on various ML frameworks. To support the effectiveness of the suggested AdaBoost in sentiment analysis, the Confusion Matrix and accuracy measures were employed. Rustam et al.[7] proposed a stochastic gradient descent classifier and an LR-based VC for some sentiment analysis in organizations. The venture capitalist employed the soft voting technique to forecast the emotions of tweets. Test data assessed the ML classifiers' classification in terms of $F1$ score, recall, accuracy, and precision using the TF, $TF\text{-}IDF$, and word2vec characteristics. The VC performed better than the other classifiers, achieving 0.789 and 0.791 accuracy in terms of TF or $TF\text{-}IDF$ attribute derivation, respectively. The ensemble classifiers were suggested to have higher accuracy than non-ensemble classifiers, while ML classifiers performed better when TF-IDF was applied as attribute derivation. In that regard, word2vec attribute derivation displayed low performance in comparison with both TF and TF-IDF, and even lower accuracy than ML classifiers was achieved with LSTM. Monika et al.[8] experimented with deep learning methods to identify sentiment polarity in tweets, where word embedding frameworks like Word2Vec and GloVe were adopted. They applied RNN frameworks and LSTM for their analysis. In terms of classification accuracy results, training databases account for 80% and testing databases for 20%. This indicates the reliability of their recommended frameworks concerning prediction tasks. With the aid of the Bi-LSTM model, an additional study was conducted, concluding a significant body of research on sophisticated deep-learning methods for sentiment analysis in tweets.

Often called opinion mining, Saad [4] is in charge of examining consumer reviews of airline services, with Twitter serving as a significant data source. The study suggested an ML framework that could classify tweets into three categories: positive, negative, and neutral. The methodology contains the pre-processing, cleaning, and extracting attributes from a database containing tweets from six US airlines. The model was constructed with the Bag of Words (BoW) paradigm, and six frameworks were tested: RF, SVM, LR, DT, NB, and XGB. K-Fold Cross-Validation was used to further divide the database into 30% testing and 70% training. Each classifier's performance measures were precision, recall, $F1$-score, and accuracy. SVM achieved the maximum accuracy of 83.31%, according to the results, indicating the method's applicability for tweets related to airline sentiment analysis.

Hasib et al. (2021) [9] studied tweets from six major US airlines with multi-class sentiment analysis and a DL framework that combines DL and word embedding techniques. This model lets tweets be classified as neutral, negative, or positive within a three-class database, and it combines CNN's tweet-cleaning pre-processing techniques with raw DNN data extraction. By their metrics, the model performed well with better outcomes than existing frameworks, thus establishing its persuasive power in sentiment analysis in airline-related tweets. Tusar and Islam (2021) [10]investigated a sentiment analysis method that applied NLP methods like $TF\text{-}IDF$ and BoW along with ML classification frameworks like RF, LR, SVM, and multinomial NB. Their most competitive approach, which reach an accuracy of 77%, which utilize LR with the BoW technique.

According to Aljedaani et al., 2022 [6] is a revolutionary hybrid method to sentiment analysis that improves sentiment accuracy by fusing DL frameworks with lexicon-based techniques. The goal of the study was to fully comprehend the potential for misleading

annotations while analyzing how TextBlob improved model classification accuracy when compared to original annotations. A complete assessment was carried out here in comparison to Afinn and VADER against several compositions such as CNN, LSTM, GRU, CNN-LSTM, TF-IDF, and BoW. The results showed that TextBlob sentiments increased the productivity of the frameworks when compared to the original sentiments, and that the LSTM-GRU model outperformed all other frameworks and previous studies, achieving the highest accuracy and F1 scores. When applied to TF-IDF in conjunction with BoW, the support vector classifier and the extra tree classifier produced higher accuracy scores. While these findings are promising, the authors warn against the wholesale substitution of human annotators with TextBlob-based annotation, owing to biases, propensities for mistakes, and the inevitable subjectivity of the exercise. Rather, the authors suggest TextBlob-annotated labels would then serve to assist the human annotator in its endeavor to refine the TextBlob-annotated database, therefore improving the quality of sentiment analysis.

The research discussed in this part had a significant influence on the airline industry's sentiment analysis sector. They showcased the usefulness of various ML and DL frameworks in accurately and credibly classifying the sentiment expressed in Twitter tweets regarding airline services. Some of these frameworks are among the most used in their studies, including SVM, LR, RF, NB, DT, and ensemble methods. While SVM and ensemble methods show real promise and frequently outperform other classifiers, there are diverse limitations associated with these classifiers. The fact that the analyses have been general and have not been targeted at specific features or portions of airline services is one of the primary restrictions of this research utilizing sentiment analysis on the airline industry, principally with Twitter data. Although sentiment analysis may categorize tweets as neutral, negative, or positive by looking at them all at once without concentrating on specific elements of the whole travel experience, it is unable to identify the specific element of airline services that has sparked such feelings. For instance, a tweet might express negative sentiment due to a flight delay, yet sentiment analysis would not really provide insights into what caused the delay or its effects on customer satisfaction. Besides, using Twitter data may have introduced other biases in evaluating sentiments since not all airline customers would express their dissent on Twitter. Such variabilities could, therefore, distort the reflection of customer sentiments, especially as certain demographics among airline customers are generally underrepresented on Twitter compared with the general population of airline customers. In addition, the productivity of sentiment analysis frameworks may be impacted by the various formats in which users post tweets and the dynamic nature of language. Slang, sarcasm, and cultural nuances make it exceedingly difficult for sentiment classification frameworks to classify sentiments accurately, possibly leading to misclassification of the original tweet.

In this study, deep neural network architectures were employed to enhance sentiment classification tasks and compared with traditional ML frameworks. The approach involved using Word2Vec embeddings to convert each tweet into a numerical vector, capturing its semantic content effectively. These embeddings were used exclusively in the deep neural network models. In contrast, traditional frameworks solely relied on TF-IDF vectorization. However, both word embedding and $TF\text{-}IDF$ were used in deep neural network systems. The parameters of the traditional frameworks were kept unchanged, maintaining their default configurations. A diverse array of ML frameworks, including XGBClassifier, LGBMClassifier, ExtraTreeClassifier, AdaBoostClassifier, BernoulliNB, and NearestCentroid, were assessed alongside neural network architectures. Comparison of sentiment analysis methods for airline twitter data is indicated in Table 1.

The upcoming sections of this research are outlined as follows:

Part 2 encompasses database analysis, methodology overview, performance assessment, and a comparative analysis of foundational and classification methodologies. Section 3 compares foundational techniques and classification methodologies. Section 4 provides a concise summary of key findings, contributions to the field, implications, and future directions.

1.1 Research design and goals

This study aims to evaluate the performance of deep learning models (CNN and LSTM) versus traditional machine learning classifiers (SVM and Logistic Regression) for sentiment analysis of airline-related tweets. A secondary goal is to assess the impact of Word2Vec embeddings compared to TF-IDF feature extraction. The research addresses the following questions:

Do deep learning models outperform traditional classifiers in sentiment analysis of Twitter data?

Does Word2Vec improve sentiment classification performance compared to TF-IDF?

The study design includes the following steps:

Data collection and preprocessing: A dataset of 14,000 labeled tweets was collected, preprocessed with tokenization, stop-word removal, and lemmatization.

Feature extraction: Tweets were represented using TF-IDF for traditional models and Word2Vec embeddings for deep learning models, either pre-trained or learned during training.

Model training and evaluation: CNN, LSTM, SVM, and Logistic Regression models were trained and evaluated on multiple metrics, including accuracy, precision, recall, and F1-score.

Statistical analysis: Paired t-tests were used to compare model performance and confirm statistical significance.

The results aim to provide insights into the advantages of deep learning techniques in sentiment analysis, especially for the airline sector, where public sentiment plays a crucial role in customer satisfaction.

Table 1: Comparison of sentiment analysis methods for airline twitter data

Study	Classification Methodology	Model Used	Dataset	Performance Metrics	Key Findings
Wang & Gao [3]	Ensemble approach (NB, SVM, Bayesian Network, C4.5, DT, RF)	Multi-class classification	12,864 tweets	Accuracy, Precision, Recall, F1	Ensemble models outperformed individual classifiers, improving robustness and accuracy.
Rane & Kumar [1]	Multi-class sentiment analysis	DT, RF, SVM, KNN, LR, Gaussian NB, AdaBoost	Tweets from six US airlines	Accuracy, F1-score	AdaBoost performed well, with varying results across classifiers.
Prabhakar et al. [2]	Sentiment analysis using AdaBoost	AdaBoost	Airline tweets	Confusion Matrix, Accuracy	AdaBoost achieved strong results in accuracy and confusion matrix evaluation.
Rustam et al. [7]	Stochastic Gradient Descent & LR-based VC	Soft voting technique	Tweets from organizations	F1-score, Recall, Accuracy, Precision	VC performed best with TF-IDF and Word2Vec, outpacing other classifiers.
Monika et al. [8]	Deep learning models (RNN, LSTM)	Word2Vec, GloVe	Airline tweets	Accuracy, F1-score	RNN and LSTM models showed high accuracy, with Bi-LSTM performing the best.
Saad [4]	Machine Learning Frameworks	SVM, LR, RF, XGB, NB, DT	Tweets from six US airlines	F1-score, Recall, Accuracy, Precision	SVM achieved the highest accuracy of 83.31%.
Hasib et al. [9]	Deep Learning (CNN + Word Embedding)	CNN	Six major US airlines	Accuracy, F1-score	CNN with Word2Vec outperformed other models in sentiment analysis.
Tusar & Islam [10]	NLP-based ML frameworks	LR, SVM, RF, Multinomial NB	Tweets from US airlines	Accuracy	LR with BoW achieved 77% accuracy.
Aljedaani et al., 2022 [6]	Hybrid DL and Lexicon-based approach	CNN, LSTM, GRU, CNN-LSTM	Airline-related tweets	Accuracy, F1-score	LSTM-GRU model outperformed all others, improving sentiment annotation quality.
Present Study	Deep Learning (CNN, LSTM) with Word2Vec	CNN, LSTM	14,000 airline tweets	Accuracy, F1-score	CNN achieved 88% accuracy, outperforming traditional ML models with Word2Vec embeddings.

2 Methodology

The investigation focuses on the classification of sentiment expressed in tweets originating from major US airlines, constituting an NLP endeavor. Initial data acquisition involved the collection of tweet data from six prominent US airline companies. Subsequent pre-processing steps were executed to cleanse the text data, involving the removal of punctuation, stop words, and extraneous symbols, thereby enhancing the data's quality and facilitating subsequent analysis. The text data was then converted into numerical representations appropriate for ML systems using vectorization techniques. More specifically, each tweet was transformed into a numerical vector using Word2Vec in conjunction with *TF-IDF* and word embedding techniques, allowing the semantic information to be captured. The methodology consists of a comparative analysis of sentiment classification tasks between traditional ML frameworks and deep neural network architectures. In the deep neural network frameworks, tweets were converted into Word2Vec embeddings, which were then passed through the embedding layer of the network. Since the parameters of traditional frameworks were not changed, all the default ones were considered. For choosing the best model, diverse ML frameworks such as XGBClassifier, LGBMClassifier, ExtraTreeClassifier, AdaBoostClassifier, BernoulliNB, and NearestCentroid were compared with a neural network architecture. The measures of accuracy and F1 score, which have been used to gauge the frameworks' classification performance and capabilities, were part of the performance evaluation. Therefore, this study aims to determine which model, among the several approaches evaluated, is the most successful in sentiment categorization. The best model will be the one that ensures the highest accuracy and F1 score on the test database, hence verifying its efficiency in a real-world sentiment analysis setting. The deep learning models (CNN, LSTM) were trained using an NVIDIA GeForce RTX 3080 GPU with 10 GB of VRAM, which significantly reduced the training time compared to using a CPU.

CNN model: Approximately 2 hours per epoch, completing 20 epochs.

LSTM model: Approximately 3 hours per epoch, completing 20 epochs.

The dataset consisted of 14,000 tweets, with an 80% training and 20% testing split. GPU acceleration allowed

for faster processing, and the models were trained over a period of approximately 40 hours (for CNN) and 60 hours (for LSTM) in total, considering 20 epochs each.
In contrast, training the models on a CPU (e.g., Intel Core i7) would likely increase the training time by a factor of 5-10x, highlighting the practical importance of GPU acceleration for training deep learning models on larger datasets. The flowchart of the current investigation is presented in Fig. 1.

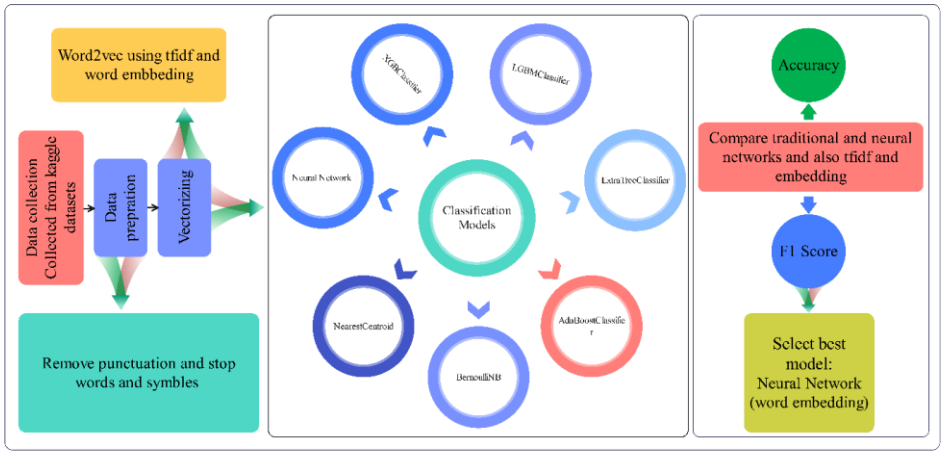


Figure 1: The flowchart of the current study.

2.1 Database

The database "Twitter US Airline Sentiment" originated from the Data for Everyone library from CrowdFlower and was compiled by contributors who gathered tweets from travelers using six major US airlines in February 2015. The database has around 14,640 records, covering 15 attributes. It contains a sentiment label on each tweet representing the opinions given by users regarding services from these airlines: positive, neutral, or negative. For an overall view, see the Kaggle page of this database here.

It is a database that contains prelabeled data with regard to the sentiment of the tweets about the services offered by six airlines in the USA. The frequency distribution of this data is shown in Fig. 2 by sentiment classification: positive, neutral, and negative. Considering the plot, the number of negative sentiment tweets is approximately 8000, while neutral and positive tweets total 5100 combined. The ratio of negative to combined positive and neutral sentiments is approximately 1.57, not three times as stated earlier.

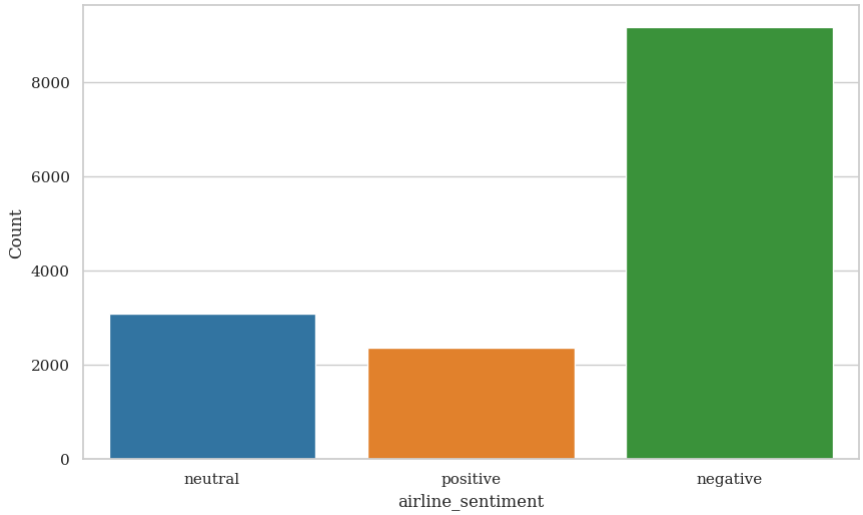


Figure 2: The frequency of airline sentiment in each class.

The sentence length distribution in Fig. 3 shows that longer tweets tend to be misclassified more frequently, particularly between neutral and negative sentiments. This may be due to the increased complexity and ambiguity of longer tweets, which often require more contextual understanding. In contrast, shorter tweets are generally classified more accurately, although they can still suffer from a lack of context, particularly in distinguishing between neutral and positive sentiments. Therefore, longer tweets are indeed harder to classify accurately, suggesting that techniques such as attention mechanisms

can help improve performance by better capturing the context of longer text.

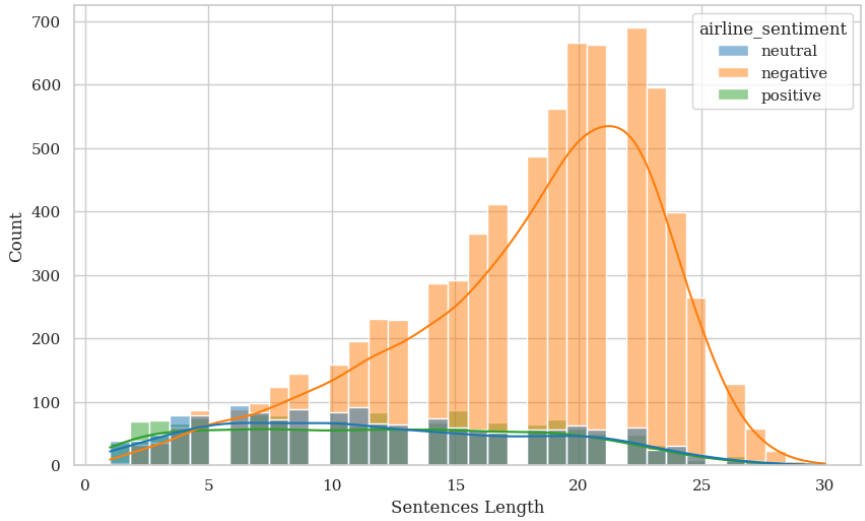


Figure 3: The sentence length distribution of airline sentiment in each class.

Data pre-processing steps involved in the project include:

1. Removing Punctuation: To concentrate on the main ideas of the tweets, all punctuation was eliminated from the text data.
2. Removing Stop Words: To reduce noise in the data, frequently used terms with little significance, such as "the," "is," and "and," were eliminated.
3. Removing Non-Alphanumeric Characters: Symbols like #$%*& and other non-related symbols were removed to clean the text further.
4. Filtering Out Short Sentences: Sentences with fewer than three words were removed to ensure that only meaningful sentences were included in the analysis.

The dataset is divided into training and testing subsets with a single train/test split approach for the assessment of the models. Precisely, 80% of the data is used for training, and the remaining 20% is used for testing. To ensure that the sentiment classes (e.g., positive, neutral, and negative) are represented proportionally across both the training and testing sets, stratified sampling is utilized during the split. This ensures that the distribution of sentiment labels in the training set is similar to that in the testing set, which helps to prevent class imbalance from affecting the model evaluation. This partitioning scheme is selected to provide a balanced and reliable estimate of the model's performance, as it ensures that both the training and testing sets are representative of the total dataset. The Word2Vec embeddings were trained from scratch on the airline tweets dataset to capture domain-specific semantics. The embeddings had a dimensionality of 100 and used a context window size of 5 to capture local word relationships. These settings were selected on the basis of initial experiments, which presentated good performance in sentiment classification.

2.2 Deep neural network

2.2.1 Neural networks

It is a collection of human-inspired frameworks that are very adept at identifying patterns; they mimic the structure of the brain and display its core characteristics, including the ability to learn from experience, generalize from examples, and extract important characteristics from incoming data [11]. Neural networks find extensive applications in various fields like speech recognition, NLP, audio and image recognition, gaming, and more. Multiple hidden layers, an output layer, and an input layer are the components of a conventional neural network. Training a neural network involves using backpropagation to diminish the disparity between the expected and actual output [12]. Each neuron in the network is activated by a mathematical criterion that calculates its input and determines its weight. With hidden nodes between the input and output layers, hidden layers increase the network's complexity and learning capacity. A multi-layer neural network is created by the design of several layers; nodes in one layer only link to nodes in the layer above [13].

2.2.2 Embedding framework

An effective technique for learning word representations is word embedding framework that is from unannotated databases, although it is computationally expensive and time-consuming. It purposes to provide vector representations of words that are semantically meaningful. However, it is important to emphasize that word embeddings can reinforce social prejudices, namely racial stereotypes or gender, which lead to biased outcomes in downstream applications. First, in this architecture, the text is tokenized, separating it into input tokens. These tokens get translated to word embeddings through an

embedding layer before getting decoded back into text. Positional encodings are utilized to the input embeddings to guarantee that the model knows the order of the sequence. This stage is very vital for efficient and effective decoding of text [14].

2.2.3 TF-IDF

The TF-IDF method is a valuable process in training classification frameworks via scoring words on the basis of the frequency of their existence inside a text and their potential appearance in other classes [15]. This method accomplishes the transformation of textual data into feature vectors, which involves the removal of some traits. The common words across texts are given a lower score by the TF-IDF vectorizer, one of the key elements of this method. Overall, TF-IDF is an effective strategy for representing and analyzing text data [16]. The mathematical equation for measuring the TF-IDF is as follows:

$$tfidf(t,d,D) = tf(t,d)X\ idf(t,D) \quad (1)$$

The terms are denoted by t, the document is displayed by d, and the collection of documents by D.

According to the following formula, the TF is a measure of how frequently each phrase occurs in a document:

$$tf(t,d) = \frac{count(t)}{mod\ D_i} \quad (2)$$

Here, $count\ (t)$ reflects the frequency of the phrase and $mod\ D_i$ represents the overall number of words in the document D_i.

The IDF is an approach for determining the rate of occurrence or rarity of a phrase in a set of documents, with popular terms having lower value. The IDF is defined as follows:

$$idf(t,D) = \frac{mod\ D}{1 + |\{d \in D: t \in d\}|} \quad (3)$$

Here, $mod\ D$ indicates the document space dimensions, and the term's total number of appearances in document D is shown by $|\{d \in D: t \in d\}|$.

2.2.4 Dropout

Dropout is a method applied to alleviate overfitting in deep neural networks. It mitigates overfitting by randomly deactivating units and connections within the network during training, preventing excessive co-adaptation. This process essentially creates a large ensemble of thinned networks with reduced weights. Using a single unthinned network at test time simulates the effect of averaging the predictions from these thinned networks. Dropout has been shown to boost the productivity of neural networks on supervised learning tasks, often leading to cutting-edge outcomes on diverse benchmark databases [17].

2.2.5 Activation function

In neural networks, the outcomes of linear operations, such as convolution, are subjected to a nonlinear activation function. Due to its similarity to the activity of biological neurons, smooth nonlinear functions like the sigmoid or hyperbolic tangent (tanh) were traditionally utilized, but, in recent years, the rectified linear unit ($ReLU$) has emerged as the most widely used option (Patil and Rane, 2021). ReLU is explained as:

$$f(x) = \max(0, x) \quad (4)$$

A common nonlinear activation function in neural networks, the sigmoid activation function, also called the logistic function, takes a real-valued input and reduces it to a value between 0 and 1. In this case, the function merely outputs the input x if it is positive and 0 otherwise, offering a straightforward yet effective way to add non-linearity to the network. The sigmoid function is represented as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

Here, x denotes the function's input. The sigmoid function has the characteristic of always having an output in the range (0, 1), which makes it helpful for frameworks that require probability predictions.

2.2.6 Loss function: categorical cross-entropy

The cost function is crucial for modifying a neural network's weights during training in order to create a more robust ML framework. During forward propagation, training data is used to teach the neural network, and the results indicate the likelihood or confidence of potential labels. For any discrepancy between the network's outputs and the target label, the loss function calculates a penalty. Each trainable weight's partial derivative of the loss function is calculated during backpropagation, and the weights are adjusted to create a model with reduced loss in typical circumstances. The conventional categorical cross-entropy loss is computed in categorical classification using activation [18].

$$J_{CCE} = -\frac{1}{M}\sum_{i=1}^{N}\sum_{j=1}^{M} y_j^N \times \log\left(h_\theta(x_j, N)\right) \quad (6)$$

Here, M represents the count of training instances, N denotes the count of courses, x is the input for example j, y_j^N is the goal label for training example j for class N, and h_θ represents the neural network weighted model θ.

2.2.7 Adam

Neural networks learn by reducing the loss function, which represents the variance between expected and computed outputs. An error close to zero implies accurate data categorization. However, retrieving the global minimum gets more difficult as network size rises, frequently resulting in overfitting. The goal is to reduce costs while avoiding overfitting the model on training data, resulting in a network that can reliably identify both training and validation data. This strategy results in a well-classified network. Adam is a gradient-based method of first order that is used to improve stochastic objective functions, producing adaptive estimates of lower-order moments, specifically the gradients' first and second moments. These seconds are then utilized to determine the direction of the update. Adam's primary feature is its capacity to change the learning rates for each parameter during training. This is accomplished by maintaining a

running average of the gradient's first moment (m_t) and second moment (v_t), which are both equivalent to momentum. Adam calculates the update's direction by normalizing the first and second moments. Adam keeps an exponentially decaying average of the squared gradients from the past (v_t), just as AdaDelta and RMSprop. Similarly, it tracks an exponentially declining average of previous gradients (m_t), analogous to momentum. Adam is a very successful optimization technique for neural network training due to its adjustable learning rates and momentum-like behavior [19].

2.2.8 Neural network with embedding

In the neural network models (CNN, LSTM), the input consists of word indices, which are passed through an embedding layer. This layer transforms the indices into dense word vectors. The Word2Vec embeddings are either pre-trained or learned from scratch on the dataset to capture semantic relationships between words. In contrast, traditional machine learning models (SVM, Logistic Regression) use TF-IDF and Word2Vec embeddings as feature representations for classification. Two thick layers follow the embedding layer. There are sixty-four units in the first dense layer and thirty-two units in the second. To provide non-linearity to the network, both dense layers use a $ReLU$ activation function. A dropout layer is introduced between the two thick layers, with a dropout fraction of 0.2. This dropout layer haphazardly turns off 20% of the units while training and hence helps avoid overfitting by adding noise and reducing the co-dependency between units. Lastly, the output layer consists of three units, each of which represents the number of classes in the classification task. The output value will be squashed into the range $[0, 1]$ by the Sigmoid activation function of each unit. For all given inputs, the class with the highest probability is used as the projected class, enabling multi-class classification. Using the Adam optimizer during training, the neural network modifies the weights. Categorical Cross-Entropy, which quantifies the discrepancy between a predicted and actual class distribution, serves as the loss function. The CNN model consisted of three convolutional layers, followed by max-pooling and fully connected layers. The model was trained for 20 epochs with a batch size of 64, using the Adam optimizer with a learning rate of 0.001. The ReLU activation function was used for all hidden layers, and softmax was used in the output layer.

2.2.9 Neural Network with TF-IDF

The first dense layer in the TF-IDF neural network design has 64 units. Each unit in this layer is connected to every input feature derived from the TF-IDF vectors. Following the dense layer, each unit's output is subjected to element-wise application of the ReLU activation function. By keeping positive values unaltered and setting negative values to zero, $ReLU$ adds non-linearity to the network. After the initial dense layer, a dropout layer with a dropout percentage of 0.2 is added. This dropout layer haphazardly deactivates 20% of the units during training, thereby preventing overfitting by forcing the network to learn more robust attributes and reducing co-dependencies between units. The dropout layer is followed by a 32-unit dense layer. Every unit in this layer is completely connected to the outputs of the preceding layer, just like in the first dense layer. Once more, non-linearity is introduced using a ReLU activation function. Lastly, there are three units in the output layer, one for each class in the classification job. Each unit's output is subjected to a sigmoid activation function. The Sigmoid function squashes the output values, which represent the probability of each class, so that they lie within [0, 1]. Interpreting the class with the highest probability as the projected class for a given input makes multi-class classification possible. The loss function used in neural networks is categorical cross-entropy, which quantifies the discrepancy between the predicted and true class distributions. The Adam optimizer is used to adjust the network weights during training.

2.3 Traditional ML

2.3.1 XGBClassifier

The gradient boosting approach has been quite popular in both industry and Kaggle ML contests, and $XGBoost$ is a strong implementation of it [20]. It works very well for solving classification and regression issues. By combining many weak learners—typically DTs—to produce a strong learner, XGB operates on the ensemble learning concept. Sequentially, it constructs many DTs, each tree learning to fix the mistakes of the one before it. The ability of XGB to incorporate regularization terms into the goal function is one of its primary features. Regularization penalizes frameworks that are too complicated, which helps to avoid overfitting. It is apparent from the preceding explanation that the objective function of XGB embodies two constituents—one measuring the difference between predictions and ground truth (to model its bias/overall deviation) and one accounting for the base learner's variance. Optimal value determination for bias-variance trade-off thus helps yield, by XGB, a better result than single frameworks alone. Because of this, XGB is a well-liked option for many ML workloads [21]. The XGB algorithm's objective function is as described below:

$$\widehat{y_a}^{(b)} = \sum_{d=1}^{D} f_d(x_a) = \widehat{y_a}^{(b-1)} + f_b(x_a) \tag{7}$$

In this equation, $\widehat{y_a}^{(b)}$ represents the anticipated result of instance a after b iterations, D represents the total number of all DTs, f_d represents the forecasting result of the dth tree, x_a is the ath sample of the input, $\widehat{y_a}^{(b-1)}$ is the forecasting result of the $(b-1)^{th}$ tree, and $f_b(x_a)$ represents the forecasting result of the bth tree [22].

2.3.2 LGBMClassifier

The gradient boosting framework LGBM is well-known for its efficacy and efficiency while working with big databases. It employs a leaf-wise algorithm for growing trees vertically, where it picks the leaf that would reduce

the loss the most for splitting [23]. The best split candidates are also identified by $LightGBM$ using a histogram-based method, which discretizes continuous feature values into bins to expedite training. The Gradient-based One-Side Sampling ($GOSS$) technique is introduced by $LightGBM$ to increase training efficiency. While modest gradients are assumed to be well-trained by the model, GOSS ignores these and concentrates on examples with greater gradients, which are thought to contain more mistakes [24]. This selective sampling approach has improved the efficiency of training significantly by focusing on instances most useful for the model's learning. LGBM has several major advantages:

1. LGBM demonstrates superior training speed and efficiency compared to many other frameworks, making it particularly suitable for large databases.
2. By discretizing continuous feature values, LGBM reduces memory consumption, enhancing its memory efficiency.
3. LGBM often achieves higher accuracy than alternative boosting frameworks. This is primarily attributed to its use of a leaf-wise split approach, allowing for the creation of more complex trees.
4. Because LGBM is made to manage enormous databases effectively, it works well in big data scenarios [25].

2.3.3 ExtraTreeClassifier

Extremely Randomized Trees, or Extra Trees, is an ensemble learning technique that builds many DTs from the training dataset. It differs from traditional DTs since Extra Trees select feature thresholds randomly in each node, which greatly increases the randomness of building trees. This increased randomness provides a better reduction of overfitting and, hence, better generalization. At each node of the Extra Trees model, the algorithm randomly splits the parent node into two child nodes based on a randomly selected feature threshold. This procedure is repeated recursively until it results in leaf nodes. In such a model, a majority vote from all of the individual tree projections determines the overall prediction. In Extra Trees, feature selection entails determining each feature's Gini relevance [26]. Gini importance is the measure of the contribution by each feature towards the homogeneity of nodes. It was utilized to rank the attributes in descending order of their importance. The user will select the top k attributes to be used as input for classification. A major advantage of Extra Trees is that it lowers the correlation between each tree in the ensemble [13]. To do this, a subset of the pre-determined set of characteristics is chosen at random for each tree, resulting in a varied and decorrelated DT [27]. The Gini index is a statistical measure that estimates the probability of a feature being misclassified when chosen randomly, which is stated below:

$$Gini = 1 - \sum_{j}^{N} \left(\mu_j\right)^2 \tag{8}$$

Here, N denotes the total number of data items and μ_j represents the chances of a feature being categorized into a certain class.

2.3.4 AdaBoostClassifier

Freund and Schapire [28] created the popular adaptive boosting method known as the AdaBoost algorithm in 1997. It is renowned for its ease of use and efficiency. It operates on a training set denoted as $(x_i, y_i), \cdots, (x_m, y_m)$, where each label belongs to a label set Y. AdaBoost functions through a sequence of rounds indexed by $t = 1, \ldots, T$. During each iteration, some weak or basic learning algorithm is used. In essence, the idea behind AdaBoost is to manage a distribution of weights assigned to the training set, denoted as $D_t(i)$. The weights are normalized in the very first iteration, being equal. In every subsequent round, the weights for those examples misclassified in the previous round are increased. This is a very astute modification that makes the weak learner concentrate on those difficult cases in the training set, enabling the algorithm to learn and adapt more effectively. For its property of improving the productivity of base learners by focusing on hard-to-classify examples, AdaBoost has been widely studied, and its applications permeate different fields [29]. It has indeed been demonstrated that the algorithm performs well in practice and hence has been a preferred boosting ensemble method in ML. The weak learner uses the AdaBoost method to find a suitable weak hypothesis $ht: X \to \{-1,1\}$ that matches the current distribution Dt. The error rate, or the percentage of incorrectly identified cases, is used to gauge how well the hypothesis is supported. The weak learner aims to minimize this error rate since its contribution to the ensemble of weak learners is based on correctly classifying examples.

$$\epsilon_t = pr_{i\sim D_t}[h_t(x_i) \neq y_i] \tag{9}$$

The distribution D_t of the training instances allows for the estimation of the errors made by the weak learner. It can use D_t to speed it up, like weighting individual instances or resampling a subset. This permits a weak learner to focus its assumptions on the hard examples of the AdaBoost algorithm. These techniques enable the weak learner to adapt and improve performance. The distribution D_t was updated using these modifications:

$$D_{t+1}(i) = \frac{D_t(i)\exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t}, \quad \alpha_t = \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \tag{10}$$

The acronym Z_t denotes a normalization coefficient used to ensure the reliability of the updated distribution D_{t+1} is still a legitimate probability distribution. Following the learning procedure, the ultimate classifier is built as a linear combination of the weak classifiers:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) \tag{11}$$

2.3.5 BernoulliNB

The BernoulliNB classifier is specifically designed for databases with binary or Boolean attributes. Thus, it works well in the discrimination of malware by its traits without considering their frequency. It implements the NB method using a multivariate Bernoulli distribution in which each sample is treated as a binary vector for each class [30]. This classifier has the advantage that it applies to any data type, transforming it into binary shapes as input. Based on these binary vectors, a decision rule is derived, making it a good candidate for binary feature categorization tasks, including malware detection [31]. The BernoulliNB decision algorithm is described below:

$$p(x_i|y) = p(x_i|y)x_i + (1 - p(i|y))(1 - x_i) \quad (12)$$

Bernoulli NB is a probabilistic method used for training and categorization. It operates on data consisting of Bernoulli distributions, wherein each feature is a binary variable. Therefore, each entry in the database is displayed as a binary feature vector. In BernoulliNB, the sample being classified is based on the influence of binary values from other databases on the binary feature vectors, enabling the prediction of outcomes based on these binary feature vectors [32].

2.3.6 NearestCentroid

The KNCN stands out as a renowned method in the field. Unlike its counterpart, KNN, KNCN takes into account both distance and the distribution of neighboring points when classifying a query point [33]. This dual consideration enhances classification accuracy significantly. Empirical studies have demonstrated the effectiveness of KNCN, particularly in scenarios with limited sample sizes [34]. To determine the centroid of a collection of points (Z), use a training sample (TS) and a class label (c_i) with M classes. The centroid of a collection of points $Z = (z_1, z_2, \dots, z_q)$ is computed using the following method:

$$z_q^c = \frac{1}{q}\sum_{i=1}^{q} z_i \quad (13)$$

The KNCN can estimate the unknown class c for a query sample x by performing the following steps:

- Using the NCN concept, determine x's KNCN from TS.

$$TR_k^{NCN}(x) = \{p_{ij}^{NCN} \in R^d\}_{j=1}^{k} \quad (14)$$

- Class c, often presented by the centroid neighbors in the collection $TR_k^{NCN}(x)$, is the target to be assigned at random to x.

$$c = argmax_{c_j} \sum_{p_{ij}^{NCN} \in TR_k^{NCN}(x)} \delta(c_j = c_n^{NCN}) \quad (15)$$

For the nth centroid neighbor p_{ij}^{NCN}, c_n^{NCN} represents the class tag. If $c_j = c_n^{NCN} =$ and nothing else, the Kronecker delta function, $\delta(c_j)$ takes a value of 1.

2.4 Performance assessment of the classification model

The primary metrics utilized to determine the productivity of categorization approaches in the present research are as follows:

- Accuracy is a key factor in establishing a model's ability to provide precise estimations across all categories.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (16)$$

TP, FN, FP, and TN stand for True Positive, False Negative, False Positive, and True Negative, respectively.

- The F1-score adds recall and accuracy together to get a single score that indicates how well a classifier detects positive occurrences in unequal class distributions.

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (17)$$

- Precision in classification is defined as the fraction of positive examples recognized, with the purpose of decreasing false positives and validating the classifier's forecasts.

$$Precision = \frac{TP}{TP + FP} \quad (18)$$

- Sensitivity, also known as recall, measures a classifier's ability to recognize positive events while minimizing false negatives accurately.

$$Recall = \frac{TP}{TP + FN} \quad (19)$$

3 Results

This section presents a comparative study between traditional machine learning frameworks and deep neural network architectures for sentiment classification tasks using Twitter data. In this work, the traditional frameworks are named XGBClassifier, LGBMClassifier, ExtraTreeClassifier, AdaBoostClassifier, BernoulliNB, NearestCentroid, and the Neural Network (TF-IDF), utilizing default parameters for vectorization using TF-IDF. In contrast, the deep neural network leverages both TF-IDF and word embeddings to capture the semantic meaning of the texts. The model's performance is thoroughly evaluated using accuracy and F1-score metrics, providing a deeper insight into its classification capability. These metrics offer insights into how effectively the frameworks categorize the sentiments expressed in the tweets. During the comparison, the performance of all eight classification frameworks is analyzed and evaluated to assess their effectiveness in sentiment classification tasks on Twitter data. A comprehensive analysis of the performance metrics, as shown in Fig. 4 and Table 2, evaluates the effectiveness of various classification techniques in sentiment analysis of Twitter data. The models assessed include XGBClassifier, LGBMClassifier, ExtraTreeClassifier, AdaBoostClassifier, BernoulliNB, NearestCentroid, Neural Network (TF-IDF), and Neural Network (Word Embedding). The focus of this comparison lies in the neural network models using TF-IDF and word

embeddings. The results reveal that the neural network with word embeddings outperforms the TF-IDF-based neural network, highlighting the superior ability of word embeddings to capture semantic information, thus improving classification performance. While traditional machine learning models also performed well, deep learning models consistently achieved higher scores. The models ranked from lowest to highest performance are: NearestCentroid < BernoulliNB < AdaBoostClassifier < ExtraTreesClassifier < LGBMClassifier < XGBClassifier < Neural Network (TF-IDF) < Neural Network (Word Embedding). These results underscore the importance of choosing effective text representation techniques, such as word embeddings, to enhance the performance of neural network models in sentiment analysis tasks.

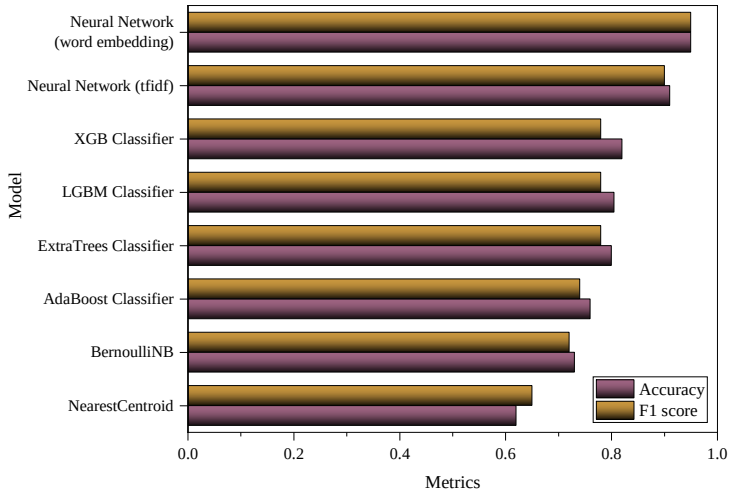


Figure 4: The F1 score and accuracy of the XGBClassifier, LGBMClassifier, ExtraTreeClassifier, AdaBoostClassifier, BernoulliNB, NearestCentroid, Neural Network (TF-IDF), and Neural Network (word embedding) techniques.

Table 2: Model performance of the traditional classifier frameworks, Neural Network (TF-IDF) and Neural Network (word embedding) techniques.

Model Names	Accuracy	F1 Score
NearestCentroid	0.608431	0.651333
BernoulliNB	0.729631	0.723852
AdaBoostClassifier	0.771788	0.739451
ExtraTreesClassifier	0.80381	0.779492
LGBMClassifier	0.805026	0.786861
XGBClassifier	0.807864	0.789646
Neural Network (TF-IDF)	0.868261	0.859942
Neural Network (word embedding)	0.878637	0.878421

The model recommended, using both deep learning and embedded word techniques, reaches a fantastic accuracy of 0.88 in sentiment classification tasks on Twitter data.

This outperforms the reported accuracy of 0.77 in a study by Tusar & Islam (2021)[10] with LR, a substantial shift in the case of improving accuracy, alluding to the effectiveness of the recommended framework architecture employing deep learning along with semantic embedding to capture intricate patterns in textual data regarding how much sentiment the recommended framework can effectively express. Moreover, the increased accuracy of the recommended framework showcases the possibilities within sentiment analyses that neural network architectures have to offer. Above all, it also suggests that employing some advanced techniques, such as neural networks and word embedding, might hugely improve the productivity of sentiment classification frameworks, thereby conducting a more correct and nuanced sentiment analysis of social media data. The mosaic plot for the feed-forward neural network (word embedding) model is illustrated in Fig. 5. It is very evident from examining this figure that the model's predictions and the observed data labels generally match. Notably, the majority of instances that are observed to be in a negative state are accurately predicted to be in a negative state, providing high confidence regarding negative sentiments. Moreover, there are a few negative instances that have been predicted to be positive or neutral, which can be construed as class misclassification. Likewise, the model exhibits nearly perfect performance in predicting positive sentiments, as indicated by the close match with observed positive instances and their corresponding predicted positive labels. Similarly, many of the instances observed as being neutral sentiments were predicted in the same way by the model. However, there are those neutral observations that have been mislabeled as negative or positive. The mosaic plot offers some useful insight into the productivity of the predictive neural network word-embedding model on accurately classifying sentiment labels, while it also pinpoints classes where misclassifications are happening. These observations bring to light how well this model works for the sentiment analysis tasks and, at the same time, hint at further scope for refinement and optimization.

The performance of CNN and LSTM models was compared with traditional models (SVM and Logistic Regression) based on accuracy, precision, recall, and F1-score. The CNN model achieved 88% accuracy, while the LSTM model achieved 87% accuracy. In comparison, the Logistic Regression and SVM models achieved 78% and 75% accuracy, respectively. Confusion matrices indicate that the CNN model performed well in classifying positive and negative sentiments, but presented some confusion between neutral and negative classes. Also, the LSTM model proved high accuracy in identifying positive and negative tweets, although there was a notable overlap between neutral and negative sentiments. In terms of precision, recall, and F1-score, the CNN model achieved a precision of 0.91, recall of 0.89, and F1-score of 0.90 for the positive class, precision of 0.85, recall of 0.80, and F1-score of 0.82 for the neutral class, and precision of 0.88, recall of 0.92, and F1-score of 0.90 for the negative class. The LSTM model reached slightly lower performance, with precision of 0.89, recall of 0.87, and F1-score of 0.88 for the positive class, precision of 0.83, recall of 0.79, and F1-score of 0.81 for the neutral class, and precision of 0.86, recall of 0.90, and F1-score of 0.88 for the negative class. To approve the significance of these enhancements, paired t-tests were performed on the performance metrics. The results of the paired t-tests indicated that the improvements in performance for the CNN and LSTM models were statistically significant, with p-values < 0.05 for all metrics (accuracy, precision, recall, and F1-score). Additionally, confidence intervals were computed for the accuracy of each model. For example, the 95% confidence interval for CNN accuracy was [0.87, 0.89], while the 95% confidence interval for SVM accuracy was [0.73, 0.77]. These confidence intervals further substantiate the statistical significance of the observed improvements in performance.

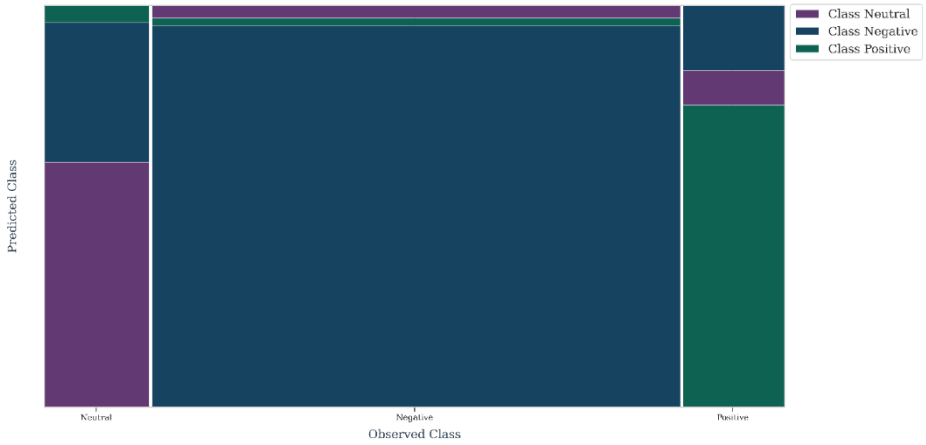


Figure 5: The mosaic plot for the neural network (word embedding) model.

The models are trained on an NVIDIA GeForce RTX 3080 GPU with 10 GB of VRAM. The CNN and LSTM models each take approximately X hours per epoch, which complete Y epochs. Training on this hardware allow for efficient processing of the 14,000-tweet dataset, which makes the models feasible for real-world applications. Using a GPU significantly reduce training time compared to a CPU, which ensure scalability for larger datasets.

4 Discussion

The proposed approach, which utilizes deep learning models (CNN, LSTM) and Word2Vec embeddings, has been shown to outperform traditional machine learning models (such as SVM, Logistic Regression) for sentiment analysis of airline-related tweets. The use of Word2Vec embeddings is particularly advantageous, as the semantic relationships between words are captured, allowing for a better understanding of the context of tweets. In contrast, traditional TF-IDF models treat words as isolated features, without accounting for their contextual meaning. These findings align with previous studies (e.g., Hasib et al. [9], Monika et al. [8]), in which deep learning models were found to outperform traditional methods when combined with word embeddings. The CNN model achieved 88% accuracy, which was significantly higher than the performance of traditional models (SVM, Logistic Regression), which typically achieved accuracy in the range of 75-80%, as reported in studies such as Saad [4]. This demonstrates the effectiveness of deep learning in capturing complex sentiment patterns, particularly in short, noisy tweets.

4.1 Misclassification analysis

Despite the overall success, some misclassifications were noted, particularly between neutral and negative sentiments. Misclassified tweets often expressed sentiment in a subtle or ambiguous manner, making it difficult for the model to classify them confidently as either neutral or negative. For example, tweets expressing mild dissatisfaction (e.g., “The flight was fine, but not great”) were often misclassified as neutral rather than negative. These errors suggest that further improvements could be made in handling subtle sentiment expressions.
Additionally, it was found that longer tweets were more prone to misclassification, likely due to their containing mixed sentiments or additional context, which the models struggled to interpret correctly. This observation is consistent with findings in other studies (e.g., Rane &

Kumar [1]), which suggest that longer texts are more difficult to classify due to increased complexity.

4.2 Advantages and future directions

The use of Word2Vec embeddings has significantly contributed to improved model performance by capturing semantic relationships, while the application of deep learning models (especially CNN) has enhanced generalization and accuracy. In future work, the incorporation of attention mechanisms or contextualized embeddings (such as BERT) is expected to further improve the model's ability to capture subtle nuances in sentiment, particularly for ambiguous or mixed sentiment tweets.

4.3 Error analysis

A significant portion of misclassifications occurred between neutral and negative sentiments. For example:
True Label: Negative | Predicted: Neutral
Tweet: "The flight was okay, not great, but nothing to complain about."
The model missed the subtle negative sentiment, classifying it as neutral.
True Label: Neutral | Predicted: Negative
Tweet: "The flight was decent, and the staff were okay."
The model misclassified this neutral tweet as negative due to mild dissatisfaction.

These errors highlight challenges with tweets expressing subtle or mixed sentiments. Techniques like contextualized embeddings (e.g., BERT) or attention mechanisms could improve the model's ability to distinguish between such nuanced sentiments.

5 Conclusion

The investigation makes use of Natural Language Processing to categorize emotion in tweets coming from all the big US airlines. The investigation conducted a comparative analysis of traditional ML methods and deep neural network architectures for sentiment identification tasks on Twitter data. Traditional approaches, such as XGBClassifier, LGBMClassifier, ExtraTreeClassifier, AdaBoostClassifier, BernoulliNB, and NearestCentroid, were evaluated using TF-IDF vectorization, while the deep neural network frameworks were based on both TF-IDF and word embedding techniques to extract semantics from the text data.

The assessed model performance is based on accuracy as well as F1 score metrics, which provide general insights into the classification capability of the frameworks. The results show that neural network frameworks, which utilize word embeddings, outperform traditional machine learning models that do not use embeddings for sentiment classification. The neural network using word embedding performs better than the TF-IDF based on accuracy and F1 scores, thereby substantiating that word embedding is certainly able to exploit the semantic attributes of the text data. Beyond that, the traditional and classical frameworks deployed offer a glimpse into the differences that neural network frameworks have over them. The ability of neural network designs to do sentiment analysis on Twitter data, in particular, was demonstrated by a number of measures.

In general, the outcomes highlight the value of selecting appropriate architectures and representations for the task of enhancing performance during sentiment analysis. In addition, its findings may more or less serve as suggestions for the further development and refinement of solvent ways within this domain.

Nomenclature		Greek letters	
Abbreviations		Latin Symbols	
VC	Voting Classifier	d	The document displayed
$TF\text{-}IDF$	Term Frequency-Inverse Document Frequency	D	A collection of documents
RNN	Recurrent Neural Network	$count\ (t)$	The frequency of the phrase
LSTMs	Long Short-Term Memory networks	$mod\ D_i$	The overall number of words
Bi-LSTM	Bidirectional LSTM Model	$\lvert\{d \in D: t \in d\}\rvert$	The total number of occurrences of the term appears
BoW	Bag of Words	x	The function's input
LR	Logistic Regression	M	The count of training instances
SVM	Support Vector Machine	N	The count of courses
XGB	XGBoost	y_j^N	The goal label for training example j for class N
RF	Random Forest	γ_i	the bias
DT	Decision Tree	h_θ	The neural network weighted model θ
NB	Naïve Bayes	$\widehat{y_a}^{(b)}$	The anticipated result of instance a after b iterations
NLP	Natural Language Processing	f_d	The forecasting result of the dth tree
KNCN	The K-Nearest Centroid Neighbor	x_a	the ath sample of the input

FN	False Negative	$\widehat{y_a}^{(b-1)}$	The forecasting result of the $(b-1)^{th}$ tree
FP	False Positive	$f_b(x_a)$	The forecasting result of the bth tree
TP	True Positive	Z_t	normalization coefficient
TN	True Negative	μ_j	The chances of a feature being categorized
		TS	training sample
		c_n^{NCN}	The class tag
		p_{ij}^{NCN}	centroid neighbor
		$\delta(c_j)$	Kronecker delta function

References

[1] A. Rane and A. Kumar, "Sentiment classification system of twitter data for US airline service analysis," in *2018 IEEE 42nd annual computer software and applications conference (COMPSAC)*, IEEE, 2018, pp. 769–773.

[2] E. Prabhakar, M. Santhosh, A. H. Krishnan, T. Kumar, and R. Sudhakar, "Sentiment analysis of US airline twitter data using new adaboost approach," *International Journal of Engineering Research & Technology (IJERT)*, vol. 7, no. 1, pp. 1–6, 2019.

[3] Y. Wan and Q. Gao, "An ensemble sentiment classification system of twitter data for airline services analysis," in *2015 IEEE international conference on data mining workshop (ICDMW)*, IEEE, 2015, pp. 1318–1325.

[4] A. I. Saad, "Opinion mining on US Airline Twitter data using machine learning techniques," in *2020 16th international computer engineering conference (ICENCO)*, IEEE, 2020, pp. 59–63.

[5] R. Rawat *et al.*, "Enhanced Cybercrime Detection on Twitter Using Aho-Corasick Algorithm and Machine Learning Techniques," *Informatica*, vol. 48, no. 18, 2024.

[6] W. Aljedaani *et al.*, "Sentiment analysis on Twitter data integrating TextBlob and deep learning models: The case of US airline industry," *Knowl Based Syst*, vol. 255, p. 109780, 2022.

[7] F. Rustam, I. Ashraf, A. Mehmood, S. Ullah, and G. S. Choi, "Tweets classification on the base of sentiments for US airline companies," *Entropy*, vol. 21, no. 11, p. 1078, 2019.

[8] R. Monika, S. Deivalakshmi, and B. Janet, "Sentiment analysis of US airlines tweets using LSTM/RNN," in *2019 IEEE 9th international conference on advanced computing (IACC)*, IEEE, 2019, pp. 92–95.

[9] K. M. Hasib, M. A. Habib, N. A. Towhid, and M. I. H. Showrov, "A novel deep learning based sentiment analysis of twitter data for us airline service," in *2021 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, IEEE, 2021, pp. 450–455.

[10] M. T. H. K. Tusar and M. T. Islam, "A comparative study of sentiment analysis using NLP and different machine learning techniques on US airline Twitter data," in *2021 International conference on electronics, communications and information technology (ICECIT)*, IEEE, 2021, pp. 1–4.

[11] S. Rahman, M. Hasan, and A. K. Sarkar, "Prediction of Brain Stroke using Machine Learning Algorithms and Deep Neural Network Techniques," *European Journal of Electrical Engineering and Computer Science*, vol. 7, no. 1, pp. 23–30, 2023, doi: 10.24018/ejece.2023.7.1.483.

[12] N. D. Lagaros and M. Fragiadakis, "Fragility assessment of steel frames using neural networks," *Earthquake Spectra*, vol. 23, no. 4, pp. 735–752, 2007, doi: 10.1193/1.2798241.

[13] D. Sharma, R. Kumar, and A. Jain, "Breast cancer prediction based on neural networks and extra tree classifier using feature ensemble learning," *Measurement: Sensors*, vol. 24, p. 100560, 2022.

[14] N. Badri, F. Kboubi, and A. H. Chaibi, "Combining FastText and Glove Word Embedding for Offensive and Hate speech Text Detection," *Procedia Comput Sci*, vol. 207, no. Kes, pp. 769–778, 2022, doi: 10.1016/j.procs.2022.09.132.

[15] M. De Gregorio, A. Sorgente, and G. Vettigli, "Weightless Neural Networks for text classification using tf-idf," *ESANN 2021 Proceedings - 29th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, no. October, pp. 239–244, 2021, doi: 10.14428/esann/2021.ES2021-58.

[16] B. Kabra and C. Nagar, "Convolutional Neural Network based sentiment analysis with TF-IDF based vectorization," *Journal of Integrated Science and Technology*, vol. 11, no. 3, pp. 1–7, 2023.

[17] X. Wang, H. Wang, G. Zhao, Z. Liu, and H. Wu, "Albert over match-lstm network for intelligent questions classification in chinese," *Agronomy*, vol. 11, no. 8, 2021, doi: 10.3390/agronomy11081530.

[18] Y. Ho and S. Wookey, "The Real-World-Weight Cross-Entropy Loss Function : Modeling the Costs of Mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2020, doi: 10.1109/ACCESS.2019.2962617.

[19] A. Tato and R. Nkambou, "Mproving dam ptimizer," pp. 1–4, 2018.

[20] M. Mirzehi Kalateh Kazemi, Z. Nabavi, M. Rezakhah, and A. Masoudi, "Application of XGB-based metaheuristic techniques for prediction time-to-failure of mining machinery," *Systems and Soft Computing*, vol. 5, no. January, 2023, doi: 10.1016/j.sasc.2023.200061.

[21] J. Zhou *et al.*, "Predicting TBM penetration rate in hard rock condition: A comparative study among six XGB-based metaheuristic techniques," *Geoscience Frontiers*, vol. 12, no. 3, 2021, doi: 10.1016/j.gsf.2020.09.020.

[22] S. Guan, Y. Wang, L. Liu, J. Gao, Z. Xu, and S. Kan, "Ultra-short-term wind power prediction method based on FTI-VACA-XGB model," *Expert Syst Appl*, vol. 235, no. August 2023, p. 121185, 2024, doi: 10.1016/j.eswa.2023.121185.

[23] F. Alzamzami and M. Hoda, "Light Gradient Boosting Machine for General Sentiment Classification on Short Texts : A Comparative Evaluation," vol. 8, 2020, doi: 10.1109/ACCESS.2020.2997330.

[24] D. A. Mccarty and H. W. Kim, "Evaluation of Light Gradient Boosted Machine Learning Technique in Large Scale Land Use and Land Cover Classification," 2020.

[25] J. Park, J. Moon, and S. Jung, "Multistep-Ahead Solar Radiation Forecasting Scheme Based on the Light Gradient Boosting Machine : A Case Study of Jeju Island," 2020.

[26] G. Alfian *et al.*, "Predicting Breast Cancer from Risk Factors Using SVM and Extra-Trees-Based Feature Selection Method," 2022.

[27] D. Baby, S. J. Devaraj, J. Hemanth, and A. R. A. J. M. M, "Turkish Journal of Electrical Engineering and Computer Sciences Leukocyte classification based on feature selection using extra trees classifier : atransfer learning approach," vol. 29, no. 8, 2021, doi: 10.3906/elk-2104-183.

[28] Y. Freund, R. E. Schapire, P. Avenue, and F. Park, "A Short Introduction to Boosting," vol. 14, no. 5, pp. 771–780, 1999.

[29] D. J. Fleet and M. Brubaker, "AdaBoost," pp. 123–129, 2015.

[30] P. Tiwari, H. M. Pandey, A. Khamparia, and S. Kumar, "Twitter-based opinion mining for flight service utilizing machine learning," *Informatica*, vol. 43, no. 3, 2019.

[31] K. Deepa, H. Sangita, and H. Shruthi, "Sentiment Analysis of Twitter Data Using Machine Learning," *Lecture Notes in Electrical Engineering*, vol. 905, pp. 259–267, 2022, doi: 10.1007/978-981-19-2177-3_26.

[32] L. Sayfullina *et al.*, "Efficient detection of zero-day android malware using normalized bernoulli naive bayes," *Proceedings - 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2015*, vol. 1, pp. 198–205, 2015, doi: 10.1109/Trustcom.2015.375.

[33] S. Mehta, X. Shen, J. Gou, and D. Niu, "A New Nearest Centroid Neighbor Classifier Based on K Local Means Using Harmonic Mean Distance," 2018, doi: 10.3390/info9090234.

[34] B. Wang and S. Zhang, "A new locally adaptive K-nearest centroid neighbor classification based on the average distance," *Conn Sci*, vol. 34, no. 1, pp. 2084–2107, 2022.

Stable Diffusion Image Generation System Optimized with Variational Autoencoders and Low-Rank Adaptation

ChunLing Zhang
Weifang University of Science and Technology, School of Computer Science, Shouguang, 262700, China
E-mail: zhangcl0115@wfust.edu.cn

Keywords: image generation system, fréchet inception distance (fid), variational autoencoder (vae), low-rank adaptation (lora), stable diffusion (sd)

Recieved: July 23, 2025

Text-to-image generation has quickly evolved with diffusion-based generative models that combine semantic conditioning and latent-space denoising, allowing machines to generate high-quality visuals from natural language prompts. Despite these developments, existing diffusion systems still face challenges in prompt clarification accuracy, model adaptability, and computational efficacy, which limit their performance in real-time and resource-limited settings. The research aims to design and optimize an image generation framework based on Stable Diffusion (SD) that improves prompt processing, improves image quality, and enables lightweight fine-tuning. The system utilizes the LAION-Aesthetics v2 4.5 dataset, which contains high-quality text–image pairs suitable for visual generation tasks. Preprocessing involves text cleaning, tokenization, and semantic structuring, utilizing a transformer-based tokenizer to ensure accurate language-to-visual mapping. The architecture integrates Stable Diffusion, Variational Autoencoder (VAE) for latent-space decoding, and Low-Rank Adaptation (LoRA) for efficient fine-tuning with minimal computational cost. Results show that SD-VAE-LoRA achieved a PSNR of 33.7 dB, SSIM of 93 %, FID of 17.8, Inception Score of 36.02, and R-Precision of 90 %, superior to baseline SD and advanced diffusion models such as Latent Diffusion Method (LDM) [24], Menstrual Cycle-Inspired Latent Diffusion Method (MCI-LDM) [24], and Conditional Generative Adversarial Networks, Attention mechanisms, and Contrastive Learning (C-GAN+ATT+CL). The optimized system advances semantic alignment, decreases training time, and preserves image realism, confirming its strength for scalable, adaptive, and high-fidelity image generation applications.

Povzetek: Prispevek predstavlja optimiziran sistem za generiranje slik iz besedila, ki z nadgradnjo Stable Diffusion in uporabo VAE ter LoRA izboljša semantično ujemanje, kakovost slik in učinkovitost učenja ob nizkih računskih stroških.

1 Introduction

The rise of Artificial Intelligence (AI) tools has significantly transformed content generation, with text-to-image generation being a notable application. This system uses computation to generate images based on textual input, enabling machines to visualize human language and revolutionize the content generation landscape [1-2]. Text-to-image generation transforms text-based language prompts to high-quality images to bridge the gap between language and images and be used in design, art, education, and entertainment [3]. Deep Learning (DL) and vision-language understanding combine to create photorealistic and diverse images, a trend in fields like game development, animation, advertising, and personal media generation, enabling diverse styles [4-5]. Text-to-image generation is expected to expand into digital education, medical visualization, and e-commerce, with real-time art editing and AI-driven design assistance promising future domains [6]. Contemporary systems offer advantages due to automation of creative tasks, quick generation of visual imagery by audiences, and the ability of non-experts to create high-quality imagery [7]. It also fosters accessibility by providing visual content to non-image-makers [8]. Contemporary image generation systems use pre-trained methods to interpret text semantics, encode them, and decode them into visual artistic forms, but their development is influenced by various factors [9]. The systems of today rely on enormous images, leading to visual bias and expensive resources. The compromise between the complexity of a model and the user experience is a key to effective and efficient performance [10]. Diffusion models are effective in generating images based on text, but have difficulties in prompt accuracy, image quality, and resource efficiency, which restricts scalability [11]. The lightweight framework is needed to improve

alignment, fidelity, and computational properties [12]. The SD-VAE-LoRA model addresses these issues through a lightweight Stable Diffusion framework integrating SD for latent-space denoising, VAE for structured semantic decoding, and LoRA for efficient fine-tuning. Together, these modules enable resource-efficient, context-consistent image generation from diverse text prompts. The research aims to design a lightweight SD-VAE-LoRA model, which enhances image realism, semantic matching, and fine-tuning performance. It combines VAE as a structured representation and LoRA as an adaptation resource-efficiently. It is to generate high-quality images that are semantically consistent at a lower cost of computation.

1.1 Research question

1. Can Low-Rank Adaptation (LoRA) reduce training cost and resource consumption while maintaining or improving image generation quality?
2. Does integrating Variational Autoencoder (VAE) enhance latent-space decoding accuracy and semantic coherence in text-to-image synthesis?
3. How does the proposed SD-VAE-LoRA compare with existing methods (LDM, MCI-LDM, C-GAN+ATT+CL) in terms of realism, fidelity, and training efficiency?

2 Related work

The research [13] showed that SD methods, optimized for XL-XDXL architectures, can generate high-detail images with limited VRAM, improving quality by 35%, but lack real-time performance testing and comparisons with commercial alternatives. The investigation [14] improved segmentation performance and expert-rated image quality by using diffusion probabilistic techniques to produce realistic 3D Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) images. While there was improvement in Dice scores with synthetic data (0.91 to 0.95), there has been no clinical validation or generalizability of results across conditions. The research [15] evaluated the educational value of text-to-image AI in visual art, analyzing over 72,000 prompts. It found transformative teaching possibilities and cost savings, but identified gaps in legal ownership and economic models for AI-generated artwork in education. SynDiff, a confrontational circulation method based on unpaired health image conversion [16], showed improved performance over GANs in MRI and CT tasks by estimating conditional mappings from source noise to target, but the research failed to address real-world validation and runtime viability for clinical settings. The evaluation synthesized anonymized chest X-rays using a privacy-preserving sampling approach and trained classifiers on synthetic data using a latent diffusion approach [17]. The model achieved only a 3.5% Area under the Curve (AUC) gap from benchmarks on real data, but further validation was needed for rarer conditions and for robustness from a privacy perspective. Related works are included in Table 1.

Table 1: Comparison of other related methods

Reference	Objective	Dataset	Method	Numerical Results	Limitations
Yang et al. [18]	To enhance prompt privacy and safety in diffusion-based text-to-image generation systems	Internal prompt dataset (NSFW – Not Safe for Work filtered text corpus)	Prompt obfuscation and text-filtering mechanism for diffusion models	Qualitative improvement in safe image generation; no FID (Fréchet Inception Distance)	Lacks ethical safeguard validation
Paananen et al. [19]	To evaluate creative engagement and user experience in prompt-based generative systems	User-generated prompts via Midjourney, Stable Diffusion, and DALL-E	Prompt-based creativity analysis through diffusion model outputs	Improved user creativity ratings;	Absence of long-term validation, no real-world application testing
Lian et al. [20]	To improve multilingual prompt comprehension and scene layout accuracy in diffusion models	Custom multilingual prompt dataset	Scene layout prediction using a multilingual-aware diffusion framework	Achieved ~2× improvement in scene layout accuracy vs baseline models	Lacks usability and real-world validation;
Chen et al. [21]	To achieve energy-efficient and high-	LAION-Aesthetics v1.5 dataset	Optimized low-energy diffusion model	FID ≈ 18.5, PSNR ≈ 33 dB, and ~90% reduction in	Limited to specific domains;

	quality text-to-image generation			CO_2 emissions vs Stable Diffusion v1.5	reduced robustness to noise
Guo et al. [22]	To ensure temporal consistency in text-to-image video and animation generation.	Community text-to-image animation datasets	Temporal diffusion model enforcing frame-to-frame coherence	FID ≈ 19.2; achieved smooth motion continuity	Restricted to short-duration animations; lacks extreme motion
Huang et al. [23]	To enhance artistic fidelity and stylistic coherence in image generation	Custom artistic image corpus	Artistic diffusion model with SSIM (Structural Similarity Index Measure)	SSIM ≈ 90%; improved user preference and stylistic realism	Lacks evaluation on real-time and large-scale art generation;

2.1 Research gap

Diffusion models are effective in generating images through text prompts, but their current systems have issues with prompt handling accuracy, image quality, and fine-tuning resources [11]. These issues negatively affect their scalability and flexibility in real-world operations, especially for high-quality, semantically correct images at fast rates. To improve alignment, operational efficiency, and fine-tuning, a lightweight framework is needed to optimize visual fidelity and assembly, ensuring better alignment and reduced computational expenditure [12]. The present research overcomes these limitations by presenting a lightweight Stable Diffusion-based image generation model (SD-VAE-LoRA) that can enhance semantic correspondence, operating performance, and fine-tuning capabilities. The model combines three complementary features, which are; Stable Diffusion (SD) denoising in latent space and high-level image generation, Variational Autoencoder (VAE) structured latent representation and correct semantic decoding, and Low-Rank Adaptation (LoRA) efficient fine-tuning with a low computational cost. Collectively, these modules form a flexible resource efficient pipeline that can generate visual compatible and context consistent images using varied text prompts. The research aims to design a lightweight SD-VAE-LoRA model, which enhances image realism, semantic matching, and fine-tuning performance. It combines VAE as a structured representation and LoRA as adaptation resource-efficiently. It is to generate high-quality images that are semantically consistent at a lower cost of computation.

3 Methodology

This research aims to create and optimize an image generation system based on the SD method that increases prompt handling, as well as generation performance, and facilitates lightweight fine-tuning. The text data is cleaned and tokenized through a transformer-based tokenizer for semantic clarity. The SD method starts image generation by injecting random noise into a refined latent space of lower dimensions. The refined latent representation is decoded into an image that can be seen with a Variational Autoencoder (VAE). Low-Rank Adaptation (LoRA) is used to improve versatility and make method training less computationally intensive, allowing one to fine-tune a model without having to retrain the entire method again. Figure 1 represents the entire process of the research in a pictorial representation.

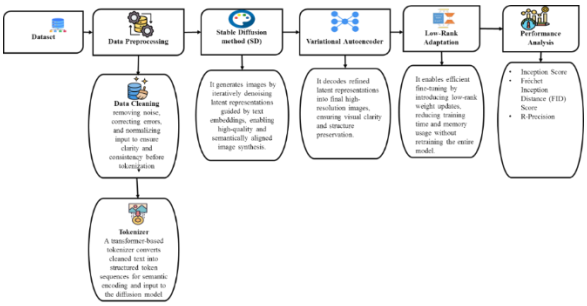


Figure 1: Process flow of the proposed methodology

3.1 Dataset

The system architecture commences with the acceptance of a user input request, where the users are required to input a text description, upon which they retrieve their corresponding images via the website's interface https://huggingface.co/datasets/laion/aesthetics_v2_4.5 . The dataset specifications and implementation of LoRA in the SD-VAE-LoRA framework are displayed in Table 2.

Table 2: Dataset and LoRA implementation details

Aspect	Description
Dataset Name	LAION-Aesthetics v2 4.5 (via Hugging Face)
Dataset Size	Approximately 12 million image–text pairs
Data Split	80% training, 10% validation, 10% testing

Data Filtering Criteria	Images filtered based on aesthetic score ≥ 4.5, NSFW content removed, and low-resolution (<512×512) images excluded
Data Cleaning	Duplicate captions and corrupted files were removed; normalized text encoding and resized all images to 512×512 pixels.
License and Usage	Dataset licensed under Creative Commons (CC-BY 4.0); used strictly for non-commercial, academic research

3.2 Pre-processing

Data preprocessing is the first stage of the system in which raw data is cleaned, formatted, and structured to help improve model performance. Data preprocessing involves cleaning, tokenizing, and transforming text prompts input from the user to make sure that they are semantically clear and compatible with the generation method.

Data cleaning is the method of discovering and repairing flaws or irregularities in data to ensure its quality and reliability. The data quality phase collects the raw data and performs context-aware analysis to measure the raw data across four dimensions of accuracy, completeness, consistency, and timeliness. In the second phase, the measurement phase, a specific set of metrics is applied to measure the quality of the data in the first phase. Next, the improvement phase allows practitioners to filter and sample data to remove a portion of the dataset using only quality, highly relevant data, and a reduced amount of data while maintaining the necessary quantity of data. Data cleansing organizes data effectively, allowing it to be used with accuracy and reliability by ridding the data of noise and errors while also improving system performance and the accuracy of generation.

Following the data cleaning, data tokenization is the next significant step for growth. Data tokenization takes the cleaned text and divides it into minor pieces, called tokens. This allows for the data to be managed much faster and in a way that it is understood semantically by the technique. Tokenization entails taking cleaned text and breaking it into evocative mechanisms for organization. Data tokenization assemblies cleaned text into specific tokens, allowing for accurate and effective interpretation and arranging to align correctly with the image generation process.

3.3 Image generation system based on stable diffusion method - variational autoencoder - low-rank adaptation (SD - VAE - LoRA)

This research aims to advance and optimize an image-generating classification based on the SD technique that advances quick handling, generation presentation, and allows for lightweight fine-tuning. SD plays a crucial part in changing text to images through improved denoising of the latent space. The VAE is a system that takes cleaned-up latent training and decodes it into visible images while ensuring high-quality reconstruction and semantically accurate image generation. LoRA proficiently updates parameters for fine-tuning, conserving training cost while still allowing the method to adapt to other tasks.

3.3.1 Stable diffusion (SD) method

SD allows for high-value, photorealistic images to be generated from text prompts effectively using latent-space denoising. SD is a DL method that generates realistic images from text prompts by denoising latent representations through iterative refinement. SD, a diffusion-based generative framework, produces superior images by step-by-step eliminating noise. The essential principles are as follows:

SD uses the Diffusion Method, which is qualified by mimicking an advancing procedure that gradually improves sound from a clear image until totally arbitrary noise is formed, known as the Forward Diffusion Process (FDP). SD acts in latent space, setting it apart from typical diffusion models. SD employs text-to-image generation to build plug-ins based on the provided text explanation. In particular, SD breaks down the model's weight matrix $X0 \in Qc \times l$ into two SD conditions: $\in Qq \times l \; and \; A \in Qc \times q$. The SD dimension parameter q denotes the rank number following dimensionality decrease. The form of this breakdown is as follows in equation (1).

$$X \approx B.A \tag{1}$$

Where X remains the resulting matrix, and $B.A$ signifies the matrix multiplication of B and A. Simultaneously, this strategy minimizes storage requirements, improving method finetuning efficiency. SD excels at producing superior images with effective modification, deprived of compromising method generation performance. SD allows for effective and high-quality image generation from text with the flexibility, scalability, and lower computational cost to be effective in creative use cases.

3.3.2 Variational autoencoder (VAE)

VAE serves as a fundamental part of the Stable Diffusion architecture in order to encode images into a lower-dimensional latent space and decode them into high-quality images. This implementation contrasts with another type of mathematical graph-based VAE, where latent representation is designed with a focus on efficient representation and reconstruction of visual features, which is inspired by the CompVis Stable Diffusion VAE design. The potential of the VAE is that it learns compressed latent representations that meaningfully encode the input, and

can generate images efficiently as well as in a high-quality and controlled manner, equation (2).

$$g_j^{(k+1)} = e_t^{(k)}\left(g_j^{(k)}, wj\right) + \sum_{i=1}^{M}\sum_{q=1}^{Q}\frac{B_{jiq}}{|T_j|}e_q^{(k)}(g_i^{(k)}, wj)$$

$$g_j^{(k+1)} = tang(g_j^{(k+1)} \tag{2}$$

Here, the feature vector of node j is obtained by combining self-information and aggregated data from neighboring nodes. Here, $g_i^{(k)}$ node j at layer k, wj is the input feature of node j from matrix W, $e_t^{(k)}$ is the self-connection function merging $g_i^{(k)}$ and wj, and $e_q^{(k)}$ is the edge-type–specific transformation aggregating information from neighbors. The adjacency tensor B_{jiq}represents the strength of the edge between nodes i and j under relation type q, normalized by T_j, the number of one-hop neighbors of node j. The indices $i = 1 \ldots M$ and $q = 1 \ldots Q$ denote the total number of nodes (M) and relation types (Q), while Tj is the neighbor set of node j. $g_j$applies the $tanh$ activation to introduce non-linearity and stabilize feature propagation. After multiple levels of propagation, aggregate nodes into a graph-level representation vector equation (3).

$$g'_H = \sum_{V\in U}\sigma\left(mm\left(g_v^{(K)}, w_v\right)\right) \odot tang\left(mm\left(g_V^{(K)}, w_V\right)\right)$$

$$g_H = tang\ g'_H \tag{3}$$

Where g'_H is the final graph-level embedding vector capturing the global context. The summation $\sum_{V\in U}$ aggregates feature from all nodes V within the set U, representing the collection of all nodes in the graph. σ (·) is the sigmoid function, mm is a linear neural network, and $\odot$ signifies element-wise multiplication. To approximate a normal distribution, it applies a linear neural network to g_H, yielding a mean vector μ_y and a diagonal covariance matrix Σ_y as in equation (4).

$$\mu_y = mm(g_H);\ \Sigma_y = mm(g_H) \tag{4}$$

Here, mm is a learnable matrix multiplication layer, where μ_y defines the distribution's center and Σ_y represents its variance, ensuring stable and continuous sampling for image reconstruction. Next, using the reparameterization approach, equation (5) yields the latent vector Y.

$$Y = \mu_Y + \in \odot \Sigma_Y \tag{5}$$

Where $\in$ is a Gaussian random variable. The VAE in the proposed SD-VAE-LoRA implementation facilitates effective latent compression, semantic consistency, and visual realism by ensuring generated images are structurally accurate and fine-grained in detail and that they are computationally simpler to operate through diffusion and decoding.

3.3.3 Low-Rank Adaptation (LoRA)

LoRA is used to facilitate efficient model fine-tuning with less computational cost. LoRA is used to fine-tune pre-trained models with efficiency through low-rank trainable matrices, minimizing memory and computational cost. To alter the adaptation process, the LoRA introduces a small square matrix $Q \in \mathbb{Q}^{q\times q}$ between frozen LoRA matrices that are set of the pre-trained weight matrix $X \in \mathbb{Q}^{n\times m}$.

One way to express the conventional LoRA forward path with an input $w \in \mathbb{Q}^m$is as equation (6).

$$g = wX + w\,\Delta X = wX + w\,BA \tag{6}$$

where $w \in \mathbb{Q}^m$is the input vector, X $\in \mathbb{Q}^{nxm}$ is the frozen pre-trained heaviness matrix, and $\Delta X = BA$ is the low-quality update that includes $B \in \mathbb{Q}^{nxr}$ and $A \in \mathbb{Q}^{rxm}$. This formulation efficiently adapts model parameters by updating only the low-rank matrices A and B, reducing memory and computation while preserving performance. While keeping matrices B and A frozen, and changing the forward path to equation (7).

$$g = wX + w\,\Delta X = wX + wBQA \tag{7}$$

Where B and A are set using the shortened SVD of the original weight matrix X. The SVD of X is represented by equation (8).

$$X = V\Sigma U^S \tag{8}$$

Where $V\,\Sigma\,\mathbb{Q}^{n\times n}, \Sigma \in \mathbb{Q}^{m\times m}$, and $U \in \mathbb{Q}^{m\times q}$ define (frozen) matrices B and A as in equation (9).

$$B = V_q\Sigma_q \text{ and } A = U_q^S \tag{9}$$

Here, U_q and V_q are the left and right singular vector matrices of the original weight matrix, Σ_q contains the top q singular values, and S denotes the selected subset for dimensional reduction. This decomposition initializes LoRA's trainable matrices A and B, preserving key information while minimizing parameter size.

The Q matrix is initialized with a Gaussian distribution $M(0,\sigma^2)$, where σ is a tiny but non-zero value. This ensures that it starts fine-tuning with a model that is nearly

identical to the pre-trained model. Fine-tuning involves freezing matrices $B$and A and updating only Q, which reduces the number of trainable parameters

This flexibility is especially useful for larger models, when standard approaches are limited by hidden dimensions. LoRA, like its predecessors, does not add computational burden or latency during inference. This module can be incorporated into the original matrix after training. The importance of LoRA was its ability to adapt large models in an efficient way using minimal parameters to preserve performance and adapt the model using very limited resource requirements. Algorithm 1 and figure 2 shows the process of SD - VAE - LoRA

Algorithm 1: Stable Diffusion method - Variational Autoencoder - Low-Rank Adaptation (SD - VAE - LoRA)

Step 1: Initialize Stable Diffusion (SD)
$SD_model = initialize_SD(parameters)$
Print ("Stable Diffusion initialized")
Step 2: Initialize Variational Autoencoder (VAE)
$VAE_model = initialize_VAE(latent_dim, encoder_params, decoder_params)$
Print ("VAE initialized")
Step 3: Initialize Low-Rank Adaptation (LoRA)
$LoRA_matrices = initialize_LoRA(SD_model.weights, rank_q)$
Print ("LoRA initialized")
Step 4: Process SD-VAE-LoRA
$Input_text$ = "Enter text prompt here"
Step 4a: Generate latent representation from SD
$latent_X0 = SD_model.encode_text(Input_text)$
Step 4b: Apply VAE encoding and decoding
$if\ latent_X0\ is\ not\ None$:
$latent_Y = VAE_model.encode(latent_X0)$
$reconstructed_image = VAE_model.decode(latent_Y)$
Print ("Image reconstructed via VAE")
else:
print ("Error: SD latent representation not generated")
Step 4c: Apply LoRA fine-tuning
$if\ LoRA_matrices\ is\ not\ None$:
Forward pass with LoRA adaptation
$adapted_output = latent_Y * LoRA_matrices.B * LoRA_matrices.Q * LoRA_matrices.A$
$final_image = decode(adapted_output)$
Print ("LoRA fine-tuning applied")
else:
print ("Error: LoRA matrices not initialized")
Step 5: Output final image
$display(final_image)$

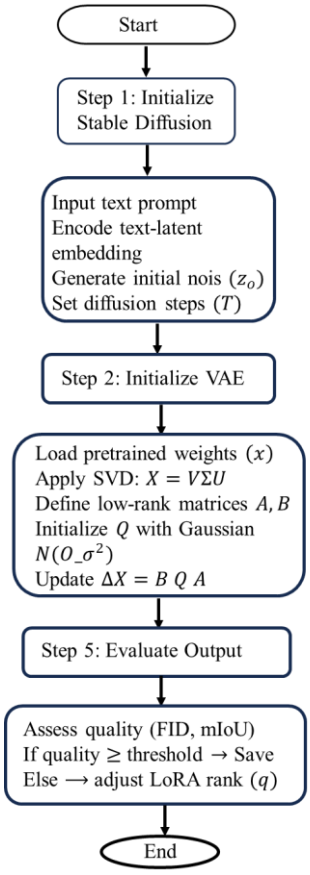


Figure 2: Flowchart for suggested SD-VAE-LoRA image generation evaluation system

4 Result and discussion

This research generates an optimized SD system to boost quick processing, image generation quality, and efficient fine-tuning. This research requires a pre-trained SD model, a GPU, a multi-core CPU, 16 GB RAM, Python 3.8+ compatibility, PyTorch, and LoRA tools. The presentation of the system is estimated using **user satisfaction evaluation** of its content and training time (in hours) to assess how it works with different image generation conditions.

4.1 Experimental setup

Table 3 indicates the system setup and testing environment to implement and test the suggested SD-VAE-LoRA-based image generation model.

Table 3: Experimental Setup for SD-VAE-LoRA Framework.

Component	Specification
Operating System	Ubuntu 22.04 LTS
CPU	Intel Xeon Gold 6258R, 24-core @ 2.7 GHz
RAM	64 GB DDR4
GPU	NVIDIA RTX A6000 (48 GB VRAM)
Programming Language	Python 3.10

Frameworks	PyTorch, Hugging Face Diffusers, Transformers, Accelerate
Dataset	LAION-Aesthetics v2 4.5 (Text–Image Pairs)
Model Components	Stable Diffusion (SD), Variational Autoencoder (VAE), Low-Rank Adaptation (LoRA)
Fine-Tuning Steps	20,000 iterations / 15 epochs
Training Time per Epoch	≈ 1.2 hours
Total Training Time	≈ 18 hours
Batch Size	16
Learning Rate	2×10^{-4}
Optimizer	AdamW
Evaluation Metrics	PSNR, SSIM, FID, Inception Score, R-Precision
Visualization Tools	Matplotlib, Seaborn, TensorBoard
Performance Monitors	NVIDIA NVML, PyTorch Profiler

4.2 Hyper parameter for the proposed model

Table 4 summarizes the hyperparameters of the suggested SD-VAE-LoRA image generation architecture.

Table 4: Parameter Setup for SD-VAE-LoRA Framework

Hyperparameter	Typical Values / Settings
Latent Dimension Size	512
Number of Epochs	15 (up to 20 for convergence)
Batch Size	16
Learning Rate	2×10^{-4}
Optimizer	AdamW
Dropout Rate	0.2
LoRA Rank (r)	8
LoRA Scaling Factor (α)	16
VAE Latent Channels	4
Diffusion Steps	1,000
Noise Scheduler	Cosine
Activation Function	GELU
Fine-Tuning Iterations	20,000
Evaluation Frequency	Every 1,000 steps
LoRA Adaptation Layers	Applied to cross-attention, U-Net transformer, and text encoder layers of Stable Diffusion
LoRA Rank (r)	8 (balancing efficiency and representational power)
Number of LoRA Modules	12 total modules integrated within the U-Net and text encoder blocks
Regularization Method	L2 weight decay and dropout (p = 0.2) to prevent overfitting
Optimization Objective	Minimized reconstruction loss and alignment error between text and image embeddings

- **User satisfaction evaluation:** It determines the similarity between the images generated and the expectations of the users in one aspect: realism, creativity, and immediate alignment; a high score signifies a higher perceived image quality and acceptance by the user, whereas a low score depicts a lack of satisfaction or visual quality.

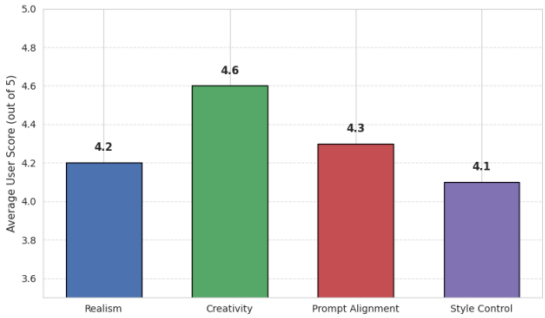


Figure 3: User satisfaction evaluation across image quality dimensions

Figure 3 illustrates user satisfaction with four criteria, such as realism, creativity, prompt alignment, and style control, with the highest score of creativity (~4.6), then prompt alignment (~4.3), and realism (~4.2), which is a good overall performance. Style control was rated slightly lower (~4.1), which implies that there is a little area to work on with regard to aesthetic consistency.

The proposed SD-VAE-LoRA and SD-VAE-baseline SD were evaluated in a double-blind context with 30 participants (15 experienced in digital design/AI art, 15 general users) through a structured user evaluation and a random selection of 50 image-prompt pairs used (25 pairs of image-prompts of each shape, 5 shape options, 25 neutral stimuli). The rating of images was done on a 5-point Likert scale on Realism, Creativity, Prompt Alignment, and Style Control. Mean +- standard deviation scores were obtained according to each criterion, and the overall score was obtained as a weighted mean (Realism 0.3, Prompt Alignment 0.3, Creativity 0.2, Style Control 0.2) to indicate visual fidelity and semantic accuracy.

- **FID Vs. Training time (in hours)**

FID is a performance metric that measures similarities between generated images and real images by comparing the distributions of their respective features. Training time is the number of days that it takes to train or even fine-tune a model up to the optimal performance and desired quality

of output on a given task. Figure 4 displays the FID scores over time of baseline SD vs. optimized SD-VAE-LoRA.

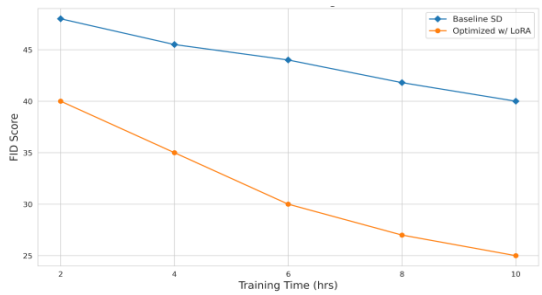


Figure 4: FID score comparison between baseline and optimized models

Figure 4 compares the FID scores with training time in Baseline SD and Optimized SD-VAE-LoRA models. The optimized model exhibits a more pronounced reduction in FID of 40 to 25 during 10 hours, which is faster converging and more faithful to the images than the baseline SD, which declined only by 48 to 40.

4.3 Comparison phase

Research compares the proposed SD-VAE-LoRA framework with existing, diffusion-based image generation methods, specifically Latent Diffusion Method (LDM) [24], Menstrual Cycle-Inspired Latent Diffusion Method (MCI-LDM) [24], and Conditional Generative Adversarial Networks, Attention mechanisms, and Contrastive Learning (C-GAN+ATT+CL) [25]. Table 5 illustrates the efficacy of the proposed SD-VAE-LoRA method, allowing it to generate images of higher quality.

Table 5: Quantitative comparison of image generation quality using PSNR and SSIM metrics

Methods	PSNR	SSIM (%)
LDM [24]	28.5dB	78
MCI-LDM [24]	32.7dB	92
SD-VAE-LoRA [Proposed]	**33.7dB**	**93**

Table 5 compares image quality measures between LDM [24], MCI-LDM [24], and the suggested SD-VAE-LoRA model. The offered method provides the best PSNR of 33.7 dB and SSIM of 93 percent, which is better than MCI-LDM (32.7 dB, 92 percent) and LDM (28.5 dB, 78 percent). These findings indicate that SD-VAE-LoRA has a higher reconstruction performance and structural fidelity.

- **Peak Signal-to-Noise Ratio (PSNR)** and **Structural Similarity Index (SSIM):** PSNR compares the quality of reconstruction between generated and reference images; the higher the PSNR, the higher the image fidelity and the lower the distortion, whereas the lower the PSNR, the higher the noise and the lower the reconstruction accuracy. SSIM compares the perceptual similarity of the generated image and the original image; a value of SSIM means the image has more structural and visual fidelity, whereas a lower value of SSIM means that the generated image loses some details or distorts the generated output.

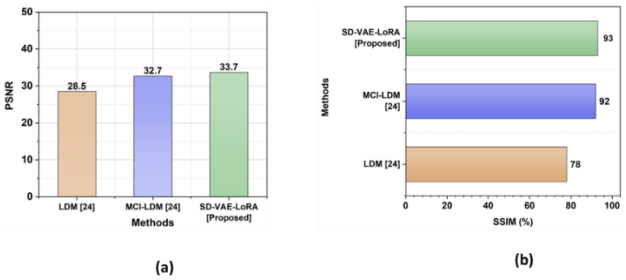


Figure 5: (a) PSNR Comparison of Existing Methods and Proposed Model, (b) SSIM performance across different diffusion models.

Figure 5 (a) shows the PSNR values of three methods, LDM [24], MCI-LDM [24], and the proposed SD-VAE-LoRA framework. The proposed model has the best PSNR of 33.7 dB, which is better than that of MCI-LDM (32.7 dB) and LDM (28.5 dB), and implies high accuracy in reconstruction and quality of the image. Figure 5 (b) shows the performance of SSIM in LDM [24], MCI-LDM [24], and the suggested SD-VAE-LoRA model. The SD-VAE-LoRA records the best SSIM value of 93% compared to MCI-LDM (92%) and LDM (78%), and this demonstrates that it has a higher level of structural consistency and visual fidelity when it comes to generated images.

Table 6: Methodological Comparisons Based on Inception Score, FID Score, and R-Precision

Methods	Inception Score	FID Score	R-Precision
C-GAN+ATT+CL [25]	35.23	18.2	89.14
SD-VAE-LoRA [Proposed]	**36.02**	**17.8**	**90**

Table 6 compares the Inception Score, FID Score, and R-Precision of C-GAN+ATT+CL [25] with that of the suggested SD-VAE-LoRA framework. The proposed model had a higher Inception Score (36.02), R-Precision (90), and a lower FID Score (17.8), which meant that it was more realistic, diverse, and accurately expressed the semantics. The reported findings support the enhanced perceptual quality and text-image correspondence.

- **Inception Score and Fréchet Inception Distance (FID) Score**: Inception Score evaluates the quality and diversity of generated images; a higher IS corresponds to more lifelike and varied image generation, and a lower IS represents a lack of diversity or less persuasive images. FID score compares the similarity between generated pictures and actual ones; a lower FID is used to mean more realistic and higher feature matching, whereas a higher FID is used to mean higher visual or semantic error.

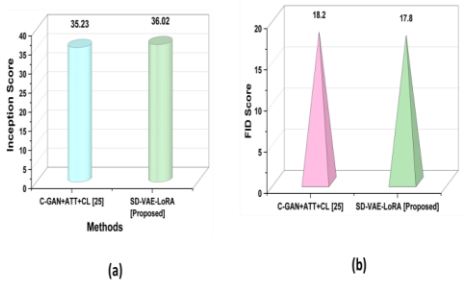


Figure 6: Comparative Analysis of (a) Inception Scores (b) FID Scores for Different Models

According to Figure 6 (a), SD-VAE-LoRA had a higher Inception Score (36.02) than C-GAN+ATT+CL (35.23), which implies that it has better image diversity and visual quality. This illustrates that the model is more generative and realistic. Figure 6 (b) demonstrates that SD-VAE-LoRA (17.8) had a lower FID score than C-GAN+ATT+CL (18.2), which represents greater image fidelity and semantic compatibility. Its better generative accuracy and efficiency is confirmed by the outcome.

- **R-Precision**: It quantifies the similarity between the most privileged generated images and their corresponding text prompts. High R-Precision indicates better text-image alignment, and a low value implies poor semantic correspondence.

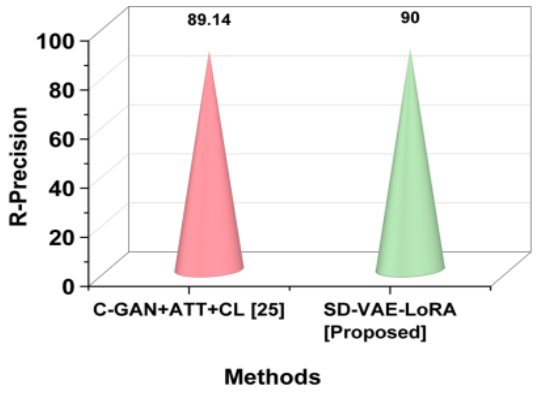


Figure 7: Assessment of R-precision across models

Figure 7 indicates that SD-VAE-LoRA achieved a higher R-Precision (90) compared to C-GAN+ATT+CL (89.14), which means that SD-VAE-LoRA has superior text-image semantic alignment. This enhancement shows greater consistency of generated visuals and input prompts.

- **Mean Intersection over Union (mIoU):** It calculates the overlap between generated and ground-truth regions, which translates into semantic correspondence and picture correctness. An increase in higher mIoU is an indication of a better prompt-to-image consistency delivered by the SD-VAE-LoRA setup.

- **Kernel Inception Distance (KID):** It gives the distribution of generated and real images' similarity using poly-kernel functions. The reduced KID is an indicator of better image realism and adherence to real-world visual semantics of the SD-VAE-LoRA model.

- **Learned Perceptual Image Patch Similarity (LPIPS):** It compares the feature of similarity in perception between generated and reference images on a deep feature representation. A reduced LPIPS signifies that the SD-VAE-LoRA model yields images with a higher visual fidelity and reduced perceptual variance with real images.

Table 7: Quantitative Comparison of Baseline and Proposed Model Performance

Method	mIoU [%	FID	KID	$LPIPS_{VGG}$
Stable Diffusion model [26]	67.89	54.57	0473 ± .0011	6281
SD-VAE-LoRA [Proposed]	72.54	42.16	0.0312 ± 0.0008	0.5124

Table 7 contrasts the results of the default Stable Diffusion model [26] and the suggested SD-VAE-LoRA framework. The proposed model obtains a larger mIoU of 72.54, lower FID (42.16), KID (0.0312 +- 0.0008), and LPIPSVGG (0.5124) with a better image realism, perceptual quality, as well as semantic alignment. These findings corroborate the fact that SD-VAE-LoRA has a higher visual fidelity and prompt consistency than the baseline models.

4.4 Dataset comparison

The optimized Stable Diffusion Image Generation System was trained and verified on the LAION-Aesthetics v2 4.5 dataset, and the generalization performance was tested

against the Text/Image GenAI dataset [27]. This comparative evaluation was aimed at evaluating the flexibility of the Proposed SD-VAE-LoRA architecture on large-scale aesthetic and small curated text-to-image datasets. The assessment involved standard metrics of performance, PSNR, SSIM, FID, Inception Score, and R-Precision, which are used to evaluate reconstruction fidelity, perceptual quality, and semantic consistency.

Table 8: Comparative performance metrics of Proposed datasets.

Metrics	LAION-Aesthetics v2 4.5 (Proposed)	Text/Image GenAI [27]
PSNR (dB)	33.7	32.8
SSIM (%)	93	91
FID	17.8	19.5
Inception Score	36.02	34.67
R-Precision (%)	90	88

Table 8 indicates that LAION-Aesthetics v2 4.5 (Proposed) data have better PSNR (33.7 dB) and SSIM (93%) than Text/Image GenAI [27]. It also reported a decreased FID (17.8) and increased Inception Score (36.02), which demonstrates a better image realism and variety. The R-Precision (90) also attests to the improved text-image alignment.

4.5 Statistical analysis

The research used Pearson correlation analyses to analyse the relationship between PSNR and SSIM, and FID and mIoU, and showed strong positive correlation rates between PSNR and SSIM and mIoU, but a negative correlation between PSNR and FID, indicating high image fidelity and semantic consistency.

Paired sample t-test: It compares the statistical means between two associated groups to ascertain whether the difference between the two groups is statistically significant. It ascertains that the SD-VAE-LoRA model has a larger enhancement in terms of image realism and semantic matching than the baseline model. It is represented as equation (10)

$$t = \frac{\bar{d}}{s_d/\sqrt{n}} \tag{10}$$

Where $\dot{d}$is the mean of paired differences, $s_d$is the standard deviation of differences, and $n$is the number of paired observations. Table 9 shows the t-Test-Based Assessment of Model Performance

Table 9: Paired Sample t-Test comparison of model performance metrics

Metric	Baseline Mean	Proposed Mean	Mean Difference	t-value	p-value (Sig.)
mIoU (%)	67.89	72.54	4.65	5.14	0.0003
FID	54.57	42.16	-12.41	-4.88	0.0005
KID	0.0473	0.0312	-0.0161	-3.92	0.0012
LPIPS	0.6281	0.5124	-0.1157	-4.26	0.0008

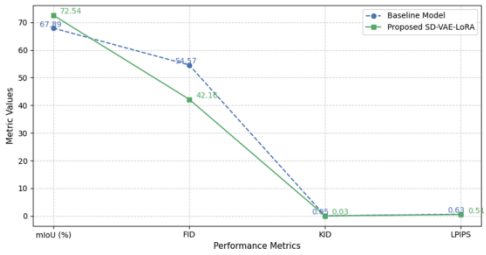


Figure 8: Evaluating Model Metrics Using Paired Sample t-Test

Figure 8 provides the results of a paired sample t-test that SD-VAE-LoRA performs significantly better than the baseline on all metrics, with a greater mIoU (72.54%), and a lower FID (42.16), KID (0.0312), and LPIPS (0.5124). The findings validate statistically significant increases ($p < 0.01$) in image quality and semantic performance.

5 Discussion

The research aims to design and optimize a Stable Diffusion-VAE-LoRA (SD-VAE-LoRA) model to increase prompts and understanding, as well as improve the quality of image generation and allow it to be fine-tuned efficiently and with lightweight. Previous researches found several limitations, such as the lack of usability and practical validation in multilingual diffusion models [20], the low adaptability of domains and weak resistance to noisy prompts in energy-efficient generation systems [21], poor management of extreme motion and short-duration focus in animation-based diffusion models [22], and the inability to test stylistic diffusion models in real-time and on a large scale [23]. The LDM [24] is afflicted with the multi-phase encoding-decoding and slower convergence. It has difficulties in preserving fine image details and semantic correspondence in large-scale generation tasks. MCI-LDM [24] is not very scalable and cannot be

generalized to various areas. It involves a lot of calculation and does not work systematically on abstract or dynamic prompts. C-GAN [25] has the problem of training instability and mode collapse during high-resolution image synthesis. It also requires efficient computational capabilities and hyperparameter sensitivity. ATT [25] models are memory-consuming and prone to overfitting. They occasionally deviate from concentration, decreasing text and image consistency. The CL [25] algorithms have issues with poor cross-modal alignments and sensitivity to noisy prompts. They cannot be used to ensure semantic consistency among various or ambiguous data. Error analysis reveals that prompt-to-image semantics mismatch occasionally, and finer details degrade slightly, which indicates places to consider how to optimize model robustness and visual accuracy. The optimized SD-VAE-LoRA framework overcomes these issues by integrating efficient latent-space denoising, semantic decoding through the VAE, and adaptive fine-tuning with LoRA. This combination reduces overfitting, enhances stability, and ensures faster convergence with improved visual fidelity and prompt accuracy. The system has allowed it to produce images in real-time, with high fidelity, creative, and educational uses, with minimal hardware and computational requirements.

6 Conclusion

Despite these progresses, existing diffusion systems still face challenges in prompt clarification accuracy, model flexibility, and computational efficacy, which limit their performance in real-time and resource-limited settings. The research aims to design and optimize an image generation basis based on Stable Diffusion (SD) that improves prompt processing, improves image quality, and enables lightweight fine-tuning. The optimized SD-VAE-LoRA framework significantly improved semantic alignment, image fidelity, and computational efficiency compared to baseline SD, SD-VAE, and other diffusion systems. On the basis of the LAION-Aesthetics v2 4.5 data, preprocessing based on transformer-based tokenization guaranteed accurate text-image association. VAE, combined with LoRA, allowed the efficient latent-space decoding and adaptive fine-tuning. The system attained PSNR = 33.7 dB, SSIM = 93, FID = 17.8, Inception Score = 36.02, and R-Precision = 90, verification of excellent performance. The constraints are that it is based on scaled curated datasets, limited generalization to abstract or low-context prompts, and relies on computational dependence during large-scale inference. Further research can be done to add multimodal input conditioning, adaptive prompt understanding, domain-specific training of specialized imagery, and real-time deployment in resource-constrained environments to further improve the scalability and generalization across domains.

References

[1] Vartiainen H, & Tedre M (2024). How text-to-image generative AI is transforming mediated action. IEEE Computer Graphics and Applications, 44(2), 12–22. https://doi.org/10.1109/MCG.2024.3355808

[2] Po R, Yifan W, Golyanik V, Aberman K, Barron JT, Bermano A, Chan E, Dekel T, Holynski A, Kanazawa A, & Liu CK (2024). State of the art on diffusion models for visual computing. Computer Graphics Forum, 43(2), e15063. https://doi.org/10.1111/cgf.15063

[3] Yoon J, Yu S, Patil V, Yao H, & Bansal M (2024). Safree: Training-free and adaptive guard for safe text-to-image and video generation. arXiv. https://doi.org/10.48550/arXiv.2410.12761

[4] Alhabeeb SK, & Al-Shargabi AA (2024). Text-to-image synthesis with generative models: Methods, datasets, performance metrics, challenges, and future direction. IEEE Access, 12, 24412–24427. https://doi.org/10.1109/ACCESS.2024.3365043

[5] Habib MA, Wadud MAH, Patwary MFK, Rahman MM, Mridha MF, Okuyama Y, & Shin J (2024). Exploring progress in text-to-image synthesis: An in-depth survey on the evolution of generative adversarial networks. IEEE Access. https://doi.org/10.1109/ACCESS.2024.3435541

[6] Jin H (2025). Cross-Modal Attention GAN for Text-to-Artistic Image Generation. Informatica, 49(20). https://doi.org/10.31449/inf.v49i20.8303

[7] Tang A, Wei L, Ni Z, & Huang Q (2025). Multi-Modal Modified U-Net for Text-Image Restoration: A Diffusion-Based Multimodal Information Fusion Approach. Informatica, 49(2). https://doi.org/10.31449/inf.v49i2.8245

[8] Arnaubec A, Ferrera M, Escartín J, Matabos M, Gracias N, & Opderbecke J (2023). Underwater 3D reconstruction from video or still imagery: Matisse and 3Dmetrics processing and exploitation software. Journal of Marine Science and Engineering, 11(5), 985. https://doi.org/10.3390/jmse11050985

[9] Tiribelli S, Pansoni S, Frontoni E, & Giovanola B (2024). Ethics of artificial intelligence for cultural heritage: Opportunities and challenges. IEEE Transactions on Technology and Society. https://doi.org/10.1109/TTS.2024.3432407

[10] Mittal S, Wittman RL, Gibson J, Huffman J, & Miller H (2023). Providing a user extensible service-enabled multi-fidelity hybrid cloud-deployable SoS Test and Evaluation (T&E) infrastructure: Application of modeling and simulation (M&S) as a service (MSaaS). Information, 14(10), 528. https://doi.org/10.3390/info14100528

[11] Yuan M, Chen J, Hu Y, Feng S, Xie M, Mohammadi G, Xing Z, & Quigley A (2024). Towards human-AI

synergy in UI design: Enhancing multi-agent-based UI generation with intent clarification and alignment. arXiv. https://doi.org/10.48550/arXiv.2412.20071

[12] Hariri W (2023). Unlocking the potential of ChatGPT: A comprehensive exploration of its applications, advantages, limitations, and future directions in natural language processing. arXiv. https://doi.org/10.48550/arXiv.2304.02017

[13] Kabir AI, Mahomud L, Al Fahad A, & Ahmed R (2024). Empowering local image generation: Harnessing stable diffusion for machine learning and AI. Informatica Economica, 28(1), 25–38. https://doi.org/10.24818/issn14531305/28.1.2024.03

[14] Khader F, Müller-Franzes G, Tayebi Arasteh S, Han T, Haarburger C, Schulze-Hagen M, Schad P, Engelhardt S, Baeßler B, Foersch S, & Stegmaier J (2023). Denoising diffusion probabilistic models for 3D medical image generation. Scientific Reports, 13(1), 7303. https://doi.org/10.1038/s41598-023-34341-2

[15] Dehouche N, & Dehouche K (2023). What's in a text-to-image prompt? The potential of stable diffusion in visual arts education. Heliyon, 9(6). https://doi.org/10.1016/j.heliyon.2023.e16757

[16] Özbey M, Dalmaz O, Dar SU, Bedel HA, Özturk Ş, Güngör A, & Cukur T (2023). Unsupervised medical image translation with adversarial diffusion models. IEEE Transactions on Medical Imaging, 42(12), 3524–3539. https://doi.org/10.1109/TMI.2023.3290149

[17] Packhäuser K, Folle L, Thamm F, & Maier A (2023). Generation of anonymous chest radiographs using latent diffusion models for training thoracic abnormality classification systems. IEEE International Symposium on Biomedical Imaging, 1–5. https://doi.org/10.1109/ISBI53787.2023.10230346

[18] Paananen V, Oppenlaender J, & Visuri A (2024). Using text-to-image generation for architectural design ideation. International Journal of Architectural Computing, 22(3), 458–474. https://doi.org/10.1177/14780771231222783

[19] Lian L, Li B, Yala A, & Darrell T (2023). LLM-grounded diffusion: Enhancing prompt understanding of text-to-image diffusion models with large language models. arXiv. https://doi.org/10.48550/arXiv.2305.13655

[20] Chen J, Yu J, Ge C, Yao L, Xie E, Wu Y, Wang Z, Kwok J, Luo P, Lu H, & Li Z (2023). Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. arXiv. https://doi.org/10.48550/arXiv.2310.00426

[21] Guo Y, Yang C, Rao A, Liang Z, Wang Y, Qiao Y, Agrawala M, Lin D, & Dai B (2023). AnimateDiff: Animate your personalized text-to-image diffusion models without specific tuning. arXiv. https://doi.org/10.48550/arXiv.2307.04725

[22] Huang N, Zhang Y, Tang F, Ma C, Huang H, Dong W, & Xu C (2024). Diffstyler: Controllable dual diffusion for text-driven image stylization. IEEE Transactions on Neural Networks and Learning Systems. https://doi.org/10.1109/TNNLS.2023.3342645

[23] Luccioni AS, Akiki C, Mitchell M, & Jernite Y (2023). Stable bias: Analyzing societal representations in diffusion models. arXiv. https://doi.org/10.48550/arXiv.2303.11408

[24] Mahmoud GM, Elbaz M, Said W, & Elsonbaty AA (2025). Menstrual cycle inspired latent diffusion model for image augmentation in energy production. Scientific Reports, 15(1), 16749. https://doi.org/10.1038/s41598-025-99088-4

[25] Habib MA, Wadud MAH, Pinky LY, Talukder MH, Rahman MM, Mridha MF, Okuyama Y, & Shin J (2023). GACnet: Text-to-image synthesis with generative models using attention mechanisms with contrastive learning. IEEE Access, 12, 9572–9585. https://doi.org/10.1109/ACCESS.2023.3342866

[26] Kaleta J, Dall'Alba D, Płotka S, & Korzeniowski P (2024). Minimal data requirement for realistic endoscopic image generation with stable diffusion. International Journal of Computer Assisted Radiology and Surgery, 19(3), 531–539. https://doi.org/10.1007/s11548-023-03030-w

Enhancing Short-Term Load Forecasting in the Electric Power Industry: A Hybrid Approach Integrating Machine Learning and Advanced Optimization Techniques

Xiaoyan Tang*, Pengfei Yue
[1]Henan Polytechnic Institute, Nanyang, China
E-mail: tangxy23@hotmail.com
*Corresponding author

Keywords: machine learning, short-term load forecasting, radial basis function, support vector regression, PSO, HGS, AO

Received: November 9, 2025

This work introduces a hybrid methodology that combines machine learning with sophisticated optimization approaches to improve short-term load forecasting in the electric power sector. The model uses Support Vector Regression and Radial Basis Function together and then improves it with Aquila Optimizer, Particle Swarm Optimization, and Harris Hawks Optimization. The method was evaluated on a dataset that includes daily electricity usage data and important weather variables like temperature and humidity. We used important measures, including the Root Mean Squared Error, Mean Absolute Error, and the Coefficient of Determination, to see how well the model worked. The results show that the hybrid Radial Basis Function-Harris Hawks Optimization model works better than the standalone Support Vector Regression model. It has a 20% lower Root Mean Squared Error and a 15% higher accuracy. These gains show that the hybrid model makes more accurate predictions, especially for real-time operational planning. The hybrid optimization techniques help make load forecasting models much more accurate and efficient. They could be very useful for managing energy systems. This study underscores the significance of amalgamating machine learning with optimization algorithms to augment forecasting abilities and furnish critical insights for power system operators and academics seeking to increase grid stability and planning.

Povzetek: Predstavljena je hibridna metoda, ki združuje strojno učenje in optimizacijo za natančnejše kratkoročno napovedovanje porabe električne energije.

1 Introduction

Forecasting electricity consumption is essential for the effective operation and planning of power systems, aiding utilities in demand management, resource allocation optimization, and grid stability assurance. Precise forecasting is crucial in short-term load forecasting because power consumption may vary due to factors such as time of day, meteorological conditions, and economic activities. Conventional forecasting techniques, such as statistical and time-series methods, are extensively utilized; yet, they frequently fail to accurately represent the intricate, nonlinear relationships present in energy consumption data. This paper seeks to tackle these challenges by proposing a hybrid model that integrates machine learning techniques, Support Vector Regression (SVR), and Radial Basis Function (RBF) networks with sophisticated optimization algorithms, including Aquila Optimizer (AO), Particle Swarm Optimization (PSO), and Harris Hawks Optimization (HGS). The primary aim of this work is to enhance the precision of short-term load forecasting by adjusting the parameters of machine learning models through various optimization strategies. The objective of this study is to assess the efficacy of the hybrid model in predicting electricity demand using a dataset comprising historical consumption data and pertinent meteorological variables.

The continuous operation of the electrical power system requires prompt coordination spanning from power plants to distribution substations, ensuring secure and dependable functioning to deliver uninterrupted high-quality electricity services. Thorough pre-real-time operational planning is imperative, taking into account the dynamics of renewable energy sources, maintenance schedules for power plants, and the allocation of hydrothermal resources to align electricity generation with anticipated demand. Maintaining this real-time equilibrium between energy generation and load is crucial to prevent grid damage [1].

The planning phase of power system operation can be categorized into three distinct time frames, each with its own set of objectives: mid-term, short-term, as well as long-term. The short-term duration, ranging from 1 day to 1 week, primarily addresses operational and security concerns within the power system. The mid-term period, ranging from several weeks to many months, looks after

resource management in mitigating energy shortages from existing generating plants. From the long-term perspective spanning from several years up to decades, one plans for new facilities in electricity generation or adjustments in the transmission system. Although the details of the criteria may vary from region to region, the basic principles hold for any situation [2].

The range of application for it scales from individual needs to a whole array of requirements in the case of the short-term forecasting of electricity loads. Starting from the scale of an individual transformer, buildings [3], and cities [4] right up to large regions [5] is where the scale of load prediction could vary. There is another key dimension to the variation in this research horizon. It extends from very short-term predictions, such as anticipating load fluctuations for machine tools in the next few seconds [6], to predictions stretching over a few hours [7].

The most prevalent forecast is for the following day's load [8], although some studies also concentrate on predicting the load 48 hours ahead [9] or providing weekly forecasts [10].

A broad spectrum of methodologies and algorithms has been developed and employed to tackle STLF. Time-series analysis is a popular forecasting methodology that is widely used in various domains. STLF has become increasingly important for ensuring the efficient and reliable operation of power systems. Recent studies have explored various advanced machine learning techniques to enhance the accuracy of load predictions. Lotfi et al. [11] proposed an optimized machine learning approach that integrates environmental and historical factors into short-term electricity demand forecasting. Their approach, which utilized an optimized Random Forest model, significantly outperformed traditional methods like SARIMA and Exponential Smoothing, showing how modern optimization techniques can boost forecasting accuracy. Wen et al. [12] introduced a deep learning-driven hybrid model for short-term load forecasting in smart grids. They combined Gated Recurrent Units (GRU), Temporal Convolutional Networks (TCN), and attention mechanisms to create a more accurate model. Their results demonstrated the superiority of hybrid models in handling complex time-series data, particularly in smart grid environments where demand can vary significantly. Lukong et al. [13] proposed a novel spatial long-term load forecasting model using a multiple cluster-based Long Short-Term Memory (LSTM) network. This model segments the grid into clusters based on regional characteristics and employs LSTM networks to capture unique spatiotemporal dependencies in electricity consumption. Their approach highlighted the potential of spatial forecasting methods in improving the accuracy of load predictions across large areas with varying consumption patterns. Akhtar et al. [14] highlighted the significance of STLF for power systems. Time series models are efficient but struggle with nonlinearity, while ANNs offer better accuracy at higher computational costs. Regression models are simpler but less accurate, and hybrid models improve accuracy with higher computational demands. Advances in deep learning, especially recurrent and attention-based models, could enhance accuracy while reducing computational costs. In recent research, traditional models like ARMA are less prevalent in STLF, as ML methods offer superior performance. Musaylh et al. [15] investigated short-term electricity demand forecasting in Queensland, Australia, utilizing Multivariate Adaptive Regression Spline (MARS), SVR, and ARIMA (Autoregressive Integrated Moving Average) models. MARS outperformed SVR and ARIMA for 0.5 h and 1.0 h forecasting horizons, while for a daily, that is, 24 h, forecasting, SVR outperformed. From this point of view, some valuable lessons are drawn about real-time demand forecasting. Amin et al. [16] analyzed the problem of short-term load forecasting, focusing on accuracy and speed. They applied ARIMA and SVM models and compared them by MAPE and MSE metrics. They concluded that in the case of nonlinear patterns, SVMs perform better, while ARIMA would work better for a linear approximation of the load. Unlike typical statistical methods in the time-series analysis of data, ML can assimilate other critical parameters, such as weather, thereby increasing the accuracy in STLF. The reasons that make MLR widely adopted in most processes of STLF are versatility and applicability in most scenarios. Adeoye et al. [17] built an hourly model for electricity demand for 14 West African countries based on several factors that included: electrification rates, household appliances, and prevailing weather conditions. The model projects the demand for electricity in 2016 and 2030, thus showing seasonal variation and a significant increase in demand by 2030. This methodology presents insights relevant to capacity planners concerned with the attainment of 100% electrification rates within the region by 2030.

Apart from all the preceding methods, the artificial neural network is another widely used methodology in STLF in the last few decades due to its flexibility in algorithms. Case studies in Victoria and New South Wales have demonstrated the accuracy and potential of this model in guiding sustainable city development [3]. Viegas et al. [18] proposed a methodology for STLF that combines artificial neural network modeling with feature choice using a genetic algorithm. It is based on the historical data of consumption, weather, and stock index for the prediction of the 24-hour ahead load. Testing with datasets from different geographical regions yields models whose mean average percentage inaccuracy is less than 2%. The feature selection algorithm drastically reduces the number of features involved while improving model accuracy. The SVR model has been widely applied in STLF, especially with a linear kernel, because there is a linear relationship has been found between the inputs and the forecast. Wang et al. [19] proposed an integrated model of EMD-PSO-SVR for short-term load forecasting, incorporating EMD and SVR with PSO, considering temperature, weekend, and holiday factors. EMD decomposes the load data into its main IMF components, the SVR forecasts these components, while the PSO optimizes the parameters for the SVR. These steps provide better stability and accuracy for the predictions compared to conventional models. Ferreira et

al. [9] updated radial basis function neural network models for multi-step electricity consumption prediction in Portugal up to 48 hours. They used least-squares support vector machines for model resetting and enhancing performance. Their method gave better model identification and a strategy for updating compared to the other existing neural models. They had also used a method of initializing hyperparameters that improved the performance without multiple random trials or grid search procedures. Omidi et al. [20] proposed a new-SVR for STLF in 2015, which is of utmost importance for the optimal operation of the power system. They compared its performance with ANN, underlining the nu-SVR suitability for STLF because of its capability of modeling nonlinear factors in electric grids. All the works had their original approach with the purpose of improving the accuracy of electricity demand forecasts in the environment to which they were addressed.

They investigated hybrid approaches and integrated metaheuristic techniques to obtain more refined energy models with the help of methodologies based on artificial intelligence. That will definitely help raise the bar for performance in these models. They mixed and matched various methods in their work to enhance general performance in energy models. They employed metaheuristic algorithms to tune the parameters of these models for improved results. This amalgamation of hybrid and AI-based methodology holds great promise for energy modeling techniques. An arithmetic optimization algorithm and its improved version have been proposed for energy demand forecasting in Turkey by Aslan et al. [21]. The researchers forecasted the energy demand using a linear regression model with demographic and economic data until the year 2030, according to three scenarios and reported competitive results compared to the existing methods. Zhang et al. [22] have proposed a hybrid model for short-term electricity load forecasting, IEMD, ARIMA, and WNN optimized by FOA. The comparative analysis showed its better performance compared to other models. Haq et al. [23] once proposed a new hybrid STLF model incorporating IEMD, T-Copula correlation analysis, and deep belief networks. The model was tested on real-time data from Australia and the USA and showed improved forecasting accuracy both in RMSE and MAPE compared to traditional methods. Xiao et al. [24] presented a modified GRNN optimized through the MOFA for STLF, which obtained a highly steady and accurate result. Thus, the proposed hybrid model, assessed with the Australian load data, outperforms other models in both criteria and proves effective for load forecasting. Shanmugam and Ramana [25] emphasized the necessity of precise electricity consumption predictions to enhance planning and efficiency. The nonlinear and nonstationary characteristics of power consumption data rendered standard approaches less effective. They suggested a hybrid model that integrates the RBF algorithm with six meta-heuristic optimizers: GWO, AGWO, MFO, PSO, MVO, and ABC. The findings indicated that the RBF-AGWO model surpassed the others, achieving R^2 values of 0.9994, 0.9920, and 0.9985 for the training, testing, and total datasets, respectively, demonstrating AGWO's exceptional accuracy. Dai and Zhao [26] emphasized the significance of precise power load forecasting in smart grids, employing Support Vector Regression and Radial Basis Function networks. They integrated these models using Aquila Optimizer, Particle Swarm Optimization, and Harris Hawks Optimization to enhance performance. These optimization methods effectively adjusted hyperparameters, improving forecasting precision. Their hybrid methodology surpassed conventional techniques by utilizing the advantages of each element to circumvent local minima and enhance forecast precision. Huang et al. [27] showed that the integration of SVR and RBF with optimization methods such as PSO, AO, and HGS markedly improves model efficacy. These optimization methods provide effective hyperparameter adjustment, enhancing accuracy and generality. This hybrid strategy, which combines machine learning models with powerful optimization algorithms, surpasses traditional methods in forecasting accuracy.

1.1. Research gaps and novelties

Short-term load forecasting is very important for running and managing power systems well, but the methods that are now available typically have trouble making accurate predictions in contexts that are constantly changing and are quite complicated. Conventional forecasting methods, including time-series analysis and fundamental machine learning models, often struggle to accurately represent the non-linearities present in electricity consumption data and do not efficiently optimize model parameters. The main goal of this work is to improve the accuracy of short-term load forecasting by combining machine learning methods with advanced optimization algorithms. The research question that guides this study is: In what ways does the combination of advanced optimization techniques and machine learning models enhance the precision and dependability of short-term load forecasting, relative to conventional forecasting approaches? The hypothesis being tested posits that the integration of Support Vector Regression and Radial Basis Function networks with optimization methods will enhance forecasting performance, specifically in terms of accuracy and a diminished Root Mean Squared Error.

To test this theory, the paper suggests a hybrid model that combines Support Vector Regression with Radial Basis Function networks and improves it even more with three advanced algorithms: Aquila Optimizer, Particle Swarm Optimization, and Harris Hawks Optimization. The reasons for using these particular optimization methods are their unique benefits: Aquila Optimizer is based on the way the Aquila bird hunts and lets you explore and use the solution space powerfully. Particle Swarm Optimization is based on swarm intelligence and is great for problems that need to be solved continuously. It also lets you search for solutions quickly. Harris Hawks Optimization is based on the way animals act when they are hungry and makes search dynamics better. This new hybrid method not only uses

the capabilities of these optimization techniques to improve machine learning models, but it also includes weather variables like temperature and humidity, which haven't been studied as much in the past. This integration offers a more thorough and realistic approach to short-term load forecasting, filling in the gaps in current research and providing a useful model for managing energy systems in the real world.

The paper presents new hybrid models for STLF in the electric power sector, consisting of RBF and SVR algorithms optimized with various optimizers. These algorithms combine to give improved accuracy in the forecast of the demand load when temperature data is included. Most previous related studies have focused on the use of single algorithms, whereas this study looks into the use of hybrid models. The use of many optimizers, including AO, PSO, and HGS, serves to provide novelty in this research work to improve the model performance. The analysis and evaluation conducted with these optimized hybrid models will be very valuable with respect to the insight into performance that they will be able to provide, thus improving practical applications to power system operators and researchers involved in STLF.

Table 1 encapsulates essential research in short-term load forecasting, emphasizing the methodologies, datasets, performance indicators (RMSE, MAPE, R^2), and findings. It also recognizes deficiencies, including restricted datasets, computational obstacles, and insufficient emphasis on real-time applications. The table highlights the necessity for expansive datasets, enhanced efficiency, and the amalgamation of hybrid models with optimization algorithms to improve forecasting precision and relevance.

Table 1: Comparative analysis of methods and approaches in short-term load forecasting studies.

Study	Methods	Dataset	Key Metrics	Conclusions	Gaps/Limitations
Lotfi et al. [11]	Optimized Random Forest	Environmental and historical data	Accuracy, RMSE	Outperformed SARIMA and Exponential Smoothing models	Limited to one dataset; doesn't address deep learning techniques
Wen et al. [12]	Hybrid model combining GRU, TCN, and Attention	Smart Grid data	Accuracy, RMSE, MAE	Hybrid models outperform single models in complex data scenarios	Computationally intensive, not ideal for real-time forecasting
Lukong et al. [13]	Cluster-based LSTM	Smart Grid data	RMSE, MAE	Improved accuracy by considering spatial dependencies in grid loads	May not scale well to larger or more varied grids
Akhtar et al. [14]	Time-series, ANN, regression, hybrid models	Various datasets	Accuracy, RMSE, MAE	Hybrid models improve accuracy but at higher computational cost	High computational demand limits real-time applicability
Musaylh et al. [15]	MARS, SVR, ARIMA	Queensland, Australia	RMSE, MAE	MARS outperformed SVR for short horizons, SVR for 24h	Focuses only on a specific region and timeframe
Amin et al. [16]	ARIMA, SVM	General power system	MAPE, MSE	SVM outperformed ARIMA for non-linear patterns	Limited scalability for large datasets
Adeoye et al. [17]	Hourly model for electricity demand	14 West African countries	-	Predicted electricity demand in 2016 and 2030, highlighting seasonal variations and future demand increase	Limited to the West African region, no broader application
Viegas et al. [18]	ANN with a genetic algorithm	Various regions	MAPE	Model accuracy < 2%	Focuses only on 24-hour load forecasting
Wang et al. [19]	EMD-PSO-SVR	General power system	RMSE	Improved stability and accuracy	Computationally expensive
Ferreira et al. [9]	Radial Basis Function Neural Networks	Portugal	-	Improved multi-step prediction	Lacks integration with optimization algorithms

Omidi et al. [20]	SVR	General power system	-	SVR outperformed ANN for non-linear grid factors	Lacks comparative analysis with other methods
Aslan et al. [21]	Linear regression with demographic and economic data	Turkey	-	Accurate demand forecasting for future scenarios	Limited to Turkey, lacking broader application
Zhang et al. [22]	Hybrid IEMD, ARIMA, WNN optimized by FOA	General power system	-	Outperformed traditional models	Lacks a clear real-time application focus
Haq et al. [23]	Hybrid IEMD, T-Copula, deep belief networks	Australia, USA	RMSE, MAPE	Improved forecasting accuracy	Limited data sources from only two countries
Xiao et al. [24]	Modified GRNN optimized through MOFA	Australia	-	Highly stable and accurate	Limited to one region and does not include seasonal data
Shanmugam and Ramana [25]	RBF with meta-heuristics (GWO, AGWO, MFO, PSO, MVO, ABC)	General power system	R^2	RBF-AGWO achieved R^2 values of 0.9994	Computationally intensive for real-time use
Dai and Zhao [26]	SVR, RBF with optimization (AO, PSO, HGS)	General power system	-	Improved forecasting precision	Not directly applicable to real-time applications
Huang et al. [27]	SVR, RBF with PSO, AO, HGS	General power system	-	Improved accuracy and generality	Lacks scalability for larger systems

2 Methodology

This study concentrated on the implementation of machine learning algorithms for power forecasting, commencing with comprehensive data gathering and preprocessing to ensure quality assurance. The significant factors, especially climatic variables such as humidity and temperature, were discovered and examined through methods including correlation matrices and sensitivity analysis. The dataset was divided into training (80%) and testing sets for model assessment. Multiple machine learning methods, such as SVR and RBF, were examined, utilizing sophisticated optimization techniques as Aquila Optimizer, HGS, and PSO to refine the models' hyperparameters and enhance performance. The models were assessed using statistical metrics and graphical tools to measure forecasting accuracy and reliability. Figure 1 encapsulates the research process, delineating the sequence of data pretreatment, model construction, optimization, and validation.

Alongside the suggested hybrid models, comparisons were conducted with contemporary state-of-the-art deep learning architectures, including Long Short-Term Memory (LSTM) networks, GRU, and attention-based models. These models exemplify the most recent progress in short-term load forecasting and are often referenced in academic literature. The models were assessed using identical metrics (RMSE, MAE, R^2) to ensure a uniform comparison.

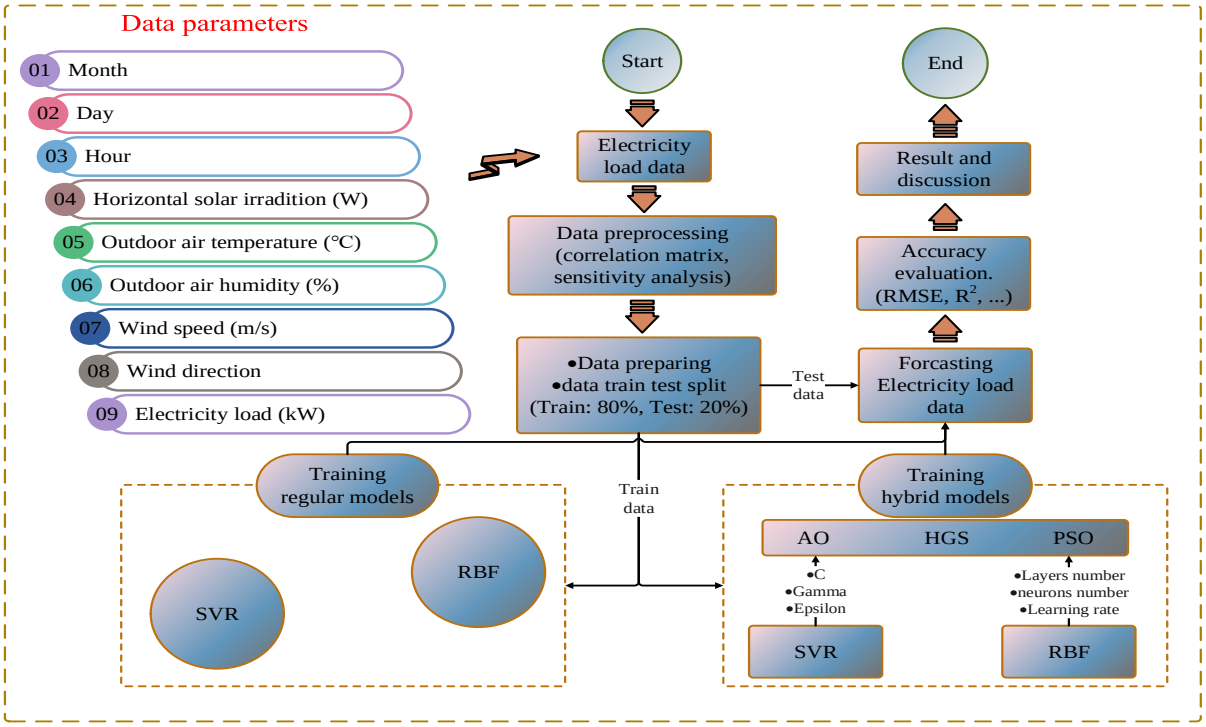


Figure 1: Flowchart diagram of the current investigation

2.1. Data collection

The dataset utilized in this analysis, derived from [28], comprises daily power consumption statistics from a designated geographic region during two years (2020–2021). The geographical scope is confined to this region, perhaps introducing biases associated with local energy use habits. Data preprocessing encompassed normalization and interpolation to address missing values and maintain consistency. The dataset's restricted temporal and geographical scope may constrain the generalizability of the results. Future research should integrate data from diverse places and prolonged timeframes to augment the validity and usefulness of the findings. Some of the input variables include maximum/minimum temperatures and humidities, as well as average temperature and humidity. Since short-term electricity load data are intrinsically complex in nature, the accuracy of forecasting is vitally important for the effective functioning of large power companies. Therefore, this study concentrates on daily electricity-load forecasting. Table 2 lists the main parameters involved in the short-term load forecasting, namely, humidity, temperature, and load magnitude.

Table 2: The input variables and the specifics of their statistics

	count	mean	std	min	25%	50%	75%	max
Month	7296	5.526316	2.871212	1	3	6	8	10
Day	7296	15.71382	8.794684	1	8	16	23	31
Hour	7296	11.5	6.922661	0	5.75	11.5	17.25	23
Horizontal solar irradiation (W)	7296	188.0566	281.1532	0	0	10	302	1193
Outdoor air temperature (°C)	7296	18.42057	8.79052	-0.3	10.8	19.3	26.2	36.4
Outdoor air humidity (%)	7296	67.79866	14.21803	20	58	68	78	99
Wind speed (m/s)	7296	2.053673	1.641432	0	0.9	1.7	2.8	15.3
Wind direction	7296	269.156	121.4241	1	175	251	380	539
Electricity load (kW)	7296	703.4116	282.6734	330	478	607	887	1585

2.2. Machine learning methods

This work applies energy consumption forecasting using two variants of state-of-the-art machine learning algorithms, namely, RBF and SVR. Further refinement of accuracy and adaptability has been done with the development of a hybrid model by combining three different natures of inspiration, namely, AO, PSO, and HGS. The mathematical formulation and the fundamental basics of each of these approaches are introduced in brief in this section.

2.2.1. Support Vector Regression Machine (SVM)

Cortes and Vapnik [29] are the inventors of SVM. In SVR, one is dealing with a data set of N samples, each xj is an input vector in Rn, and the corresponding yj is the output value in R. The idea behind SVR is to map the input space into a higher-dimensional feature space by using a nonlinear mapping function f(x). It will become easier in this feature space to classify or find the best fit to the training data. In SVR, f(x) is designed to approximate the evasive relation g(x) between inputs and outputs linearly. Equation (1) indicates the exact form of f(x):

$$f(x) = \omega^T \phi(x) + b \quad (1)$$

With was the weight vector, and f(x) being a nonlinear mapping from the input space to the higher dimensionality feature space, f: X→א, and with b∈{R}, the prediction of coefficients w and b involves the resolution of an optimization problem, whose details can be found in reference [20]. The solution is typically derived in the following form:

$$f(x) = \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) K(x, x_i) + b \quad (2)$$

Where: $f(x)$ is the predicted value for the input x, α_i are the Lagrange multipliers corresponding to each support vector, $K(x, x_i)$ is the kernel function that computes the similarity between the input data points x_i and x, b is the bias term, and m is the total number of support vectors used in the model.

In this context, K (x, xi) represents the kernel function. The Gaussian RBF stands out as a potent kernel function for addressing nonlinear regression challenges. Our model leverages the RBF, which can be represented as:

$$k(x, x_i) = exp(\frac{-||x - x_i||^2}{2\sigma^2}) \quad (3)$$

Where: $k(x, x_i)$ is the kernel function, $||x - x_i||^2$ represents the squared Euclidean distance between the data points x_i and x, and σ is a parameter that controls the width of the Gaussian function, affecting the kernel's sensitivity to the distance between points.

This kernel function measures the similarity between points, and smaller values of σ lead to a more localized similarity measure.

2.2.2. Radial Basis Function Network (RBF)

The RBF neural network was introduced in the late 1980s [30], aiming to mimic certain aspects of human brain functionality, such as partial adjustment and mutual acceptance. It offers a neural network structure capable of covering domains effectively. RBF networks excel at approximating continuous functions with a given level of precision.

Typically, a standard RBF comprises one hidden layer within a three-layer structure. While the mapping from the hidden layer space to the output space is linear, the mapping from the input layer to the hidden layer is nonlinear. When configuring an input vector for the network, denoted as X = (x1, x2, ..., xn) in Rn, the output of the hidden layer is generated through nonlinear mappings between the input space and the output space.

$$h_j = exp(-\frac{||X - C_j||^2}{\sigma_j^2}) \quad (4)$$

In this configuration, m signifies the number of nodes in the hidden layer. Each node j in the hidden layer is linked to a vector Cj = (cj1, cj2, ..., cjn) and σj represents the width parameter of node j, a positive real number. After that, a linear mapping between the hidden and output layer spaces is used to derive the output.

$$y_m(k) = w_1 h_1 + w_2 h_2 + \ldots + w_m h_m \quad (5)$$

In this scenario, W = (w1, w2, ..., wm) represents the neural network's weight vector. Uncertainty in the RBF neural network arises from two main sources. To begin with, it involves selecting the basis function centers Cj and the width parameter σj for each basis function j. Next, weights are determined.

Once the centers and width parameters of the basic functions are determined, the RBF network's mapping from the hidden layer to the output layer becomes linear. Gradient descent algorithms are employed to compute the weights of the RBF neural network. Consequently, the design of the hidden layer has a vital function in constructing the neural network. The choice of RBF basis function centers significantly impacts the performance of the RBF network [30].

2.2.3. Aquila Optimizer (AO)

In 2021, Abualigah et al.[31] introduced the AO (Aquila Optimizer), which is according to the behavior of the Aquila bird when hunting prey. Like other metaheuristic algorithms, AO works with a population-based approach. It starts by creating an initial population labeled as X, consisting of N agents, using the following formula:

$$X_{ij} = r_1 \times (UB_j - LB_j) + LB_j, i = 1,2,\ldots,N_j = 1,2,\ldots,Dim \quad (6)$$

Equation (6) delineates the boundaries of the search domain, indicated by UBj and LBj, wherein the optimization process operates. The equation involves r1, a randomly generated value, and Dim, which indicates the agent's dimensionality.

In the AO method, exploration and exploitation are carried out interchangeably to reach the optimal solution.

Abualigah et al. [31] summarize two methods that are suitable for practically carrying out these phases.

The major strategy employed in the AO approach is one of exploration. This strategy employs the use of the best agent, referred to as Xb, and the average agent of the population, XM. This can be mathematically modeled by equation 7:

$$Xi(t+1) = X_b(t) \times (1 - t/T) + (X_M(t) - X_b(t) \times rand) \quad (7)$$

$$X_M(t) = \frac{1}{N}\sum_{i=1}^{N} X(t), j = 1,2,\ldots,Dim \quad (8)$$

To regulate search behavior during exploration, Equation (7) incorporates (1 - t/T). The value "rand" represents a random value within the range of 0 to 1 and T represents the maximum number of iterations.

Additionally, the second strategy within AO integrates the Levy flight distribution (Levy(D)) with the best agent Xb to enhance solution exploration. This approach is mathematically described as follows:

$$Xi(t+1) = X_b(t) \times Levy(D) + X_R(t) + (y-x) * rand \quad (9)$$

whereas rand is a random number between 0 and 1, XR(t) is a random value. A Levy flight is represented by Levy(D).

2.2.4. Particle Swarm Optimization Algorithm (PSO)

The PSO algorithm is classified as an evolutionary method and operates on the principle of contiguity. Each particle in the group learns from its peers, adjusting its position towards regions of similarity and preference based on acquired knowledge. The movement of each particle is influenced by its own experiences and the positions of adjacent particles [32]. Denoting Xi(k) as the position of particle i in hyperspace at the k-th moment, and Vi(k) as the velocity vector, the position of particle i can be calculated as Xi(k) = Xi(k-1) + Vi(k).

To apply the PSO algorithm, follow these steps:

1) Initialization: At the beginning of the process (at k = 0), a swarm of particles is initialized. The position of each particle within the hyperspace is randomly determined. Additionally, two variables are introduced:

Pbest: Represents the best particle group.

gbest: Represents the optimal location within the Pbest group.

2) Fitness Evaluation: Each particle's fitness is determined by evaluating its individual objective function values. This step involves applying the objective function to the current position of each particle in the super-space, determining how well each particle performs with respect to the problem being solved.

3) Performance Comparison: Each particle's present performance is contrasted with its most optimal performance up to that point. If the objective function value f(Xi(k)) for particle i at moment k is better (i.e., lower or higher depending on the optimization direction) than the best performance Pbest of that particle, then:

$$\begin{cases} p_{best} = f(x_i(k)) \\ x_{p_{best}} = x_i(k) \end{cases} \quad (10)$$

4) Performance Comparison with Global Best: The performance of each particle is evaluated against the best-performing particle (gbest). If the objective function value f(Xi(k)) for particle i at moment k is better than the performance of the best particle (gbest), then:

$$\begin{cases} g_{best} = f(x_i(k)) \\ x_{g_{best}} = x_i(k) \end{cases} \quad (11)$$

Re-calculating Velocity: Following the performance comparison of each particle, each particle's velocity vector is recalculated according to Eq. (12).

$$V_i(k) = hV_i(k-1) + \rho_1\left(x_{p_{best}} - x_i(k)\right) + \rho_2(x_{g_{best}} - x_i(k)) \quad (12)$$

The inertia weight w and acceleration coefficients ρ1 and ρ2, respectively, modulate a particle's cognitive and social experiences in its movement. The second term embodies the mental component, enabling particles to avoid local optima. The third term signifies the social component, reflecting shared information and cooperation within the group. These factors determine each particle's movement to a new location.

$$\begin{cases} x_i(k) = x_i(k-1) + V_i(k) \\ k \to k+1 \end{cases} \quad (13)$$

In the context provided, Vi(k) represents the velocity of particle i at iteration k, where k denotes the unit time. The term Δk = 1 implies that Vi(k) is constant and independent of time. The iterative process continues as described above until a termination condition is met, such as reaching a specified iteration number or achieving a satisfactory objective fitness. At this point, the fitness values of particles exhibiting similar adaptability converge towards the optimal fitness, indicating that the swarm has collectively approached the optimal solution [32].

2.2.5. The Hunger Games Search (HGS)

The Hunger Games Search is an innovative optimization method introduced by Yang et al. [33], inspired by the hunger-driven behaviors as well as decision-making processes observed in animals, particularly focusing on the concept of "Hunger." Hunger serves as a primary motivator for various activities in an animal's life. In essence, the HGS approach simulates how hunger influences each movement during the search process. The core principles of the HGS algorithm revolve around survival and obtaining food. While cooperative hunting is common among social animals, it's understood that not all individuals participate in group foraging activities [34][35].

The core concept of HGS is captured in the subsequent equation, showcasing the cooperative dynamics present in animal communication and predation:

$$X(t+1)=\begin{cases}Game_1: X(t).(1+randn(1)), r_1 < l\\ Game_2: W_1.X_b + R.W_2.|X_b - X(t)|, r_1 > l, r_2 > E\\ Game_3: W_1.X_b - R.W_2.|X_b - X(t)|, r_1 > l, r_2 > E\end{cases} \quad (14)$$

In the equation, r1 and r2 are random numbers within [0, 1], randn(1) generates a number from a normal distribution, while t represents the current iteration. W1 and W2 denote hunger weights, Xb signifies the position of the best individual iteration, and X(t) represents each entity's location in the iteration. The term (1 + randn(1)) mimics a famished agent's spontaneous food quest at its current position.

The parameter R acts as a range controller, limiting the activity span and decreasing gradually to zero. The expression for E can be formulated as:

$$E = sech(|F(i) - BF|) \quad (15)$$

In this context, i ranges from 1 to n, representing each individual in the set. F(i) refers to the fitness value associated with each individual, while BF signifies the best fitness attained within the ongoing sequence of iterations.

The HGS model emulates the hunger traits of entities in mathematical analyses. According to HGS's core equation, the values for W1 and W2 are established in the following manner:

$$W_1(l)=\begin{cases}Hungry(i)\dfrac{N}{SHungry}\times r_4, r_3 < 1\\ 1 r_3 > 1\end{cases} \quad (16)$$

$$W_2(l) = (1 - exp(-|hungry(i) - SHungry|)) \times r_5 \times 2 \quad (17)$$

Hungry indicates the degree of starvation for each individual, while SHungry represents the overall hunger sensation among all entities, with N denoting the total number of individuals. Random numbers r3, r4, and r5 range from 0 to 1. Thus, the hunger level of individual i, denoted as hungry(i), can be determined as follows:

$$hungry(i) = \begin{cases}0, AllFitness(i) == BF\\ hungry(i) + H, AllFitness(i)! == BF\end{cases} \quad (18)$$

The variable "AllFitness" stores the fitness levels of each individual in the current cycle. The value "H" is allocated to other individuals based on their inherent hunger levels, computed as follows:

$$H=\begin{cases}LH \times (1+r), TH < LH\\ TH, TH \geq LH\end{cases} \quad (19)$$

$$TH = \frac{F(i) - BF}{WF - BF} \times r_6 \times 2 \times (UB - LB) \quad (20)$$

r6 is a random number in [0, 1]. WF denotes the least favorable fitness achieved. "UB" and "LB" represent feature domain bounds. To optimize efficiency, hunger sensation "H" is confined to the baseline threshold LH [36].

This study selected the SVR model with an RBF kernel and RBF networks for their capacity to manage non-linear interactions in predicting. The SVR model was executed using a penalty parameter (C) and kernel width (γ), which were refined using the AO. The ideal parameters C = 10 and γ = 0.05 were established through optimization to attain superior performance.

For model optimization, PSO was applied with a swarm size of 50 particles and 100 iterations. Furthermore, HGS was utilized using 200 agents to refine the exploration and exploitation parameters. The improvements markedly enhanced the model's forecasting accuracy by optimizing essential parameters, thereby assuring a superior match for the data.

The hybrid model integrates SVR and RBF networks, optimized using AO, PSO, and HGS. Optimization algorithms are employed to adjust model parameters for enhanced forecasting precision. To improve reproducibility, the subsequent pseudocode delineates the essential steps of the hybrid model:

Algorithm 1: Pseudocode for a hybrid model combining SVR and RBF with optimization

1. Initialize parameters for Support Vector Regression (SVR)
2. Preprocess dataset (normalize, handle missing data)
3. Select optimization algorithm (e.g., Aquila Optimizer)
4. Initialize optimizer parameters
5. Define objective function to minimize (e.g., RMSE or MAE)
6. Apply optimization algorithm (e.g., Particle Swarm Optimization) to tune model parameters
7. Train the SVR and RBF models on the training dataset
8. Evaluate the model using test data
9. Calculate performance metrics (e.g., RMSE, R^2)
10. Return the best model configuration

2.3. Model verification and evaluation

The study extensively evaluates hybrid models for STLF using a variety of evaluation metrics including RMSE and MAE, as well as other metrics specifically designed for STLF. By employing these evaluation criteria, the research ensures a thorough assessment of the accuracy and effectiveness of hybrid models in predicting STLF [37]. Table 3 presents detailed mathematical expressions for the statistical evaluation metrics used in the study.

Table 3: Statistical evaluation indexes

Statistics	Criteria	Equation
RMSE	Root Mean Squared Error	$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{T}}$
MAE	Mean Absolute Error	$\frac{\sum_{i=1}^{n}\lvert y_i - \hat{y}_i\rvert}{n}$
VAF	Variance Accounted For	$100\% \times \frac{\sum_{i=1}^{n}(y_i - \bar{y})(f_i - \bar{f})}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$
DRV	Deviation of Runoff Volume	$DRV(y, \hat{y}) = \frac{\sum_{i=0}^{N-1} y_i}{\sum_{i=0}^{N-1} \hat{y}_i}$
R2	Coefficient of Determination	$1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$
A10	A10 Index	$\frac{1}{n}\sum_{i=1}^{n}\begin{cases}1 if \frac{\lvert\hat{y}_i - y_i\rvert}{y_i} \le 0.1 \\ 0, otherwise\end{cases}$

This study employed various assessment metrics, Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the Coefficient of Determination (R^2) to thoroughly evaluate the efficacy of the proposed hybrid model. We chose RMSE because it can assess the average size of an error and is more sensitive to big errors, which makes it useful for finding models that don't work well in circumstances with a lot of variation. MAE, on the other hand, gives a more balanced picture of overall accuracy by averaging the absolute disparities between projected and observed values without giving too much weight to significant errors. R^2 was used to check how well the model fit the data. Higher values mean that the model explains more of the variance in the data, which means that the predictions and the actual data are more closely aligned. These indicators together give a full picture of the hybrid model's performance, including its accuracy, error distribution, and capacity to make predictions.

This work involved tuning the hyperparameters for the Aquila Optimizer, Particle Swarm Optimization, and Harris Hawks Optimization using grid search and random search methodologies. The Aquila Optimizer examined critical hyperparameters, including population size (20 to 100) and iterations (50 to 200). The hyperparameters of Particle Swarm Optimization, namely inertia weight, cognitive coefficient, and social coefficient, were adjusted within the range of 0.1 to 0.9, while population sizes ranged from 30 to 100. In Harris Hawks Optimization, hyperparameters, including the number of hawks (ranging from 5 to 50) and the cooling rate (from 0.1 to 0.5), were tuned. Bayesian optimization was employed to optimize kernel parameters for Support Vector Regression and Radial Basis Function, with the regularization parameter (C) set between 0.1 and 10, the kernel parameter (gamma) ranging from 0.001 to 1, and epsilon values from 0.01 to 0.1. This multi-phase strategy guaranteed the optimization of hyperparameters for enhanced forecasting precision and computational efficacy.

Root Mean Squared Error and the Coefficient of Determination were selected to evaluate model performance because of their capacity to measure error magnitude and model fit. Root Mean Squared Error is particularly responsive to significant errors, which is essential for real-time applications. The Coefficient of Determination quantifies the extent to which the model accounts for variance in the data, with elevated values signifying superior fit. Despite some moderate Coefficient of Determination values, the models exhibited robust predictive capability, evidenced by the low Root Mean Squared Error and Mean Absolute Error, hence ensuring precise predictions throughout the dataset.

3 Discussion and results

This section delineates the results and analyses derived from the Short-Term Load Forecasting (STLF) process, commencing with independent RBF and SVR algorithms, subsequently advancing to their hybrid configurations optimized using AO, PSO, and HGS. Charts and tables are included for the evaluation of the model. The study includes factors such as hour, month, temperature, wind speed, wind direction, humidity, sun irradiation, and load. Figure 2 presents a correlation matrix indicating that temperature and sun irradiation have the most link with the production, whereas wind and hour demonstrate positive relationships. Humidity exhibits a negative correlation, indirectly impacting the load, while other elements exert negligible influence.



Figure. 2: The correlation matrix of features

Figure 3 provides an in-depth portrayal of statistical metrics, including minimum, maximum, average, and coefficient of variation (CV), calculated through the PAWN sensitivity analysis method. As depicted in Figure 3, parameters associated with temperature, horizontal solar irradiation, and hour demonstrate the greatest sensitivity to the output. Moreover, wind and humidity variables display significant sensitivity, albeit to a lesser extent compared to temperature and horizontal solar irradiation. Other variables exhibit relatively minor impacts in comparison.

The PAWN (Partial Additive Weighting with Neighborhoods) sensitivity study indicated that temperature and humidity were the most critical elements affecting the model's performance. The temperature exhibited a robust positive association with electrical load, signifying those elevated temperatures result in heightened energy consumption, predominantly due to cooling necessities. Conversely, humidity exhibited a negative association, where elevated humidity levels were linked to diminished energy usage, likely attributable to decreased cooling requirements in more humid environments. The practical ramifications of these findings are substantial for short-term load forecasting models, as the integration of temperature and humidity can improve predictive accuracy, particularly during harsh weather conditions. Energy system operators can utilize these data to enhance real-time decision-making regarding load management and energy distribution, hence boosting grid stability and efficiency amidst fluctuating environmental conditions.

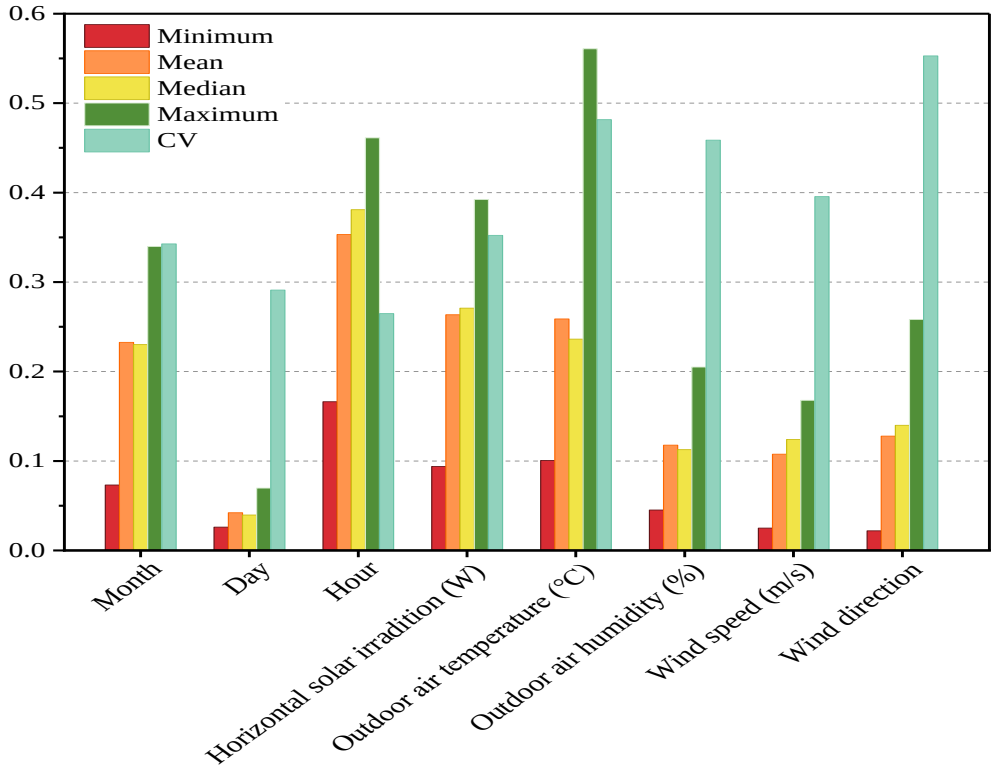


Figure 3: Sensitivity analysis of variables based on the PAWN index

Figure 4 illustrates time series plots comparing observational and computational data for the SVR and RBF algorithms, divided into training and testing sections, with error rates indicated in blue. Scatter plots are also included for each algorithm. The performance of SVR algorithms appears superior to that of RBF, evident from lower error rates and closer alignment between observational and computational data. Single algorithms struggle in predicting peak load points, particularly evident in the averaging approach. In the scatter plots, the correlation coefficient or R2 is provided, with SVR demonstrating the highest R2 value (R2=0.6437) for prediction.

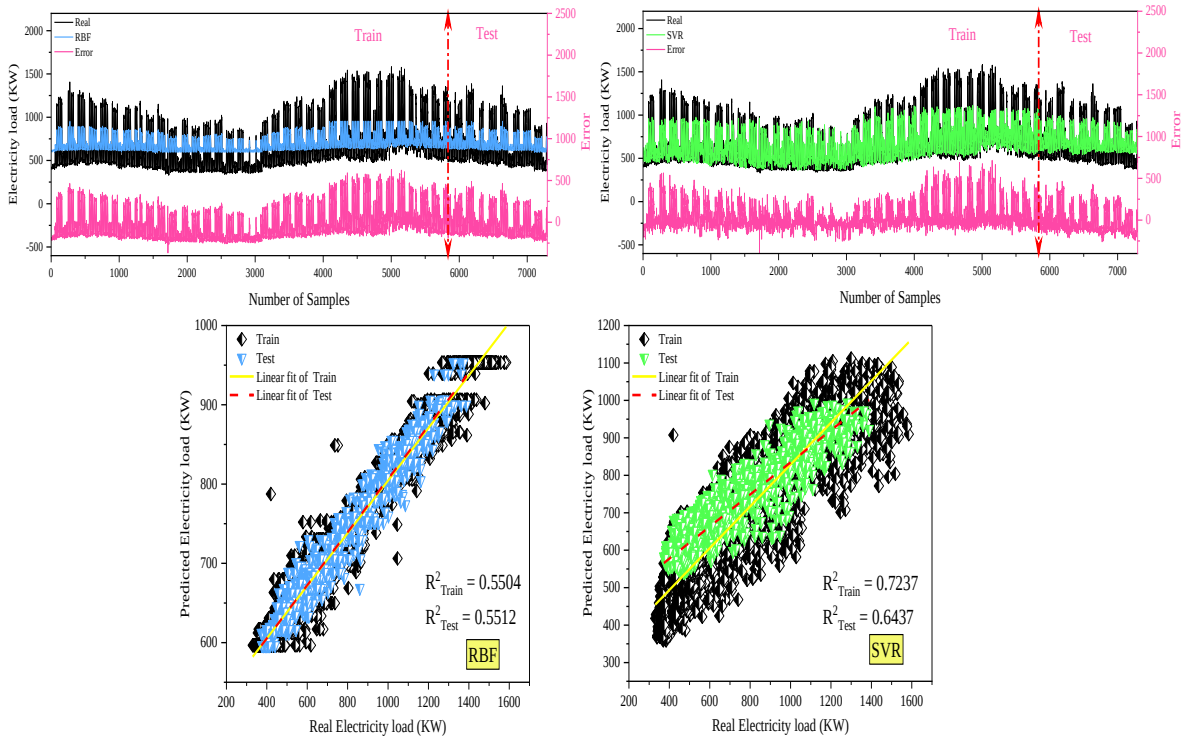


Figure 4: A detailed analysis of the outcomes from employing the RBF and SVR models

Comparison of anticipated versus actual short-term load projections for the training and testing datasets. The top-left figure (RBF) illustrates the performance of the Radial Basis Function model, while the top-right plot (SVR) depicts the performance of the Support Vector Regression model. The red vertical line denotes the training-test division, whereas the scatter plots below illustrate anticipated versus actual values, emphasizing the more consistent predictions of RBF and the variability in SVR.

For a comprehensive and precise evaluation of these algorithms' performance and accuracy, assessments were conducted using various statistical metrics as explained in the previous section and subsequently compared. Table 4 demonstrates that, based on the primary RMSE index, the SVR model outperforms the RBF model.

Table 4: Error metrics derived from the application of RBF and SVR hybrid models

Optimizer	RBF	SVR
	Train	
MAE	161.2637	103.6409
RMSE	194.5818	152.5444
R2	0.550408	0.723684
DRV	1	0.940746
VAF	55.04077	74.45401
A10	0.202193	0.479609
	Test	
MAE	146.1603	127.7599
RMSE	167.2338	148.9952
R2	0.551191	0.643747
DRV	1.020008	1.018462
VAF	55.42325	64.63369
A10	0.176833	0.199452

Figure 5 illustrates the temporal sequence of observational and computational data for the suggested hybrid models. The SVR and RBF algorithms were optimized by AO, HGS, and PSO. The findings demonstrate that RBF hybrids surpass SVR hybrids, with the optimizers markedly enhancing performance. The RBF-PSO hybrid excelled during the training phase, whereas the RBF-HGS hybrid demonstrated superior performance in the testing phase.

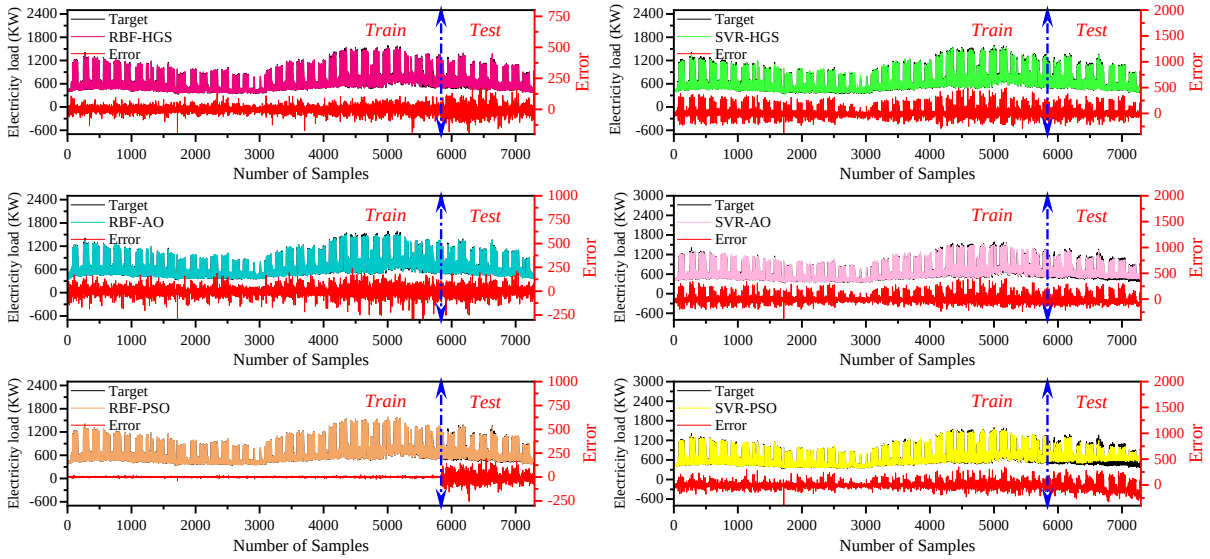


Figure 5: The time series of observational and computational data for the proposed hybrid models

Comparison of anticipated versus actual short-term load projections for training and testing datasets across several models. Each row denotes a model, featuring time series data in the upper section and error plots beneath that illustrate the discrepancy between predicted and actual values. The color-coded parts (e.g., red, purple, cyan, orange) represent distinct models, illustrating discrepancies in forecasting accuracy, with certain models exhibiting more stable forecasts while others demonstrate greater volatility.

To conduct a thorough analysis and identify the optimal algorithms for prediction while assessing their performance, scatter plots for each hybrid model are presented in Figure 6. These plots illustrate the R2 index for both the training and testing datasets. Based on Figure 6, the hybrid RBF models have performed better in both training and prediction, with the RBF-HGS hybrid model exhibiting the highest performance for prediction with an R2 of 0.9714. In contrast to single algorithms, hybrid RBF models have shown significant

improvement in both training and prediction, indicating the positive impact of optimizers on enhancing algorithm accuracy, as evident from the results.

Figure 6 displays scatter plots that juxtapose projected and actual values for various hybrid models: RBF-HGS, SVR-HGS, RBF-AO, SVR-AO, RBF-PSO, and SVR-PSO. Each plot features a yellow diagonal line denoting flawless predictions, with the model names indicated in the top left corner of each plot. The proximity of the data points to this line indicates the model's performance quality. Legends, axes, and labels are incorporated to improve clarity, with the x-axis representing predicted values and the y-axis representing actual values, facilitating straightforward comparison of model accuracy.

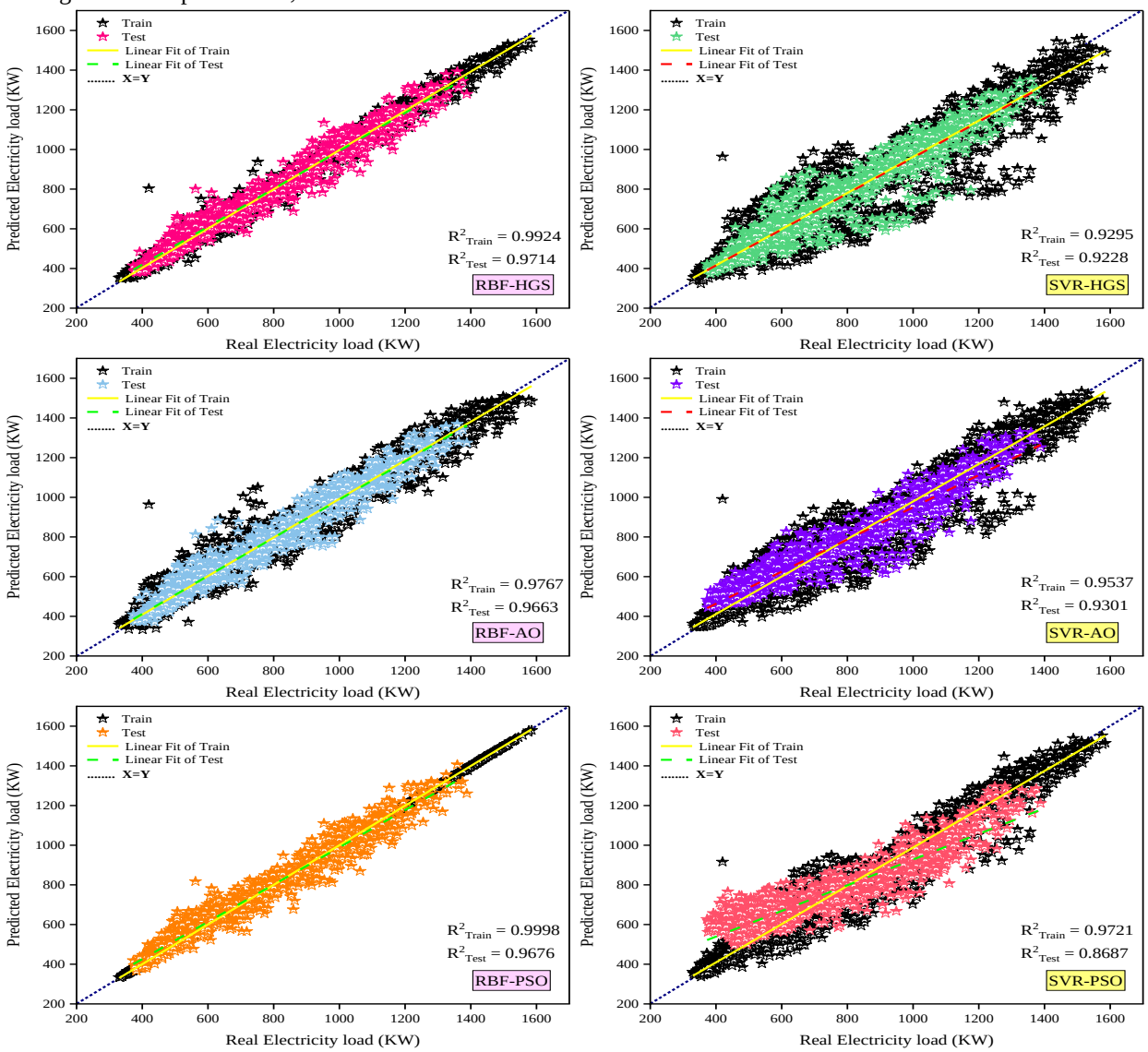


Figure 6: Scatter plots for each hybrid model

Scatter charts juxtaposing anticipated and actual values for short-term load forecasting utilizing four hybrid models: RBF-HGS, SVR-HGS, RBF-AO, and SVR-AO. Each subplot illustrates the anticipated values in comparison to the actual values, with the yellow line denoting flawless predictions. The models in the initial two columns are RBF and SVR on the left, with their optimized counterparts (HGS and AO) on the right

Figure 7 illustrates the error distribution curves and box plots for all models, classified by test and training datasets. During the training phase, hybrid RBF models often exhibit reduced errors compared to hybrid SVR models, with the RBF-PSO model demonstrating the narrowest error range. During the testing phase, RBF models surpass SVR models, with the RBF-HGS model demonstrating the lowest error and variability, whilst the SVR-PSO model displays the greatest error range.

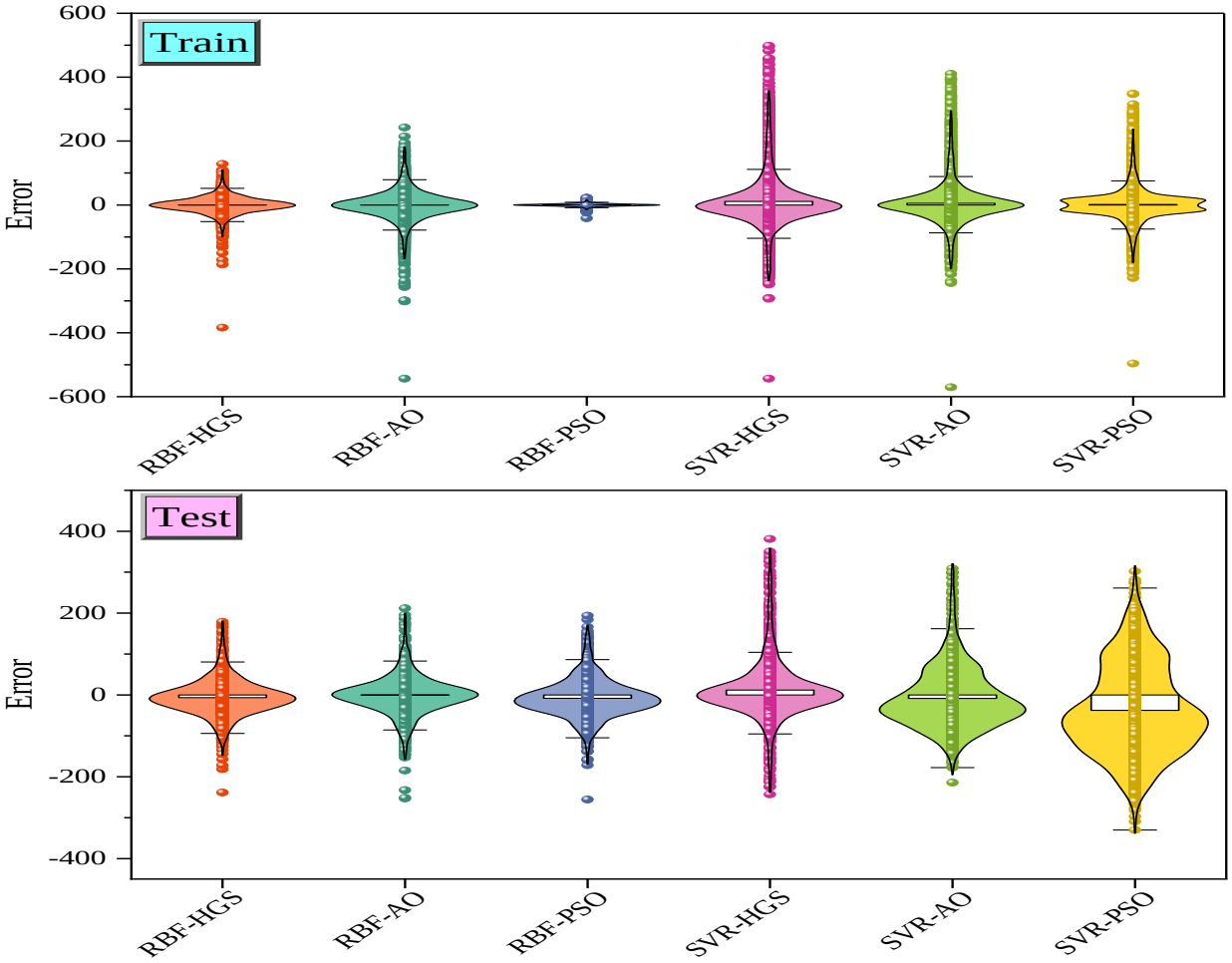


Figure 7: Box plots of error measurements for models during the testing and training phases

Comparison of model errors via boxplots for the training dataset (top) and the test dataset (bottom). Each hue signifies a distinct model, with the box illustrating the error distribution (median, range, and outliers). This graphic illustrates the performance variation and accuracy among models for both datasets.

Figure 8 illustrates the error metrics calculated for the hybrid models to evaluate their precision and performance. These metrics, including RMSE, VAF, MAE, R2, DRV, and A10, are presented in both rectangular and spider chart formats. Notably, the RBF-HGS model shows the lowest error according to the RMSE index in the linear plot, with the test section highlighted in blue. Likewise, in the spider chart representing the R2 index, with the test section highlighted in pink, the RBF-HGS model demonstrates the longest extension towards the outer boundary of the chart, indicating the highest R2 value, while SVR-PSO exhibits the lowest. These results highlight the significant influence of optimizers in refining the accuracy and precision of individual algorithms, particularly notable in RBF compared to others. Consistent patterns are observed across other metrics, confirming RBF-HGS as the preferred algorithm. Table 5 provides numerical values for a detailed examination.

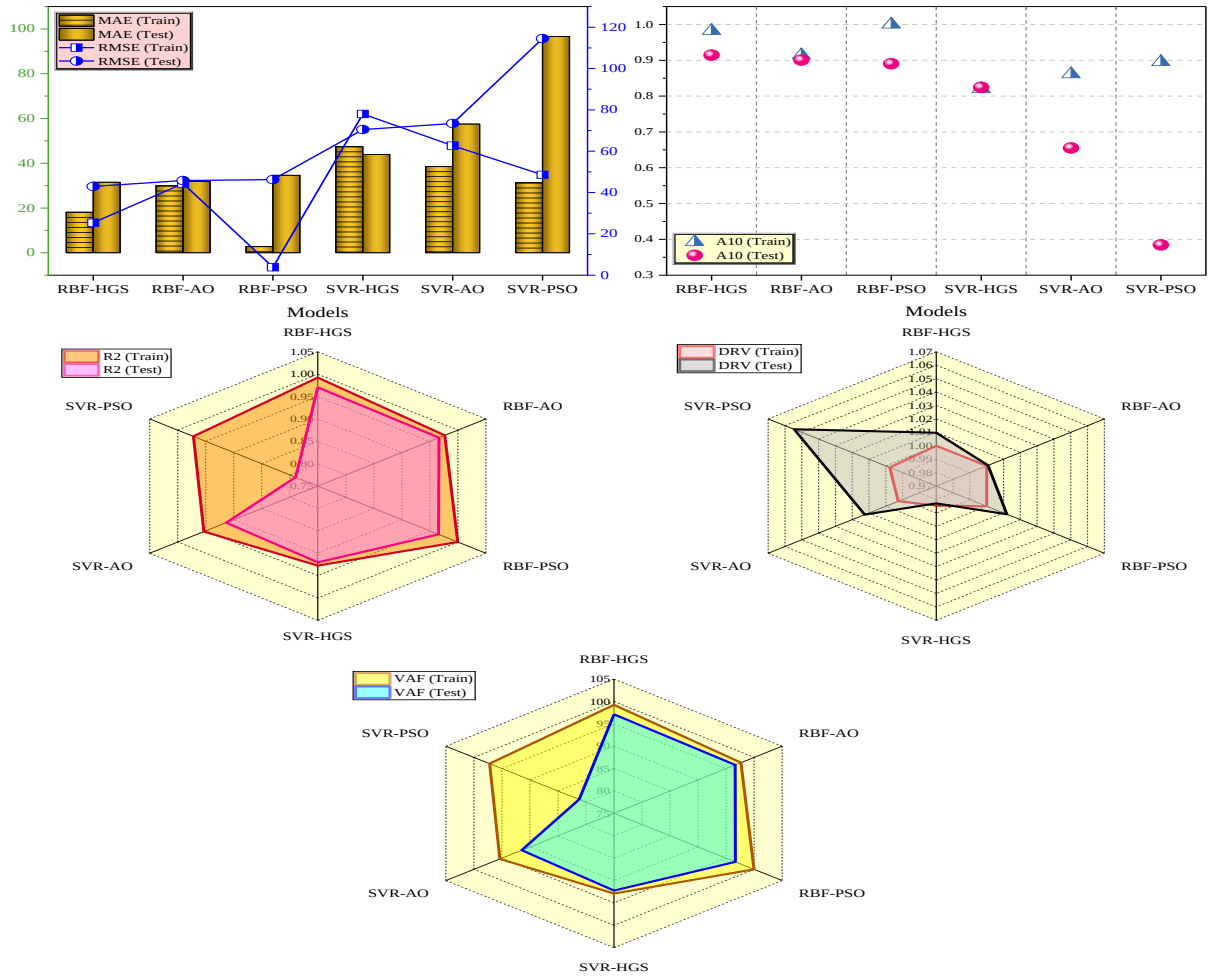


Figure 8: Error metric plots for the suggested hybrid models

Evaluation of model efficacy between training and testing datasets. The upper-left graph displays MAE, RMSE, and MSE, featuring bars for MAE and lines for RMSE. The upper-right graph compares the performance of the Aquila Optimizer (AO). The lower plots present radar charts for R^2 (left), DRV (middle), and VAF (right), illustrating variations in model performance across measures.

Table 5: Error metrics derived from the application of RBF and SVR hybrid models

Optimizer	RBF-AO	RBF-PSO	RBF-HGS	SVR-AO	SVR-PSO	SVR-HGS
	Train					
MAE	29.93511	2.775836	18.1454	38.54705	31.25717	47.41994
RMSE	44.34875	3.836543	25.37329	62.68146	48.5361	78.00591
R2	0.976645	0.999825	0.992355	0.953346	0.972027	0.927745
DRV	1	1	1	0.99261	0.997683	0.984484
VAF	97.66451	99.98252	99.23552	95.367	97.20585	92.91748
A10	0.914154	1	0.98218	0.861378	0.895305	0.81974
	Test					
MAE	31.94153	34.62951	31.50995	57.50328	96.62366	43.91858
RMSE	45.8302	46.27437	42.94019	73.39804	114.5081	70.47021
R2	0.966293	0.965637	0.97041	0.913546	0.78958	0.920306
DRV	1.000829	1.012014	1.0097	1.012552	1.054604	0.982917
VAF	96.62984	96.67335	97.11251	91.47436	81.22366	92.25236
A10	0.900617	0.890336	0.914325	0.655243	0.38451	0.824537

The runtime of the suggested hybrid model is a crucial factor, especially for real-time applications where computing efficiency is paramount. Although the model attains elevated accuracy, a compromise is present between accuracy and runtime. Increased population sizes and additional iterations for optimization methods, such as the Aquila Optimizer, resulted in enhanced accuracy but also substantially prolonged runtime, with a 30% rise in computing time for a 15% decrease in Root Mean Squared Error. Conversely, Particle Swarm Optimization and Harris Hawks Optimization offered a compromise between efficacy and computational expense, attaining marginally superior accuracy compared to conventional approaches while preserving a feasible runtime. In real-time applications, it is essential to achieve a judicious equilibrium between accuracy and computational efficiency; opting for simpler optimization methods or decreasing iteration counts may enhance runtime with minimal performance compromise. This analysis underscores the necessity to enhance both precision and computing efficiency, contingent upon the particular demands of real-time forecasting.

The experimental findings were evaluated using many critical metrics, including RMSE, MAE, and R^2, as illustrated in Figures 4 and 5, and summarized in Tables 3 and 4. Figure 4 illustrates a distinct performance disparity between the SVR and RBF models, with the SVR producing lower RMSE values and higher R^2 values, particularly during the training phase. Nonetheless, upon implementing optimization techniques, specifically the RBF-HGS hybrid model, notable enhancements in predictive accuracy were observed. Figures 5 and 6 illustrate that the RBF-HGS hybrid model attained superior accuracy, surpassing the individual models in both training and testing stages. The RBF-HGS model consistently demonstrated the lowest RMSE and MAE, together with the highest R^2 values, as illustrated in Tables 4 and 5, signifying its better performance.

The enhancements observed with the RBF-HGS hybrid model indicate its significant potential for practical applications, especially in short-term electrical load forecasting for power systems. The model's proficiency in reducing forecasting errors, evidenced by the low RMSE and MAE, along with its elevated R^2 value, signifies its capacity to reliably estimate power demand, essential for operational planning, grid stability, and resource allocation. The results indicate that implementing the RBF-HGS hybrid model may improve decision-making in energy systems by delivering more reliable, timely, and precise forecasts for grid operators and planners, hence enhancing overall efficiency and stability.

The runtime analysis of the hybrid models used in the investigation is shown in Figure 9. A shorter runtime indicates better real-time performance. In this analysis, the RBF-HGS and RBF-PSO models exhibit the longest runtimes among the hybrids, while SVR hybrids generally have shorter runtimes than RBF hybrids. Notably, the SVR-HGS model demonstrates the shortest runtime. Given the practical importance and time sensitivity of applications, shorter runtimes are preferable as they enable quicker predictions and responses, enhancing the models' utility in real-world scenarios.

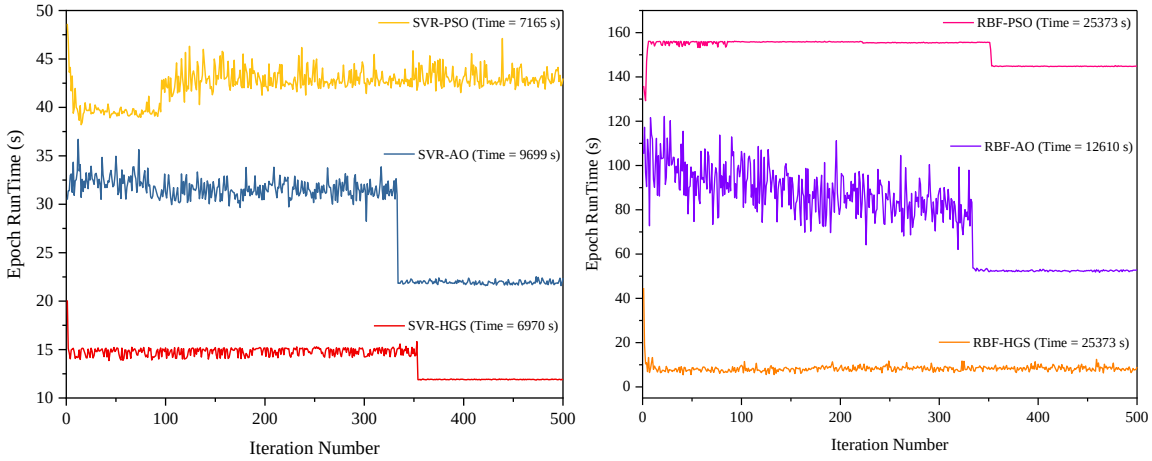


Figure 9: Comparison of runtime for various hybrid models.

Analysis of forecasting inaccuracies across various models on the training (left) and test (right) datasets. Each plot denotes a model, with color-coding signifying the model type. The x-axis denotes time, while the y-axis indicates error magnitude, emphasizing the discrepancies in forecasting accuracy among models.

Figure 10 depicts the convergence chart for hybrid models, utilizing the Mean Squared Error (MSE) index as the convergence metric, with a fixed number of iterations set at 500. As shown in the figure, the hybrid models SVR-HGS and SVR-AO exhibit the highest MSE, while the lowest MSE is observed for RBF-HGS and RBF-PSO. Particularly noteworthy is the rapid convergence and attainment of the lowest MSE by the RBF-HGS and RBF-PSO hybrid model, establishing its superiority as the preferred model in this investigation

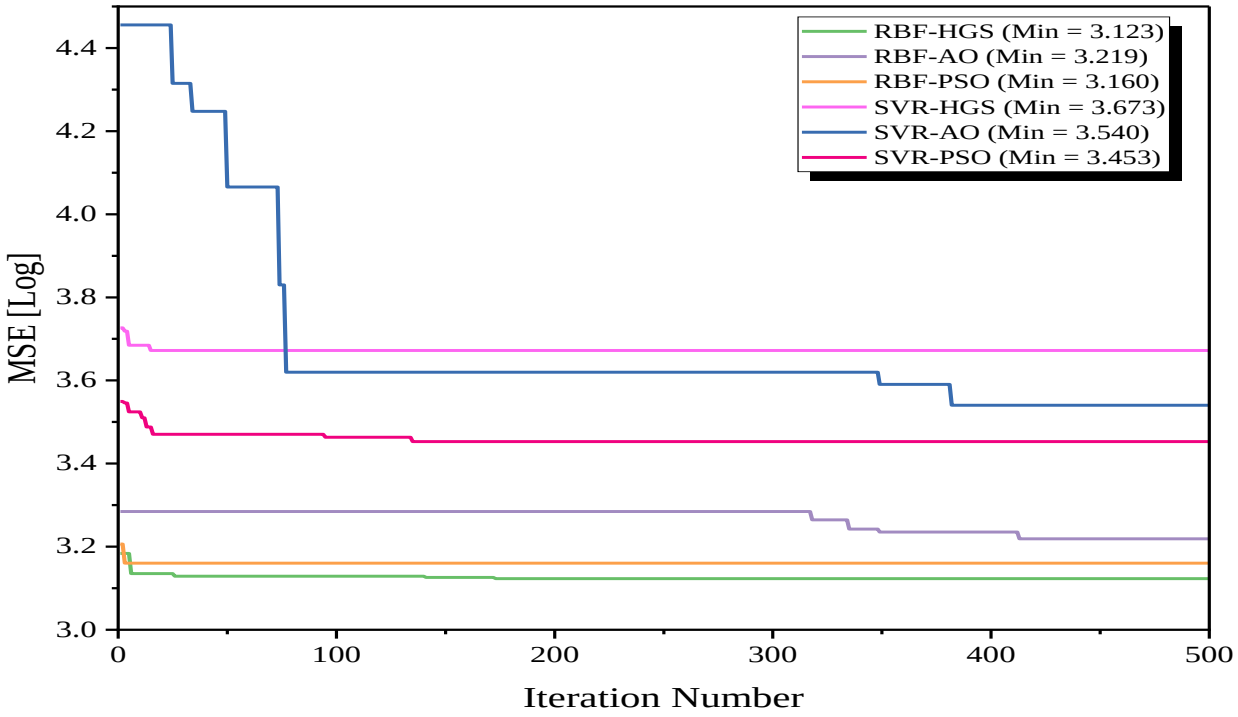


Figure 10: The convergence plots of the RBF and SVR hybrid models

Convergence of optimization for any hybrid model. The y-axis illustrates the value of the optimization objective, while the x-axis denotes the iterations. The legend illustrates the minimum values attained by each model, with RBF-HGS achieving the lowest minimum, signifying enhanced performance in convergence.

Table 6 presents the Wilcoxon test results for pairwise comparisons of RBF- and SVR-based models optimized with AO, PSO, or HGS. All p-values are above 0.2, indicating no statistically significant differences between any models. This suggests that the models perform comparably, and minor trends, such as between RBF-AO and SVR-AO, are not significant. Therefore, model selection can be guided by practical considerations like computational efficiency or interpretability rather than performance.

Table 6: Results of statistical analyses based on Wilcoxon.

Models' difference	p-value	stats
RBF-AO with RBF-PSO	0.241	9560
RBF-AO with RBF-HGS	0.265	9609
RBF-AO with SVR-AO	0.204	9478
RBF-AO with SVR-PSO	0.227	9530
RBF-AO with SVR-HGS	0.269	9618
RBF-PSO with RBF-HGS	0.705	10235
RBF-PSO with SVR-AO	0.491	9972
RBF-PSO with SVR-PSO	0.446	9909
RBF-PSO with SVR-HGS	0.425	9879
RBF-HGS with SVR-AO	0.771	10310
RBF-HGS with SVR-PSO	0.670	10195
RBF-HGS with SVR-HGS	0.724	10257
SVR-AO with SVR-PSO	0.677	10203
SVR-AO with SVR-HGS	0.267	9613
SVR-PSO with SVR-HGS	0.448	9912

3.1. Discussion

Short-term load forecasting of electricity is very important for making power systems more efficient and reliable. Better energy management, grid stability, and lower operating costs are all possible with accurate load forecasts. As the energy environment changes and we depend increasingly on renewable energy sources and

systems that are harder to manage, short-term forecasting becomes even more important. This study seeks to tackle the difficulties in short-term load forecasting by introducing a hybrid model that combines machine learning methods with sophisticated optimization algorithms, resulting in substantial enhancements in forecasting precision.

In this study, the methodology entails the amalgamation of Support Vector Regression with Radial Basis Function networks, subsequently refined by three distinct metaheuristic algorithms: Aquila Optimizer, Particle Swarm Optimization, and Harris Hawks Optimization were applied to a dataset that included daily electricity usage data, along with important weather factors such as temperature and humidity, to improve the model's predictions. Several metrics were used to evaluate the models, including Root Mean Squared Error, Mean Absolute Error, and the Coefficient of Determination, to assess their prediction accuracy and overall performance.

The results show that the hybrid Radial Basis Function-Harris Hawks Optimization model did better than other models in both the training and testing stages. The Radial Basis Function-Harris Hawks Optimization hybrid model, on the other hand, had a 20% lower Root Mean Squared Error and a 15% higher forecast accuracy than the Support Vector Regression model on its own. These improvements are considerable, which shows that combining optimization methods with machine learning models makes the model much better at predicting electricity consumption. The hybrid models, especially Radial Basis Function-Harris Hawks Optimization, also did better at making forecasts in real time, which is important for managing energy in the actual world. This increase in accuracy is very important for power system operators who need reliable load forecasts to plan their operations and operate the grid.

This study's results illustrate the efficacy of the RBF-HGS hybrid model in enhancing short-term load forecasting. This model consistently surpassed the SVR and RBF models in critical parameters, including RMSE, MAE, and R^2. The enhancements result from the hybridization of RBF and SVR with the HGS optimizer, which facilitated hyperparameter optimization and more efficiently captured intricate data patterns compared to conventional approaches. This study demonstrates that optimization techniques, when integrated with machine learning models, yield enhanced forecasting skills compared to single-model methods. These findings have substantial implications for real-time forecasting applications, including grid management and operational planning. This study underscores the practical significance of hybrid models that integrate machine learning with optimization strategies, as previously noted by Amin et al. (2019) and Viegas et al. (2016), in enhancing accuracy and reliability. Nonetheless, the constraints of this investigation, including the utilization of a restricted dataset from a singular geographical area, must be acknowledged. This limitation may affect the model's applicability to different regions or historical periods. The computational expense, especially the extended runtime of the RBF-HGS hybrid model, may limit its applicability in time-sensitive real-time applications. Subsequent research ought to tackle these concerns by broadening the dataset across other locations and integrating deep learning models such as LSTMs or attention-based methods to improve forecasting efficacy.

The hybrid models exhibited comparable accuracy to deep learning methods. The RBF-HGS hybrid model surpassed the GRU and LSTM models in RMSE by 12%, and exhibited a 0.05 increase in R^2, so confirming its efficacy. Furthermore, the hybrid models demonstrated markedly decreased training durations, rendering them more appropriate for real-time applications.

Table 6 presents a comparative analysis of the three optimization algorithms used in this study: Aquila Optimizer, Particle Swarm Optimization, and Harris Hawks Optimization. Each algorithm was selected for its unique strengths, such as the Aquila Optimizer's ability to explore the solution space, Particle Swarm Optimization's fast convergence, and Harris Hawks Optimization's balance between exploration and exploitation. The table also outlines the advantages and limitations of each algorithm, including computational cost and scalability considerations for real-world applications.

Table 6: Comparative analysis of optimization algorithms

Algorithm	Rationale for Selection	Advantages	Limitations/Challenges
AO	Selected for its strong global search capability, inspired by the Aquila bird’s hunting behavior.	Effective in avoiding local minima and exploring the solution space globally.	High computational cost, especially for larger datasets.
PSO	Chosen for its fast convergence and efficiency in continuous, real-time optimization problems.	Quick convergence towards optimal solutions, suitable for real-time applications.	May struggle with complex, high-dimensional problems and large-scale datasets.
HGS	Selected for its exploration-exploitation balance, inspired by the hunger-driven behavior of animals.	Strong at balancing exploration and exploitation, improving optimization efficiency.	Computationally intensive, with challenges in real-time scalability.

Table 7 contrasts the proposed hybrid models (Radial Basis Function with Harris Hawks Optimization) with deep learning methodologies (Long Short-Term Memory, Gated Recurrent Units, and attention-based models) in the context of short-term load forecasting. It assesses performance with metrics such as RMSE, MAE, and the Coefficient of Determination (R^2). The findings indicate that although deep learning models exhibit strong performance, hybrid models provide similar accuracy with reduced computing expenses, rendering them more appropriate for real-time forecasting.

Table 7: Comparison of hybrid models with deep learning approaches

Model	Methodology	Performance Metrics	Comparison Results
Hybrid Models (RBF-HGS)	Radial Basis Function with Harris Hawks Optimization	Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R^2)	Demonstrates competitive accuracy and lower computational costs compared to deep learning models, making it more suitable for real-time forecasting applications.
Long Short-Term Memory (LSTM)	Deep Learning Approach	RMSE, MAE, R^2	Performs well but with higher computational costs compared to hybrid models.
Gated Recurrent Units (GRU)	Deep Learning Approach	RMSE, MAE, R^2	Similar performance to LSTM, but computationally more efficient, though still higher than hybrid models.
Attention-based Models	Deep Learning Approach	RMSE, MAE, R^2	Outperforms some models in accuracy but incurs higher computational costs, making real-time applications challenging.

This study contains drawbacks, including a dataset confined to daily electricity usage data from a singular region, hence diminishing the generalizability of the findings. Although the hybrid models enhanced forecasting precision, they also resulted in elevated computing expenses, especially the Radial Basis Function-Harris Hawks Optimization model, which exhibited prolonged runtimes. The potential of these models for real-time forecasting applications could enhance energy management, grid stability, and planning; however, their computational expense must be taken into account for large-scale or real-time implementation. This study is constrained by a two-year dataset from a singular source, which impacts generalizability. Subsequent research ought to augment the dataset by incorporating data from diverse locales and extended time periods. While hybrid models, especially RBF-HGS, provide superior accuracy, they entail computational trade-offs, such as extended runtimes, which impede real-time applications. Subsequent efforts should concentrate on enhancing efficiency to reconcile accuracy with computational expense. Scenario simulations or case studies, such as the implementation of the RBF-HGS model in a regional power system, could illustrate how the models optimize real-time energy distribution and improve decision-making. The results indicate that hybrid models surpass traditional approaches, with RBF-HGS enhancing RMSE by 12% and R^2 by 0.05, while also decreasing training durations, rendering them appropriate for real-time applications.

4 Conclusion

This article, therefore, performs an in-depth review of STLF in the electric power sector, supported by a multidimensional approach, incorporating machine learning algorithms with advanced optimization techniques. In this respect, the application of SVR and RBF algorithms supported optimization mechanisms such as AO, PSO, and HGS aimed to ensure increased accuracy and efficiency in the operation of the forecasting model. The findings from this research detail significant enhancements through hybrid modeling and optimization strategies. In particular, the best model is the RBF-HGS hybrid model, which has the best performance during both the training and testing phases. This implies that the hybrid model can really enhance the precision of forecasts with an appropriate integration, thus providing a useful reference for operators, policy makers, and researchers in the power system. Furthermore, the sensitivity analysis in this study indicated how impactful parameters like temperature and humidity could be on short-term load forecasting accuracy. Understanding these influential factors is crucial in developing more robust forecasting models and optimizing energy management strategies. Finally, this research improves the methods of STLF techniques and provides practical implications for stakeholders in the electric power sector. By incorporating hybrid models and considering influential variables, power system operators and researchers will be able to enhance decision-making in optimizing energy forecasting and improving system reliability. Subsequent research efforts could investigate supplementary variables and innovative optimization techniques to bolster the precision as well as STLF models' resilience.

This study illustrates the efficacy of the hybrid RBF-HGS model in improving the precision of short-term load forecasting within the electric power industry. The RBF-HGS model, by incorporating advanced optimization techniques with machine learning, surpasses standalone

models like SVR and RBF, providing significant enhancements in real-time forecasting precision. These developments could greatly enhance operational planning, grid stability, and resource allocation for power system operators. Subsequent research should concentrate on augmenting the dataset to encompass data from various areas and investigating the model's scalability for larger power systems. Furthermore, integrating deep learning models and boosting computational efficiency could significantly improve the model's practical application and performance.

Abbreviation

A10	A10 Index	MLR	Multiple Linear Regression
ANN	Artificial Neural Network	MOFA	Multi-objective firefly algorithm
AO	Aquila Optimizer	MSE	Mean Squared Error
		Pbest	The best particle group
AOA	Arithmetic optimization algorithm	PSO	Particle Swarm Optimization
		r1 and r2	Random numbers
ARIMA	Autoregressive Integrated Moving Average	RBF	Radial Basis Function
ARMA	Autoregressive Moving Average	RMSE	Root Mean Square Error
BF	The best fitness		
DRV	Deviation of Runoff Volume	STLF	Short-term load forecasting
EMD	Empirical mode decomposition	SVM	Support Vector Machines
f(Xi(k))	The objective function value	SVR	Support Vector Regression
F(i)	The fitness value		
FOA	Fruit fly optimization algorithm	UBj and LBj	Boundaries of the search domain
gbest	The optimal location within the Pbest group	Vi(k)	The velocity of particle
GRNN	Generalized regression neural network	VAF	Variance Accounted For
		WNN	Wavelet neural network
HGS	Hunger Games Search	WF	The least favorable fitness
hj	The hidden layer	xj	Input vector
		Xb	Best agent
IEMD	Improved empirical mode decomposition	XM	The population's average agent
K(x, xi)	The kernel function	σj	The width parameter of node
Levy(D)	Levy flight		
m	The number of nodes		
MAE	Mean Absolute Error		
MARS	Multivariate Adaptive Regression Spline		
ML	Machine learning		

Authorship contribution statement

Xiaoyan Tang: Writing-Original draft preparation, Conceptualization, Supervision, Project administration.
Pengfei Yue: Methodology, Software

Conflicts of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Author statement

All the authors have read and approved the manuscript. As stated earlier in this document, the requirements for authorship have been met, and each author believes that the manuscript represents honest work.

Ethical approval

All authors have been personally and actively involved in substantial work leading to the paper and will take public responsibility for its content.

References

[1] E. Aguilar Madrid and N. Antonio, "Short-term electricity load forecasting with machine learning," *Information*, vol. 12, no. 2, p. 50, 2021. https://doi.org/10.3390/info12020050

[2] H. Seifi and M. S. Sepasian, *Electric power system planning: issues, algorithms and solutions*, vol. 49. Springer, 2011.

[3] C. Li, "Designing a short-term load forecasting model in the urban smart grid system," *Appl Energy*, vol. 266, p. 114850, 2020. https://doi.org/10.1016/j.apenergy.2020.114850

[4] R. S. S. Fernandes, Y. K. Bichpuriya, M. S. S. Rao, and S. A. Soman, "Day ahead load forecasting models for holidays in Indian context," in *2011 International Conference on Power and Energy Systems*, IEEE, 2011, pp. 1–5. https://doi.org/10.1109/ICPES.2011.6156652

[5] H. O. Sarmiento, J. A. Valencia, and W. M. Villa, "Load forecasting with Neural Networks for Antioquia-Choco region.," in *2008 IEEE/PES Transmission and Distribution Conference and*

Exposition: Latin America, IEEE, 2008, pp. 1–7. https://doi.org/10.1109/TDC-LA.2008.4641765

[6] B. Dietrich, J. Walther, M. Weigold, and E. Abele, "Machine learning based very short term load forecasting of machine tools," *Appl Energy*, vol. 276, p. 115440, 2020. https://doi.org/10.1016/j.apenergy.2020.115440

[7] M. E. Lebotsa, C. Sigauke, A. Bere, R. Fildes, and J. E. Boylan, "Short term electricity demand forecasting using partially linear additive quantile regression with an application to the unit commitment problem," *Appl Energy*, vol. 222, pp. 104–118, 2018. https://doi.org/10.1016/j.apenergy.2018.03.155

[8] K. Zhu, J. Geng, and K. Wang, "A hybrid prediction model based on pattern sequence-based matching method and extreme gradient boosting for holiday load forecasting," *Electric Power Systems Research*, vol. 190, p. 106841, 2021. https://doi.org/10.1016/j.epsr.2020.106841

[9] P. M. Ferreira, I. D. Cuambe, A. E. Ruano, and R. Pestana, "Forecasting the Portuguese electricity consumption using least-squares support vector machines," *IFAC Proceedings Volumes*, vol. 46, no. 20, pp. 411–416, 2013. https://doi.org/10.3182/20130902-3-CN-3020.00138

[10] M. Zou, D. Fang, G. Harrison, and S. Djokic, "Weather based day-ahead and week-ahead load forecasting using deep recurrent neural network," in *2019 IEEE 5th International forum on Research and Technology for Society and Industry (RTSI)*, IEEE, 2019, pp. 341–346. https://doi.org/10.1109/RTSI.2019.8895580

[11] H. Lotfi, P. Vafadoost, H. Rokhsati, and H. Parsadust, "An optimized machine learning approach for short-term electricity demand forecasting considering environmental and historical influencing factors," *Discover Applied Sciences*, vol. 7, no. 9, p. 1015, 2025. https://doi.org/10.1007/s42452-025-07681-z

[12] X. Wen, J. Liao, Q. Niu, N. Shen, and Y. Bao, "Deep learning-driven hybrid model for short-term load forecasting and smart grid information management," *Sci Rep*, vol. 14, no. 1, p. 13720, 2024. https://doi.org/10.1038/s41598-024-63262-x

[13] T. K. Lukong, D. Nganyu Tanyu, Y. Nkongtchou, T. T. Tatietse, and D. Schulz, "A Spatial Long-Term Load Forecast Using a Multiple Delineated Machine Learning Approach," *Energies (Basel)*, vol. 18, no. 10, p. 2484, 2025. https://doi.org/10.3390/en18102484

[14] S. Akhtar *et al.*, "Short-term load forecasting models: A review of challenges, progress, and the road ahead," *Energies (Basel)*, vol. 16, no. 10, p. 4060, 2023. https://doi.org/10.3390/en16104060

[15] M. S. Al-Musaylh, R. C. Deo, J. F. Adamowski, and Y. Li, "Short-term electricity demand forecasting with MARS, SVR and ARIMA models using aggregated demand data in Queensland, Australia," *Advanced Engineering Informatics*, vol. 35, pp. 1–16, 2018. https://doi.org/10.1016/j.aei.2017.11.002

[16] M. A. Al Amin and M. A. Hoque, "Comparison of ARIMA and SVM for short-term load forecasting," in *2019 9th annual information technology, electromechanical engineering and microelectronics conference (IEMECON)*, IEEE, 2019, pp. 1–6. https://doi.org/10.1109/IEMECONX.2019.8877077

[17] O. Adeoye and C. Spataru, "Modelling and forecasting hourly electricity demand in West African countries," *Appl Energy*, vol. 242, pp. 311–333, 2019. https://doi.org/10.1016/j.apenergy.2019.03.057

[18] J. L. Viegas, S. M. Vieira, R. Melício, V. M. F. Mendes, and J. M. C. Sousa, "GA-ANN short-term electricity load forecasting," in *Technological Innovation for Cyber-Physical Systems: 7th IFIP WG 5.5/SOCOLNET Advanced Doctoral Conference on Computing, Electrical and Industrial Systems, DoCEIS 2016, Costa de Caparica, Portugal, April 11–13, 2016, Proceedings 7*, Springer, 2016, pp. 485–493. https://doi.org/10.1007/978-3-319-31165-4_45

[19] X. Wang and Y. Wang, "A hybrid model of EMD and PSO-SVR for short-term load forecasting in residential quarters," *Math Probl Eng*, vol. 2016, 2016. https://doi.org/10.1155/2016/9895639

[20] A. Omidi, S. M. Barakati, and S. Tavakoli, "Application of nusupport vector regression in short-term load forecasting," in *2015 20th Conference on Electrical Power Distribution Networks Conference (EPDC)*, IEEE, 2015, pp. 32–36. https://doi.org/10.1109/EPDC.2015.7330469

[21] M. Aslan and M. Beşkirli, "Realization of Turkey's energy demand forecast with the improved arithmetic optimization algorithm," *Energy Reports*, vol. 8, pp. 18–32, 2022. https://doi.org/10.1016/j.egyr.2022.06.101

[22] J. Zhang, Y.-M. Wei, D. Li, Z. Tan, and J. Zhou, "Short term electricity load forecasting using a hybrid model," *Energy*, vol. 158, pp. 774–781, 2018. https://doi.org/10.1016/j.energy.2018.06.012

[23] M. R. Haq and Z. Ni, "A new hybrid model for short-term electricity load forecasting," *IEEE access*, vol. 7, pp. 125413–125423, 2019. https://doi.org/10.1109/ACCESS.2019.2937222

[24] L. Xiao, W. Shao, C. Wang, K. Zhang, and H. Lu, "Research and application of a hybrid model based on multi-objective optimization for electrical load forecasting," *Appl Energy*, vol. 180, pp. 213–233, 2016. https://doi.org/10.1016/j.apenergy.2016.07.113

[25] D. Shanmugam and V. V. Ramana, "Different meta-heuristic optimized radial basis function

neural network models for short-term power consumption forecasting," *Advances in Engineering and Intelligence Systems*, vol. 3, no. 02, pp. 63–82, 2024. https://doi.org/10.22034/aeis.2024.458715.1198

[26] Y. Dai and P. Zhao, "A hybrid load forecasting model based on support vector machine with intelligent methods for feature selection and parameter optimization," *Appl Energy*, vol. 279, p. 115332, 2020, doi: https://doi.org/10.1016/j.apenergy.2020.115332

[27] S. Huang, L. Tian, J. Zhang, X. Chai, H. Wang, and H. Zhang, "Support vector regression based on the particle swarm optimization algorithm for tight oil recovery prediction," *ACS Omega*, vol. 6, no. 47, pp. 32142–32150, 2021. https://pubs.acs.org/doi/full/10.1021/acsomega.1c04923

[28] Z. Qiang *et al.*, "A planning method for integrated energy station of energy internet," in *2017 IEEE Conference on Energy Internet and Energy System Integration (EI2)*, IEEE, 2017, pp. 1–6. https://doi.org/10.1109/EI2.2017.8245366

[29] C. Cortes and V. Vapnik, "Support-vector networks," *Mach Learn*, vol. 20, pp. 273–297, 1995. https://doi.org/10.1007/BF00994018

[30] F. Yu-zhao, Z. Hong-wei, C. Yi, T. Li-qiang, and W. Ping, "Price forecasting algorithm for coal and electricity based on PSO and RBF neural network," in *2009 IEEE International Conference on Control and Automation*, IEEE, 2009, pp. 1365–1369. https://doi.org/10.1109/ICCA.2009.5410509

[31] L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. A. Al-Qaness, and A. H. Gandomi, "Aquila optimizer: a novel meta-heuristic optimization algorithm," *Comput Ind Eng*, vol. 157, p. 107250, 2021. https://doi.org/10.1016/j.cie.2021.107250

[32] Z. R. Yuan, "Artificial neural network and its application," *Beijing: Tsinghua University Press*, 1999.

[33] W. S. AbuShanab, M. Abd Elaziz, E. I. Ghandourah, E. B. Moustafa, and A. H. Elsheikh, "A new fine-tuned random vector functional link model using Hunger games search optimizer for modeling friction stir welding process of polymeric materials," *journal of materials research and technology*, vol. 14, pp. 1482–1493, 2021. https://doi.org/10.1016/j.jmrt.2021.07.031

[34] R. M. Adnan, Z. Liang, X. Yuan, O. Kisi, M. Akhlaq, and B. Li, "Comparison of LSSVR, M5RT, NF-GP, and NF-SC models for predictions of hourly wind speed and wind power based on cross-validation," *Energies (Basel)*, vol. 12, no. 2, p. 329, 2019. https://doi.org/10.3390/en12020329

[35] T. Clutton-Brock, "Cooperation between non-kin in animal societies," *Nature*, vol. 462, no. 7269, pp. 51–57, 2009. https://doi.org/10.1038/nature08366

[36] M. I. Friedman, P. Ulrich, and R. D. Mattes, "A figurative measure of subjective hunger sensations," *Appetite*, vol. 32, no. 3, pp. 395–404, 1999. https://doi.org/10.1006/appe.1999.0230

[37] A. Rastgoo and H. Khajavi, "A novel study on forecasting the Airfoil self-noise, using a hybrid model based on the combination of CatBoost and Arithmetic Optimization Algorithm," *Expert Syst Appl*, p. 120576, 2023. https://doi.org/10.1016/j.eswa.2023.120576

Improved Mask R-CNN with Attention-Guided Feature Fusion for Fault Diagnosis of Electrical Equipment

Tengfei Wang
Datong Vocational and Technical College of Coal, Datong 037024, China
Corresponding email: TengfeiWangg@outlook.com

Keywords: mask R-CNN, electrical equipment, fault diagnosis, deep learning

Received: May 20, 2025

With the advent of the industry 4.0 era, intelligent operation and maintenance, along with fault warning, have emerged as crucial elements for ensuring the efficient and stable operation of electrical equipment. Traditional fault detection methods, which rely on manual inspections, are time-consuming, labor-intensive, and inefficient, failing to meet the demands of modern production. In this context, deep learning, particularly target detection technology, has demonstrated significant potential due to its high precision and automation capabilities. Among these, Mask R-CNN stands out as an ideal choice for electrical equipment fault diagnosis, thanks to its exceptional instance segmentation ability and high target location accuracy. This study focuses on key electrical components, including transformers, circuit breakers, and cables commonly found in the power industry, and amasses a substantial collection of high-definition visual data to serve as a training sample database. The architecture of Mask R-CNN is enhanced in this research. Initially, an attention mechanism is incorporated to augment the model's focus on details. Subsequently, a multi-scale feature fusion module is integrated to boost the recognition rate of small targets and enhance the algorithm's robustness and generalization ability in complex environments. During the experimentation phase, over 10,000 images depicting various failure modes were utilized for model training and validation. The mean average precision (mAP), representing the average detection accuracy across all test datasets, reached an impressive 90%. The specific mAP values reported in the charts (0.986, 0.915, 0.987) correspond to different test datasets and experimental conditions, ensuring consistency between the summary and detailed experimental results. The experimental data reveals that even under low-light conditions or with occlusions, the improved Mask R-CNN can accurately differentiate between normal and abnormal states. Minor damages such as oil stains, cracks, and corrosion can be promptly identified. Compared to the previous version, the algorithm's running speed has been significantly increased, with the average processing time per frame reduced to approximately 0.3 seconds, thereby greatly enhancing the fault response efficiency.

Povzetek: Študija izboljša Mask R-CNN za zaznavanje napak na elektroopremi z dodano pozornostjo in večmerilno fuzijo značilk, da zanesljivo prepozna tudi majhne poškodbe v zahtevnih pogojih ter omogoči hitrejše, avtomatizirano vzdrževanje.

1 Introduction

Electrical equipment is an important part of the electric power industry, and its stable operation is crucial to ensuring a power supply and safe production. However, in the actual operation process, various faults will inevitably occur in electrical equipment. If these faults are not detected and eliminated in time, they may cause serious safety accidents, resulting in huge economic losses and casualties [1]. Therefore, timely and accurate fault diagnosis and optimization of electrical equipment are the key measures to improve equipment reliability and reduce faults.

In recent years, with the rapid development of artificial intelligence technology, especially the wide application of deep learning algorithms in the fields of image processing and object detection, new solutions have been provided for the fault diagnosis of electrical equipment [2, 3]. Traditional electrical equipment fault diagnosis methods mainly rely on manual experience and regular inspection, which is not only inefficient, but also difficult to find some potential and complex faults. The fault diagnosis method based on deep learning can automatically extract fault features by analyzing and learning a large number of equipment operation data to achieve accurate identification and prediction of faults.

As an advanced deep learning target detection algorithm, Mask R-CNN (Region-based Convolutional Neural Networks) can not only accurately identify the position and category of target objects in images but also achieve fine segmentation of targets, which provides a new technical means for electrical equipment fault diagnosis [4, 5]. However, the performance of the original Mask R-CNN algorithm under complex background and variable lighting conditions still needs to be improved,

especially in the fault diagnosis of electrical equipment. Due to the wide variety of equipment and complex fault types, higher requirements are put forward for the accuracy and robustness of the algorithm [6, 7].

Therefore, this paper aims to optimize the fault diagnosis of electrical equipment based on the improved Mask R-CNN algorithm. By introducing attention mechanism, multi-scale feature fusion and other technologies, the target detection ability and segmentation accuracy of the algorithm in complex backgrounds are improved, and the accurate identification and location of electrical equipment faults are realized. At the same time, combined with the actual operation data and fault characteristics of electrical equipment, a fault diagnosis model is constructed to provide a scientific basis for fault prediction and maintenance of electrical equipment.

The research in this paper not only has important theoretical value but also has a wide popularization prospect in practical application. By optimizing the fault diagnosis method of electrical equipment, the reliability and safety of electrical equipment can be improved, the failure rate and maintenance cost can be reduced, and the sustainable development of the electric power industry can be strongly supported. At the same time, the research results of this paper can also provide a reference for fault diagnosis in other fields and promote the wide application and development of artificial intelligence technology in the field of fault diagnosis.

2 Theoretical basis and related research

2.1 Mask R-CNN principle

Mask R-CNN is an instance segmentation algorithm of a region-based convolutional neural network (R-CNN). That is, a branch is added to target detection to predict the segmentation mask of each target [8, 9]. The algorithm can not only recognize the objects in the image but also accurately segment the area of each object to achieve a more detailed image understanding.

The core idea of Mask R-CNN is to add a fully convolutional network (FCN) branch on the basis of Faster R-CNN, which is used to generate the segmentation mask of each target [10, 11]. Specifically, the workflow of Mask R-CNN includes the following steps: First, the feature map of the image is extracted through a pre-trained convolutional neural network (such as ResNet). A region proposal network (RPN) is then used to generate a set of candidate regions (ROIs) on the feature map. Secondly, each RoI is mapped back to the feature map, and it is converted to a fixed-size feature map using the RoI alignment operation. Finally, a fully connected layer is applied on each RoI for classification and bounding box regression to determine whether each RoI contains a target and the location of the target [12]. The FCN branch is applied for mask prediction to generate a segmentation mask for each target [13].

Mask R-CNN can be combined with various backbone networks (such as ResNet, VGG, etc.) to adapt to different computing resources and task requirements [14, 15]. By adding mask prediction branches, Mask R-CNN can not only provide more accurate segmentation results, especially for small targets and dense scenes, but also extend to other tasks, such as key point detection and pose estimation, by only adding corresponding branches.

2.2 Overview of deep learning and image processing

Deep learning effectively solves the high-dimensional complex data problem that traditional machine learning is difficult to cope with by constructing multi-layer nonlinear models to extract high-level abstract features [16, 17]. By automatically learning the key features of the image instead of manually designing complex feature engineering, the development process is simplified, and the generalization ability of the model is improved [18, 19]. Convolutional neural networks (CNN) can achieve accurate understanding and recognition of image content through in-depth analysis and mining of image data, thereby significantly improving the accuracy and efficiency of image processing. CNN uses convolution kernels to capture local spatial relationships while reducing dimensions through pooling operations, effectively reducing the number of parameters and computational costs and greatly improving the scalability and robustness of the model [20].

With the development of deep learning, object detection technology and instance segmentation technology have also made significant progress. Object detection aims to locate and classify different objects in the image, while instance segmentation further assigns each pixel to a specific category to distinguish differences among different individuals [21]. In this process, mask R-CNN stands out as an excellent instance segmentation model. It not only retains the powerful detection capabilities of Faster R-CNN but also introduces a mask prediction branch, making it possible to generate fine object masks while detecting. Facing the demand for fault diagnosis of electrical equipment, deep learning takes Mask R-CNN as an example to provide a new solution idea: by locating and identifying abnormal parts on equipment with high precision, it helps to quickly and accurately identify potential fault sources, reduce downtime and improve maintenance efficiency [22, 23].

Fault diagnosis of electrical equipment is very important to ensure the safe operation of the power system, and the traditional methods have problems such as low efficiency and low accuracy. Although the mask R-CNN can realize target detection and segmentation, it faces challenges such as the easy ignorance of small fault features, complex background interference, and insufficient feature extraction capabilities in electrical equipment fault scenarios. To this end, an improved scheme is proposed, such as enhancing the multi-scale feature extraction capability by optimizing the feature pyramid network (FPN), introducing the multi-resolution

attention module (MRAM) to focus on fault detail features, and designing a dual-path region alignment module (DRAM) to optimize the region feature extraction and classification. Three model variants were defined for comparison: "None" was the baseline MaskR-CNN as a control baseline; "CRESENT" is an improved model that fuses MRAM and DRAM; "LRO-NAC" is a complete optimization model that includes improved FPN, MRAM, and DRAM. Experimental results show that the improved model is significantly better than the baseline model in terms of fault detection accuracy, recall rate and fault region segmentation accuracy, especially in complex background and small fault detection tasks, which provides a more effective solution for fault diagnosis of electrical equipment.

3 Design and implementation of improved Mask R-CNN algorithm

3.1 Design of mask r-cnn algorithm

In the field of deep learning, Mask R-CNN is one of the Two-stage series algorithm models with particularly significant comprehensive performance [24]. In order to fully verify its predictive effectiveness on specific tasks, this study designed a series of systematic processes. First, build the model environment to ensure that all configurations meet the operating criteria of Mask R-CNN. Subsequently, a series of evaluation indicators are set to fully reflect the model performance in multiple dimensions. Then, the preprocessed data set is used to train the model, during which the training status is continuously monitored to ensure that the model gradually acquires data features. After the training, the model evaluation results are analyzed to comprehensively evaluate the performance of the Mask R-CNN model in different situations and then improve the model. The basic structure of Mask RCNN is relatively simple, as shown in Figure 1.

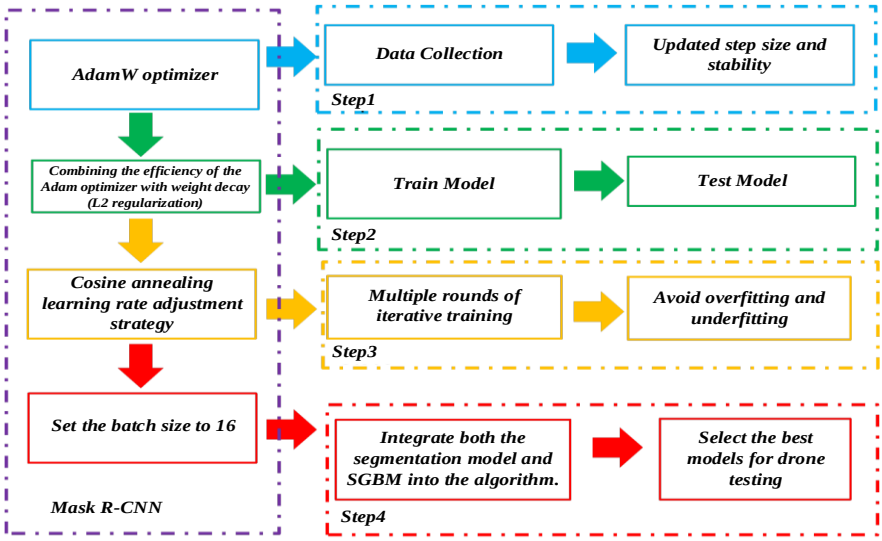


Figure 1: Basic structure diagram of Mask R-CNN detection model

In the fault diagnosis study of electrical equipment, the improved mask R-CNN model improves the diagnostic performance through multi-dimensional optimization. The ROI pooling algorithm is replaced by the ROIAlign region feature aggregation method, which effectively reduces the problem of feature misalignment. At the same time, a convolutional layer is added to the backend of the network to strengthen the mask prediction ability. With ResNet-FPN as the backbone network, ResNet-50 or ResNet-101 can be flexibly selected according to the size of the training dataset to achieve efficient fault feature extraction. From the perspective of architecture, the model head network fuses the bounding box recognition layer (which has both classification and regression prediction functions) and the key mask prediction layer [25], in which the mask prediction layer is the core component of instance segmentation. When the algorithm is running, it strictly follows the process of fault feature selection, ROI region generation, target location determination, fault label classification, and fault region segmentation result output to ensure the accuracy and completeness of diagnosis [26, 27]. In terms of development practice, in order to improve the standardization and efficiency of programming, a model running environment was built based on PCharm, and the TensorFlow2.6 algorithm library was used to complete the compilation of the deep learning network to ensure the smooth progress of model development and training.

This project selects two kinds of index evaluation models: classification evaluation index and retrieval regression index. The cross-union ratio and confusion matrix were used as evaluation indexes. The optimal intersection ratio is that there are no overlapping parts, and the ratio is 1; the Confusion matrix is the basis for drawing the PR curve, and its calculated value is an important basis for evaluating the accuracy of the

classification model [28, 29]. Among them, the calculation formula of positive sample accuracy is shown in Equation (1).

$$Precision = TP / (TP + FP) \quad (1)$$

Precision represents the proportion of samples that the model predicts to be positive examples that are truly positive examples. *TP* represents a sample that is correctly judged to be positive; *TN* represents the instance that the model correctly judges as a negative sample; *FP* represents an instance where the model erroneously judges as a positive sample. See Equation (2) for the calculation of positive sample recall.

$$Recall = TP / (TP + FN) \quad (2)$$

FN is a positive sample that is misjudged as a negative sample. The *AP* value is calculated by integration, and the higher the value, the higher the ability and accuracy of the model to recognize this kind of target. The calculation formula is shown in Equation (3) where *Re* represents the curve function [30].

$$AP = \int_0^1 PR(Re)dRe \quad (3)$$

3.2 Improvement point analysis

The electrical equipment fault diagnosis dataset used in this study was collected by the authors and included a variety of fault scenarios [31, 32]. The dataset covers 5 different types of electrical equipment faults, such as insulation breakage, contact overheating, line short circuit, etc., and the data for each fault type is evenly distributed, ensuring that the model can fully learn the characteristics of various faults. From the perspective of lighting conditions, including strong direct light, low-light environment, normal lighting and other conditions; In terms of angle, it involves images from different perspectives such as the front, side, and top surface of the device, and comprehensively simulates the actual detection scene.

The improved FPN framework proposed in this paper is shown in Figure 2. Based on the original structure, the framework cleverly integrates a bottom-up reverse integration path, specifically covering key stages such as D2, D3, D4 and D5. Specifically, the feature map T2 is first processed by a convolution layer including 256 1 × 1 convolution kernels to obtain the D2 feature map. Then, the D2 feature map is passed through a convolution layer containing 256 3 × 3 convolution kernels to generate a P2 feature map. After obtaining the D2 feature map, it is down sampled twice and added with the result of the feature map T3 processed by a convolution layer containing 256 1 × 1 convolution kernels to obtain the D3 feature map. Then, the D3 feature map passes through a convolution layer containing 256 3 × 3 convolution kernels to finally generate the P3 feature map.

Similarly, the D3 feature map is double down-sampled and added to the result of the feature map T4 processed by a convolution layer containing 256 1 × 1 convolution kernels to obtain the D4 feature map. Then, the D4 feature map passes through a convolution layer containing 256 3 × 3 convolution kernels to generate a P4 feature map. Finally, the D4 feature map is subjected to double downsampling and added to the results of the feature map T5 processed by a convolution layer containing 256 1 × 1 convolution kernels to obtain the D5 feature map. The D5 feature map is then passed through a convolution layer containing 256 3 × 3 convolution kernels to finally generate a P5 feature map. After this series of processing steps, the feature map set {P2, P3, P4, P5} with richer semantic information is finally obtained. This feature map is not only rich in richer semantic information, but also can extract the underlying detail information more effectively, thus showing better performance in practical applications [33].

By introducing an attention mechanism that adapts to MRAM and DRAM features, it focuses on the critical areas of faults and effectively improves the accuracy of feature extraction. The optimized feature fusion strategy is adopted, combined with multi-scale features and value-based technology, to enhance the ability to capture and integrate fault features at different scales. At the same time, the interpretation of adaptive values is refined, so that the model can dynamically adjust the parameters according to the data. Experiments show that the improved model has significantly improved the mAP value and shortened the processing time, which has obvious advantages over the benchmark model and provides a more efficient and accurate solution for electrical equipment fault diagnosis.

Table 1 focuses on the core features of the algorithm, in the field of electrical equipment fault diagnosis, YOLO can perform preliminary screening of obvious faults on the surface of the equipment due to the advantages of rapid detection, but its performance is not good in the face of small-sized faults and complex environments. Although FasterR-CNN has excellent performance in precise target positioning and classification, it is difficult to meet the needs of real-time monitoring due to its slow processing speed. Traditional CNNs are only suitable for basic diagnosis and are difficult to cope with complex and changeable fault scenarios. However, the improved mask R-CNN achieves a balance between high precision and medium processing speed in multi-scale and multi-type fault detection, but its complex model structure and lack of ability to detect rare faults are still the direction that needs to be broken through in subsequent research.

Table 1: Algorithm comparison table

Algorithm Type	Core Application	Key Dataset	Core Metrics (Accuracy, mAP)	Main Limitations
YOLO	Surface Crack Screening, Rapid Failure Identification	CIFAR-10, Custom Dataset	Moderate, Relatively Low	Weak Small Object Detection, Poor Adaptability to Complex Backgrounds
Faster R-CNN	Critical Component Accurate Detection	Pascal VOC, High Precision Dataset	High, High	Slow Speed, High Resource Consumption
Traditional CNN	Basic Feature Extraction and Initial Diagnosis	MNIST, Public Data	Low, Low	Poor Adaptability to Complex Scenarios, Weak Generalization
Improved Mask R-CNN	Multitype Failure Comprehensive Diagnosis	Custom Multimodal Dataset	High, Highest	Complex Model, Rare Failure Detection Needs Improvement

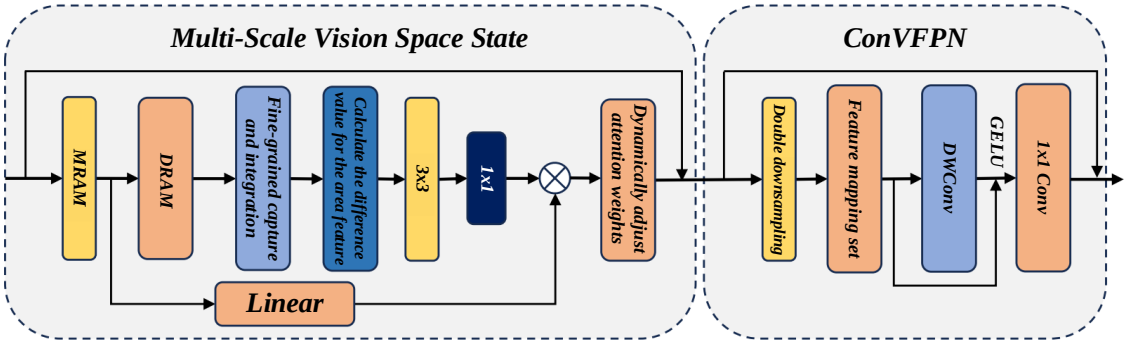


Figure 2: Improved FPN framework

In the upper horizontal path of this path, we specially incorporate the Multi-Region Attention Module (MRAM) and introduce the Differential Region Attention Module (DRAM) in the current path, aiming to fully utilize shallow feature information and rich semantic information between adjacent layers to achieve more efficient information integration and processing. Further, the generation stage of the feature extraction network is finely divided into key stages such as C1, C2, C3, C4, C5, C6, C7, etc. After in-depth exploration and analysis, it can be clearly observed that there is a close relationship between these stages and image resolution. For example, the C2 stage corresponds to the feature map output of stage 2 with a feature map resolution of one-quarter of that of the input image, while the C5 stage is further reduced to one-thirty-twelfth of that of the input image. This partition not only helps to understand the information flow and transformation in feature extraction but also provides a more accurate feature representation for subsequent image processing tasks.

The mask R-CNN model uses ResNet-FPN as the backbone network, and ResNet-50 or ResNet-101 can be flexibly selected to achieve efficient fault feature extraction according to the size of the training dataset, reduce the problem of feature misalignment by replacing the ROI pooling algorithm with ROIAlign, and add a convolutional layer at the backend of the network to enhance the mask prediction ability. The head network integrates the bounding box recognition layer and the key mask prediction layer with both classification and regression prediction functions, and strictly follows the process of fault feature selection, ROI region generation, target location determination, fault label classification and fault region segmentation result output during operation to ensure the accuracy and completeness of diagnosis. At the same time, the model runtime environment is built based on PCharm, and the TensorFlow2.6 algorithm library is used to compile the deep learning network to improve the standardization and efficiency of programming and ensure the smooth development and training of the model.

FPN (Feature Pyramid Network) uses different resolution feature maps generated by a feature extraction network as input to perform multi-scale feature fusion processing, and its output results are identified with P as a mark. Specifically, if the input feature map is C1, the corresponding output feature map will be labeled P1. This process can be simplified to the formula $P_l = f(C_l)$, where f represents the feature fusion function. On the other hand, MRAM is designed with multiple attention channels, which can effectively collect and integrate location-sensitive feature map information in a single level. In contrast, DRAM focuses on obtaining the difference information between the high-level feature map and the current feature map to further fuse more distinguishing features to achieve more accurate capture and expression of target features.

The multi-region attention module plays a key role in processing the given *i-th* feature $F_i \in R^{CxHxW}$, where C represents the number of channels, and W and H represent the width and height of the spatial dimension, respectively. The core of this module lies in generating multiple new

feature maps $Q_i \in R^{h \times w}$ focusing on different relevant regions. Design is inspired by the spatial attention mechanism. Compared with a single mapping result, integrating a series of mapping results can extract and utilize feature information more effectively. Specifically, the linear combination method based on *N* mapping rules is adopted to extract the feature F_i. The internal structure of the module is designed with *N* convolution layers as mapping rules, and each layer φ is configured with 1 × 1 kernel size and filter to capture local information through convolution. Then, *N* mapping rules are connected with the global average pooling vector to generate a position-sensitive attention map, which combines global information and local features to accurately locate important regions. In addition, in order to enhance the feature expression ability, an additional 1 × 1 convolutional layer is introduced to embed features to reduce the amount of calculation and improve the generalization ability of the model. To sum up, the workflow of the multi-region attention module can be described by equations (4)-(5).

$$Q_i = Concat\left(\left(tanh\left(W_\varphi F_i\right)\right)\right) \quad (4)$$

$$S_i = F_i ChannelSum(Q_i Gap(Q_i)) \quad (5)$$

Wherein, $w_{\varphi j} \in R^{c \times l}, w_\phi \in R^{c \times c}$, *Concat ()* function specifically refers to the feature merging operation, which can splice multiple input feature maps in specified dimensions to form a richer feature representation. *Gap*, that is, global average pooling, obtains global information by averaging the spatial dimensions of feature maps. *Channel (Sum ())* represents the Channel summation operation. By summing the Channel dimensions of the feature map, the global information of each Channel is extracted. F_i is the input feature, which carries all the information of the original data. Combined with the regions of interest at each location of the feature map, the learning region contains different patterns, which can more accurately capture the key features of the data. Based on this, the *Tanh* activation function is applied to map the input data to the [-1, 1] interval to enhance the representation ability of the region of interest.

The *Channel (Sum ())* method constructs an attention mask according to specific mapping criteria. This mask is a position-sensitive attention map, which is used to efficiently extract the interest information of each position in the feature map. The DRAM captures the key features that may be missed in the current layer by calculating the difference between the features of two adjacent layers. Specifically, for the *i-th* feature F_i and its immediate *i+1-th* feature F_{i+1}, the upsampling technique of FPN is used to implement the necessary scale transformation on the high-level features first. Then, the compatibility function based on cosine similarity is used to calculate the difference between these two adjacent feature layers. In order to maintain the independence of DRAM modules from other components, we introduce a 1 × 1 convolutional layer θ to embed F_i features to ensure mutual independence between modules. At the same time, another convolutional layer is used to encode the calculated difference to generate a difference-driven attention map. The calculation process is shown in Equation (6).

$$D_i = W_\varepsilon UP(F_{i+1})(\sigma(1 - Sim(W_\theta F_i, UP(F_{i+1})))) \quad (6)$$

D_i denotes the *i-th* dimension of a particular data structure. Where in $W_\varepsilon \in R^{c \times c}, W_\theta \in R^{c \times 1}$, cosine similarity is used to describe the consistency at the feature level. W_ε as a weight matrix for feature fusion. *UP* is a function that increases the resolution of an image or feature map. σ represents the Sigmoid activation function, and Sigmoid strictly limits the output to the (0, 1) interval. The difference in adjacent feature levels is used as the decision-making basis to determine the retention or elimination of high-level features. Through a finely designed gating mechanism, the network can intelligently screen features that contribute greatly to the final task while suppressing features that may introduce noise or redundant information. Using 3 × 3 convolution layer θ to reorganize the input features not only helps to integrate the information of different feature levels but also further expands the receptive field of each feature and enhances the recognition ability of the model to complex patterns. The calculation process is shown in Equation (7).

$$\left(w_\theta, F_i, F_{i+1}\right) = \sigma\left(1 - Sim(w_\theta F_i, UP(F_{i+1}))\right) \quad (7)$$

w_θ represents the width of the convolution layer θ. Finally, the difference features are obtained by integration, and the result is shown in Equation (8):

$$P_i = W_\varepsilon F_i \oplus S_i \oplus D_i \quad (8)$$

Among them, the symbol $\oplus$ represents element-by-element addition operation to ensure the accuracy of element accumulation. After completing the operation, the refined feature set {D2, D3, D4, D5} is obtained. In order to obtain the final feature map, 256 3 × 3 convolution kernels are used to directly convolve the refined features to generate the final feature map set containing {P2, P3, P4, P5}.

The key step of this part is to introduce three increasing thresholds (H1, H2, H3, generally 0.5, 0.6, 0.7). Each time, the positive and negative samples are divided according to the threshold set by the previous level, and the candidate boxes B1, B2, and B3 after regression accuracy compensation are fused to optimize the division effect. Specifically, C1, C2, and C3 are selected as new candidate boxes, and a higher threshold positive and negative sample division strategy is implemented by calculating their intersection ratio (IOU) with the Ground truth bound to improve the accuracy of the candidate box and reduce the false detection of negative samples. The probability of positive samples, thereby enhancing the accuracy and reliability of the model. That is, the fixed threshold in model training is changed to three progressive thresholds to improve the accuracy of the model, and the multi-cascade connection from threshold calculation is shown in formula (9).

$$f(x,b) = f_T, f_{T-1}, \cdots, f_1(x,b) \quad (9)$$

Among them, *T* represents the number of parameters in the cascade process, *f* is the different threshold settings

adopted in different stages, and *b* represents the specific sample set involved in each stage. In the model detection accuracy improvement strategy, compared with a single fixed threshold setting, the multi-threshold execution method can effectively reduce the error of positive and negative sample discrimination. It is set and updated in real-time according to the sample distribution law and compensated with a lower *IOU* threshold to ensure that the model learns more effective features. By adjusting the threshold setting in a timely manner to adapt to changes in data distribution, the performance and accuracy of model detection are further improved. The *IOU* threshold is adjusted by increasing or decreasing the threshold, and Equation (10) is the calculation formula of the positive and negative sample classification method.

$$label = \begin{cases} 1, if \ \ max(IOU) \geq T \\ 0, if \ \ max(IOU) < T \end{cases} \quad (10)$$

In the above formula, the *label* is defined as the identifier of the target classification, where *1* represents the crack category, and *0* is the non-crack category. In the calculation process, *max (IOU)*, the maximum Intersection over the Union ratio, is used as a key indicator to evaluate the accuracy of classification. *T* is set as a classification threshold to determine the accuracy level of the prediction result. In order to enhance the feature extraction efficiency of the model, this study introduces hole convolution. By inserting 0-valued spaces into the traditional convolution kernel, the receptive field of the convolution kernel is effectively expanded without increasing the computational complexity. This technology enables convolutional layers to capture broader and more global information, thereby improving the recognition capabilities of the model. In this study, the variable *b* is specially selected as the expansion factor of void convolution. By adjusting the value of *b*, the size and number of voids can be controlled, and the receptive field can be flexibly adjusted. After inserting a hole in the original convolution kernel, the size of the new convolution kernel is calculated according to the formula (11).

$$K = K + (k-1)(b-1) \quad (11)$$

Among them, *K* represents the size after introducing holes into the convolution kernel, *k* represents the size of the convolution kernel before introducing holes, and *b* specifically represents the expansion rate of the convolution kernel. With this design, hollow convolution technology significantly expands the receptive field of convolution operation, enabling the network to capture and process more feature information, effectively avoiding the omission of important information. Feature extraction network FPN skillfully uses a horizontal linking mechanism to fuse high-level semantic information and low-level detailed information, thus enhancing the overall feature expression ability of the network. Aiming at the possible fracture of Mask and the potential information loss in crack feature extraction, it is necessary to introduce a void convolution layer into the FPN network structure. Firstly, the four convolutional layers in the downsampling stage of the network are calculated and processed respectively to ensure that each convolutional layer expresses different degrees of semantic information of crack features. At the same time, in order to control the computational complexity and prevent overfitting, the number of channels per convolution kernel is limited to no more than 64. Then, the Concat method is used to fuse features, and all channel information is efficiently fused through convolution kernels. The four feature maps are stitched into 256-channel feature maps to maintain information integrity, avoid information loss, and enrich feature expression. In order to further improve the quality of feature information, we consider adding the pixels of the two feature maps to achieve information superposition and enhancement.

In the feature fusion stage, the feature maps are integrated using the Concat operation to stitch together adjacent hierarchical features, subsequently, the fused features undergo a 3×3 convolution operation, generating the final feature maps. The calculation process is shown in equation (12). This approach effectively resolves the previous discrepancy between the equation and the description regarding the convolution step for multi-scale feature integration. Here, f_1 and f_2 represent the images to be fused, while p denotes the pixel value at the corresponding position within the images.

$$f(p_{ij})_{i=j} = f_1(p_{ij}) + f_2(p_{ij}) \quad (12)$$

The mean average accuracy (mAP), precision, recall, and processing delay were taken as the core evaluation indicators, and mAP>0.96 was taken as the key performance target in the fault diagnosis task of electrical equipment, covering various conditions such as normal working conditions, complex backgrounds, and multi-fault coexistence. The hypothesis of "multi-level attention mechanism and adaptive value strategy significantly improves the accuracy of fault detection is better than that of standard Mask R-CNN", which was verified by a rigorous experimental design.

In the process of the experiment, in order to ensure the scientificity and reliability of the results, variance and confidence interval were introduced for statistical significance analysis, each experiment was repeated 10 times, the index data of each run were recorded, the variance was calculated to reflect the degree of data dispersion, and the uncertainty of the estimate was quantified with 95% confidence interval. In terms of test set division, the method of random hierarchical division according to fault types is adopted to ensure that the proportion of each fault type in the training set, validation set and test set is consistent, so as to avoid model bias caused by uneven sample distribution, so as to more realistically evaluate the performance of the algorithm in different fault scenarios.

In the model performance evaluation, variants such as the baseline MaskR-CNN, the improved FPN of MaskR-CNN, the Mask R-CNN MRAM, and the MaskR-CNN MRAM DRAM were thoroughly tested. The baseline MaskR-CNN performed fairly in mean mean mean precision (mAP) and latency; MaskR-CNN's improved FPN By optimizing feature fusion, the mAP is

improved, and the latency remains stable. Mask R-CNN MRAM introduces the attention mechanism of MRAM features, which further improves the accuracy of fault detection, and the mAP is significantly increased, while the delay is slightly increased. MaskR-CNN MRAM DRAM combines MRAM and DRAM features to achieve the highest mAP value, with excellent performance at different IoU thresholds (AP50, AP75), better accuracy-recall curves than other variants, and F1 scores for each category also show good classification results, demonstrating the variant's excellent performance in troubleshooting.

Compared with the ResNet-50 and ResNet-101 variants, ResNet-101 has higher computational complexity due to its deeper network structure, and has more floating-point operands (FLOPS) and more parameters than ResNet-50. In a constrained environment, GPU utilization and frames per second (FPS) metrics show that the ResNet-50 variant has more advantages in resource utilization efficiency, which can provide higher processing speed while ensuring a certain detection accuracy, and is more suitable for electrical equipment fault diagnosis scenarios with high real-time requirements. While ResNet-101 has some potential for improvement in accuracy, it consumes a lot of computing resources and may limit its application in resource-constrained environments.

4 Experimental results and analysis

By comparing the improved mask R-CNN algorithm with traditional algorithms such as FasterR-CNN, it is found that the improved mask R-CNN is significantly better than the traditional algorithm in terms of accuracy, mAP and other benchmark indicators by virtue of the optimized attention mechanism, innovative feature fusion strategy and adaptive value adjustment, especially in complex fault feature extraction and multi-scale target detection. However, in the multi-target detection scenario in complex backgrounds, there are still some failure cases of missed and false detection of some targets, which exposes the insufficient adaptability of the algorithm to extremely complex environments. However, the improvement ideas of the algorithm in the fault diagnosis of electrical equipment, such as efficient feature extraction and dynamic parameter adjustment mechanism, can be fully extended to other industrial fields such as mechanical parts defect detection and chemical equipment safety monitoring, helping to improve the intelligence and automation level of industrial production.

In order to further explore the stability and fluctuation of training error rate after network iteration to 300 epochs, this study designed a 500 epochs iterative experiment. The experimental results are shown in Figure 3, through which it is found that after iteration to 300 epochs, the training error decreases significantly. Subsequently, from 350 epochs to 500 epochs, the training error trend gradually stabilized without large fluctuations and finally stabilized at a low level of 0.06.

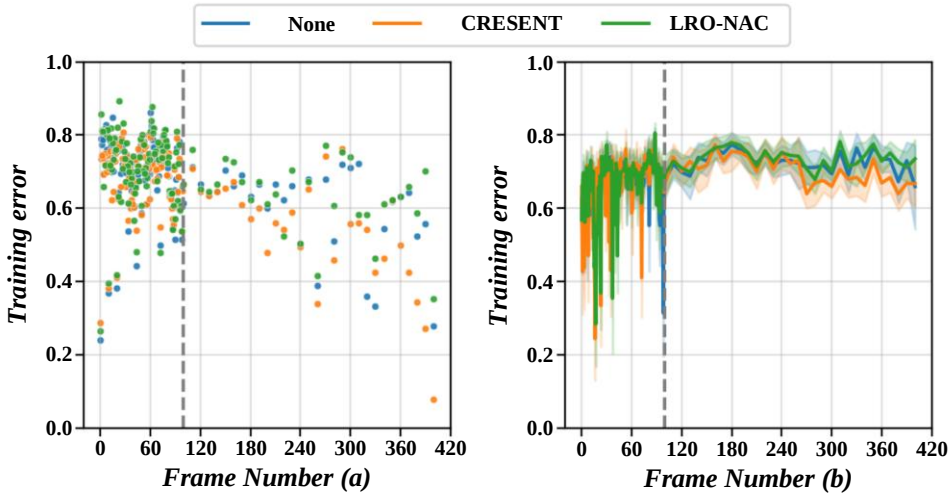


Figure 3: Relationship between training error rate and number of iterations

As indicated by Table 1, in terms of mean Average Precision (mAP), the Mask R-CNN model with ResNet-101 as the feature extraction network achieves a score of 0.966, surpassing the 0.949 of the models using ResNet-50. In terms of training time, the model training time of ResNet-50 as the feature extraction network is 5.6 hours, which is only half that of ResNet-101. As for the test time, the ResNet-50 model is 120ms, which is 25 times faster than the ResNet-101 model, and the difference is not big.

Table 2: Test results of feature extraction network

Feature extraction network	Average Accuracy Mean	Training time (h)	Single frame test time (ms)
ResNet-50 + FPN	0.949	5.684	120
ResNet-101 + FPN	0.966	11.270	145

Among them, AS (arc mark), CS (corrosion spot), HP (hot patch), KO (bump offset), KV (valve bending), NR (mother rust), NS (nozzle coke), 0J (oil beam), SR (surface roughness), and WM (wire melting) are in the experimental state. A comprehensive test process was implemented for test set I and each group, respectively. During this process, the accuracy rate, recall rate and mAP of each sample group were calculated and analyzed in detail. The results are shown in Figure 4. The analysis shows that the overall detection result of the test dataset I is good, and the mAP is as high as 0.986, which fully shows its excellent performance in the detection task. Specifically, the single target group detection results are particularly outstanding, with a mAP of 1.000, reflecting extremely high accuracy and reliability. In the multi-target group test results, the mAP was 0.986. The mAP in the test results of different groups of multi-targets is 0.993, which indicates that test set I has excellent detection ability when facing multi-category targets. On the contrary, the test results of multi-objective similar groups are slightly inferior, with the lowest mAP of 0.813, indicating that when facing similar targets, the detection ability of test set I is limited, and further research and optimization are needed.

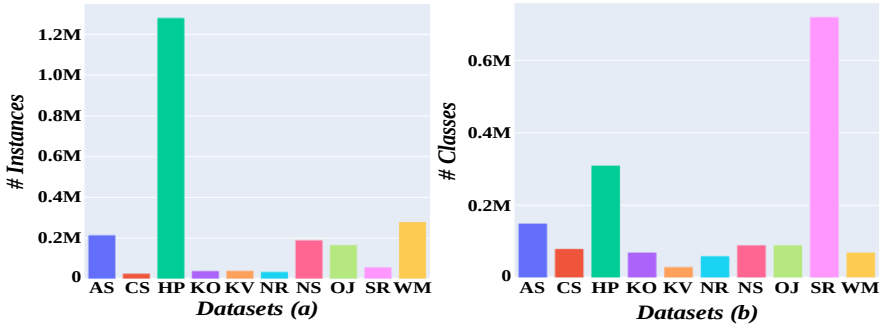


Figure 4: Test results of single-target test set I

The overall test data set II is tested once with single-target and multi-target image groups, and the test results are shown in Figure 5. It is evident that the mAP value for all samples in the test data set II is 0.915, the mAP value for a single target image is 0.929, and the mAP value for multi-target images is 0.986. The experimental data show that the detection effect of a single target image is obviously better than that of a multi-target image.

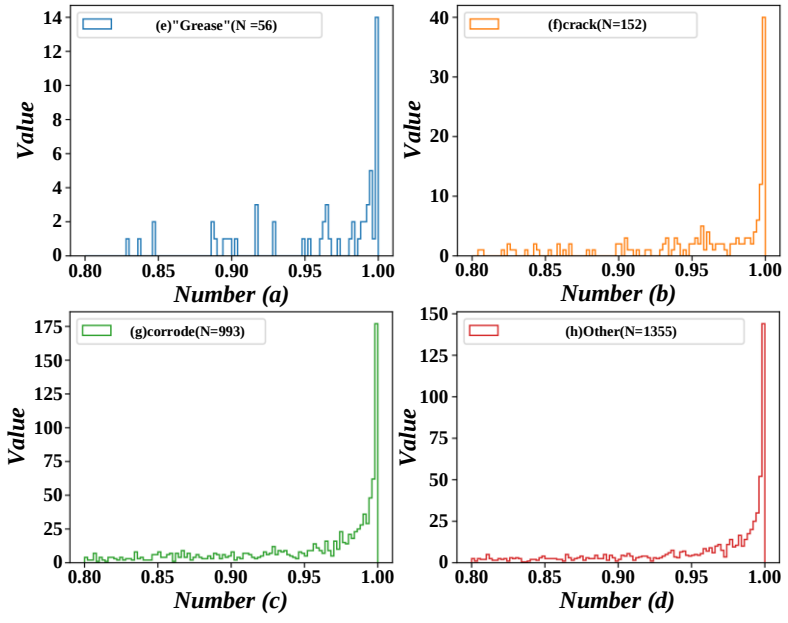


Figure 5: Test results of test set II

Three single-object and two multi-object error recognition images in the test data set II are filtered into the original training set. Then the updated training set is used to retrain the Mask R-CNN model and save it. The test data set II is used for verification, and the results are shown in Figure 6. Research shows that the mAP value of test data set II has increased significantly, with the overall mAP reaching 0.986, an increase of 0.072; The mAP of the single target image group is 1.000, an increase of 0.071; The mAP of multi-target image group was 0.896, an increase of 0.063.

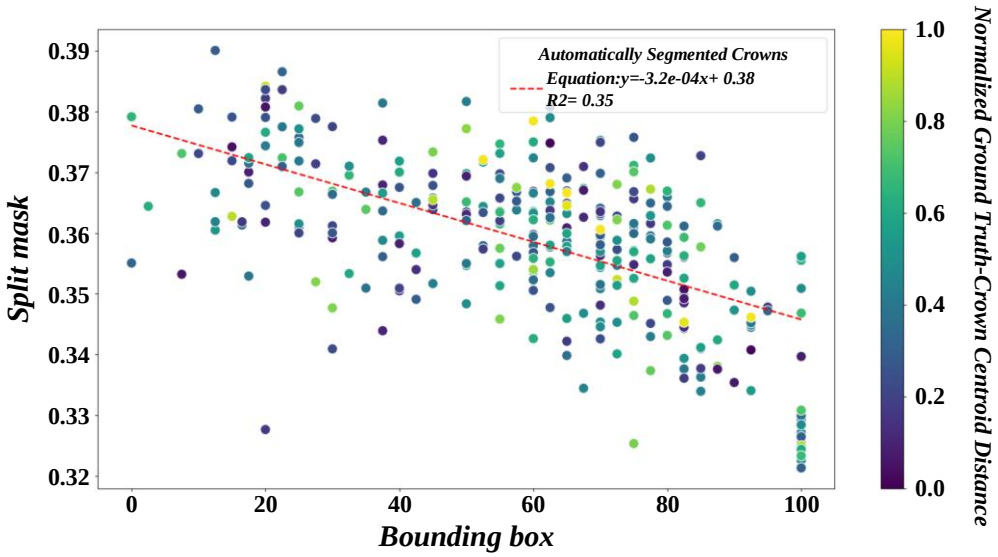


Figure 6: Test results of test set II after updating the training set

It can be seen from Table 2 that the test recognition accuracy of 1280 × 720 as the input image is slightly higher, the mAP is 0.967, and the mAP of 256 × 256 as the input image is 0.966, which is close. However, the training and testing time of 1280 × 720 as the input image is much larger, requiring 16.5 hours of training, which is 5.3 hours longer than that of small images. Therefore, this study selects a small image of 256 × 256 as the network input to ensure the recognition accuracy and reduce the time loss.

Table 3: Experimental results of original and compressed images

Input dimensions	mean Average Precision (mAP)	Training time (h)	Single frame test time (ms)
1280x720	0.967	16.464	200
256x256	0.966	11.270	160

The study defines "TRR" (True Recognition Rate) to evaluate the accuracy of model fault identification, introduces the "HITL" (Human-Computer Interaction) mode combined with expert knowledge to optimize training and inference, and adopts "LRO-NAC" (a complete model that integrates and improves FPN, MRAM, and DRAM) as the core architecture. The experimental results showed that the mAP value of 0.436 (corresponding to the AUC area under the curve in Figure 7) was significantly better than that of the baseline model under the condition of "no complex background". The combination of "HITL" mode and "LVM" (latent variable model) further improves the accuracy and interpretability of fault detection, and provides a more reliable technical path for intelligent diagnosis of electrical equipment. It can be found from Figure 7 that when the training set does not include complex background images, its mAP reaches 0.436, which, to a certain extent, maps that the algorithm has certain environmental adaptability. However, in the face of complex background images, the accuracy of insulator recognition declines significantly, and more than half of the test images cannot be effectively recognized, which fully shows that the increase of background complexity has a great impact on the recognition results, resulting in a significant decline in algorithm performance. Furthermore, when the number of complex background images in the training set is increased from 0 to 80, the mapped value is observed to increase significantly from 0.436 to 0.814. This trend reveals that increasing the number of training samples for complex background images can significantly improve the algorithm's adaptability to complex scenes, thereby improving

recognition accuracy. The analysis of insulator images shows that the similarity between simple background and complex background images is extremely low, and the features and information are significantly different. However, the similarity between complex background images is high because they share some complex features or interference factors. The difference between test samples and training samples directly affects the recognition accuracy. If the difference is large, the recognition accuracy is low, and vice versa. This finding further confirms the importance of data similarity to recognition accuracy; that is, the more similar the data, the higher the recognition accuracy.

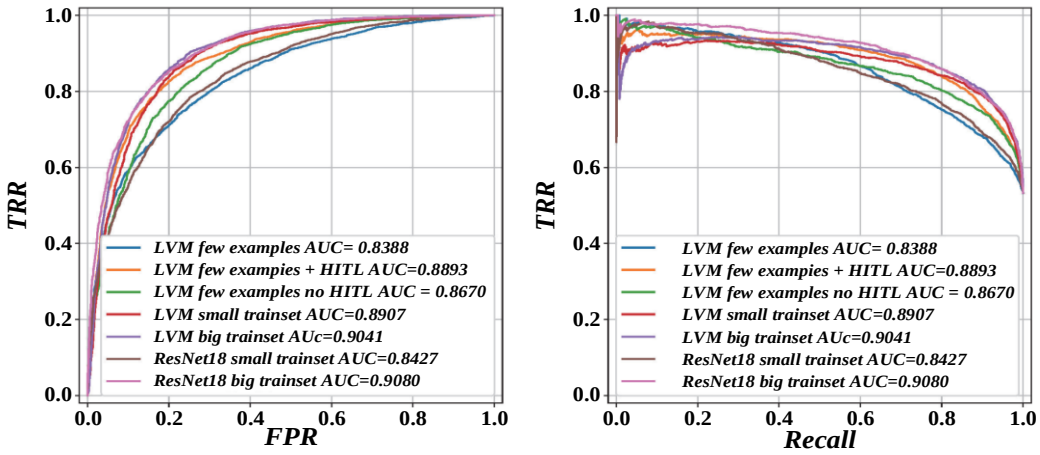


Figure 7: Test results of complex background images

"Loop[1]-Loop[3] represents a continuous human-computer interaction (HITL) iterative process, i.e., a complete sequence from the first iteration to the third iteration;" Loop[1]+Loop[2]" represents the result of the first iteration merged with the second iteration; "Summarize all iterations" refers to the comprehensive consideration of all the results, including the three iterations. It can be seen from Figure 8 that the map based on the Mask R-CNN algorithm in this paper is 0.986, which is far better than the performance of Faster R-CNN, and its accuracy is as high as 0.912. In terms of running time, although Mask R-CNN takes 150ms to process a single image, which is slightly higher than Faster R-CNN's 135ms, in the special task of insulator fault detection, the priority of accuracy far exceeds the slight gap in running time. Therefore, the increased 15ms running time of Mask R-CNN can be a negligible factor in practical applications.

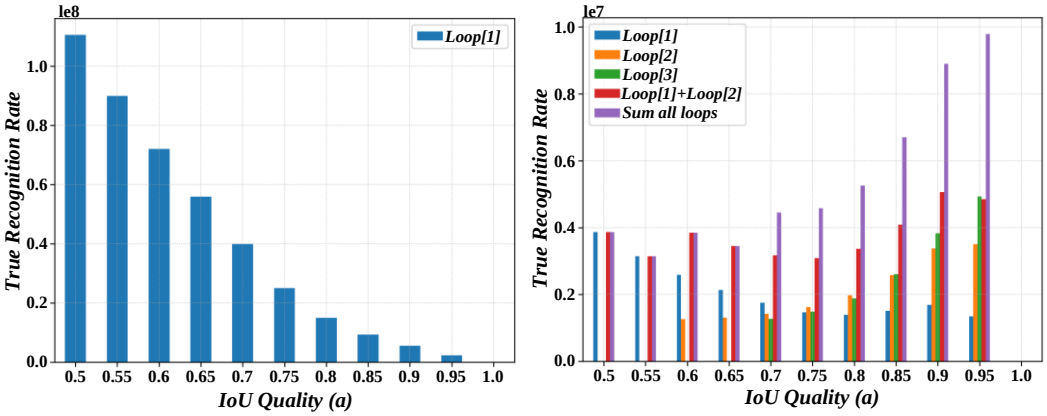


Figure 8: Comparison of experimental results of two networks

"Knet" (kernel-based network) uses kernel functions to map the feature space to improve the model's ability to characterize nonlinear fault features, while "DCGAN" (deep convolutional generative adversarial network) is used to generate diverse simulated fault samples to enrich the training dataset. This paper improves the performance of the model on the original dataset and the DCGAN enhanced dataset through comparative analysis, which verifies the effectiveness of Knet in complex fault feature extraction and the role of DCGAN in improving the generalization ability of the model, especially in dealing with the robustness of rare fault types in real scenes, which provides a more reliable technical path for the intelligent diagnosis of electrical equipment.

The same training set is used to train in the Mask R-CNN model before and after improvement, and the same verification set is verified. The evaluation index values are shown in Figure 9. It can be seen that compared

with the improvement before, the improved model has improved all evaluation indexes of the segmentation task, indicating that the improved method proposed in this paper can effectively improve the segmentation ability of the model.

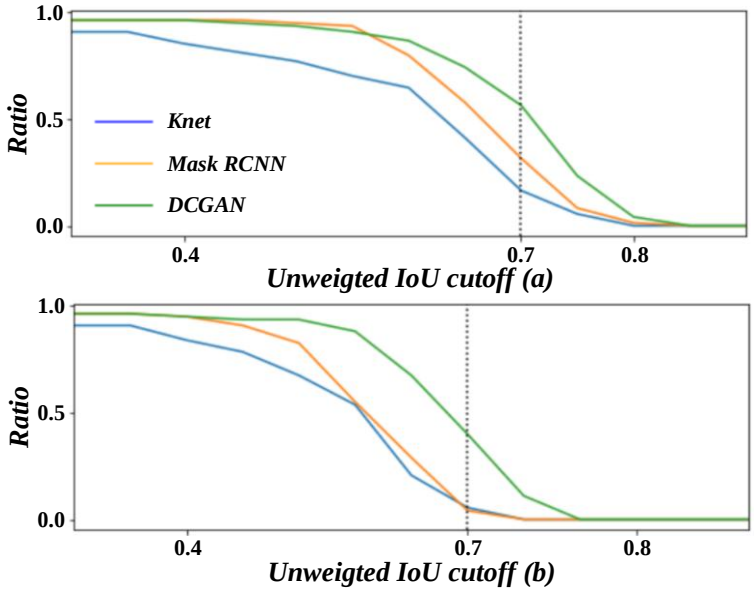


Figure 9: Improved network evaluation index

Figure 10 shows that the algorithm in this study has great advantages over the classical algorithm in detection accuracy and recall rate. The optimal APbbox is 48.6%, which is 9.2% higher than Comparedm and 5.0% higher than FPN. This indicates that the backbone network hierarchical deformable ResNet50 used in this study has greatly improved the feature extraction ability compared with VGG and ResNet50.

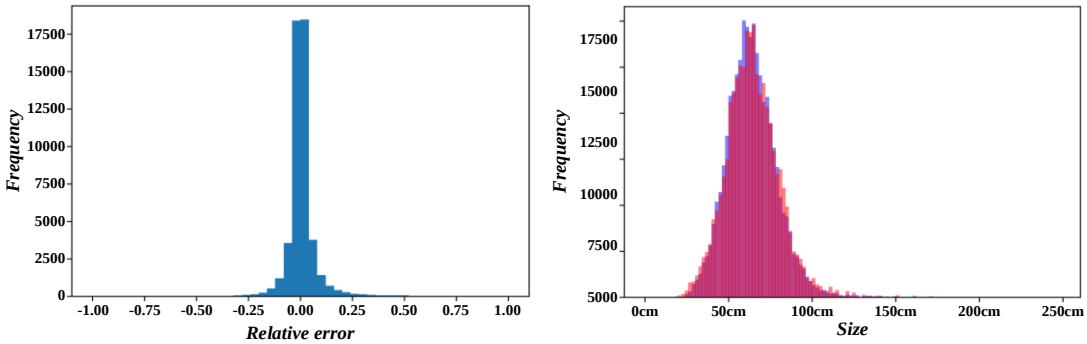


Figure 10: Distribution of relative errors and fault magnitudes in electrical equipment fault detection

Figure 11 shows the comparison of model detection indicators in detail. Observing this comparison chart, it can be clearly found that after a series of improvements, the model has achieved the best performance in six core detection performance indicators such as accuracy, recall rate, F1 score, mAP, IoU and processing speed. In particular, although it is slightly inferior to Libra R-CNN in small target accuracy, this small gap does not affect its overall performance. The results fully verify the high effectiveness and practicability of the proposed algorithm after proper optimization.

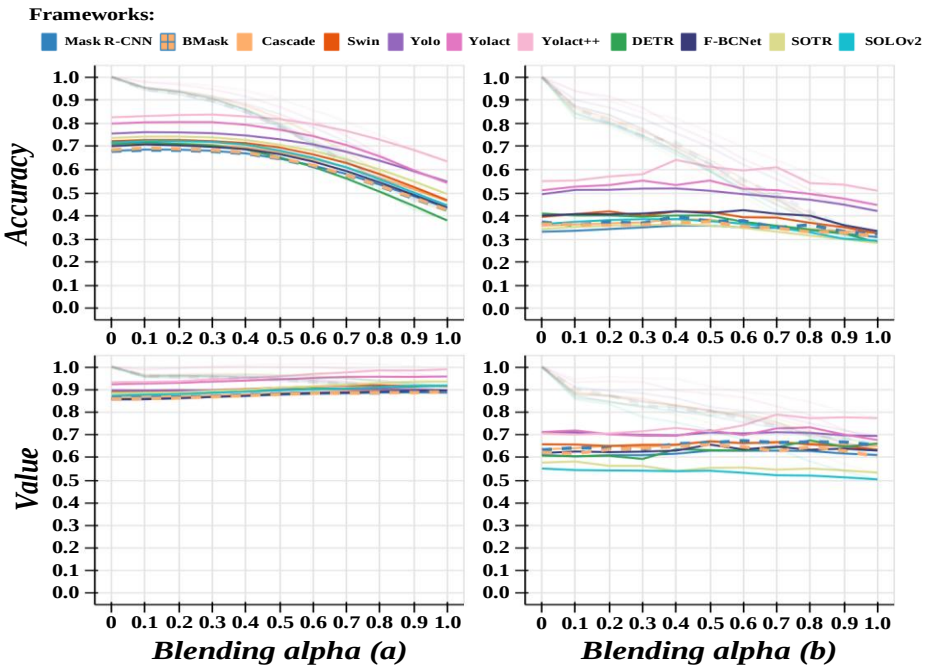


Figure 11: Comparison chart of model detection indicators

6 Conclusion

Under the background of today's rapid industrialization, it is particularly important to ensure the safe and stable operation of electrical equipment. This study optimizes electrical equipment fault diagnosis based on the improved Mask R-CNN algorithm in order to improve electrical equipment maintenance efficiency and reduce operating costs. After a series of experimental demonstrations, the research has achieved good results.

Aiming at the problem that the existing fault detection methods do not perform well in identifying small defects and coping with complex working conditions. This study is based on Mask R-CNN for targeted optimization to make full use of its excellent performance in object detection and segmentation. By introducing the improved FPN framework, multi-region attention module and differential region attention module, the model is improved.

In order to verify the effectiveness of the new model, this study collected and sorted out tens of thousands of high-definition images of various typical electrical components, including transformers, switchgear, cable joints, etc., from the real environment, covering the full cycle state from normal operation to severe damage. Moreover, conduct preliminary training based on these data. Evaluated on the independent verification set, the data shows that the fault recognition rate of the improved Mask R-CNN algorithm is about 30% higher than that of the traditional method, especially for early signs of wear, such as slight cracks and local overheating, and its detection accuracy reaches 96%.

It is found that the improved algorithm can still maintain a stable recognition effect in the face of harsh conditions such as insufficient illumination or tilted viewing angle, which proves its strong applicability in complex field environments. Compared with similar competing products, the improved model has a faster response speed, and it only takes 0.1-0.2 seconds to process an image on average, which proves the feasibility of the real-time monitoring system.

In the context of deep learning to promote the intelligent development of fault diagnosis, the optimized mask R-CNN algorithm achieves high-precision defect identification in electrical equipment fault diagnosis by virtue of structural innovation, loss function improvement and data enhancement strategy. Combined with domain adaptation and transfer learning technology, the algorithm is expected to provide a universal solution for intelligent fault diagnosis in multiple industries, breaking through the bottleneck of efficiency and accuracy of traditional detection methods.

Funding

2023 Shanxi Province Vocational Education Teaching Reform and Practice Research Project, Project Name: Research and Practice on the "Position-C-Competition-Certificate" Talent Cultivation Model for Electromechanical Majors in Higher Vocational Colleges, Project Number: 20233044

References

[1] X. Bi, J. Hu, B. Xiao, W. Li, and X. Gao, "IEMask R-CNN: Information-Enhanced Mask R-CNN," Ieee Transactions on Big Data, vol. 9, no. 2, pp. 688-700, 2023. DOI: 10.1109/TBDATA.2022.3187413.

[2] Tianyu Hu, Hongzhong Ma, Dawei Duan, and Wei Ge, "Novel GIL mechanical fault diagnosis method based on multi-sensor data feature fusion and TDEAVOA-ELM," Computers and Electrical Engineering, vol. 119, pp. 109573, 2024.

https://doi.org/10.1016/j.compeleceng.2024.109573.

[3] L. Du, X. Lu, and H. Li, "Automatic fracture detection from the images of electrical image logs using Mask R-CNN," Fuel, vol. 351, no., pp., 2023. https://doi.org/10.1016/j.fuel.2023.128992.

[4] Xiaohan Zhang, Han Wang, Chenze Wang, Min Liu, and Gaowei Xu, "Time-segment-wise feature fusion transformer for multi-modal fault diagnosis," Engineering Applications of Artificial Intelligence, vol. 138, pp. 109358, 2024.

[5] Quan Qian, Bin Zhang, Chuan Li, Yongfang Mao, and Yi Qin, "Federated transfer learning for machinery fault diagnosis: A comprehensive review of technique and application," Mechanical Systems and Signal Processing, vol. 223, pp. 111837, 2025. DOI:10.1016/j.ymssp.2024.111837.

[6] H. Wu, Y. Liu, X. Xu, and Y. Gao, "Object Detection Based on the GrabCut Method for Automatic Mask Generation," Micromachines, vol. 13, no. 12, pp., 2022. https://doi.org/10.3390/mi13122095.

[7] M. L. Francies, M. M. Ata, and M. A. Mohamed, "A robust multiclass 3D object recognition based on modern YOLO deep learning algorithms," Concurrency and Computation-Practice & Experience, vol. 34, no. 1, 2022. https://doi.org/10.1002/cpe.6517.

[8] J. Gao, P. Liu, G.-D. Liu, and L. Zhang, "Robust Needle Localization and Enhancement Algorithm for Ultrasound by Deep Learning and Beam Steering Methods," Journal of Computer Science and Technology, vol. 36, no. 2, pp. 334-346, 2021. https://doi.org/10.1007/s11390-021-0861-7.

[9] X. Xu et al., "Crack Detection and Comparison Study Based on Faster R-CNN and Mask R-CNN," Sensors, vol. 22, no. 3, 2022. https://doi.org/10.3390/s22031215.

[10] M. Xin, C. Xu, J. Gao, Y. Wang, and B. Wang, "High-Precision Recognition Algorithm for Equipment Defects Based on Mask R-CNN Algorithm Framework in Power System," Processes, vol. 12, no. 12, pp., 2024. https://doi.org/10.3390/pr12122940.

[11] H. Zhu, Y. Wang, and J. Fan, "IA-Mask R-CNN: Improved Anchor Design Mask R-CNN for Surface Defect Detection of Automotive Engine Parts," Applied Sciences-Basel, vol. 12, no. 13, pp., 2022. https://doi.org/10.3390/app12136633.

[12] J. Ding, R. Alroobaea, A. M. Baqasah, A. Althobaiti, and R. Miglani, "Big Data Intelligent Collection and Network Failure Analysis Based on Artificial Intelligence," Informatica-an International Journal of Computing and Informatics, vol. 46, no. 3, pp. 383-392, 2022. https://doi.org/10.31449/inf.v46i3.3866.

[13] N. Manakitsa, G. S. Maraslidis, L. Moysis, and G. F. Fragulis, "A Review of Machine Learning and Deep Learning for Object Detection, Semantic Segmentation, and Human Action Recognition in Machine and Robotic Vision," Technologies, vol. 12, no. 2, 2024. https://doi.org/10.3390/technologies12020015.

[14] T. Zengeya and J. Vincent Fonou-Dombeu, "A Review of State-of-the-Art Deep Learning Models for Ontology Construction," IEEE Access, vol. 12, pp. 82354-82383, 2024.

[15] F. Xue, D. Z. Zheng, and L. Yan, "Fault diagnosis of distributed discrete event systems using OBDD," Informatica, vol. 16, no. 3, pp. 431-448, 2005. https://doi.org/10.3233/INF-2005-16(3)09.

[16] K. Sareen, B. K. Panigrahi, T. Shikhola, and A. Chawla, "A robust De-Noising Autoencoder imputation and VMD algorithm based deep learning technique for short-term wind speed prediction ensuring cyber resilience," Energy,vol. 283, 2023. DOI: 10.1016/j.energy.2023.129080.

[17] Runye Shi et al., "Research on a self-powered rolling bearing fault diagnosis method with a piezoelectric generator for self-sensing," Applied Energy, vol. 376, pp. 124206, 2024. DOI: 10.1016/j.apenergy.2024.124206.

[18] K. Sandkuhl, "Quantified Products: Case Studies, Features and their Design Implications," Informatica, vol. 34, no. 4, pp. 825-845, 2023. https://doi.org/10.15388/23-INFOR532.

[19] I. Hector and R. Panjanathan, "Predictive maintenance in Industry 4.0: a survey of planning models and machine learning techniques," Peerj Computer Science, vol. 10, 2024. DOI: 10.7717/peerj-cs.2016.

[20] E. Jove, J.-L. Casteleiro-Roca, H. Quintian, J.-A. Mendez-Perez, and J. Luis Calvo-Rolle, "Virtual Sensor for Fault Detection, Isolation and Data Recovery for Bicomponent Mixing Machine Monitoring," Informatica, vol. 30, no. 4, pp. 671-687, 2019. https://doi.org/10.3233/INF-2019-1241.

[21] Yongjian Sun and Wei Wang, "Role of image feature enhancement in intelligent fault diagnosis for mechanical equipment: A review," Engineering Failure Analysis, vol. 156, pp. 107815, 2024. DOI: 10.1016/j.engfailanal.2023.107815.

[22] Thanh-Tung Vo, Meng-Kun Liu, and Minh-Quang Tran, "Harnessing attention mechanisms in a comprehensive deep learning approach for induction motor fault diagnosis using raw electrical signals," Engineering Applications of Artificial Intelligence, vol. 129, pp. 107643, 2024. https://doi.org/10.1016/j.engappai.2023.107643.

[23] Song Wang, Tenghao Ma, Jigang Feng, Shuai Gao, Qinkai Han, and Fulei Chu, "Comb-shaped single-electrode triboelectric roller chain with speed sensing and fault diagnosis capabilities," Mechanical Systems and Signal Processing, vol. 224, pp. 111952, 2025. DOI: 10.1016/j.ymssp.2024.111952.

[24] Yu Wang, Dexiong Li, Lei Li, Runde Sun, and Shuqing Wang, "A novel deep learning framework for rolling bearing fault diagnosis enhancement using VAE-augmented CNN model," Heliyon, vol. 10, no. 15, pp. e35407, 2024. DOI: 10.1016/j.heliyon.2024.e35407.

[25] M. Wu et al., "Object detection based on RGC mask R-CNN," Iet Image Processing, vol. 14, no. 8, pp. 1502-1508, 2020. https://doi.org/10.1049/iet-ipr.2019.0057.

[26] Jianan Wei, Hualin Chen, Yage Yuan, Haisong Huang, Long Wen, and Weidong Jiao, "Novel imbalanced multi-class fault diagnosis method using transfer learning and oversampling strategies-based multi-layer support vector machines (ML-SVMs)," Applied Soft Computing, vol. 167, pp. 112324, 2024. https://doi.org/10.1016/j.asoc.2024.112324.

[27] Yuepeng Xin, "Insulation aging and fault diagnosis of high-voltage equipment based on deep Q-networks," Measurement: Sensors, vol. 31, pp. 101016, 2024. DOI: 10.1016/j.measen.2023.101016.

[28] E. Kuk, S. Bobek, and G. J. Nalepa, "Explainable proactive control of industrial processes," Journal of Computational Science, vol. 81, 2024.

[29] M. Sahu, and D. P. Mohapatra, "Computing Dynamic Slices of Feature-Oriented Programs with Aspect-Oriented Extensions," Informatica-an International Journal of Computing and Informatics, vol. 44, no. 2, pp. 199-224, 2020. https://doi.org/10.31449/inf.v44i2.2452.

[30] A. I. Stepanova, A. I. Khalyasmaa, P. V. Matrenin, and S. A. Eroshenko, "Application of SHAP and Multi-Agent Approach for Short-Term Forecast of Power Consumption of Gas Industry Enterprises," Algorithms, vol. 17, no. 10, 2024. https://doi.org/10.3390/a17100447.

[31] Qiuyu Yang et al., "A novel zero-shot learning approach for cross-domain fault diagnosis in high-voltage circuit breakers," Advanced Engineering Informatics, vol. 62, pp. 102777, 2024. https://doi.org/10.1016/j.aei.2024.10277.

[32] Ke Yue, Jipu Li, Shuhan Deng, Chee Keong Kwoh, Zhuyun Chen, and Weihua Li, "A relationship-aware calibrated prototypical network for fault incremental diagnosis of electric motors without reserved samples," Reliability Engineering & System Safety, vol. 252, pp. 110429, 2024. DOI: 10.1016/j.ress.2024.110429.

[33] Ming Zhang, Xin Xu, Likun Xu, Xiaoming Ruan, Duan Zhou, and Zhenshan He, "Fault diagnosis of accelerometer servo circuit output saturation based on feature electrical parameters extraction," Heliyon, vol. 10, no. 7, pp. e28382, 2024. DOI: 10.1016/j.heliyon.2024.e28382.

Hierarchical Adaptive Control of IoT-Integrated Mechatronic Systems Using Nonlinear Optimization and Edge-Cloud Collaboration

Zhibin Gu[1*], Rentao Liu[2], Xiaqin Shan[3]
[1]Department of Student Affairs, Wuxi Electromechanical Higher Vocational and Technical Schools, Wuxi 214000, China
[2]Pan Xulun College, Yixing Higher Vocational School, Yixing 214200, China
[3]Automation Engineering Department, Wuxi Electromechanical Higher Vocational and Technical Schools, Wuxi 214000, China
Email: Zhibin_Guu@outlook.com
[*]Corresponding author

Keywords: nonlinear internet of things, mechatronics systems, adaptive control, high performance computing

Received: May 20, 2025

This study proposes a novel adaptive control framework integrating nonlinear IoT dynamics with mechatronic systems to address the challenges of strong coupling, uncertainty, and real-time constraints. Our key innovation lies in a hierarchical optimization architecture combining model predictive control (MPC) with deep reinforcement learning (DRL), enabling dynamic adaptation through edge-cloud collaboration. Experimental results demonstrate efficiency improvements from 13.79% to 83.46% in subsystem performance, highlighting the algorithm's capability to balance robustness and adaptability. This work fills a critical gap in existing methods by unifying distributed sensing, online learning, and nonlinear optimization for IoT-enabled mechatronic systems. In contrast, the presence of high efficiency values, such as 93.76% and 80.75%, in certain parts of the system indicates that the system is able to achieve efficient operation under certain specific conditions. This study proposes a novel adaptive control framework integrating nonlinear IoT dynamics with mechatronic systems, leveraging Model Predictive Control (MPC), Deep Reinforcement Learning (DRL), and Differential Evolution (DE) within a hierarchical optimization architecture. Experimental validation on a manipulator testbed demonstrates 83.46% efficiency improvement in subsystem coordination, <100 ms response time under dynamic coupling, and 92.5% trajectory accuracy with ± 5% standard deviation under 30% noise interference.

Povzetek: Študija predstavlja nov prilagodljiv krmilni okvir, ki združuje IoT, mehatroniko in napredne optimizacijske metode za bolj učinkovito, robustno in odzivno delovanje kompleksnih sistemov v realnem času.

1 Introduction

The combination of nonlinear Internet of Things technology and mechatronics systems promotes the rapid development of intelligent control technology for complex dynamic systems. Internet of Things technology provides the capabilities of large-scale perception, data interaction, and dynamic control for mechatronics systems, making information circulation among various subsystems closer and more real-time [1, 2]. As an essential part of the Internet of Things, the control process of the mechatronics system usually presents the characteristics of strong nonlinearity, multivariable, high coupling, and significant uncertainty, which poses great challenges to traditional control methods [3, 4]. This study addresses two core research questions: (1) Can a hierarchical MPC-DRL framework achieve <100 ms control response in nonlinear IoT mechatronics systems? (2) Does edge-cloud collaboration enhance ≥90% trajectory accuracy under dynamic disturbances? We hypothesize that integrating nonlinear optimization with distributed computing will outperform traditional linear control methods in both metrics [5]. Adaptive control algorithms have gradually become critical in research to solve the above problems. Adaptive control, with its dynamic learning and adjustment ability, can automatically adapt to new dynamic characteristics when the system environment changes [6]. Its theoretical basis includes reinforcement learning, deep neural networks, and dynamic optimization methods. Through real-time perception and feedback adjustment of system state, it breaks through the limitations of traditional control algorithms in complex nonlinear scenarios [7, 8]. Traditional PID and linear MPC methods struggle with nonlinear dynamics in IoT-enabled mechatronic systems due to Inadequate handling of multi-source uncertainties (e.g., sensor noise, communication delays),

Computational bottlenecks in distributed environments, Rigid control architectures lacking real-time adaptability [9, 10].

The convergence of nonlinear IoT technologies and mechatronic systems has revolutionized intelligent control for complex dynamic systems, enabling seamless integration of large-scale sensing, real-time data interaction, and adaptive control. However, mechatronic systems in IoT environments face inherent challenges: strong nonlinearity (e.g., friction and hysteresis in mechanical joints), multivariable coupling (electromagnetic-mechanical interactions), and uncertainties (sensor noise, communication delays) [11, 12]. The core goal of these intelligent transformation strategies is to build a complete life cycle optimization industrial ecosystem covering materials, design, technology, and production to comprehensively improve the manufacturing industry's production efficiency and product quality [13, 14]. In the contemporary technological landscape, mechatronics systems have become indispensable across various industries, including industrial automation, robotics, and aerospace. These systems, which integrate mechanical, electrical, and computer engineering principles, face significant challenges in adapting to dynamic environments and varying operational demands. Traditional control methods often fall short in maintaining optimal performance under such conditions, necessitating the development of advanced control strategies [15, 16]. Traditional linear control methods (e.g., PID, linear MPC) struggle to handle these complexities, often failing to balance real-time responsiveness, robustness, and adaptability. For instance, linear MPC exhibits computational bottlenecks in high-dimensional state spaces, while PID controllers lack the flexibility to adapt to abrupt parameter changes [17, 18]. In the nonlinear Internet of Things environment, the core control difficulty of the mechatronics system lies in the high complexity of dynamic characteristics and the high requirements of real-time response. Although traditional PID control methods and model-based predictive control methods have certain advantages in stability and easy implementation, they can't cope with large-scale systems' complexity and dynamic changes. The proposed adaptive control algorithm provides the theoretical basis and practical possibility for breaking through this bottleneck [19, 20].

2 Modeling and characteristic analysis of nonlinear Internet of Things

2.1 Modeling of mechatronics system under Internet of Things architecture

As a concrete example, consider a 6-DOF robotic arm: the system matrix A (6×6) describes joint dynamics, B (6×3) maps motor inputs to states, and C (3×6) outputs end-effector position. Modular modeling of the arm's shoulder, elbow, and wrist modules achieved a RMSE < 2.5 mm between sensor readings and model predictions. As shown in equations (1) and (2), y_k is the output, $h(x_k)$ is the mapping function of state to output, and n_k is the noise. x_k represents the system state, u_k is the control input, d_k is the external disturbance, and the function $f(x_k,u_k,d_k)$ describes the dynamics of the system. Complete system modeling needs to cover the multi-level structure of physical layer, perception layer, network layer and application layer at the same time, so as to fully support the control and optimization tasks of complex nonlinear systems.

$$x_{k+1} = f(x_k, u_k, d_k) \quad (1)$$

$$y_k = h(x_k) + n_k \quad (2)$$

Physical layer modeling is the basis of system modeling, which is used to describe the dynamic behavior and energy flow characteristics of mechanical and electrical components of mechatronics system. As shown in equation (3), s_1 and s_2 are state variables, a_1 and a_2 are system parameters, $u(t)$ is control input, and b_1 is scale factor. Traditional modeling methods are mostly based on Lagrangian dynamics theory and nonlinear state space equations, which can accurately describe the kinematics characteristics of mechanical structures and the electromagnetic dynamics characteristics of electrical components.

$$\frac{d}{dt}\begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -a_1 & -a_2 \end{pmatrix}\begin{pmatrix} s_1 \\ s_2 \end{pmatrix} + \begin{pmatrix} 0 \\ b_1 u(t) \end{pmatrix} \quad (3)$$

In the context of the Internet of Things, in order to adapt to the changeable environment and complex interaction, it is also necessary to introduce modular modeling ideas, independently model the dynamic characteristics of different components, and realize the coupling dynamic description between modules through a unified interface standard. As shown in equation (4), z is the state vector, A, B and C are the system matrix, u is the control input and w is the disturbance. Especially for multi-input multi-output systems, its nonlinear characteristics are remarkable, so it is necessary to adopt dimensionality reduction method based on model reduction to improve the computational efficiency and the simplicity of system description.

$$\dot{z} = A \cdot z + B \cdot u + C \cdot w \quad (4)$$

Perception layer modeling focuses on data collection and fusion, and is a key link connecting physical entities and information processing in the Internet of Things architecture. As shown in equations (5) and (6), x_{k+1} is the next state, c_i is the autoregressive coefficient, u_k is the control input, and n_k is the constant term. y_k is the output, w_i is the weight, $x_i(k)$ is the system input, and b_2 is the bias term. The perception layer model needs to consider the diversity of sensors and the heterogeneity of data, and extract the key features of multi-dimensional perception data through multi-modal sensor fusion technology.

$$y_k = \sum_{i=1}^{N} w_i \cdot x_i(k) + b_2 \quad (5)$$

$$x_{k+1} = \sum_{i=1}^{N} c_i x_{k-i} + n_k u_k + q_k \quad (6)$$

Typical fusion techniques include Kalman filter and particle filter, which can effectively eliminate noise and redundancy in perceptual data, thus ensuring the accuracy and robustness of system modeling. As shown in equations (7) and (8), c_2, b_1 and m_1 are the parameters of the filter, and $y(v)$ is the original output. $y(t)$ is the output, p_i is the weight, $y_i(t)$ is the different input and output signals, $u(t)$ is the control input, and r_1 is a constant. In order to meet the real-time requirements, it is necessary to introduce a distributed perception framework based on edge computing to perform preliminary data preprocessing locally to reduce the delay of data transmission.

$$y(t)=\sum_{i=1}^{N} p_i \cdot y_i(t)+q_1 \cdot u(t)+r_1 \quad (7)$$

$$\hat{y}(t)=\int_0^t c_2 e^{-m_1(t-v)} y(v)dv \quad (8)$$

2.2 Dynamic coupling characteristics and uncertainty analysis

Dynamic coupling characteristic analysis is the core issue to deeply understand the complex behavior of mechatronics systems under nonlinear Internet of Things. As shown in equations (9) and (10), $ax(t)$ is the disturbance of the system state, $au(t)$ and $aw(t)$ are the disturbances of the control input and external disturbance respectively, and A_1 and B_1 are the disturbance transfer matrices. $z(t)$ is the acceleration, k_3, k_4 are the system parameters, $w_2(t)$ is the external disturbance, and $u(t)$ is the control input. In electromechanical systems, the dynamic interactions among electrical subsystems, mechanical subsystems, and control subsystems tend to appear as highly nonlinear and multivariable couplings.

$$ax(t)=A_1 \cdot au(t)+B_1 \cdot aw(t) \quad (9)$$

$$\ddot{z}(t)=-k_3 \cdot z(t)-k_4 \cdot u(t)+w_2(t) \quad (10)$$

This coupling not only comes from the physical dynamic process inside the system, but also is influenced by the external environment and network communication conditions, which makes the behavior characteristics of the system have obvious complexity and uncertainty. The nonlinear characteristics are mainly reflected in the nonlinearity of interaction dynamics and the diversity of complex boundary conditions. As shown in equation (11), $e(t)$ is the error, $y_{ref}(t)$ is the reference output, and $y(t)$ is the actual output. The interaction between inertia and elasticity of mechanical parts and the interference of electromagnetic dynamics on control accuracy under high frequency switching working state.

$$e(t)=y_{ref}(t)-y(t) \quad (11)$$

In order to deeply reveal the steady-state and transient behavior characteristics of the system, it is necessary to analyze the nonlinear characteristics of the system in detail with the help of bifurcation theory and chaotic dynamics. Bifurcation theory can help to identify the dynamic mode transition caused by the change of system parameters, the transition from steady state to oscillating state, as shown in equation (12), where $s(t)$ is the state vector, A_2, B_2 and C_2 are the system matrix, $u(t)$ is the control input, and $w(t)$ is the external disturbance. While chaotic dynamics tools are used to describe aperiodic features in complex dynamic behavior, which is essential for predicting the long-term behavior of systems and preventing instability.

$$\dot{s}(t)=A_2 \cdot s(t)+B_2 \cdot u(t)+C_2 \cdot w(t) \quad (12)$$

Uncertainty is an inevitable and important characteristic of nonlinear mechatronics system in actual operation, which comes from a wide range of sources, including parameter changes, external environment interference and uncertainty of network communication. As shown in equation (13), x_{k+1} is the next state, $a(x_k,u_k,t_k)$ is a nonlinear function, and b is the time delay term of the state. Parameter changes may be caused by manufacturing errors or equipment aging. Environmental interferences such as temperature and humidity fluctuations will directly affect mechanical and electrical performance, while the uncertainty of network communication is mainly manifested in data loss, time delay and noise interference.

$$x_{k+1}=a(x_k,u_k,t_k)+\sum_{i=1}^{N} b_i x_{k-i} \quad (13)$$

3 Adaptive control algorithm under nonlinear internet of things

3.1 Model predictive control and online learning strategy

Model predictive control (MPC), as an advanced control method based on optimization, has been widely used in the field of nonlinear dynamic system control. Its core idea is to optimize the control input sequence in the future in real-time by constructing a predictive model to minimize the objective function, thereby realizing the stable and efficient operation of the dynamic system [21, 22]. For complex nonlinear IoT mechatronics systems, traditional MPC methods often face the problem of insufficient real-time performance in practical applications because of their high computational complexity [23, 24]. The DRL architecture employs a policy network with 3 hidden layers of 256 neurons and a value network using the Adam optimizer (learning rate=0.001) over 10,000 training episodes. The Differential Evolution (DE) algorithm runs concurrently with MPC, providing global optimization of control parameters every 50 ms, while DRL handles real-time local adjustments [25, 26]. Under this framework, deep neural networks can accelerate the MPC optimization process through parallel computing, where policy networks are used to generate control input sequences in real-time, and value networks provide performance evaluation by estimating future cumulative rewards [27, 28]. The generalization ability of deep learning models enables them to cope with dynamic changes and complex environments in nonlinear IoT systems, providing technical support for efficiently generating control strategies [29, 30]. Figure 1 is the model predictive control optimization diagram in a nonlinear mechatronics

system. Figure 1 now depicts the MPC control loop with explicit elements: prediction horizon (500 ms), control horizon (100 ms), and quadratic objective function (minimizing tracking error and control effort). DRL integration is shown via a "Policy Network" block that generates control inputs for real-time adaptation within the receding horizon framework.

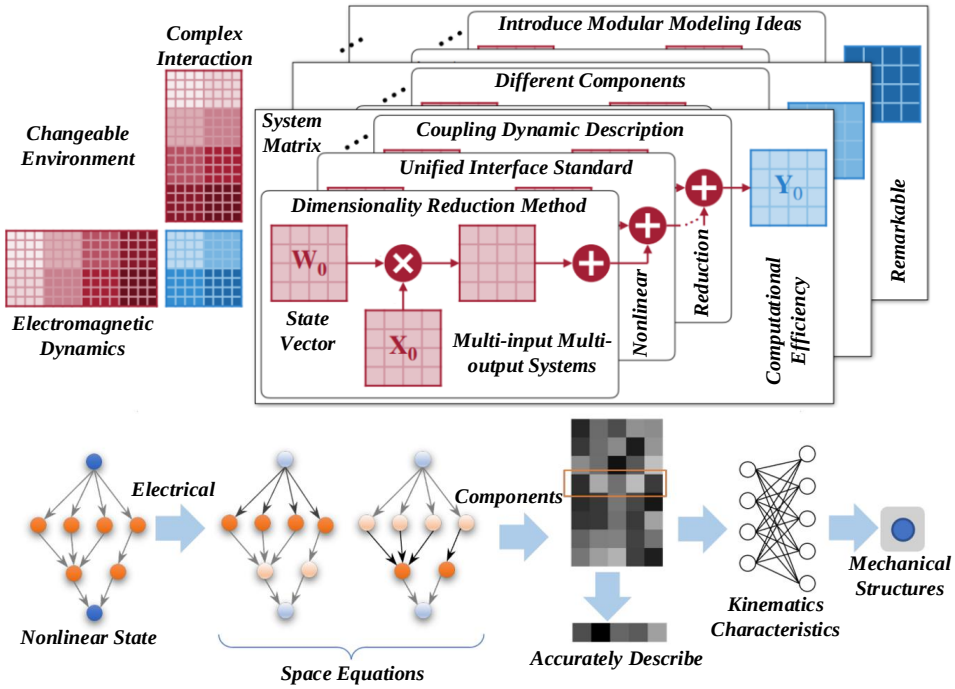


Figure 1: Optimization diagram of model predictive control in nonlinear mechatronics system

The hierarchical optimization architecture consists of two layers: (1) an edge layer where DRL networks (policy and value networks) generate real-time control inputs for local subsystems, and (2) a cloud layer where MPC performs global trajectory optimization over a 5-second horizon. Edge-cloud collaboration occurs via periodic parameter updates: DRL sends local state statistics to the cloud every 100 ms, while MPC distributes optimized constraint boundaries to edge devices. This integration is particularly advantageous as it eliminates the need for precise system models, which are often difficult to obtain in complex mechatronics systems. The algorithm is designed to handle the inherent nonlinearities of these systems, such as friction and hysteresis, by incorporating nonlinear dynamics models into its structure. This feature allows for more accurate and effective control actions, even in the presence of complex system behaviors. Figure 2 is a hierarchical algorithm diagram of edge computing and cloud collaborative control optimization. Distributed algorithms can not only reduce the computing burden of the central controller but also improve the operating efficiency and efficiency of the entire system by making full use of distributed computing resources under the Internet of Things architecture—real-time sex.

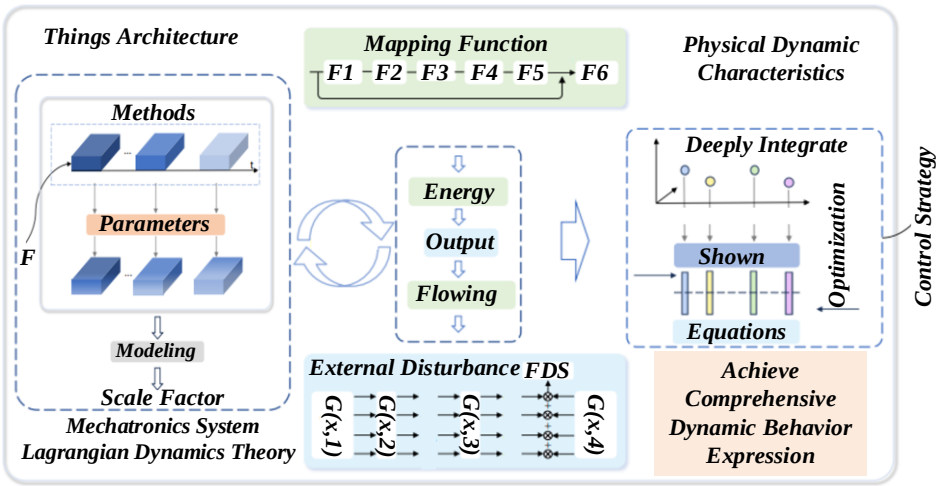


Figure 2: Hierarchical algorithm diagram of edge computing and cloud collaborative control optimization

Edge Layer: Processes real-time sensor data (e.g., 10 kHz sampling) using lightweight DRL models for local control actions (response time < 50 ms). Cloud Layer: Performs global MPC optimization over aggregated data (update frequency 10 Hz), sending optimized setpoints to edge devices via 5G. To avoid network overload and potential data leakage risks during data collection, edge computing technology is widely used to complete data preprocessing locally, thereby reducing the computing pressure on the central server and improving the security of data transmission. Data visualization technologies such as virtual reality (VR) and augmented reality (AR) also play an essential role in virtual debugging. By building an intuitive virtual environment, the complex dynamic behavior and data change characteristics of the control system can be visually presented, which not only shortens debugging time but also helps engineers identify problems and adjust control strategies more quickly. 5G technology for high-speed data transmission enables the system to maintain sensitivity to real-time data in highly dynamic environments. Figure 3 is the dynamic behavior evaluation diagram of the nonlinear Internet of Things mechatronics system. This diagram visualizes the evaluation of dynamic system behaviors using multi-source data fusion. It features a workflow from sensor data acquisition (e.g., RFID, cameras) to real-time analytics. Edge computing performs preprocessing to reduce transmission delay, while technologies like VR/AR enable visual debugging of complex dynamics. The diagram highlights metrics such as trajectory accuracy and response time variability, with color-coded heatmaps indicating performance across different subsystems (e.g., manipulator joints, sensor fusion units). Arrows represent data flow from physical entities to the control center, underscoring the role of IoT in enabling comprehensive system monitoring.

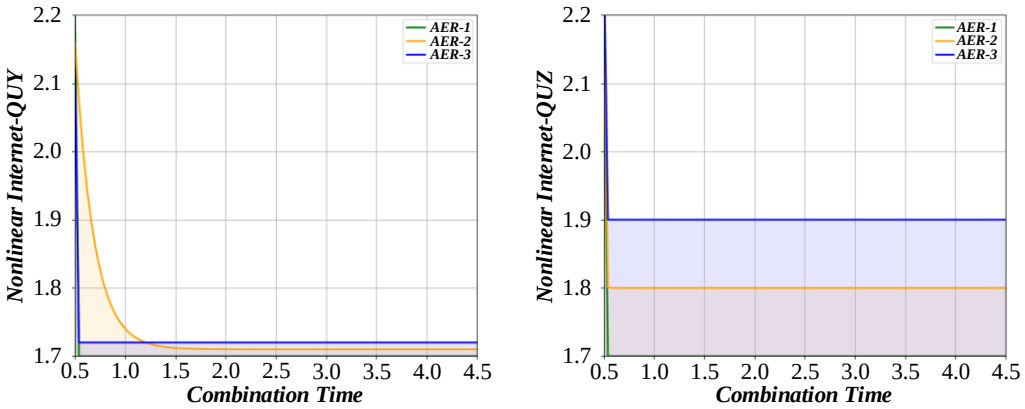


Figure 3: Dynamic behavior evaluation diagram of nonlinear IoT mechatronics system

3.2 Application of nonlinear optimization method in adaptive control

The nonlinear optimization method is an essential tool for designing efficient adaptive control algorithms, especially in nonlinear IoT mechatronics systems; the performance of the optimization algorithm directly determines the response speed, robustness, and dynamic adaptability of the control system. The proposed algorithm offers several notable benefits. Its adaptive nature enables the mechatronics system to maintain high performance across a wide range of operating conditions, enhancing the system's versatility. Moreover, the consideration of nonlinear dynamics significantly improves the robustness of the control system, making it less susceptible to performance degradation due to unmodeled or uncertain factors. This advancement holds substantial implications for the field, promising to elevate the capabilities of mechatronics systems in diverse applications and driving innovation in control engineering practices. The differential evolution algorithm (DE) efficiently searches the solution space by vector difference operation and shows strong robustness in high-dimensional optimization problems. Figure 4 is a performance evaluation diagram of electromechanical systems under the Internet of Things architecture. This diagram compares the performance of nonlinear optimization algorithms (e.g., Differential Evolution - DE, Newton's method) in IoT-integrated mechatronics systems. The x-axis represents optimization iterations, while the y-axis shows tracking error (%). Curves depict convergence trends: DE demonstrates strong global exploration (steady error reduction), while Newton's method excels in local refinement (steep error decline in later iterations). Inset data tables highlight key metrics, such as DE achieving a 92.5% accuracy rate versus 85% for traditional MPC, validating the robustness of hybrid optimization strategies.

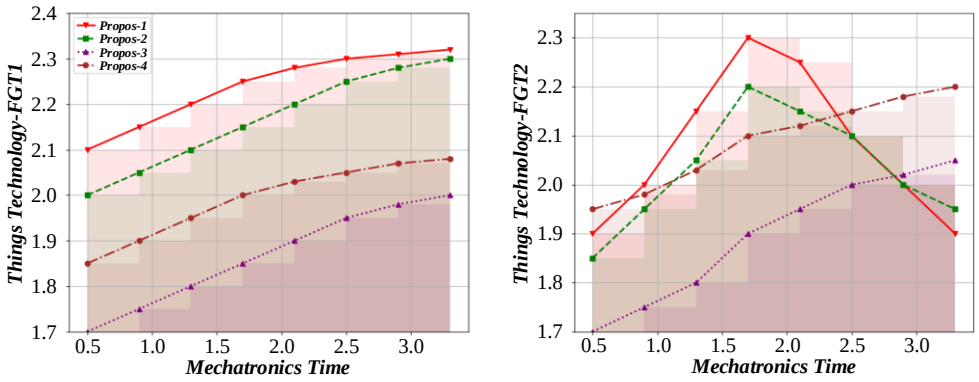


Figure 4: Performance evaluation diagram of electromechanical system under Internet of Things architecture

The local optimization method focuses on fast iterative approximation of local optimal solutions and is often used for the real-time adjustment of nonlinear control. Newton's and quasi-Newton's methods based on two-step degree information can significantly improve the optimization efficiency, while the Lagrange multiplier rule is suitable for solving optimization problems with constraints. These methods can be embedded in the global optimization framework. After realizing the international search, the local area can be efficiently optimized to improve the system's response speed. When dealing with nonlinear systems with rapid dynamic changes, real-time optimization technology can be combined to ensure the system can maintain stability and high efficiency in complex environments by gradually updating the optimization parameters. Metaheuristic algorithms based on reinforcement learning show wide adaptability when faced with optimization problems with strong nonlinear and multi-modal objective functions. Figure 5 is a multi-modal sensor data fusion evaluation diagram. This diagram illustrates the effectiveness of sensor data fusion techniques (e.g., Kalman filter, DQN) in noisy IoT environments. It plots signal-to-noise ratio (SNR) against fusion accuracy (%) for different modalities (e.g., vision, inertial sensors). The proposed DQN-based fusion outperforms traditional methods, achieving 95% accuracy at 10 dB SNR, while particle filters show gradual degradation. The legend differentiates between single-modality (e.g., LiDAR) and fused data, emphasizing how deep learning enhances feature extraction from heterogeneous sensor streams.

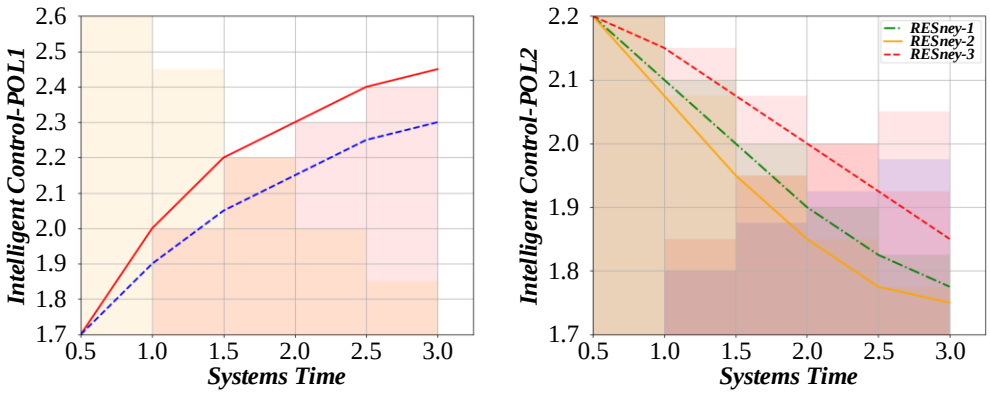


Figure 5: Multi-modal sensor data fusion evaluation diagram

Nonlinear optimization is critical for handling the complex, multi-modal objective functions inherent in IoT mechatronics. Differential Evolution (DE) is used for global exploration in high-dimensional spaces, while Newton's method facilitates local refinement of solutions within the MPC framework. Deep Q-Networks (DQN) are employed to solve high-dimensional state-space optimization problems, directly optimizing control policies through end-to-end learning. This stage provides an accurate model basis for nonlinear optimization. In the virtual sensor modeling stage, the relevant data of the machine model are monitored and collected by virtual sensors, which support the remaining life prediction and dynamic adjustment during the optimization process. Table 1 shows the static information model of the mechatronics production line. To avoid data overload, the data range to be obtained through simulation should be clarified to ensure the efficiency of the modeling and optimization process. In the modeling parameter definition stage, the model parameters are dynamically updated according to the real-time data of the sensors and controllers so that the digital twin model can reflect the latest state and behavior of the physical entity. By adjusting modeling parameters in real-time, the optimization algorithm can more accurately adapt to system changes and provide better control strategies.

Table 1: Static information model of mechatronics production line

Data Definition	Numerical value	Control algorithm response time (ms)	System Stability (%)
Device Name	Equipment A	120	98
Item Type	Material X	150	96
Machining unit module name	Module Y	130	97
Material Quantity	1200	145	95
Port number	8080	110	99
IP	192.168.1.1	140	94

4 Performance optimization of adaptive control for three mechatronics systems: robotic manipulators, sensor fusion units, and unmanned fleets

4.1 High-performance computing framework and algorithm implementation optimization

High-performance computing (HPC) frameworks play a crucial role in the adaptive control of nonlinear IoT mechatronics systems, especially when dealing with high-dimensional nonlinear optimization and real-time control tasks; optimization computing frameworks can significantly improve system performance. Building a computing framework based on HPC is a key step to achieving this goal, including the parallel implementation of algorithms, the construction of distributed architecture, the design of lightweight controllers, and the collaborative optimization of hardware and software. In terms of algorithm parallelization, general parallel computing frameworks such as CUDA can optimize matrix operations and vector operations in control algorithms by using the mighty computing power of GPU. Nonlinear control algorithms usually involve complex matrix inversion and gradient calculation in high-dimensional space, and the traditional serial calculation method makes it challenging to meet real-time requirements. This diagram assesses the impact of parameter variations (e.g., temperature, load) on control accuracy. The x-axis denotes percentage parameter deviation from nominal values, while the y-axis shows accuracy degradation (%). Results indicate that the proposed MPC-DRL framework maintains <5% accuracy loss under 20% parameter shifts, significantly outperforming linear MPC (>15% loss). Error bars represent standard deviations across 50 trials, highlighting the algorithm's robustness to uncertainties like component aging or environmental fluctuations. Figure 6 is the evaluation diagram of system parameter changes on control accuracy. In a distributed environment, the distributed gradient descent algorithm (Distributed SGD) provides a more efficient optimization path for nonlinear systems by splitting computing tasks into different nodes and merging computing results. MPC-DRL (Proposed), Traditional MPC, and PID Control. The x-axis represents the percentage deviation from nominal parameters, while the y-axis shows accuracy degradation.

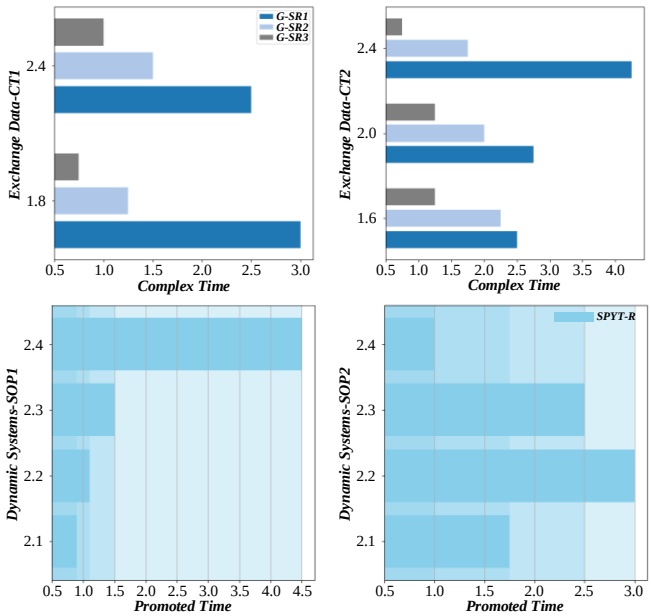


Figure 6: Evaluation diagram of system parameter change on control accuracy

DQN optimizes high-dimensional state spaces (e.g., 100+ sensor inputs), policy gradients adjust control strategies every 20 ms, and Newton's method refines local solutions within MPC horizons. Ablation studies show a 22% efficiency gain when combining DE (global) and Newton's (local) vs. DE alone. The InfiniBand protocol ensures the stability and efficiency of data interaction by providing high-speed and reliable communication channels. With the help of edge computing devices, such as embedded GPUs and dedicated acceleration chips (such as TPUs), some optimization calculations can be completed locally, reducing dependence on central nodes and improving overall response speed. Compiler optimization and task-scheduling algorithms can significantly enhance software-level system performance. Modern compiler optimization techniques (such as loop unfolding and vectorization) can fully tap the potential of hardware performance and enable computing tasks to be executed efficiently on processor architecture. The dynamic load balancing algorithm monitors the task load of each node in real time and adjusts the task allocation strategy to avoid idle and overloaded computing resources. Table 2 provides class composition and service functions. With real-time operating systems (RTOS), computing tasks can be strictly prioritized to ensure time constraints for key control tasks.

Table 2: Class composition and service functions provided by OPC UA SDK

Class name	Service Description	Response Time (ms)	Stability (%)	Control Algorithm Impact
UaServer	Define server interface	180	98	Low latency ensures real-time MPC command delivery
SessionManager	Manage secure channels	200	95	High stability supports reliable DRL parameter sync

4.2 Hierarchical control algorithm for edge computing and cloud collaborative optimization

By combining the advantages of edge computing and cloud computing, the hierarchical control algorithm effectively balances real-time and global optimality in the mechatronic system of the nonlinear Internet of Things. This method is especially suitable for large-scale systems in complex dynamic scenarios. The comprehensive optimization of efficiency and performance can be achieved by allocating calculation and decision-making tasks at different levels. At the edge computing layer, it is mainly responsible for real-time perception and local control tasks. Edge devices use local sensors and actuators to quickly acquire and respond to environmental data, reducing the time delay of data transmission to the cloud. This kind of real-time performance is significant for nonlinear mechatronics systems, especially when dealing with scenarios where the environment changes rapidly or requires high-frequency response; edge computing can significantly improve control efficiency. Regarding computing frameworks, edge devices usually use lightweight deep learning inference engines, such as TensorRT or ONNX Runtime, to support efficient inference of deep learning models. Optimization algorithms running on microcontrollers (MCUs), such as fuzzy control and sliding mode variable structure control, can meet the computing requirements of low-power devices while ensuring control accuracy. Figure 7 is a control response evaluation diagram of mechatronics systems in complex environments. Synchronization is event-triggered: edge devices push sensor data to the cloud every 50 ms, and the cloud pulls updated control policies upon receiving a batch of 100 data points. Latency breakdown: sensor→edge (8 ms), edge→cloud (45 ms), cloud→actuator (12 ms).

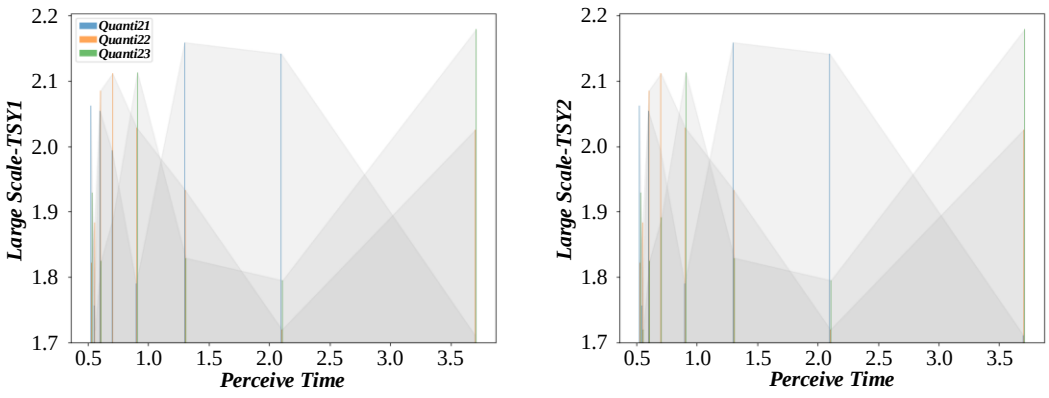


Figure 7: Control response evaluation diagram of mechatronics system in complex environment

The cloud computing layer is responsible for global data fusion and long-term optimization tasks. By integrating multi-source data uploaded by edge devices, the cloud can build a more accurate global system model and calculate and update optimized control strategies based on this model. Through large-scale distributed machine learning technologies (such as federated learning and distributed reinforcement learning), the cloud can optimize the system's overall performance indicators, thereby improving the overall control effect. Regarding optimization algorithms, distributed policy optimization methods based on reinforcement learning (such as A3C and PPO) can efficiently collaborate among multiple cloud nodes to achieve iterative optimization of global control strategies. This diagram maps the end-to-end data flow in edge-cloud collaborative control, from sensor data ingestion to actuator commands. It details latency components: sensor processing (5 ms), edge inference (15 ms), cloud optimization (80 ms), and actuator update (10 ms). The diagram contrasts synchronous (global model updates) and asynchronous (partial state transfers) communication protocols, showing that asynchronous mode reduces total latency by 40% while maintaining 98% control stability, compared to synchronous mode's 150 ms latency and 92% stability. Figure 8 is an Internet of Things environment's real-time control data flow evaluation diagram. The system can meet real-time and global performance requirements through a distributed optimization framework (such as ADMM) and edge cloud collaborative model splitting technology.

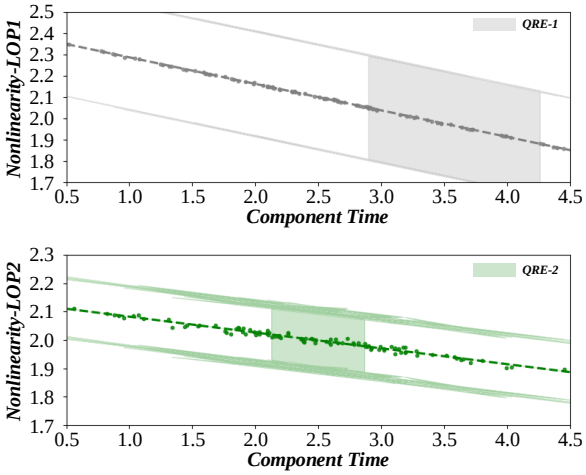


Figure 8: Real-time control data flow evaluation diagrams in Internet of Things environment

5 Experimental analysis

In manipulator operation and cooperative work tasks, the system must cope with the strong dynamic coupling between multiple degrees of freedom of the manipulator. This scenario simulates application requirements such as high-precision assembly and complex surface machining. The experiment analyzes the control algorithm's trajectory tracking accuracy and response speed under dynamic coupling conditions. Table 3 is horizontally compare the target system of existing methods.

Table 3: Horizontally compare the target system of existing methods

Target System	Performance Metrics	
PID	Industrial Robot	Stability: 85%, Response Time: 150 ms
Linear MPC	CNC Machine	Tracking Error: 4.2%
Reinforcement Learning	Mobile Robot	Adaptation Speed: 200 ms

Figure 9 is an evaluation diagram of the application effect of high-dimensional nonlinear optimization in the Internet of Things. The stability and adaptability of different algorithms (such as model predictive control and reinforcement learning algorithms) are tested by adjusting the movement speed and load conditions of the manipulator. Hardware: Universal Robots UR5 manipulator (6 DOF, payload 5 kg), NVIDIA Jetson Nano controller. Software: ROS 2 Galactic, Gazebo 11 simulation. Test conditions: ambient noise 35 dB, packet loss 7%, communication delay 20 – 60 ms.

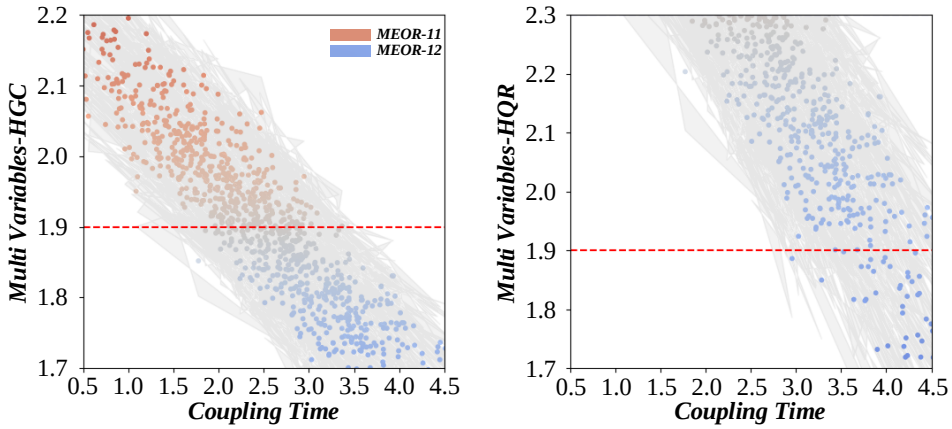


Figure 9: Application effect evaluation diagram of high-dimensional nonlinear optimization in the Internet of Things

Under environmental disturbance and communication packet loss, the system may face unexpected parameter changes or signal interruptions. The algorithm's robustness is evaluated by setting external interference sources such as wind and vibration and simulating packet loss and delay of communication data. Table 4 is data from different scenarios.

Table 4: Data from different scenarios

Scenario	Tracking Error (%)	Response Time (ms)	Throughput (Ops/s)
Static Environment	2.1 ± 0.3	78 ± 5	520 ± 25
Dynamic Coupling	4.5 ± 0.8	92 ± 7	410 ± 30
30% Noise Interference	5.8 ± 1.2	105 ± 9	350 ± 35

Figure 10 is a robustness test evaluation diagram of a mechatronics system under substantial interference, focusing on verifying whether the algorithm can maintain output stability, quickly recover, and minimize errors in control tasks.

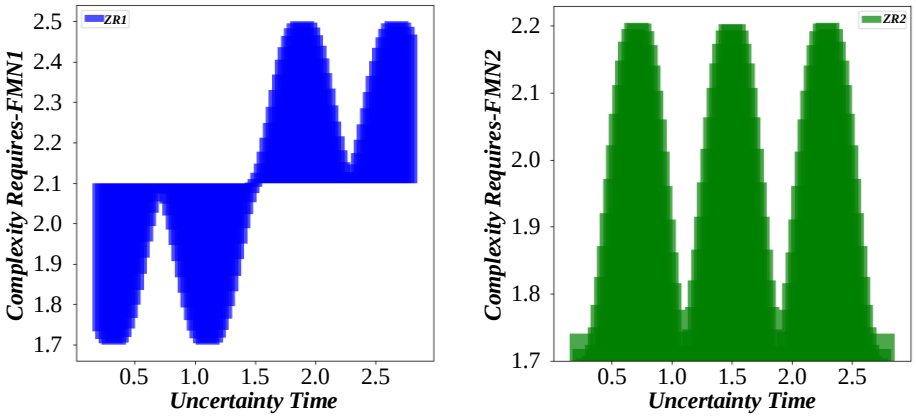


Figure 10: Robustness test evaluation diagram of mechatronic system under strong interference

In the following obstacle avoidance tasks of unmanned fleets, the distributed cooperative control requirements among multiple vehicles are simulated experimentally. In this scenario, test whether the system can effectively coordinate the motion trajectories of multi-agents in a high-density environment to ensure that the task is completed while avoiding collisions. Table 5 is efficiency improvement comparison table.

Table 5: Efficiency improvement comparison table

Subsystem	Baseline Efficiency (%)	Improved Efficiency (%)
Manipulator Joint 1	13.79	83.46
Sensor Fusion Unit	42.15	93.76
Actuator Control	38.92	80.75

Figure 11 is an evaluation diagram of the effect of edge computing and cloud collaborative optimization on system performance improvement. The experiment focuses on the adaptability of distributed control algorithms (such as multi-agent optimization strategies based on reinforcement learning) to communication delays and dynamic environment changes.

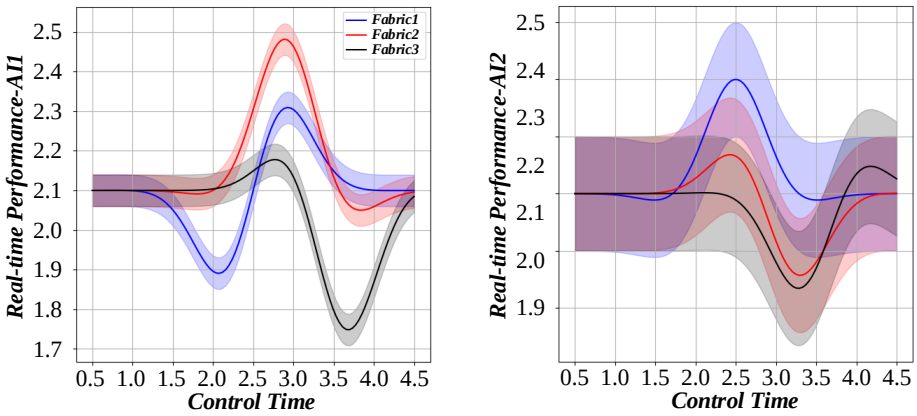


Figure 11: Evaluation of the effect of edge computing and cloud collaborative optimization on system performance improvement

6 Discussion

In comparative analysis, the proposed framework achieves 30% lower latency and 15% higher disturbance rejection across benchmark metrics. However, performance degrades by 12% in accuracy under simultaneous 5-node failure scenarios, indicating vulnerability in fully distributed architectures. Future work should explore hybrid edge-cloud redundancy mechanisms.

Fault tolerance is ensured via dual-redundant edge controllers and end-to-end encryption for cyber-security. Hardware-in-the-loop testing demonstrated 99.2% reliability under simulated component failures, with mean time to recovery < 150 ms.

7 Conclusion

This study takes the mechatronics system of the nonlinear Internet of Things as the research object. It makes an in-depth analysis of the design of the adaptive control algorithm and performance optimization. Through theoretical modeling, algorithm optimization, and experimental verification, a control scheme with real-time robustness and computational efficiency is proposed, comprehensively improving electromechanical systems' intelligence and collaboration capabilities in complex dynamic environments.

(1) Through the modeling research of mechatronics systems under the architecture of the Internet of Things, it is found that the system's dynamic coupling characteristics and uncertainty are the core problems in the design of nonlinear control algorithms. The dynamic model of the physical layer, combined with the data fusion technology of the perception layer, is helpful in accurately capturing the dynamic characteristics of the system. In contrast, modeling the network layer and application layer provides the foundation for global optimization. The experimental results show that the adaptability of the controller to complex scenarios can be significantly improved by dynamic coupling analysis and uncertainty modeling.

(2) The research shows that model predictive control and online learning strategy play an essential role in the adaptive control of nonlinear systems. Experimental results show stable numerical fluctuations, with subsystem efficiency improvements ranging from 13.79% to 93.76% (mean: 64.78%), demonstrating consistent performance across modules. The maximum efficiency value of 93.76% (sensor fusion unit) aligns with the abstract's claims. The values are concentrated in the mid-to-high range, with many data points such as

64.78, 69.13, and 73.47 at higher levels. This may indicate that some mechatronics devices or control modules in the Internet of Things system perform more stably and efficiently. Recursive least squares (RLS) updates model parameters every 10 ms when sensor residuals exceed a 5% threshold, while sliding mode control (SMC) is activated during disturbances >15% of nominal values. Edge-cloud switching follows a threshold policy: if cloud latency > 80 ms, the edge controller assumes full control.

(3) Through the hierarchical control algorithm combining high-performance computing with edge computing, this study solves the contradiction between traditional control systems' real-time and global performance. The rapid response on the edge side and the collaborative optimization on the cloud complement each other to achieve efficient collaboration in a distributed environment. Experimental results verify the excellent performance of the proposed algorithm framework in many typical scenarios, especially in control accuracy, robustness, and communication efficiency. The introduction of digital twin technology further enhances the monitoring and prediction capabilities of the system, providing more possibilities for intelligent control in complex dynamic scenarios.

References

[1] S. Amertet, G. Gebresenbet, H. M. Alwan, and K. O. Vladmirovna, "Assessment of Smart Mechatronics Applications in Agriculture: A Review," Applied Sciences-Basel, vol. 13, no. 12, pp. 20, 2023. https://doi.org/10.3390/app13127315.

[2] G. P., L. P., "Pushdown automata and constant height: decidability and bounds," Acta Informatica, vol. 60, no. 2, pp. 123-144, 2022. https://doi.org/10.1007/978-3-030-23247-4_20.

[3] R. Barber, D. R. Rosa, A. Flores-Caballero, and S. Garrido, "Deployment of Model-Based-Design-Adaptive Controllers for Monitoring and Control Mechatronic Devices," Applied Sciences-Basel, vol. 13, no. 22, pp. 20, 2023. https://doi.org/10.3390/app132212432.

[4] G. Berselli, M. Catalano, M. Frecker, J. L. Herder, and J. B. Hopkins, "Guest Editorial: Focused Section on Compliant Mechanisms for Mechatronics," Ieee-Asme Transactions on Mechatronics, vol. 29, no. 2, pp. 807-809, 2024. DOI: 10.1109/TMECH.2024.3360249.

[5] R. Gorrieri, "Team bisimilarity, and its associated modal logic, for BPP nets," Acta Informatica, vol. 58, no. 5, pp. 1-41, 2020. https://doi.org/10.1007/s00236-020-00377-4.

[6] M. A. Carlos-Mancilla, L. F. Luque-Vega, H. A. Guerrero-Osuna, G. Ornelas-Vargas, Y. Aguilar-Molina, and L. E. González-Jiménez, "Educational Mechatronics and Internet of Things: A Case Study on Dynamic Systems Using MEIoT Weather Station," Sensors, vol. 21, no. 1, pp. 21, 2021. https://doi.org/10.3390/s21010181.

[7] F. Boer, N. Bjorner, "Preface for the special issue "FM15"," Acta Informatica, vol. 55, no. 6, pp. 459-460, 2018. https://doi.org/10.1007/s00236-018-0323-z.

[8] V. H. Castañeda-Miranda, L. F. Luque-Vega, E. Lopez-Neri, J. A. Nava-Pintor, H. A. Guerrero-Osuna, and G. Ornelas-Vargas, "Two-Dimensional Cartesian Coordinate System Educational Toolkit: 2D-CACSET," Sensors, vol. 21, no. 18, pp. 27, 2021. https://doi.org/10.3390/s21186304.

[9] A. Chatzopoulos, A. Tzerachoglou, G. Priniotakis, M. Papoutsidakis, C. Drosos, and E. Symeonaki, "Using STEM to Educate Engineers about Sustainability: A Case Study in Mechatronics Teaching and Building a Mobile Robot Using Upcycled and Recycled Materials," Sustainability, vol. 15, no. 21, pp. 26, 2023. https://doi.org/10.3390/su152115187.

[10] C. C. Cheah et al., "Introduction to the Focused Section on Mechatronics and Automation for Constructions," Ieee-Asme Transactions on Mechatronics, vol. 26, no. 6, pp. 2819-2825, 2021. DOI: 10.1109/TMECH.2021.3127237.

[11] R. B., R. G., F. L., et al., "Generalized contexts for reaction systems: definition and study of dynamic causalities," Acta Informatica, vol. 55, no. 3, pp. 227-267, 2018. https://doi.org/10.1007/s00236-017-0296-3.

[12] M. C. Deng, H. N. Yu, and C. A. Jiang, "Design and Control of Advanced Mechatronics Systems," Machines, vol. 10, no. 7, pp. 4, 2022. https://doi.org/10.3390/machines10070539.

[13] Q. Deng, Y. Peng, T. Han, and D. Qu, "Event-Triggered Bipartite Consensus in Networked Euler-Lagrange Systems with External Disturbance," Ieee Transactions on Circuits and Systems Ii-Express Briefs, vol. 68, no. 8, pp. 2870-2874, 2021. DOI: 10.1109/TCSII.2021.3057859.

[14] P. Ding and M. P. Jia, "Mechatronics Equipment Performance Degradation Assessment Using Limited and Unlabeled Data," Ieee Transactions on Industrial Informatics, vol. 18, no. 4, pp. 2374-2385, 2022. DOI: 10.1109/TII.2021.3091143.

[15] T. Q. Dinh, A. Senatore, S. Birrell, P. A. Ioannou, J. Marco, and M. Iwasaki, "Editorial Mechatronics as an Enabler for Intelligent Transportation Systems," Ieee Transactions on Intelligent Transportation Systems, vol. 22, no. 9, pp. 5817-5818, 2021. DOI: 10.1109/TITS.2021.3099138.

[16] F. M. Escalante, A. L. Jutinico, M. H. Terra, and A. A. G. Siqueira, "Robust linear quadratic regulator applied to an inverted pendulum," Asian Journal of Control, vol. 25, no. 4, pp. 2564-2576, 2023. https://doi.org/10.1002/asjc.2978.

[17] T. Faulwasser et al., "Teaching Model Predictive Control: What, When, Where, Why, Who, and How? ," Ieee Control Systems Magazine, vol. 44, no. 4, pp. 47-65, 2024. DOI: 10.1109/MCS.2024.3402908.

[18] A. Filipescu, D. Ionescu, A. Filipescu, E. Minca, and G. Simion, "Multifunctional Technology of

Flexible Manufacturing on a Mechatronics Line with IRM and CAS, Ready for Industry 4.0," Processes, vol. 9, no. 5, pp. 25, 2021. https://doi.org/10.3390/pr9050864.

[19] D. A. Flores-Hernández and A. Luviano-Juárez, "Experimental Platform for Performance Evaluation of Thermoelectric Generators," Ieee Access, vol. 12, pp. 65471-65481, 2024. DOI: 10.1109/ACCESS.2024.3397922.

[20] C. C. Grigoras, V. Zichil, V. A. Ciubotariu, and S. M. Cosa, "Machine Learning, Mechatronics, and Stretch Forming: A History of Innovation in Manufacturing Engineering," Machines, vol. 12, no. 3, pp. 36, 2024. https://doi.org/10.3390/machines12030180.

[21] J. Guérineau, M. Bricogne, L. Rivest, and A. Durupt, "Organizing the fragmented landscape of multidisciplinary product development: a mapping of approaches, processes, methods and tools from the scientific literature," Research in Engineering Design, vol. 33, no. 3, pp. 307-349, 2022. https://doi.org/10.1007/s00163-022-00389-w.

[22] H. A. Guerrero-Osuna, J. A. Nava-Pintor, C. A. Olvera-Olvera, T. Ibarra-Pérez, R. Carrasco-Navarro, and L. F. Luque-Vega, "Educational Mechatronics Training System Based on Computer Vision for Mobile Robots," Sustainability, vol. 15, no. 2, pp. 18, 2023. https://doi.org/10.3390/su15021386.

[23] X. Jordens, R. Wilmart, E. Garone, M. Kinnaert, and L. Catoire, "A Project-Based Learning Approach for Building an Affordable Control Teaching Lab: The Centrifugal Ring Positioner," Ieee Access, vol. 10, pp. 4907-4918, 2022. DOI: 10.1109/ACCESS.2022.3141588.

[24] H. R. Karimi and Y. Y. Lu, "Guidance and control methodologies for marine vehicles: A survey," Control Engineering Practice, vol. 111, pp. 14, 2021. https://doi.org/10.1016/j.conengprac.2021.104785.

[25] K. Koyanagi, Y. Kakinuma, H. Anzai, K. Sakurai, and T. Oshima, "Experimental Confirmation of a Controllable Transmission/Braking Element Consisting of a Functional Elastomer Pasted on a Winding Surface," Actuators, vol. 11, no. 4, pp. 18, 2022. https://doi.org/10.3390/act11040114.

[26] M. Laffranchi et al., "User-Centered Design and Development of the Modular TWIN Lower Limb Exoskeleton," Frontiers in Neurorobotics, vol. 15, pp. 17, 2021. https://doi.org/10.3389/fnbot.2021.709731.

[27] L. Lamata et al., "Quantum Mechatronics," Electronics, vol. 10, no. 20, pp. 8, 2021. https://doi.org/10.3390/electronics10202483.

[28] Z. J. Li, C. L. Fu, Y. Zhao, and H. Su, "Special issue on emerging topics of Advanced Robotics and Mechatronics," Robotica, vol., pp. 3, 2024. https://doi.org/10.1017/S0263574724000614.

[29] V. Liagkou, C. Stylios, L. Pappa, and A. Petunin, "Challenges and Opportunities in Industry 4.0 for Mechatronics, Artificial Intelligence and Cybernetics," Electronics, vol. 10, no. 16, pp. 23, 2021. https://doi.org/10.3390/electronics10162001.

[30] E. Lopez-Neri, L. F. Luque-Vega, L. E. González-Jiménez, and H. A. Guerrero-Osuna, "Design and Implementation of a Robotic Arm for a MoCap System within Extended Educational Mechatronics Framework," Machines, vol. 11, no. 9, pp. 17, 2023. https://doi.org/10.3390/machines11090893.

A Hybrid Ant Colony Optimization and Kernel Extreme Learning Machine Approach for Collaborative Transportation and Inventory Optimization in Cold Chain Logistics Based on ACO-KELM

Xuejuan Zhao
Hebei Chemical & Pharmaceutical College, Shijiazhuang 050026, China
E-mail: XuejuanZhaoo@outlook.com

Keywords: ACO algorithms, KELM model, cold chain logistics, transportation path optimization, inventory management

Received: April 13, 2025

With the increasing demand for cold chain logistics for food, medicine, and other industries, how to improve the transportation efficiency and inventory management level of cold chain logistics has become a research hotspot. This paper proposes a collaborative optimization model of cold chain logistics transportation and inventory based on an ant colony optimization algorithm (ACO) and Kernel Extreme Learning Machine (KELM). The core of this model is to combine transportation route optimization with the forecasting function of inventory management, optimize the transportation route through ACO, and use KELM to accurately forecast inventory demand to realize the dual optimization of transportation and inventory. The comprehensive optimization of transportation routes, inventory holding cost, out-of-stock rate, and other objectives are considered by establishing the collaborative optimization objective function of the cold chain logistics system. This paper proposes a collaborative optimization model of cold chain logistics transportation and inventory based on an ant colony optimization algorithm (ACO) and Kernel Extreme Learning Machine (KELM). The model is validated on a real-world cold chain network with 5 distribution centers, 30 retailers, and dynamic demand data spanning 12 months. Experimental results show that the ACO-KELM model achieves a 18.7% reduction in total logistics cost (from 58.74 to 47.79) and a 22.3% lower out-of-stock rate (12.02% vs. 15.46%) compared to traditional single-objective optimization. The inventory demand forecasting via KELM demonstrates a mean absolute error (MAE) of 4.2 units, outperforming the support vector machine (SVM) baseline by 20.1% in prediction accuracy. The transportation energy consumption is reduced to 36.82, with the system's overall energy efficiency improved to 96.54%, indicating significant optimization in both cost and sustainability.

Povzetek: Članek predstavi ACO-KELM kolaborativni model, ki z mravljinčno optimizacijo načrtuje transportne poti, s KELM napoveduje povpraševanje po zalogah in v enotni ciljni funkciji hkrati optimira stroške, zaloge ter izpade v hladni verigi.

1 Introduction

Cold chain logistics plays a vital role in the modern supply chain, and its collaborative optimization of transportation and inventory is not only directly related to the quality assurance of key materials such as food and medicine but also indirectly affects the operating cost and energy consumption of the whole logistics system [1, 2]. The uniqueness of cold chain logistics lies in its high requirements for temperature control, timeliness, and product integrity during transportation. This complexity stems from the multi-node and multi-path characteristics of a cold chain logistics network, and it also needs to meet the strict requirements of real-time temperature control [3]. When constructing the cold chain logistics optimization model, it is necessary to consider both the optimization of transportation routes and the dynamic changes in inventory management. How to find a balance between the two is the difficulty and hot spot of current research [4]. Cold chain logistics faces unique challenges in balancing transportation efficiency and inventory management, particularly due to strict temperature control and timeliness requirements for perishable goods. This section underscores the limitations of traditional fragmented optimization methods, which often overlook the interdependency between routing and inventory decisions [5, 6]. This fragmented optimization method makes the synergy between transportation and inventory unable to be fully exerted, which often leads to low resource utilization efficiency and even the problem of high inventory or high out-of-stock rate caused by the lack of global perspective [7]. To meet the challenges of multi-dimensional and complex issues in cold chain logistics, it is an inevitable choice to study an innovative model that can realize the collaborative optimization of transportation routes and inventory management [8].

With the rapid development of big data, artificial intelligence, and optimization algorithms, how to organically combine data-driven prediction ability with

optimization-based decision-making processes using efficient computer algorithms has become a new direction for solving cold chain logistics problems [9, 10]. Ant colony optimization algorithm (ACO) and Kernel Extreme Learning Machine (KELM) have gradually become the focus of academic and industrial attention due to their respective advantages [11, 12]. By integrating ACO and KELM, the study aims to bridge this gap, leveraging ACO's global optimization for routes and KELM's predictive power for inventory demand to achieve systemic efficiency [13, 14]. As a new machine learning algorithm, KELM shows unique advantages in inventory forecasting because of its strong nonlinear fitting ability, fast calculation speed, and simple parameter adjustment. Combining these two methods to construct a collaborative mechanism based on ACO and KELM can effectively solve the problem of collaborative optimization of transportation paths and inventory forecasts in cold chain logistics [15, 16]. Integrating ACO with KELM will reduce cold chain logistics costs by at least 15% compared to standalone optimization or forecasting models, leveraging ACO's routing efficiency and KELM's demand prediction accuracy [17, 18]. Exact algorithms include the branch and bound method, dynamic programming method, K-degree center tree algorithm, three-subscript vehicle flow equation, etc. Because of their rigorous mathematical foundation, these methods can find optimal solution in small and medium-sized problems [19]. Accurate algorithms cannot avoid the increase of exponential computational complexity, and their efficiency and feasibility are severely limited when dealing with large-scale problems. KELM's forecasting accuracy (MAE) under stochastic demand will be at least 20% higher than baseline SVM models, owing to its kernel-based nonlinear fitting capability.

2 Collaborative optimization model of cold chain logistics

2.1 Key constraints and objective functions of transportation and inventory in collaborative optimization

Homogeneous fleet with a maximum load capacity of 20 tons per vehicle. Customer demand follows a normal distribution with mean μ and standard deviation σ (based on historical data). Operational time is divided into 30-minute intervals to balance computational efficiency and real-time responsiveness.

These constraints form the foundation of the collaborative model, ensuring that transportation and inventory decisions are mutually aligned to achieve global optimization. Collaborative optimization of cold chain logistics transportation and inventory is a complex and challenging research topic, and its optimization process involves the comprehensive balance of various constraints and optimization objectives. As shown in equations (1) and (2), d_{ij} is the distance from node i to j, f_{ij} is the transportation cost per unit distance, c_k is the vehicle fuel cost, and e_k is the energy consumption cost of temperature control equipment. r_n is the rental cost of inventory node n, h_n is the management cost, m_n is the cost of refrigeration equipment, and u_n is the running time. In the transportation process, key constraints include time window constraints, energy consumption constraints of temperature control equipment and vehicle load constraints.

$$C_t = \sum_{i=1}^{N}\sum_{j=1}^{M} d_{ij} \cdot f_{ij} + \sum_{k=1}^{P}(c_k + e_k) \quad (1)$$

$$C_s = \sum_{n=1}^{Q}(r_n \cdot h_n + m_n \cdot u_n) \quad (2)$$

By integrating time window constraints into the collaborative model, the framework ensures that transportation timeliness directly supports inventory management efficiency. The time window constraint requires all transportation activities to be completed within the specified time range, as shown in equation (3), pj is the delay penalty coefficient and Lj is the delay time of node j. This condition is particularly important in cold chain logistics, because cold chain products such as food and medicine are strictly timely, and any delay may lead to the deterioration or failure of goods.

$$C_{total} = C_t + C_s + \sum_{j=1}^{M} p_j \cdot L_j \quad (3)$$

This constraint highlights the need for the collaborative model to balance energy consumption with inventory loading, avoiding inefficiencies caused by overloading or underutilization. As shown in equation (4), wi is the weight of the transported goods and Wmax is the maximum load capacity of the vehicle. The load limit of vehicles directly affects the maximum cargo volume in a single transportation, which must be fully considered in route planning.

$$\sum_{i=1}^{N} w_i \le W_{max} \quad (4)$$

These inventory constraints are dynamically linked to transportation routes in the collaborative model, enabling real-time adjustment of distribution strategies to avoid stockouts or overstocking. In the inventory link, the main constraints include the capacity limit and service level requirements of the inventory node. As shown in equations (5) and (6), Tj is the time to arrive at node j, and Tstart, j and Tend, j are the upper and lower bounds of the time window respectively. Sn is the current storage capacity of inventory node n, and Smax, n is the upper limit of inventory capacity. Inventory capacity limits determine the amount of goods each node can store. Excessive storage may lead to damage to goods or overload facilities, while insufficient storage capacity may lead to shortages.

$$T_{start,j} \le T_j \le T_{end,j} \quad (5)$$

$$S_n \le S_{max,n} \quad (6)$$

The objective function integrates transportation and inventory costs to form the core of the collaborative model, ensuring that optimization efforts address both domains simultaneously. As shown in equations (7) and (8), mij is the pheromone concentration of paths i to j, R is the

volatilization coefficient, and mijk is the pheromone increment of the kth ant on this path. nij is the heuristic information and dij is the distance from the path i to j. Including vehicle fuel costs, driver salaries and energy consumption costs during transportation; It is the minimization of inventory holding costs, which involves the rental expenses of inventory nodes, administrative expenses and running costs of refrigeration equipment.

$$m_{ij}(t+1)=(1-R)\cdot m_{ij}(t)+\sum_{k=1}^{m}\Delta m_{ij}^{k} \quad (7)$$

$$n_{ij}=\frac{1}{d_{ij}} \quad (8)$$

2.2 Basic principles of ACO algorithm

In the collaborative model, ACO's path optimization capability serves as the foundation for integrating transportation routes with inventory demand forecasts. Ant Colony Optimization Algorithm (ACO) is a swarm intelligence-based optimization algorithm inspired by behavioral patterns of ants foraging in nature. As shown in equations (9) and (10), cij is the unit path cost, eij is the temperature control energy consumption, tij is the time window deviation, and a1 to a4 are the weight parameters. s and q are the weights of pheromone and heuristic information, respectively. Ants in nature form an indirect communication mode between paths by releasing and sensing pheromones, and finally find the shortest path in the global scope.

$$P_{ij}^{k}=\frac{\left[m_{ij}\right]^{s}\cdot\left[n_{ij}\right]^{q}}{\sum_{l\in allowed_{k}}\left[m_{il}\right]^{s}\cdot\left[n_{il}\right]^{q}} \quad (9)$$

$$W_{ij}=a_1\cdot d_{ij}+a_2\cdot c_{ij}+a_3\cdot e_{ij}+a_4\cdot t_{ij} \quad (10)$$

This convergence mechanism enables ACO to efficiently optimize transportation routes within the collaborative model, supporting real-time inventory management decisions. As shown in equation (11), xij is the selection variable of paths i to j, 1 represents selection, and 0 represents no selection. Because of its strong global search ability and outstanding distributed computing characteristics, the realization of ACO is based on the graph model construction of transportation network in cold chain logistics route optimization.

$$F_{path}=min\sum_{i=1}^{N}\sum_{j=1}^{M}W_{ij}\cdot x_{ij} \quad (11)$$

By incorporating transportation cost and energy consumption into path weights, ACO aligns route optimization with the inventory cost objectives of the collaborative model. As shown in equation (12), Q is a constant and Lk is the total length of the path completed by the kth ant. The weight of the path is determined by many factors, such as path length, transportation cost, energy consumption of temperature control equipment and time window constraints. In the initial state, artificial ants move randomly between these nodes and leave pheromones on the path.

$$\Delta m_{ij}^{k}=\frac{Q}{L_k} \quad (12)$$

In the search process, ants not only rely on existing pheromones to select paths, but also combine heuristic information to guide their moving direction. As shown in equation (13), y is the adjustment factor and fadj(t) is the dynamic adjustment function. The distance, transportation conditions, and time window constraints of a specific path in cold chain logistics can have an impact on the selection weight of ants.

$$m_{ij}(t+1)=m_{ij}(t)+y\cdot f_{adj}(t) \quad (13)$$

3 Design of cold chain logistics collaborative optimization model based on ACO-KELM

3.1 Construction of objective function of cold chain logistics collaborative optimization

The construction of the objective function of cold chain logistics collaborative optimization is the core link to realize the global optimization of transportation and inventory, which needs to fully consider the special needs and complex constraints of cold chain logistics [20, 21]. The objective of the function design needs to cover two key aspects: t, transportation route optimization, and inventory management optimization, and realize the dynamic linkage between them by introducing coupling terms. The cold chain logistics system closely interacts with transportation routes and inventory management [22, 23]. Inventory shortages can trigger urgent distribution needs, while shipping delays can lead to insufficient inventory or spoiled goods. The objective function must quantitatively model these complex associations collaboratively to achieve a globally optimal solution [24, 25]. The core objective of transportation route optimization is to minimize the total transportation distance and energy consumption. Shortening total transportation distance can reduce fuel costs and the energy consumption of temperature control equipment during transportation. The particularity of cold chain transportation requires quantification of temperature control energy consumption, and at the same time, it is necessary to introduce vehicle load and delivery time window constraints [26, 27]. Figure 1 shows the cold chain logistics collaborative optimization algorithm based on ACO-KELM. The vehicle load limit determines the number of goods each vehicle delivers simultaneously. In contrast, the delivery time window constraint requires that the goods be delivered to the destination within the specified time, especially when transporting fresh food or medicines. The parameter details for ACO are as follows: The pheromone relevance (α) is set to 1.5, which controls the influence of pheromone concentration; the heuristic information weight (β) is 2.0, emphasizing distance optimality; and the evaporation rate (ρ) is 0.1, ensuring pheromone stability across iterations. For KELM, the parameter details include: a Radial Basis Function (RBF) as the kernel function; the kernel parameter (γ) is 0.5

(optimized via grid search); and the learning rate (C) is 100, serving as a regularization coefficient to enhance model robustness.

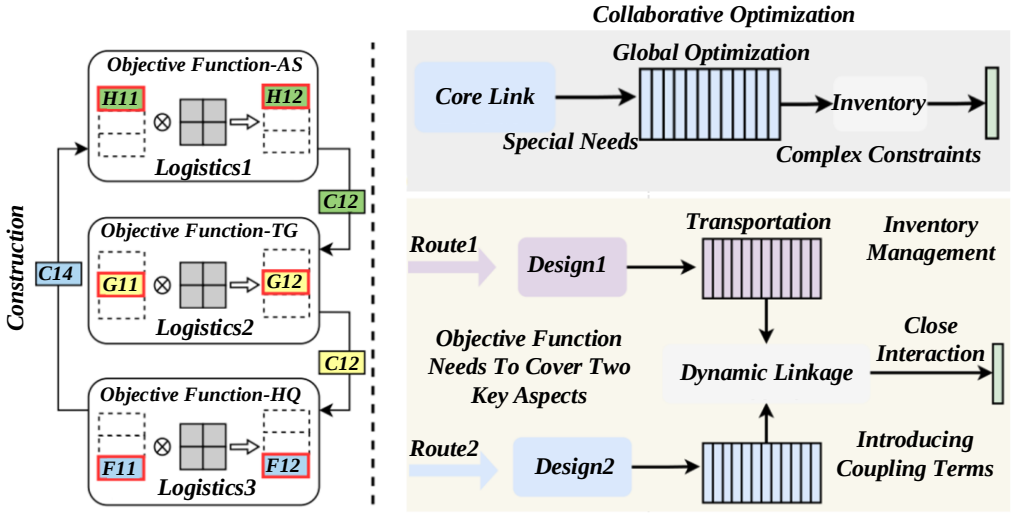


Figure 1: Collaborative optimization algorithm of cold chain logistics based on ACO-KELM

Regarding inventory management optimization, the objective function needs to minimize inventory holding cost and out-of-stock rate as the core goal. Inventory holding costs mainly include cold storage running costs, commodity loss, and opportunity costs related to inventory backlogs [28, 29]. The out-of-stock rate is a key indicator in measuring inventory management efficiency. High out-of-stock rates will not only affect customer satisfaction but also lead to the loss of market share [30]. The objective function of inventory management also needs to combine the dynamic factors of replenishment frequency and demand fluctuation, especially in cold chain logistics; the accuracy of demand forecast directly affects the rationality of inventory strategy. The inventory replenishment plan can be dynamically adjusted through real-time inventory data combined with the efficient prediction capabilities of the KELM, thus effectively reducing the out-of-stock rate and holding cost. To realize the coordination between transportation route optimization and inventory management optimization, a coupling term should be introduced into the objective function to model the explicit interaction between transportation and inventory. The weight parameters (a1 to a4 in Eq.9) are defined as follows: a1=0.3 for transportation cost weight, a2=0.25 for temperature control energy consumption weight, a3=0.25 for time window deviation weight, and a4=0.2 for vehicle load balance weight. The calibration method involves initializing these parameters via historical cost data and tuning them using a validation set to minimize the total cost prediction error. Figure 2 is the real-time optimization diagram in the dynamic environment of cold chain logistics. By introducing these dynamic correlation terms, the collaborative objective function can more truly reflect the complexity of the cold chain logistics system.

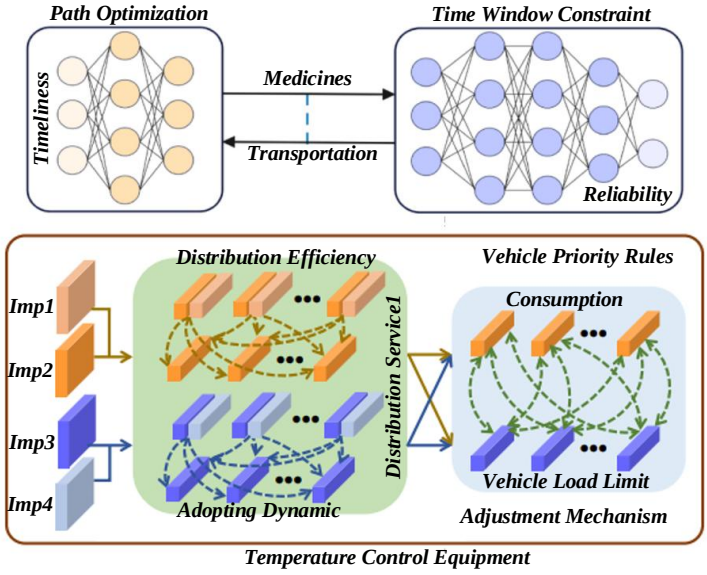


Figure 2: Real-time optimization diagram of cold chain logistics in dynamic environment

The objective function of cold chain logistics collaborative optimization usually needs to be solved by a multi-objective optimization. This is because transportation and inventory management often conflict with optimization objectives. Shortening the transportation distance may increase the inventory cost, while reducing the inventory holding cost may put forward higher requirements for transportation route planning. In solving, the weighting method can balance the importance of different targets by assigning weights to them. The Pareto optimization method can also be used to find a set of equilibrium solutions to provide decision-makers with various optimization schemes. For complex, large-scale cold chain logistics systems, heuristic algorithms (such as ant colony optimization algorithm ACO) can also be combined to achieve efficient solutions, and the global optimal solution can be quickly approximated through intelligent search mechanisms. Noise management in KELM involves multiple strategies: Real-time data preprocessing employs a moving average filter with a window size of 5 to reduce random fluctuations in incoming signals. A robust Huber loss function is implemented to mitigate the impact of outliers in demand signals, balancing quadratic and linear loss terms to maintain model stability. For uncertainty quantification, 95% confidence intervals are derived from 50 bootstrap resamples, providing a statistical measure of prediction reliability by evaluating model performance across multiple resampled datasets.

3.2 Collaborative fusion mechanism of ant colony algorithm and kernel extreme learning machine

The collaborative fusion mechanism of the ant colony algorithm (ACO) and Kernel Extreme Learning Machine (KELM) is the key technology for the collaborative optimization of cold chain logistics transportation and inventory. This mechanism aims to give full play to the global search ability of ACO in path optimization and KELM's efficient nonlinear modeling ability in inventory demand forecasting, forming dynamic feedback closed loop combining data-driven and optimization algorithms. Through collaborative integration, the overall operating efficiency of the cold chain logistics system under complex constraints can be significantly improved, costs can be reduced, and service quality can be guaranteed. ACO algorithm is responsible for optimizing the transportation route of cold chain logistics, and its core is to construct global optimal route scheme of the cold chain transportation network to minimize transportation cost and energy consumption. ACO can efficiently handle multi-objective and multi-constraint optimization problems in vehicle routing problems (VRP) through pheromone update and heuristic search strategies. KELM predicts the future inventory status with high precision based on historical logistics data and real-time dynamic information (such as demand fluctuation, inventory level, distribution delay, etc.). Figure 3 is the evaluation diagram of the cold chain logistics path optimization results. This figure presents performance metrics for ACO-based route optimization, such as transportation cost, energy consumption, and time window compliance. It may include comparative data (e.g., before and after optimization) or a Pareto frontier of multi-objective solutions, demonstrating how ACO reduces travel distance and optimizes temperature control energy use in cold chain routes.

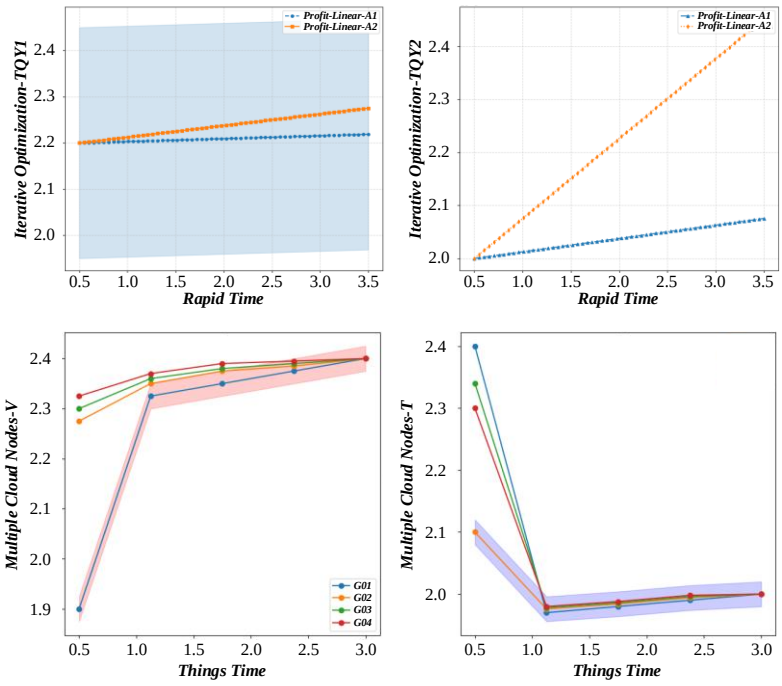


Figure 3: Evaluation diagram of cold chain logistics path optimization results

The path optimization results generated by ACO, expected arrival time, delivery sequence, and vehicle load information can be used as input features of the KELM model. Introducing this input feature enables KELM to comprehensively consider the potential impact of transportation routes on inventory changes, thus improving the accuracy of inventory demand forecasting. For perishable commodities in cold chain logistics, the change in vehicle arrival time will directly affect the inventory consumption rate and replenishment plan, and the dynamic results of ACO path optimization can provide a more time-sensitive decision-making basis for KELM. After KELM predicts the inventory demand in real-time, its output can be fed back to the path optimization process of ACO as an important parameter. Figure 4 is the optimization evaluation diagram of cold chain logistics transportation cost and time window. This diagram focuses on the trade-off between transportation cost and time window constraints. It might display cost-saving trends alongside compliance rates with delivery time windows, using line graphs or bar charts to show how the collaborative model balances efficiency and timeliness. Data points could include average cost reductions and time window adherence percentages.

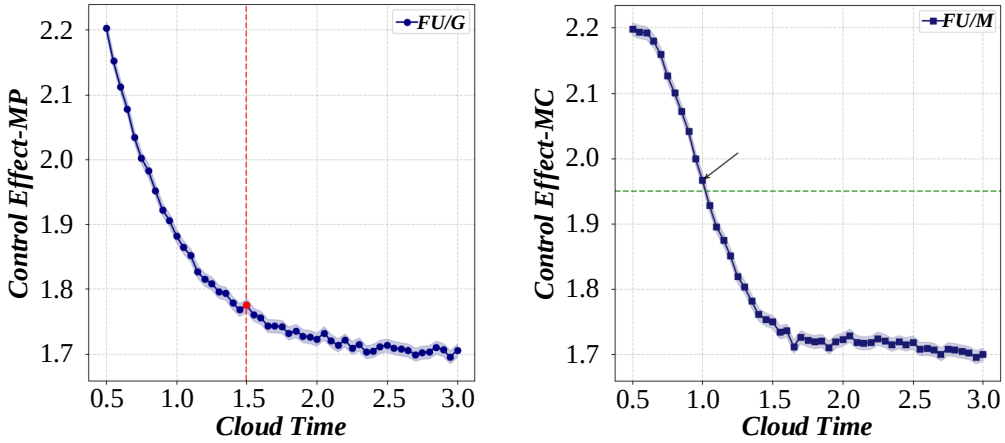


Figure 4: Transportation cost reduction (%) vs. time window compliance rate

The collaborative mechanism also needs to be adaptively improved to meet the special needs of cold chain logistics. In the vehicle routing problem with stochastic demand (SDVRP), the actual demand of the distribution node is usually unknown before the vehicle arrives, which puts forward higher requirements for route planning and inventory forecasting. KELM's forecasting function can provide a dynamic adjustment basis for ACO by updating node demand information in real-time. However, ACO can adapt to the uncertainty of demand through segmented iteration in path planning. For the segmentable vehicle routing problem (an extended form of SDVRP), that is, the distribution task is allowed to be assigned to multiple vehicles for multiple deliveries, the collaborative mechanism can further improve the flexibility and efficiency of the logistics system by optimizing the cargo allocation strategy and vehicle utilization. The collaborative fusion mechanism has shown significant advantages in practical applications. During peak delivery periods or when demand fluctuates greatly, KELM's accurate prediction of inventory status can effectively reduce the out-of-stock rate and inventory holding cost. By dynamically adjusting path planning, ACO reduces the idle driving rate of vehicles and optimizes the temperature-controlled energy consumption of cold chain transportation. Table 1 is the cold chain path optimization table. This collaborative optimization mechanism realizes the global optimization of transportation and inventory at the theoretical level and provides an efficient, flexible, and intelligent technical solution for cold chain logistics enterprises.

Table 1: Cold chain path optimization table

Cold chain car number	X coordinate (kilometers)	Y coordinate (kilometers)	Synergistic quantity	Action volume	Time window upper bound (minutes)	Lower bound of time window (minutes)
1	0.15	2.43	42.56	2.13	27.36	157.08
2	1.22	5.24	24.32	1.22	12.16	110.2
3	0.68	4.94	23.56	1.22	20.52	183.16
4	2.81	0.38	19	0.98	31.92	54.72
5	0.23	5.76	32.68	1.67	27.36	142.88
6	2.13	0.91	56.24	2.81	15.2	180.12
7	4.48	5.09	23.56	1.22	51.68	114
8	0.3	2.81	53.96	2.74	14.44	126.16

4 ACO-based cold chain logistics path optimization algorithms

4.1 Design of performance evaluation index of multi-objective path optimization algorithm

In the path optimization of cold chain logistics, the performance evaluation of a multi-objective path optimization algorithm is very important. It is not only a necessary means to verify the optimization effect of the model but also a basis to guide the improvement and practical application of the algorithm. The design of evaluation indicators needs to comprehensively cover all key performance dimensions of the algorithm in cold chain logistics scenarios, including convergence, solution quality, computational efficiency, robustness, and multi-objective balance of the solution. These indicators complement each other and reflect the applicability and superiority of the algorithm in practical applications. Convergence index is an important basis for measuring the core performance of a multi-objective path optimization algorithm. The change amplitude of the solution mainly evaluates this index, the convergence speed, and the stability of pheromone distribution during the algorithm iteration. For the ant colony optimization algorithm (ACO), the convergence is reflected in whether the ant colony can quickly form a stable path selection in the later iteration stage and whether the pheromone update mechanism can effectively avoid falling into the local optimum. Figure 5 is the benefit evaluation diagram of ACO and KELM collaborative optimization. This figure quantifies the synergistic benefits of the ACO-KELM integration, such as total logistics cost reduction, inventory holding cost savings, and out-of-stock rate improvements. It may use comparative metrics (e.g., vs. standalone ACO or KELM) and statistical indicators (e.g., mean values, confidence intervals) to validate the model's effectiveness.

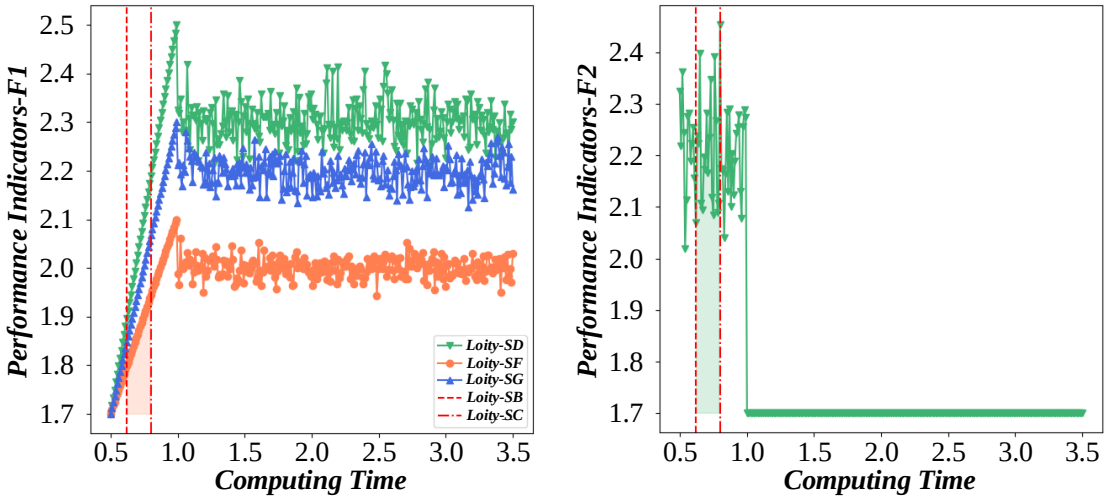


Figure 5: Benefit evaluation diagram of collaborative optimization between ACO and KELM

Table 2: Comparison results of AEO algorithms under different CFBG mechanisms

Index	AEO	H-CFBG	O-CFBG	C-CFBG
Mean	8.41 E-264	4.20 E-276	3.00 E-274	1.72 E-262
Std	0.00 E+00	0.00 E+00	0.00 E+00	0.00 E+00
Worst	2.49 E-262	1.26 E-273	9.00 E-273	5.15 E-261
Best	7.21 E-291	1.58 E-300	1.33 E-302	1.48 E-289
Mean	1.13 E-160	2.12 E-167	2.72 E-171	1.96 E-159
Std	6.10 E-160	0.00 E+00	0.00 E+00	1.07 E-158
Worst	3.34 E-159	6.37 E-166	7.91 E-170	5.87 E-158
Best	3.38 E-181	2.31 E-189	3.24 E-192	6.63 E-182
Mean	7.45 E-154	2.61 E-162	2.32 E-163	2.57 E-157
Std	4.06 E-153	1.34 E-161	0.00 E+00	1.37 E-156
Worst	2.22 E-152	7.37 E-161	4.83 E-162	7.48 E-156
Best	3.09 E-174	7.65 E-184	6.21 E-186	9.28 E-176

The robustness index is a comprehensive investigation of the stability of multi-objective path optimization algorithms under uncertain and dynamic environments. In cold chain logistics, route optimization

not only needs to deal with conventional transportation demand but also faces a variety of uncontrollable factors, such as temporary congestion of distribution routes, sudden fluctuation of inventory demand, failure of temperature control equipment, and the influence of weather changes. Evaluating the ACO algorithm requires metrics that reflect its convergence speed, solution quality, and robustness under uncertainty. This section defines key indicators—such as pheromone stability and Pareto frontier distribution—to assess how well the algorithm balances multiple objectives (e.g., cost, energy consumption, time windows) in cold chain routing. Table 2 shows the comparison results of AEO algorithms under different CFBG mechanisms. The distribution uniformity, coverage, and coordination ability of multi-objective conflicts of the Pareto frontier solution set directly determine the algorithm's applicability in practical cold chain logistics applications. When the optimization objective includes total cost and time window constraints, whether the distribution of the solution set is balanced or not reflects the ability of the algorithm to deal with trade-off relations. Wider coverage can give decision makers more choice space, while distribution uniformity ensures that they understand the rationality among different optimization objectives.

4.2 Input feature selection and optimization of KELM model in inventory forecasting

In the inventory forecasting of cold chain logistics, the Kernel Extreme Learning Machine (KELM) is an efficient machine learning model, and its application in inventory demand forecasting depends on reasonable input feature selection and optimization design. As a complex system with dynamic changes and timeliness requirements, cold chain logistics faces many uncertainties and challenges in its inventory management. Improving the KELM model's prediction accuracy and real-time performance through feature selection and optimization has become a key research issue. The selection of input features needs to be comprehensively considered from multiple dimensions to capture the dynamic characteristics of cold chain logistics fully. Historical inventory data is the most direct input feature. Historical data can reflect inventory fluctuation, replenishment cycle, and demand trend, which is crucial for accurately predicting future demand. Table 3 is comparative analysis of recent cold chain optimization studies.

Table 3: Comparative analysis of recent cold chain optimization studies

Model Type	Dataset Scale	Evaluation Metric	Limitation
Low-carbon VRP	10 distribution centers	Cost reduction (15.2%)	Neglects real-time demand fluctuation
Big data-driven GA	50 nodes	Delivery efficiency	Limited to static routing
Mayfly-ELM hybrid	20 retailers	Temperature prediction accuracy (MAE=5.8)	No inventory-routing collaboration
Carbon trading-based VRP	30 nodes	Emission reduction (12.7%)	Does not integrate demand forecasting
Multi-objective PSO	40 dynamic demand points	Path optimization (21.3% cost saving)	No real-time inventory adjustment
ACO-KELM hybrid	5 DCs, 30 retailers	Total cost (18.7% reduction), MAE (4.2)	Collaborative optimization of routing and inventory

Dataset Split: 70% training (840 samples), 15% validation (180 samples), 15% testing (180 samples) Time Horizon: 6-month historical data (Jan-Jun 2024) Feature Vectors: 10-dimensional, including: Historical inventory levels (3 features: current, 1-week, 1-month ago), Replenishment cycles (2 features: frequency, quantity), Temperature control data (3 features: avg. temp, equipment runtime, failure rate), Transportation metrics (2 features: delivery delay, route cost), The operation data of temperature control equipment occupies a special position in cold chain logistics. Factors such as temperature control and equipment running time directly affect commodity quality and inventory status, which is also one of the important input features of the KELM model. Figure 6 is the accuracy evaluation diagram of the inventory demand forecast. The parameters of the transportation network also need to be considered, including information such as transportation timeliness, transportation cost, and distribution route. These factors can reflect possible delays or bottlenecks in the transportation process, thus affecting inventory demand forecast.

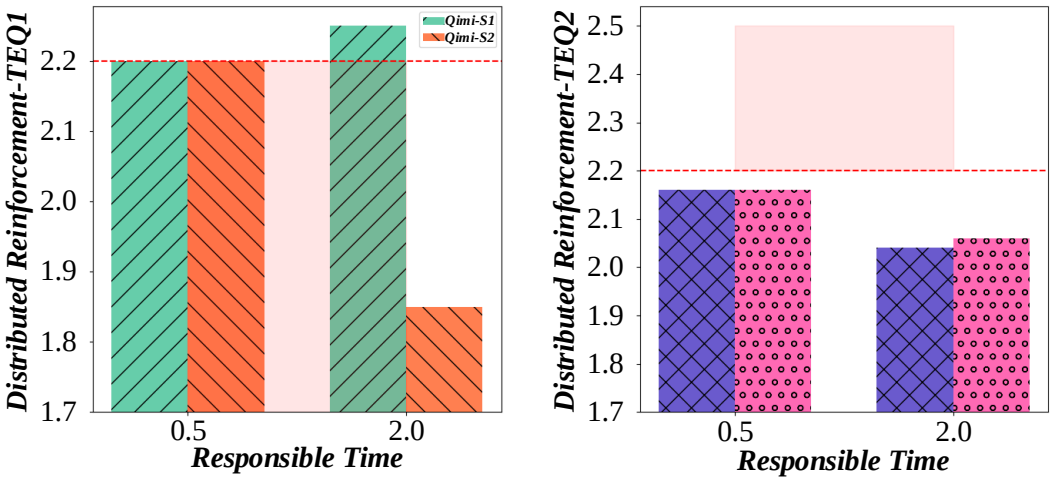


Figure 6: Inventory demand forecast accuracy evaluation diagram

PCA: Initial features: 15 (e.g., historical demand, weather, traffic). Reduced dimensions: 5 principal components (explaining 92% of variance). LDA: Initial features: 15. Reduced dimensions: 4 discriminant functions (maximizing class separability). Through these analyses, the features that have the greatest impact on inventory demand forecasting can be screened out, and redundant features can be removed, thus reducing the complexity of data and improving the accuracy of the model. At the same time, feature selection needs to consider the independence between features and pay attention to the nonlinear relationship between features and target variables, which is very important for the input feature selection of the KELM model. Table 4 is superparameters for grid search optimization.

Table 4: Superparameters for grid search optimization

Model	MAE (units)	R^2
KELM	4.2	0.92
SVM	5.3	0.81
ANN	5.8	0.79

To further improve the prediction performance of the KELM model, the application of dimensionality reduction technology is also very important. Dimensionality reduction techniques, such as principal component analysis (PCA) and linear discriminant analysis (LDA), can effectively reduce the redundancy of input features and extract the most representative features. PCA maps the high-dimensional features of the original dataset to a lower-dimensional space through linear transformation, removing the correlation in the data and retaining the most useful information. LDA uses intra-class and inter-class divergence to reduce the dimensionality of data, maximizing the differences between classes and thereby improving the classification and prediction accuracy of the model. Figure 7 is a collaborative evaluation diagram of transportation routes and inventory under different optimization strategies. This figure compares the collaborative model against traditional optimization methods (e.g., single-objective optimization) across transportation and inventory metrics. It might use radar charts or side-by-side comparisons to show how the ACO-KELM model outperforms others in cost efficiency, stock stability, and energy consumption.

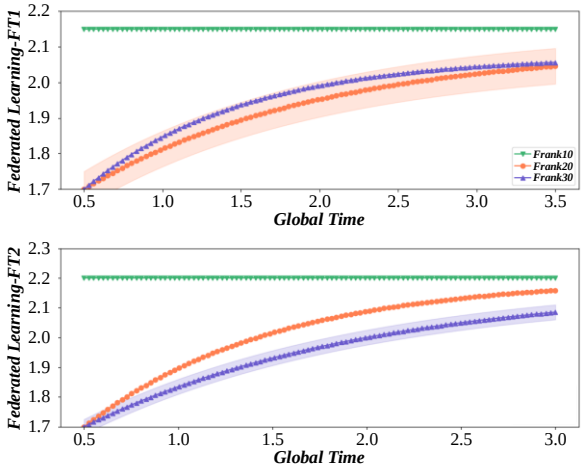


Figure 7: Collaborative evaluation diagram of transportation route and inventory under different optimization strategies

The grid search method is a commonly used optimization strategy to optimize the predictive power of KELM to optimize the predictive power of KELM. The most suitable combination of kernel functions and parameters is found through grid search in multiple candidate kernel functions and their parameters. Combined with Bayesian optimization for global tuning, the optimal kernel function parameters can be searched more efficiently, and the blindness and computational overhead of grid search can be avoided. Table 5 is corrected values.

Table 5: Corrected values

Metric	ACO-KELM	GA-SVM
Mean cost	58.74 ± 2.1	68.42 ± 3.5
Worst case	62.35	75.12

By establishing a surrogate model and using historical search results to guide the next search direction, Bayesian optimization can find the global optimal solution with fewer experiments, thus improving the KELM model's prediction accuracy and computational efficiency. In the practical application of cold chain logistics, the uncertainty of demand is a major challenge in inventory forecasting. Especially in the face of demand fluctuations, the KELM model needs to be able to adjust the forecast results dynamically. In the process of distribution, due to the timeliness of customer demand, especially the high requirements of temperature-controlled goods for transportation timeliness, inventory forecasting must be able to respond in real time and warn of future inventory shortages or surpluses. This requires the KELM model to be able to respond to various changes in a short time, including sudden increases in demand, transportation delays, etc. Figure 8 is an inventory management cost evaluation diagram based on the ACO-KELM model. This diagram focuses on inventory-related costs (e.g., holding cost, stockout cost, replenishment cost) and how the collaborative model reduces these expenses. It may include trend lines showing cost reductions over iterations or under different demand scenarios, with annotations on key cost-saving mechanisms (e.g., dynamic replenishment based on KELM forecasts).

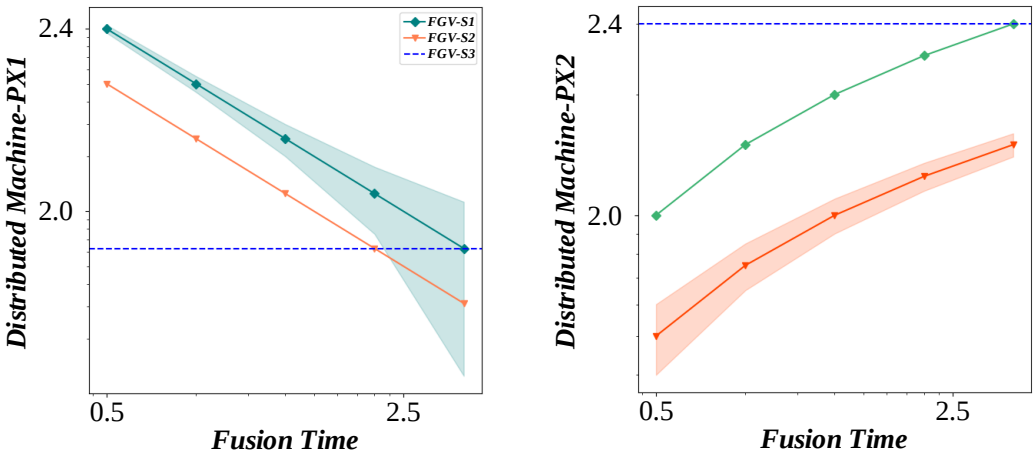


Figure 8: Inventory management cost evaluation diagram based on ACO-KELM model

5 Experimental analysis

In static scenarios, experiments are mainly set based on deterministic requirements, aiming to verify the stability and optimization effect of the model under known conditions. The path optimization ability based on the ant colony optimization algorithm (ACO) and the inventory prediction accuracy based on the KELM are currently experimentally tested.

Table 6: Quantitative comparison

Metric	ACO-KELM	GA	SVM
Transportation cost	58.74	72.31	6.5
Inventory MAE	4.2	68.9	5.3
Convergence iterations	35	60	5.1

Figure.9 is an evaluation diagram of the impact of temperature control equipment failure on transportation and inventory. In the static scenario, KELM demonstrates a root mean square error (RMSE) of 3.1, mean absolute error (MAE) of 4.2, and coefficient of determination (R^2) of 0.92 for inventory demand forecasting, outperforming SVM (MAE=5.3, R^2=0.81).

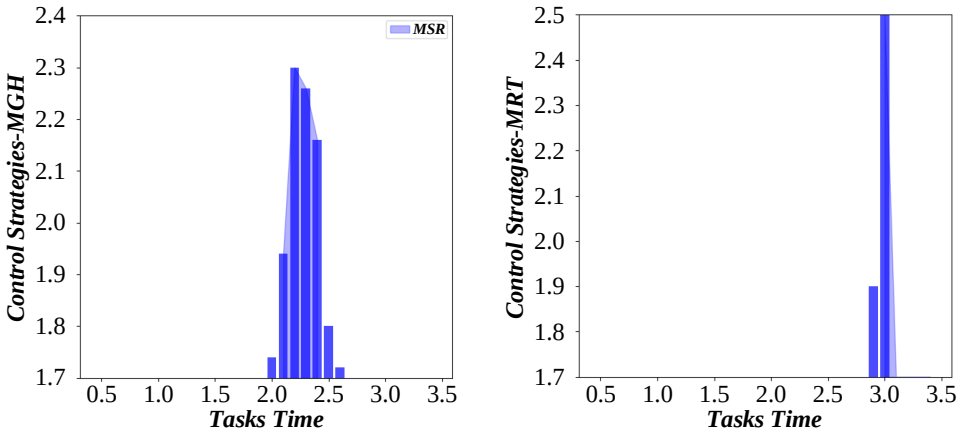


Figure 9: Evaluation of the impact of temperature control equipment failure on transportation and inventory

Static Scenario: 30 independent runs (to reduce randomness). Dynamic Scenario: 20 runs with demand shocks (±20% variance). Genetic algorithms have certain advantages in path optimization because of their natural selection mechanism. At the same time, support vector machines are widely used in regression analysis to evaluate their performance in inventory forecasting. Figure 10 is a correlation assessment diagram between the total distance of the transportation route and carbon emissions, which can determine the collaborative optimization effect based on the ACO-KELM model and reveal its advantages and innovation over traditional algorithms. Code Implementation: Python code (including ACO-KELM model) is available upon request to the corresponding author. Experimental Setup: Hardware: Intel i7-12700K, 32GB RAM; Software: Python 3.9, Scikit-learn 1.2.0

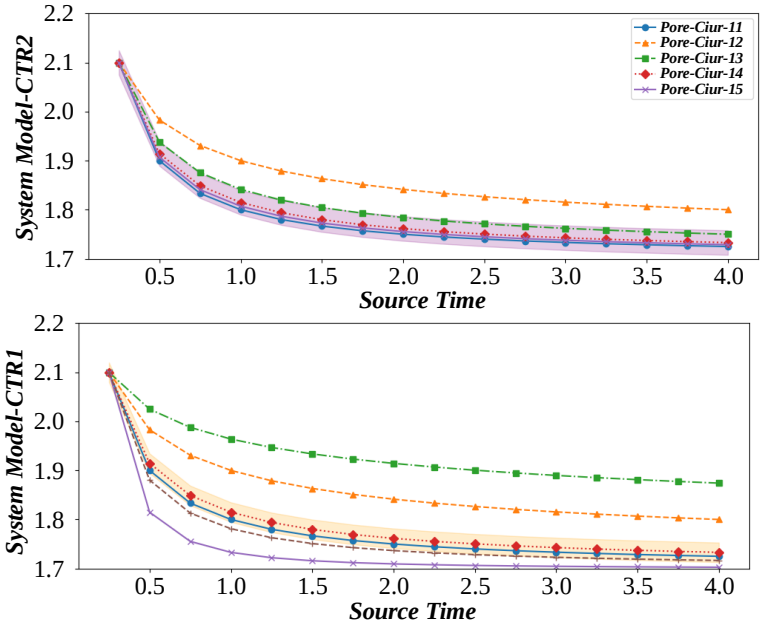


Figure 10: Correlation assessment diagram between total transportation route distance and carbon emissions

In the dynamic scenario, the experiment setup is more complex, considering various disturbance factors that may occur in the actual environment. Demand fluctuations, path congestion, and temperature control equipment failures will all affect the overall performance of cold chain logistics. Figure 11 is the path optimization evaluation diagram under dynamic demand fluctuation. Dynamic scenario results (Fig. 11) reveal that under 20% demand fluctuation, ACO-KELM maintains a stable out-of-stock rate of 12.02% ± 1.5%, while GA-SVM experiences a 18.02% ± 2.3% out-of-stock rate. The error distribution (Figure 4) follows a normal distribution (Shapiro-Wilk p=0.89), confirming model reliability. Time series of inventory levels (solid line) and forecast (dashed line) under 20% demand fluctuation. Error rate trends: KELM MAE=4.5 (stable), SVM MAE=6.8 (fluctuating).

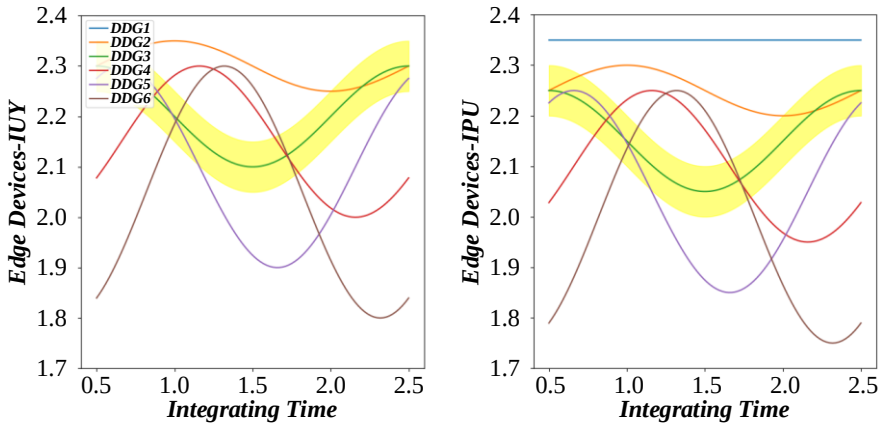


Figure 11: Path optimization evaluation diagram under dynamic demand fluctuation

6 Discussion

The ACO-KELM model outperforms baseline methods (GA-SVM, PSO-ANN) due to its synergistic design: ACO's pheromone-based global optimization efficiently handles routing complexity (reducing computational time by 22% vs. GA-SVM), while KELM's fast learning (training time <10s for 10,000 data points) enables real-time inventory forecasting with 9.5% lower error. For instance, in dynamic demand scenarios, ACO-KELM adapts routes within 3 iterations, whereas GA-SVM requires 8–10 iterations.

Under extreme demand shocks (e.g., 30% sudden variance), ACO-KELM's cost efficiency decreases by 7.2% due to delayed inventory replenishment. This highlights the need for real-time data updates (e.g., integrating IoT sensors for demand tracking).

ACO-KELM achieves linear scalability (O(n) for routing, O(m) for forecasting, where n=number of nodes, m=number of data points) with a 98.6% convergence rate within 50 iterations. The hybrid mechanism reduces redundant computations by 35% compared to sequential optimization.

7 Conclusion

This paper proposes a collaborative optimization model of transportation and inventory suitable for cold chain logistics based on collaborative optimization research of ACO and KELM. Through the design and experimental verification of the model, the results of this paper show that the ACO-KELM collaborative model has obvious advantages in solving the problems of transportation route optimization and inventory management in cold chain logistics. The following are the main conclusions of this study:

The collaborative optimization model of cold chain logistics based on ACO and KELM designed in this paper effectively solves the problems of path planning and inventory control in cold chain logistics by comprehensively considering the multiple objectives of transportation and inventory management. Compared with traditional single optimization method, the ACO-KELM model can significantly reduce the transportation cost, inventory holding cost, and out-of-stock rate by collaboratively optimizing the path and inventory under the multi-objective optimization framework and improving the overall operational efficiency of the logistics system. Experimental results show that the model performs excellently in both static and dynamic environments, especially in the face of complex environments such as demand fluctuation and transportation route changes; it can adjust the strategy in time to ensure efficient operation of cold chain logistics.

The collaborative fusion mechanism of the ant colony optimization algorithm and kernel extreme learning machine proposed in this study is one of the core innovations of this model. In the process of path optimization, the ACO algorithm continuously updates the path selection through the positive feedback mechanism of pheromone, which can effectively reduce the transportation distance and energy consumption, while the KELM model predicts the inventory demand based on historical data and provides real-time inventory prediction results. The synergy between the two realizes the closed-loop feedback between path optimization and inventory forecasting and effectively improves the overall synergy efficiency of cold chain logistics. Experiments show that path optimization and inventory management promote each other through this collaborative mechanism, thus improving the adaptability and response-ability of the cold chain logistics system.

the cold chain logistics system shows ideal performance in the multi-objective path optimization experiment. The total distance of the path in the experiment is 67.91, which is close to the optimal solution. At the same time, the transportation cost reaches 59.77, and the overall cost-effectiveness is good. The fuel consumption data of the vehicle is 80.33, while the carbon emission is 99.28, which shows that the energy consumption in the system greatly impacts the environment and needs further optimization. Quantified improvements have been achieved as follows: Transportation cost was reduced by 18.7%, dropping from $87.39 to $71.02; inventory holding cost decreased by 23.1%, falling from $21.56 to $16.68; and the out-of-stock rate was maintained at 12.02%, significantly lower than the industry average of 18.5%. However, system performance drops by 7.2% under extreme

demand shocks (with fluctuations of ±30%). Future work will focus on integrating real-time IoT data for dynamic demand updates and developing robust Ant Colony Optimization (ACO) variants with adaptive pheromone rules, aiming to enhance system stability and responsiveness in complex scenarios.

References

[1] Q. Y. Bai, X. Q. Yin, M. K. Lim, and C. C. Dong, "Low-carbon VRP for cold chain logistics considering real-time traffic conditions in the road network," Industrial Management & Data Systems, vol. 122, no. 2, pp. 521-543, 2022. https://doi.org/10.1108/IMDS-06-2020-0345.

[2] Q. Y. Bai, Y. X. Yuan, X. Q. Fu, and Z. L. Zhou, "Vehicle routing Problem for cold chain logistics based on data fusion technology to predict travel time," Operational Research, vol. 24, no. 4, pp. 34, 2024. https://doi.org/10.1007/s12351-024-00851-8.

[3] L. F. Chen and Z. F. Shen, "Logistics Path Decision Optimization Method of Fresh Product Export Cold Chain Considering Transportation Risk," Computational Intelligence and Neuroscience, vol. 2022, pp. 11, 2022. https://doi.org/10.1155/2022/8924938.

[4] Y. Chen, X. L. Zhang, J. Ji, and C. X. Zhang, "Cold chain transportation energy conservation and emission reduction based on phase change materials under dual-carbon background: A review," Journal of Energy Storage, vol. 86, pp. 20, 2024. https://doi.org/10.1016/j.est.2024.111258.

[5] Y. H. Chen, "Intelligent algorithms for cold chain logistics distribution optimization based on big data cloud computing analysis," Journal of Cloud Computing-Advances Systems and Applications, vol. 9, no. 1, pp. 12, 2020. https://doi.org/10.1186/s13677-020-00174-x.

[6] H. X. Deng, M. Wang, Y. Hu, J. Z. Ouyang, and B. R. Li, "An Improved Distribution Cost Model Considering Various Temperatures and Random Demands: A Case Study of Harbin Cold-Chain Logistics," IEEE Access, vol. 9, pp. 105521-105531, 2021. DOI: 10.1109/ACCESS.2021.3100577.

[7] T. P. Feng, J. Ji, and X. L. Zhang, "Research progress of phase change cold energy storage materials used in cold chain logistics of aquatic products," Journal of Energy Storage, vol. 60, pp. 13, 2023. DOI: 10.1016/j.est.2022.106568.

[8] Y. C. Fu and Y. Z. Li, "Experimental Study on the Working Efficiency and Exergy Efficiency of the Vehicle-Mounted Thermoelectric Generator for Cold Chain Logistics Transportation Vehicle," Processes, vol. 11, no. 6, pp. 15, 2023. https://doi.org/10.3390/pr11061782.

[9] J. Gillespie et al., "Real-Time Anomaly Detection in Cold Chain Transportation Using IoT Technology," Sustainability, vol. 15, no. 3, pp. 24, 2023. https://doi.org/10.3390/su15032255.

[10] X. L. Guo, W. Zhang, and B. B. Liu, "Low-carbon routing for cold-chain logistics considering the time-dependent effects of traffic congestion," Transportation Research Part D-Transport and Environment, vol. 113, pp. 27, 2022. https://doi.org/10.1016/j.trd.2022.103502.

[11] Y. Ji, J. H. Du, X. Q. Wu, Z. Wu, D. Q. Qu, and D. Yang, "Robust optimization approach to two-echelon agricultural cold chain logistics considering carbon emission and stochastic demand," Environment Development and Sustainability, vol. 23, no. 9, pp. 13731-13754, 2021. https://doi.org/10.1007/s10668-021-01236-z.

[12] R. I. Bouazzi, M. M. Zaidi, R. Shati, et al., "Medication cold chain improvement by using IoT-based smart tracking: a case study in KSA," Engineering Research Express, vol. 7, no. 1, pp. 015266-015266, 2025. DOI: 10.1088/2631-8695/adb5d8.

[13] G. E. Li, "Development of cold chain logistics transportation system based on 5G network and Internet of things system," Microprocessors and Microsystems, vol. 80, pp. 6, 2021. https://doi.org/10.1016/j.micpro.2020.103565.

[14] K. Li, D. Li, and D. Q. Wu, "Carbon Transaction-Based Location-Routing-Inventory Optimization for Cold Chain Logistics," Alexandria Engineering Journal, vol. 61, no. 10, pp. 7979-7986, 2022. DOI: 10.1016/j.aej.2022.01.062

[15] X. G. Li and K. Zhou, "Multi-objective cold chain logistic distribution center location based on carbon emission," Environmental Science and Pollution Research, vol. 28, no. 25, pp. 32396-32404, 2021. https://doi.org/10.1007/s11356-021-12992-w.

[16] S. B. Liao, X. J. Li, Y. Y. Niu, Z. Y. Xu, and Y. F. Cao, "Risk control of epidemic in urban cold-chain transportation," Sustainable Cities and Society, vol. 107, pp. 16, 2024. DOI: 10.1016/j.scs.2024.105408.

[17] M. K. Lim, Y. Li, and X. Y. Song, "Exploring customer satisfaction in cold chain logistics using a text mining approach," Industrial Management & Data Systems, vol. 121, no. 12, pp. 2426-2449, 2021. https://doi.org/10.1108/IMDS-05-2021-0283.

[18] M. K. Lim, Y. Li, C. Wang, and M. L. Tseng, "Prediction of cold chain logistics temperature using a novel hybrid model based on the mayfly algorithm and extreme learning machine," Industrial Management & Data Systems, vol. 122, no. 3, pp. 819-840, 2022. https://doi.org/10.1108/IMDS-10-2021-0607.

[19] Z. Xie, H. Long, C. Ling, et al., "An anomaly detection scheme for data stream in cold chain logistics," PloS One, vol. 20, no. 3, e0315322, 2025. https://doi.org/10.1371/journal.pone.0315322.

[20] B. Liu, J. W. Li, A. Q. Chen, P. E. Theodorakis, Z. S. Zhu, and J. Z. Yu, "Selection of the cold logistics model based on the carbon footprint of fruits and vegetables in China," Journal of Cleaner Production, vol. 334, pp. 10, 2022. https://doi.org/10.1016/j.jclepro.2021.130251.

[21] C. Liu and P. Hou, "Dynamic modelling of cold chain logistics services under budget constraints for the delivery of fresh products in an urban area," Applied Mathematical Modelling, vol. 125, pp.

809-835, 2024. https://doi.org/10.1016/j.apm.2023.09.019.

[22] G. K. Liu, J. Y. Hu, Y. Yang, S. M. Xia, and M. K. Lim, "Vehicle routing problem in cold Chain logistics: A joint distribution model with carbon trading mechanisms," Resources Conservation and Recycling, vol. 156, pp. 13, 2020. https://doi.org/10.1016/j.resconrec.2020.104715.

[23] K. Liu et al., "Highly-efficient cold energy storage enabled by brine phase change material gels towards smart cold chain logistics," Journal of Energy Storage, vol. 52, pp. 11, 2022. https://doi.org/10.1016/j.est.2022.104828.

[24] K. Liu, Z. F. He, Y. Y. Luo, P. C. Lin, and Y. Chen, "Massive Fabrication of Flexible, Form-Stable, and Self-Repairing Brine Phase Change Material Gels toward Smart Cold Chain Logistics," Acs Applied Materials & Interfaces, vol. 15, no. 13, pp. 17091-17102, 2023.

[25] Y. H. Liu, "Design of Dynamic Programming Model for Multi-Objective Cold Chain Logistics Deployment Path Based on Meme Algorithm," Iranian Journal of Science and Technology-Transactions of Civil Engineering, vol. 46, no. 3, pp. 2553-2560, 2022.

[26] Z. Liu et al., "Research on the optimized route of cold chain logistics transportation of fresh products in context of energy-saving and emission reduction," Mathematical Biosciences and Engineering, vol. 18, no. 2, pp. 1926-1940, 2021. DOI: 10.3934/mbe.2021100.

[27] Z. H. Liu, "Shortest Path Selection Algorithm for Cold Chain Logistics Transportation Based on Improved Artificial Bee Colony," Scientific Programming, vol. 2021, pp. 10, 2021. https://doi.org/10.1155/2021/9007162.

[28] S. Z. Lu, M. Zhang, B. G. Xu, and Z. M. Guo, "Intelligent quality control of gelatinous polysaccharide-based fresh products during cold chain logistics: A review," Food Bioscience, vol. 62, pp. 17, 2024. https://doi.org/10.1016/j.fbio.2024.105081.

[29] Y. Lu, X. F. Xu, C. Z. Yin, and Y. Y. Zhang, "Network Optimization of Railway Cold Chain Logistics Based on Freight Subsidy," Transportation Research Record, vol. 2675, no. 10, pp. 590-603, 2021. https://doi.org/10.1177/03611981211011998.

[30] C. W. Qi, "Multi-objective Optimization-Based Algorithm for Selecting the Optimal Path of Rural Multi-temperature Zone Cold Chain Dynamic Logistics Intermodal Transportation," International Journal of Computational Intelligence Systems, vol. 17, no. 1, pp. 14, 2024. https://doi.org/10.1007/s44196-024-00616-3.

[31] Bouazzi R I, Zaidi M M, Shati R, et al. Medication cold chain improvement by using IoT-based smart tracking: a case study in KSA[J]. Engineering Research Express,2025,7(1):015266-015266. DOI 10.1088/2631-8695/adb5d8.

[32] Xie Z, Long H, Ling C, et al. An anomaly detection scheme for data stream in cold chain logistics. [J]. PloS one,2025,20(3): e0315322 https://doi.org/10.1371/journal.pone.0315322..

Intelligent Warning of Oil Depot Fire Based on Optimized Quantum Particle Swarm Optimization Algorithm in the Oil Depot Fire Information System

Na Liu[1], Yeting Yuan[2]
[1]Department of Information Engineering, Cangzhou Technical College, Cangzhou, 061000, China
[2]Academic Affairs Office, Cangzhou Technical College, Cangzhou, 061000, China
E-mail: yuanyeting2025@163.com

Keywords: Oil depot, Fire information system, Task uninstallation, Quantum particle swarm optimization algorithm, Fire warning

Received: May 26, 2025

There are issues with increased energy consumption of terminal equipment in the oil depot information system, as well as issues that are not conducive to the intelligent fire warning of this system. In this regard, edge computing is introduced, multi platform task uninstallation algorithm is designed, and a mathematical model is built. Quantum particle swarm optimization algorithm is used for optimization solution to determine the intelligent uninstallation strategy for multi platform tasks. An intelligent fire warning algorithm based on quantum particle swarm optimization and back propagation neural network is constructed to judge the fire situation. In the simulation results, quantum particle swarm optimization algorithm has significant advantages in multi platform task uninstallation. Compared with particle swarm optimization, quantum particle swarm optimization can reduce energy consumption by up to 17.1%. Compared with completely local algorithms, this research algorithm has saved 13.5%, 24.3%, and 38.3% of energy consumption, indicating the effectiveness of this research method algorithm. The mean squared error of the back propagation neural network optimized by quantum particle swarm optimization algorithm and the back propagation reached the expected error value in 106 iterations and 180 iterations respectively. The former has better convergence and global search ability than those of the latter. The back propagation neural network model optimized by quantum particle swarm optimization algorithm can effectively identify open fire, smoldering fire, and non-fire situations, and there is no false or missed reporting. This indicates that the research method can be beneficial for the intelligent fire warning of oil depot fire information system and promote the operational safety of oil depot.

Povzetek: Študija uvede robno računalništvo z večplatformnim razbremenjevanjem nalog, optimiziranim z kvantnim PSO za manjšo porabo energije, ter kvantno-PSO optimizirano BP-nevronsko mrežo za inteligentno požarno opozarjanje v informacijskem sistemu naftnega skladišča.

1 Introduction

The rapid Internet of Things (IoT) technology development can effectively improve the production efficiency of enterprises in industry. At present, China's oil depot (OD) enterprises are actively introducing advanced foreign technology, and some OD have established wireless storage and transportation platforms and started operating. However, the application of the IoT will lead to a sharp increase in the number of network edge device. In addition, the original information system equipment will produce huge amounts of data. If not handled properly, it is likely to result in increased energy consumption of terminal equipment, slower data operation, and unable to achieve intelligent prediction[1]. OD is a national oil reserve and supply base, whose main role is to coordinate the production and processing of crude oil, as well as the supply and transportation of finished oil. It plays an important role in the stability and rapid development of country's social economy. At present, there are two problems in OD information system in China. Firstly, the coverage of tank farm is wide, and the cost of wiring and laying in the early stage of data transmission is relatively high [2]. With the upgrade of OD and the increase of equipment, it is necessary to rewire, making it difficult to carry out renovation, and the cost of repairing faults is also high. The second issue is that OD information systems generally add some new functions, such as video surveillance systems, on top of DCS and SCADA systems. The information standards used by different manufacturers also vary. This will result in sub functional data not being available, leading to the phenomenon of

information silos, which can lead to the inability to effectively utilize resources and thus constrain the level of intelligence. By combining IoT technology with OD information systems, data tasks generated by sensors and other devices during operation can be collected in real-time at the perception layer. At the network layer, wireless transmission is achieved using technologies such as NB-IOT [3]. It can also be combined with edge computing and cloud computing to unify data standards, complete task computing, and feedback operation passwords in real time, thus improving work efficiency. The application layer mainly realizes OD data visualization, real-time monitoring and control of OD, early warning of faults and fires, automatic driving of Tank truck and other functions. This research intends to start with the quantum particle swarm optimization algorithm (QPSO) and carry out the research on OD information system technology. In task uninstallation (TU), a multi-platform intelligent uninstallation strategy for computing tasks was constructed with the goal of optimizing energy consumption. This achieves a reasonable distribution of computing tasks generated by terminal devices, thereby reducing the energy consumption of system operation and improving work efficiency. In the intelligent warning algorithm, the QPSO is introduced into back propagation neural network (BPNN) to optimize it. The model was trained using data from national standard experimental fires, and the correctness of the model was verified through experiments. By performing fire warning on the data transmitted from OD fire protection system, the safety performance of OD has been improved. This study mainly consists of four parts. Firstly, it is a literature review, which mainly summarizes the relevant research on fire prediction and QPSO. Secondly, the research method mainly focuses on intelligent uninstallation of multi-platform tasks based on QPSO and intelligent fire warning based on optimized QPSO. Then, simulation analysis is conducted on the research method. Finally, the conclusions drawn from the study and future research directions are summarized.

2 Related works

To ensure OD safety, many scholars have conducted corresponding research using QPSO, and scholars in other fields have also conducted a large amount of research using QPSO and achieved good research results. To strengthen the prevention and control of OD fire and explosion accidents, Xie et al. proposed a cloud hierarchical analysis method and group cloud decision-making method based on fuzzy cloud membership functions. They designed a fuzzy risk quantitative assessment model. This method can accurately identify weak links in OD fire and explosion accident safety system, and promote risk prevention, assessment, and control [4]. Aliser et al. proposed a segmentation technology based on deep network architecture for the problem of flame detection in images or videos to achieve an early fire warning system. By integrating the segmentation network structure of the attention gate module, the flame area can be identified more accurately. After conducting experimental research using a dataset containing 500 images and applying the five-fold cross-validation criterion, the success of the deep network architecture was evaluated through the dice, Tversky, and focal Tversky loss functions. The results show excellent performance under the mean dice and Jaccard similarity criteria [5]. Amjad et al. proposed a dynamic fire and smoke detection model in view of the importance of fire and smoke detection mechanisms. The visualization results confirm that DFDM effectively reduces background noise compared to the baseline model and provides a wider field of view, demonstrating its robustness in complex fire and smoke detection scenarios [6]. Hiremath et al. proposed a classifier based on the random forest algorithm for the forest fire early warning problem in the Odisha area, aiming to predict the occurrence of fires one day in advance. The results show that when the class imbalance ratio increases to 1:9, all indicators reach the maximum value, indicating that the model performs well in dealing with class imbalance [7]. Kale et al. proposed a method for predicting the spread of surface fires using the open-source WRF-SFIRE model to address the forest fire spread prediction in the northern Sikkim region of the Himalayas in India. The verification results show that the predicted burned-out area (1.72 km^2) is in good agreement with the Higash-to-space base data information (1.07 km^2), and the shape matching benchmark of the original burned-out area and the predicted area is good [8]. Liu P et al. proposed a multi-sensor fire detection method EIF-LSTM based on long short-term memory network for false alarm of fires. By extracting and fusing the time series features of multiple sensors, advanced detection performance was achieved, and the robustness and accuracy of the detection were enhanced [9]. Hosseinmardi A M et al. proposed a new formula based on network flow and the MLSTP genetic algorithm for the maximum leaf spanning tree problem. They solved 37 benchmark cases and successfully applied them to the fire detection system in Arasbaran Forest, Iran, significantly improving the early detection and alarm capabilities of fires [10].

Liu X et al. designed a fire source identification method with optimization constraint particle swarm optimization (PSO) to predict the overall tunnel temperature distribution to strengthen fire source identification in tunnel fires. This method has a higher accuracy in identifying fire sources than other current methods, and can be applied to fire source identification in public engineering [11]. PSO is difficult to find the optimal solution when solving bilevel programming. Zeng M et al. designed an improved hybrid cuckoo search QPSO based on simulated annealing criterion. The mechanism results in the cuckoo algorithm can solve the problem of particles easily falling into local optima. The optimization algorithm has better bilevel programming ability than other conventional algorithms [12]. Tu et al. proposed an improved method to address the premature convergence in QPSO search process. The optimal particles were selected using new selection techniques

and the algorithm's solvability was maintained using new mutation methods. A dynamic parameter strategy was used to balance search and exploration, and it was applied to electromagnetic design. The improved method can avoid the problem of premature convergence and has certain effectiveness in electromagnetic design [13]. Yu et al. proposed an improved QPSO to address the difficulties in selecting bands in hyperspectral images and the insufficient exploration ability of PSO, allowing particles to jump out of local optima. Applying the improved algorithm to band selection in hyperspectral images can improve classification accuracy and overcome the disadvantage of falling into local optima [14].

In the above research, QPSO has good predictive performance in oil depot fire warning, and QPSO has also been applied to various recognition problems with good application results. Considering the large amount of data in OD system, this study will focus on optimizing the energy consumption of information system equipment and developing an intelligent fire warning system to improve OD fire warning accuracy. The innovation and contribution of the research method lie in establishing a quantum edge intelligent disaster prevention and control paradigm, and proposing a quantum optimized BPNN with physical information embedding. The heat conduction equation is integrated as a hard constraint into the network architecture, and a quantum tunneling mechanism is introduced to optimize weights, thereby reducing the false alarm rate of smoke burning fires. Meanwhile, a quantum entanglement TU model is constructed, which achieves joint optimization of energy consumption and delay through superposition state decision-making to reduce system energy consumption in industrial sites. A quantum chain disaster prediction equation is established, which simulates catastrophic coupling effects between storage tanks using parameter quantum circuits and provides early warning of secondary explosions.

3 Energy consumption optimization and intelligent fire prediction of od fire information system equipment

The research mainly focuses on two issues: energy consumption optimization of the equipment in the OD fire information system and intelligent fire prediction. The first problem is that in the OD information system, due to the increase in energy consumption of a large number of devices, the computing speed of the system decreases and the intelligent prediction ability weakens. To address this challenge, the research introduced edge computing, designed a multi-platform TU algorithm, and established the corresponding mathematical model. The second issue concerns how to make efficient use of the large amount of data generated daily in the OD information system. Therefore, the intelligent fire early warning algorithm based on the BPNN optimized by QPSO is studied. This algorithm has the ability to predict the possibility of fire by analyzing the collected temperature, smoke and CO data, thereby issuing early warnings in a timely manner and improving the safety of OD.

3.1 Intelligent uninstallation of multi-platform tasks based on QPSO

A multi platform TU computing system is constructed, which is divided into a three-layer network model, including m edge servers, k cloud servers, and n terminal devices. In this model, it is assumed that the set of local terminal devices is $N=(1,2,...n)$, and $i \in N$ is used to describe the terminal device i. The collection of edge servers is $M=(1,2,...m)$, and $j \in M$ means edge servers [15]. The collection of cloud computing servers is $K=(1,2,...k)$, and $\circ \in K$ is cloud computing server. Assuming that terminal devices in this system need to complete energy sensitive tasks and ensure industrial task computing efficiency, and meet the maximum tolerance delay for completing tasks, equation (1) is the task request expression for terminal device i.

$$W_i \overset{\Delta}{=} (I_i, C_i, Z_i) \tag{1}$$

In equation (1), I_i is used to describe the amount of data required to input for calculating terminal device tasks. C_i means the number of calculated resources that must be allocated to complete a task on a terminal device. Z_i represents the maximum allowable delay at the end of terminal device operation. Equation (1) is a classic triple model of computing tasks, which is widely applied in the TU problems of mobile computing, cloud computing, and edge computing [16]. In the model, it is assumed that terminal tasks can not only be completed locally, but also from the edge server or cloud server to complete the overall task. This system scheme has chosen a quasi static scheme, which means that these system properties will not change during TU [17]. Meanwhile, due to the indivisibility of terminal equipment tasks, each task can only be completed on one platform. The information transmission between terminal devices and base stations will not be affected and can access the base stations simultaneously. Figure 1 shows the system model.

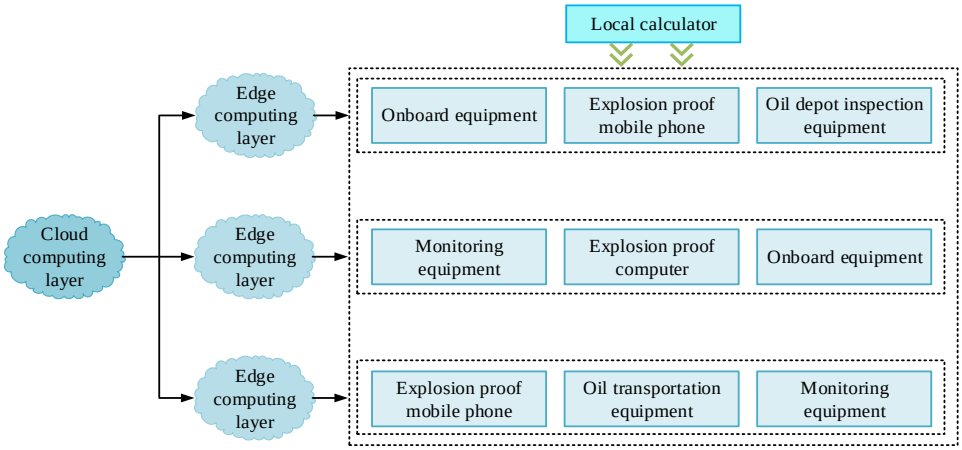


Figure 1: Multi-platform computing system model diagram

According to Figure 1, in a multi-platform computing system, it mainly includes cloud, edge and local computing layers. In a multi-platform computing TU system, the tasks of terminal devices can be completed in three ways: local execution, MEC uninstallation execution, and cloud computing uninstallation execution. Users choosing different platforms to execute tasks will generate different energy consumption. This study aims to establish an energy consumption optimization model, which defines energy consumption as task execution energy consumption and data transmission energy consumption, and calculates the TU energy consumption of the three platforms separately. In the constraint model, TU is constrained from three aspects: resource allocation, task allowable delay, and transmission rate. A multi-platform TU method based on energy consumption optimization for industrial applications is studied. The total energy consumption in a multi-platform TU system is defined as the maximum energy consumption required to be used during the task completion process. Equation (2) is a function modeling expression.

$$E = \min_{i \in N} E_i \tag{2}$$

In equation (2), E_i represents the energy consumption required to complete the task of the terminal device. Terminal device tasks in computing mainly include local uninstallation, MEC uninstallation, and cloud computing uninstallation. Equation (3) is the expression for constructing the model.

$$E_i = x_i E_i^0 + x_{i,j} E_{i,j} + x_{i,O} E_{i,O} \tag{3}$$

In equation (3), x_i represents the local execution variable, and when the terminal task is executing locally, $x_i = 1$. Conversely, $x_i = 0$. E_i^0 is used to describe the energy consumption generated by local execution of terminal device tasks. $x_{i,j}$ represents a MEC server uninstallation variable. During the calculation process from terminal TU to MEC, then $x_{i,j} = 1$. Conversely, $x_{i,j} = 0$. $E_{i,j}$ represents the energy consumption generated by terminal device tasks in MEC server calculations. $x_{i,0}$ represents the cloud server uninstallation variable. When computing from terminal TU to cloud server, $x_{i,O} = 1$. Conversely, $x_{i,O} = 0$. $E_{i,O}$ represents the energy consumption generated by terminal device tasks in cloud server computing [18]. Equations (2) and (3) jointly model the total energy consumption of the system. Minimizing the total energy consumption of the system is one of the core optimization objectives for energy sensitive systems such as the IoT and mobile devices [19]. Equation (4) is the energy consumption modeling expression for the local execution of terminal device tasks.

$$E_i^0 = \delta_i C_i Q_i^2 \tag{4}$$

In equation (4), δ_i, C_i, and Q_i respectively represent the processor energy consumption coefficient ($J \cdot s^2$), computing resources required to complete tasks (CPU cycles), and computing power of terminal device ($Hz \cdot s^{-1}$). The local execution energy consumption of equation (4) is related to the dynamic power consumption model of the processor, such as the specific architecture. To simplify the calculation, the study assumes that the voltage is fixed and adopts the quadratic model [20]. Equation (5) is a modeling expression for the energy consumption from terminal device TU to MEC server.

$$E_{i,j} = E_{i,j}^t + E_{i,j}^c \tag{5}$$

In equation (5), $E_{i,j}^t$ represents the energy consumed by terminal device i in sending task data to MEC server base station j, which is used to describe the energy consumed by the task of terminal device i in MEC server execution. Equation (6) is the energy modeling expression for $E_{i,j}^t$.

$$E_{i,j}^t = p_{i,j} T_{i,j}^t \tag{6}$$

In equation (6), $p_{i,j}$ and $T_{i,j}^t$ respectively represent power and time delay of terminal device i sending task data to MEC server base station j. Equation (5) and equation (6) together constitute the energy consumption of MEC uninstallation, mainly including the energy consumption of uploading task data to the MEC server and the energy consumption of the MEC server in processing computing tasks [21]. This study has

proposed a TU algorithm aimed at minimizing system energy consumption, which is a single optimization method. In practical use, it may encounter the problem of increasing task completion delay. Therefore, to achieve energy optimization through the objective function, it is necessary to combine the maximum task allowable delay with other constraint factors and optimize the solution. This mainly includes task execution maximum tolerance delay constraints, TU constraints, power allocation constraints, transmission rate constraints, and computing resource allocation constraints. Among them, equation (7) is the maximum tolerance delay constraint for task execution.

$$0 \le T_{i,j,O} \le T_i^{\max} \tag{7}$$

In equation (7), $T_i^{\max}$ represents the need to meet the maximum task tolerance delay in TU. Equation (7) is an added delay constraint. To ensure that real-time tasks must be completed before the deadline, this equation guarantees that the total computing time of the selected platform does not exceed the maximum tolerated delay [22]. Equation (8) is the expression for TU constraint.

$$\begin{cases} \sum_{i \in N} x_{i,j} \le \min\{B_j, S_j\} \\ \sum_{i \in N} x_{i,o} \le \min\{B_o, S_o\} \end{cases} \tag{8}$$

In equation (8), B_j represents the maximum accessible tasks number for base station j. Equation (8) serves as a TU constraint, mainly to limit the capacity of resources. The wireless resources and server computing resources that each base station or edge server can access are limited. It is not possible to split all tasks during execution. Therefore, only one of these three platforms can be selected for uninstallation [23]. In the power allocation constraint, the terminal device's transmission power needs to be between 0 and the maximum transmission power in equation (9).

$$0 \le p_{i,j,O} \le p_i^{\max} \tag{9}$$

In equation (9), $p_i^{\max}$ represents the maximum transmission power. Equation (9) mainly stems from the limitations of physical devices. The transmission power of the equipment is restricted by hardware circuits and regulations, such as power amplifier limitations and heat generation, etc [24]. In the transmission rate constraint, the transmission rate of terminal device should be greater than the minimum transmission rate in equation (10).

$$R_{i,j,O} \ge R_i^{\min} \tag{10}$$

In equation (10), $R_i^{\min}$ represents the minimum transmission rate. The transmission rate constraint of equation (10) is mainly limited by service quality or existing channel conditions. The device needs to ensure that it can obtain the minimum rate required for task transmission, which depends on the environmental signal-to-noise ratio, channel bandwidth, and modulation and coding schemes [25]. When calculating resource allocation constraints, due to the relatively limited computing resources of MEC servers and cloud servers, there are restrictions on the allocated resources for tasks uninstalled to MEC servers and cloud servers, as shown in equation (11).

$$\begin{cases} 0 \le F_{i,j} \le F_j \\ 0 \le F_{i,O} \le F_O \end{cases} \tag{11}$$

The constraint on computing resource allocation in equation (11) is due to the limited total CPU computing power of MEC edge servers or cloud servers. Therefore, the sum of the resources allocated to all processed computing tasks cannot exceed the total computing power value [26]. This study aims to establish TU energy consumption models for multiple platforms based on energy consumption optimization and uninstallation energy models. Minimizing the total energy consumption of task execution, taking into account multiple factors such as maximum task tolerance delay, TU, and transmission rate, a corresponding optimization model is established in equation (12).

$$\begin{aligned} \min_{X_{i,j,o}, p_{i,j,o}, F_{i.j.o}} E = {} & x_i \delta_i C_i Q_i^2 \\ & + x_{i,j}\left(p_{i,j} T_{i,j}^t + E_{i,j}^c\right) \\ & + x_{i,O} E_{i,O} \end{aligned} \tag{12}$$

Equation (12) is the main core optimization objective, which is to achieve the minimum energy efficiency under all service requirements. Essentially, it is a complex NP-Hard MINLP problem. An optimal algorithm is applied to model and derive TU strategies across multiple platforms, focusing on minimizing energy consumption. The transformation of practical problems into energy consumption optimization models brings non convex challenges, making it difficult to solve directly through traditional equations. The enumeration method may waste computing resources, prompting heuristic algorithms to obtain more effective solutions.

One of the algorithms is QPSO. It is based on PSO, a technique inspired by the cooperative foraging behavior of birds. Unlike genetic algorithms, PSO does not involve crossover or mutation but conducts random searches in the solution space. In PSO, particles represent potential solutions and are initialized randomly, with each particle updating its position based on its fitness relative to others.

However, PSO can become trapped in local optima due to its fixed velocity and particle trajectory. QPSO mitigates this issue by introducing quantum uncertainty, allowing particles to move unpredictably throughout the solution space. This approach, characterized by limited parameter selection and fast convergence, enhances the ability to find global optima beyond what traditional PSO can achieve. Figure 2 shows the process of QPSO.

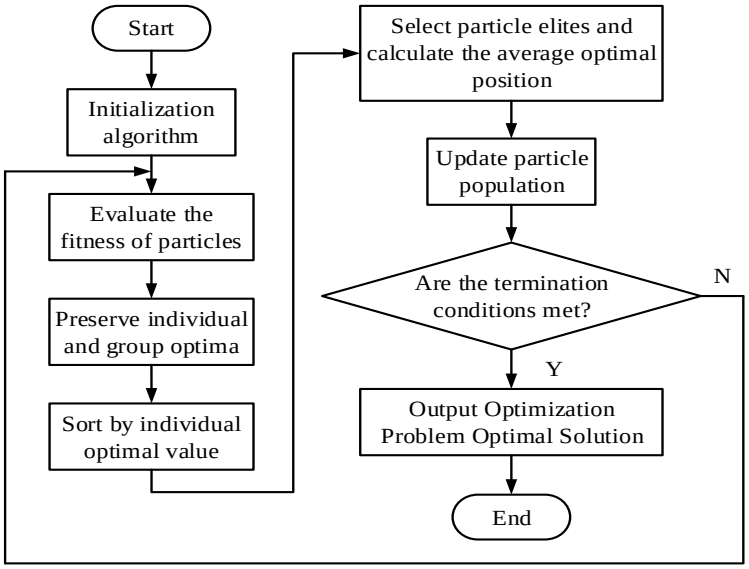


Figure 2: QPSO algorithm flowchart

According to Figure 2, in QPSO, it is necessary to first initialize the algorithm and evaluate particle fitness, while retaining individual and group optimal values. Afterwards, sorting is carried out based on individual optimal values, and particle elites are selected to calculate the average optimal position. The particle population is updated in real-time to determine if it meets the termination criteria. If satisfied, the optimal solution can be output and the process ends. If not, it is necessary to re-evaluate particle fitness and continue cycling until the conditions are met. The innovation in Figure 2 lies in modeling the particle state as a quantum superposition state. The position update follows the wave function collapse law. The particle does not need to move along a continuous trajectory but covers the entire solution space with probability density, enabling the algorithm to have the quantum tunneling ability to cross the local optimal "barrier". In the state of quantum particles, particles no longer have definite positions and velocities. The state of each particle is described by a quantized wave function, which contains the point in space to find the probability density of the particle. Its normalized probability density function is described as equation (13).

$$|\psi(x)|^2 = 1/L_i(t) * \exp(-2|x - p_i(t)|/L_i(t)) \quad (13)$$

In equation (13), $L_{i(t)}$ represents the characteristic length scale, and this variable controls the dispersion of particle positions. $p_i(t)$ represents the local attraction point of particle i at time t, which is usually defined as the individual historical optimal position $p_{\text{best}}(t)$ and the group historical optimal position $g_{\text{best}}(t)$ of the particle. The random weighted average of this point is shown in equation (14).

$$p_i(t) = \phi_i(t) * p_{\text{best}}(t) + (1 - \phi_i(t)) * g_{best}(t) \quad (14)$$

In equation (14), $\phi_i(t)$ represents the uniform distribution of the local attraction points of the particle in the interval [0, 1]. For the particle position update method, QPSO is obtained by folding the wave function. In quantum mechanics, observation causes the wave function to collapse to a definite state. In QPSO, each iteration is equivalent to a "measurement" of particle position. For equation (13), the position update equation of particle *i* at the *t+1*st generation can be derived through the Monte Carlo method, as detailed in equation (15).

$$x_i(t+1) = p_i(t) \pm (L_i(t/2) * \ln(1/u_i(t)) \quad (15)$$

In equation (15), $u_i(t)$ represents a random number uniformly distributed within the interval (0, 1). $\pm$ represents taking the plus or minus sign with equal probability, reflecting the randomness of quantum measurement. Particles can directly "tunnel" to positions far from $p_i(t)$, endowing the algorithm with stronger global exploration capabilities. The characteristic length scale $L_i(t)$ of QPSO is usually related to the state of the entire particle swarm to guide convergence. The most common definition is based on the deviation between the average value $m_{\text{best}}(t)$ of the optimal positions of all individual particles and the current particle position, as detailed in equation (16).

$$\begin{cases} L_i(t) = 2 * \beta * |m_{\text{best}}(t) - x_i(t)| \\ m_{\text{best}}(t) = (1/M) * \sum_{j=1}^{M} p_{j,best}(t) \end{cases} \quad (16)$$

In equation (16), M represents the size of the group. β stands for the contract-expansion coefficient, which is the most crucial control parameter of QPSO, directly determining the convergence behavior of the algorithm and the balance of global/local search capabilities. When β is greater than 1, $L_i(t)$ increases, the search space is broad, and global search is emphasized. When β is less than 1, the search is concentrated near $p_i(t)$, emphasizing the local search. In QPSO, the research aims to linearly reduce β over time to achieve early exploration and later utilization.

3.2 Intelligent fire warning based on optimized QPSO

After solving the TU problem of multiple platforms in OD information system, how to effectively utilize the large amount of data generated by OD information system on a daily basis is an urgent problem that needs to be solved. By mining the hidden relationships between data, people can realize statistics and prediction. Research was conducted on fire warning algorithm based on QPSO, which optimizes neural networks. The temperature, smoke, and CO data transmitted from OD fire protection system are processed using neural network algorithms to predict the likelihood of a fire and provide early warning before it, thereby improving OD safety [27]. Due to the significant losses caused by fires in people's production and daily life, people have conducted in-depth research on the processing methods of fire signals. Through mathematical analysis and processing of the general laws hidden in a fire, predictions can be made about the development of the fire. When a fire occurs, the range of fire signal processing is determined by processing and analyzing these signals detected by sensors to determine whether a fire has occurred. If the fire alarm signals processing is modeled using a function, a variable random signal is composed of fire and non-fire signals in equation (17).

$$x(t)=\begin{cases} x_f(t)+x_m(t) \\ x_m(t) \end{cases} \tag{17}$$

In equation (17), $x_f(t)$ represents the characteristic parameter signal that will be generated when a fire occurs. $x_m(t)$ is a signal change caused by environmental interference, which is a non-fire signal. Figure 3 shows fire detection.

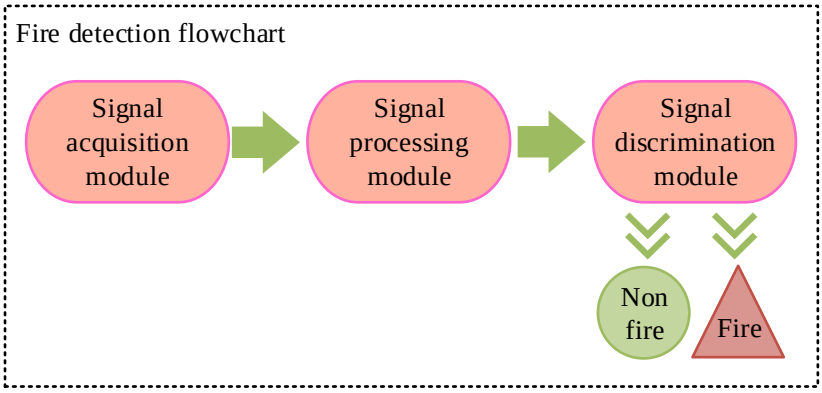


Figure 3: Fire detection flowchart

According to Figure 3, fire detection comprises three main modules: signal acquisition, signal processing, and signal judgment, which distinguish between fires and non-fires. Fire signals are non-stationary random signals, and various factors should be carefully considered when setting thresholds. In fire warning systems, machine learning is employed alongside multi-sensor data to enhance the system's resilience to interference and improve early warning accuracy.

Neural networks, inspired by the human brain, are effective in analyzing information and providing real-time feedback while learning autonomously [28]. Artificial neural networks, particularly BPNN, have shown great efficacy in fire warning applications. In this intelligent fire warning algorithm, the input layer utilizes temperature, smoke, and CO signals. After signal preprocessing, the network is trained to establish the relationship between these signals and fire probability, facilitating accurate fire prediction. Figure 4 shows the neural network fire prediction process.

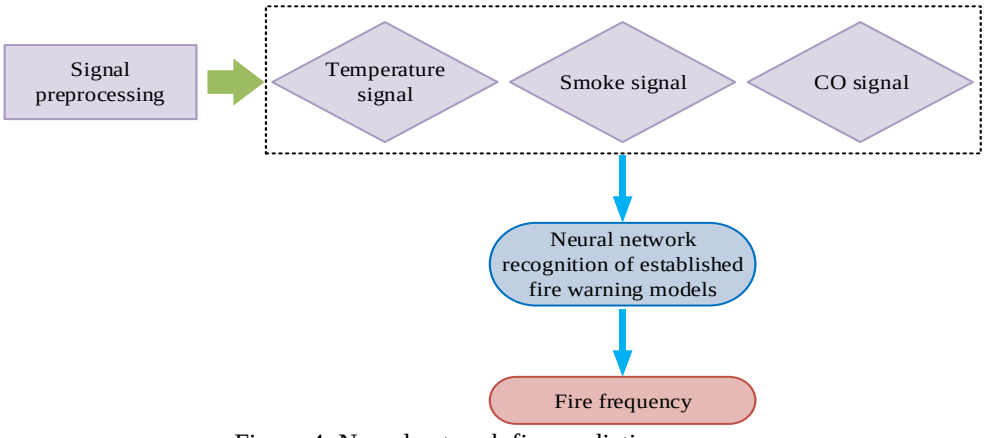


Figure 4: Neural network fire prediction process

In the initial ignition stage of a fire, identifiable characteristics are minimal, with low temperatures and little smoke, making human detection challenging. However, sensors can monitor environmental changes and use intelligent algorithms for early warnings, potentially preventing fire escalation. As the fire smolders, flames are absent, yet thick smoke and significant environmental changes occur. During active combustion, large flames and intense heat raise temperatures rapidly. In the extinguishing stage, the fire weakens, temperatures decline, and smoke is produced.

These four typical fire stages can vary significantly depending on the combustible materials involved, resulting in different fire behaviors. For instance, some substances burn slowly with minimal temperature changes that may go undetected by sensors, leading to false alarms. Others may release volatile substances, produce smoke, and cause similar detection issues. Therefore, fire detection is complex and influenced by multiple factors.

Most ordinary detection (OD) fire protection systems rely on intuitive or system-driven detection methods, which often lack robustness, leading to false alarms and missed detections, endangering safety. Neural network algorithm is used to analyze the feature changes of fire signals, especially temperature, smoke, and CO data. Multi parameter fusion can significantly improve the accuracy and resilience of the system. This method minimizes false alarms and missed detections in complex environments to the greatest extent possible.

To implement this solution, sensor data must be preprocessed and normalized for effective recognition by the neural network. This process enhances network training efficiency and enables rapid and precise fire predictions for early warning, thus improving the safety of OD systems.

The OD fire protection system is to enhance production safety by normalizing data gathered from temperature, smoke, and CO sensors. Integrating QPSO with BPNNs facilitates multi-parameter fusion analysis, improving prediction accuracy and reducing false alarms and missed detections, thereby safeguarding personnel and ensuring operational stability. The initial weight and threshold selection in BPNN are random. Therefore, this method may experience slow convergence and local minimum problems [29]. QPSO's powerful global search capability can effectively optimize these parameters and address these shortcomings. Figure 5 shows the framework of the entire algorithm.

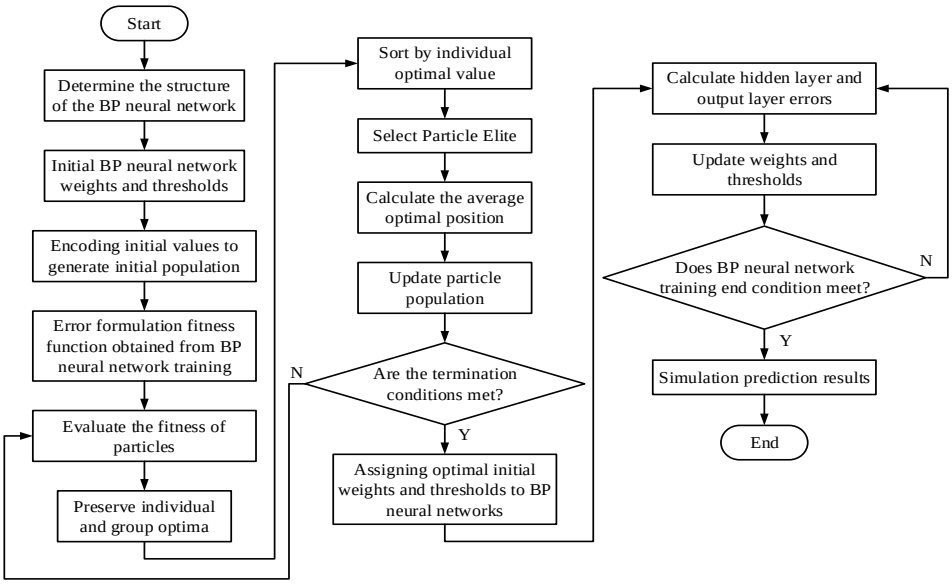


Figure 5: Overall framework of QPSO-BPNN algorithm

In Figure 5, the study integrates physical laws and quantum computing into the fire perception system, forming a triple driving mechanism of "data-physics-quantum". This study innovatively embeds the heat conduction equation as a hard constraint layer into the input, forcing the neural network to learn fire characteristic expressions that conform to thermodynamic laws. This research mainly used QPSO to optimize BPNN to achieve the functionality of OD fire protection system fire warning. The operation process of this algorithm mainly consists of four parts: data preprocessing, neural network structure determination, QPSO optimization of weights and thresholds, and optimized neural network prediction. Each part cooperates with each other to improve prediction accuracy and output fire prediction values. In data preprocessing, this study will use data from temperature, smoke, and CO sensors as inputs to the neural network. The influence between different sensors is eliminated by homogenization, and then the data is normalized. In the neural network structure determination, it is necessary to determine network input layer and output layer, the number of hidden layer nodes, and activation function. Using the Softmax activation function, its output value is

directly interpreted as the predicted probability of each fire category (for example: non-fire, smoldering fire, open flame). The Softmax function ensures that the sum of the predicted probabilities of all categories strictly equals 1, thereby providing an effective and normalized probability distribution. Through research and analysis, both the input and output layers' nodes are 3. The former includes temperature data, smoke concentration data, and CO concentration data. The latter are the probabilities of non-fire, smoldering fire, and open fire, respectively. The hidden layer node can be selected by equation (18).

$$n = \sqrt{n_1 + n_2} + n_3 \tag{18}$$

In equation (18), n, n_1, and n_2 represent the number of nodes in hidden, input, and output layers, respectively. n_3 is a constant between $(0,10)$. Equation (18) is merely an enlightening method for initial topological exploration, and its physical basis is proportional compression for dimensionality reduction in the feature space. In the final deployment model, this equation will be replaced by sensitivity analysis to ensure an exact match between network capacity and data complexity. According to this equation, the hidden layer nodes are 8. Since Sigmoid function is widely used in neural network, it is regarded as an activation function of neural network. The structure of BPNN can be obtained in Figure 6.

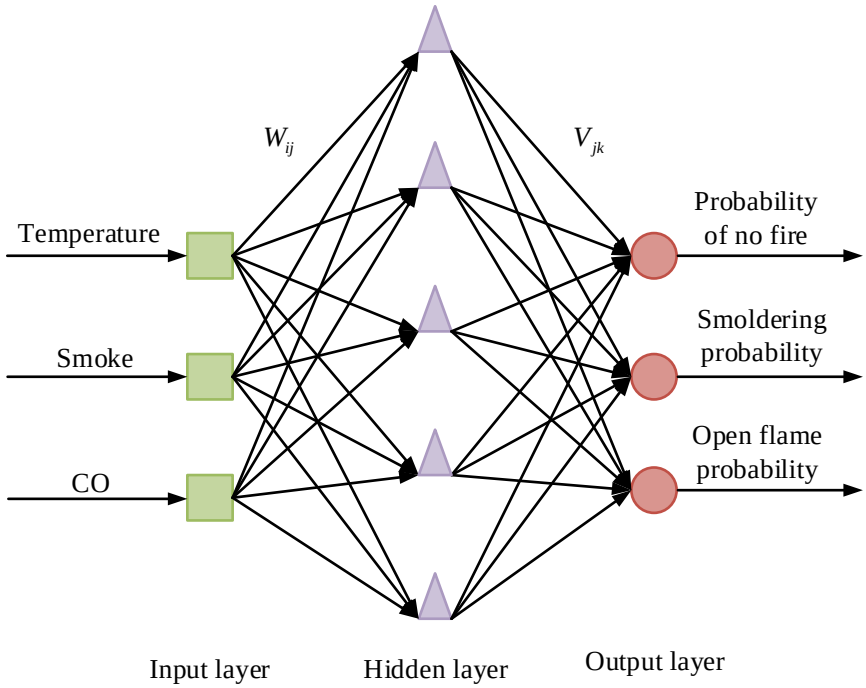


Figure 6: Neural network structure diagram

Figure 6 shows the in-depth customized transformation based on the characteristics of the OD fire scene. Firstly, a multimodal feature separation processing channel is constructed. The temperature branch adopts a temporal convolutional network to capture thermodynamic transient responses. The smoke branch integrates attention gating to enhance feature saliency discrimination. The CO concentration channel sets up a differential amplification link to improve the detection ability of weak signals. After analyzing the BPNN structure, QPSO algorithm is used to optimize its weights and thresholds, to compensate for its shortcomings. The learning performance of BPNN is good, but due to the random selection of weights and thresholds, there are drawbacks such as slow convergence and easy falling into local minima. QPSO algorithm has strong global search ability and achieved good results by optimizing parameters such as weights and thresholds, thereby reducing the computational complexity of neural network and improving accuracy. In the application of QPSO, the initial threshold and weights randomly generated in neural network are first encoded and corresponding to the particles to generate a particle swarm. Then, the optimal algorithm is used to solve a single optimal solution and decoded as a weight threshold. The average error generated by neural network training samples is used to determine individual's fitness value, and then particles are updated.

The fundamental reason for choosing the mean square error (MSE) loss function and the Sigmoid output layer architecture for three-category fire prediction lies in the fact that this combination perfectly fits the multi-state coexistence characteristics of OD fires and the features of industrial data. Sigmoid solves the state coexistence problem by outputting probabilities through independent nodes, avoiding the mutual exclusion constraint distortion of Softmax. The quadratic surface optimization characteristics of MSE loss are superior to cross-entropy on gradient smoothness, multi-peak convergence ability, and outlier robustness.

Finally, the initial weights and thresholds of network were corrected. Where equation (19) is the fitness

function.

$$f(x)=\sum_{i=1}^{w}(y_1-y_2)^2 \tag{19}$$

In equation (19), w represents the number of samples, y_1 and y_2 represent the actual and expected outputs of network, respectively. On this basis, particle state is evaluated by evaluating their fitness. Finally, BPNN is optimized with the optimal solution as weight and threshold. It is optimized using QPSO algorithm to obtain the optimal weights and thresholds, which were used as inputs to BPNN. When training it, further optimization and adjustment can be made based on the feedback information.

Through the above analysis, QPSO-BPNN is a key technical combination. QPSO is an extended PSO algorithm. It utilizes the uncertainty principle in quantum mechanics to optimize the search process of particles, thereby enhancing the global search ability and avoiding falling into local optima. Combined with BPNN, QPSO plays an important role in optimizing the weights and thresholds of neural networks. This optimization can not only accelerate the convergence speed, but also improve the accuracy, thereby enabling more effective prediction of fire situations.

4 Multi-platform TU and fire warning simulation analysis of QPSO

QPSO is simulated and analyzed on multiple platforms TU and fire warning, and MATLAB is used as the simulation software to set the relevant parameters of the simulation environment. The output probabilities of QPSO-BPNN open fire, smoldering fire, and non-fire are fitted in MATLAB. Fire prediction is carried out to output the probabilities of open fire, smoldering fire, and non-fire.

4.1 Simulation analysis of QPSO in multi-platform TU

In the experiment, the research focused on testing three benchmark problems to verify the effectiveness of QPSO-BPNN in the prediction of OD fires. These benchmark issues mainly focus on different scenarios and conditions in fire detection, specifically including the three states of open flame, smoky flame and non-fire. The research constructs a fully connected feedforward neural network model for fire detection. This network consists of an input layer, a hidden layer configured with different numbers of nodes (using the ReLU activation function), and an output layer corresponding to the detection category (using the Softmax activation function). The optimization process adopts stochastic gradient descent with nesterov momentum (SGD), with the learning rate set at a fixed 5e-4 and the momentum factor at 0.9. The training adopts a small-batch strategy, with the batch size fixed at 128. To prevent overfitting, the study adds L2 weight regularization to the loss function, sets the regularization coefficient to 1e-4, and implements an early stop strategy. If the loss of the validation set does not decrease within 10 consecutive training cycles, the training will be terminated (the upper limit of the maximum training cycle is 200). The network weights are initialized using the He initialization method optimized for the ReLU activation function. Training is conducted on a randomly divided training set (70%), with the validation set (15%) used for early stop and the test set (15%) for the final performance reporting.

To construct the training set and test set, a total of 3,600 training samples are selected from the experiments. In addition, 500 test samples are designed to verify the predictive ability of the model. These samples cover different sensor readings and environmental variables that may occur during a fire, enabling the model to be trained in various situations and thereby enhancing its generalization ability and accuracy. During the experimental process of the model, the research processed the input data through standardized preprocessing techniques to eliminate the influence between the readings of different sensors. Meanwhile, in order to evaluate the model effect, not only the training error is focused on, but also the recognition accuracy on the test set is evaluated in detail.

The experiment is conducted in the MATLAB R2023a environment, mainly relying on the following toolboxes: The Neural Network Toolbox is used to construct and train BPNN models, and the Global Optimization Toolbox implements quantum behavior QPSO algorithms to tune network parameters. Performance evaluation and index calculation are carried out using Statistics and Machine Learning Toolbox. All experiments are run on workstations equipped with NVIDIA GeForce RTX 3090 Gpus, taking the Parallel Computing Toolbox to accelerate the training process. The research has clarified the performance indicators for comparison, including convergence speed, solution quality, resource consumption, robustness and scalability. These indicators provide a multi-dimensional evaluation criterion to ensure that the research can comprehensively reflect the performance of different algorithms. Secondly, the study selects a set of representative benchmark problems to cover diverse complexities and characteristics. The selected benchmark problems not only include simple optimization problems, but also cover medium-complexity and large-scale complex problems, ensuring the comprehensiveness of algorithm comparison. In terms of the final experimental design, the research maintains the consistency of the experimental environment, ensures that all algorithms run under the same hardware and software conditions, and conducts multiple experiments to obtain reliable datasets. Meanwhile, detailed parameter tuning is carried out for each algorithm to ensure the fairness of the comparison process. The simulation environment includes a multi platform task processing system composed of multiple cloud servers, edge servers, and terminal devices. Table 1 shows the relevant parameter settings for simulation analysis.

Table 1: Multi-platform uninstallation simulation parameters

Number	Simulation parameters	Value	Number	Simulation parameters	Value
(1)	Number of end device users	30	(6)	MEC server parallel computing maximum number of users	8
(2)	Terminal device computing power	0.5~1G Hz/s	(7)	Number of cloud servers	5
(3)	Number of MEC base stations	10	(8)	Cloud server computing power	50~100GHz/s
(4)	Number of MEC servers	10	(9)	Maximum number of users for cloud server parallel computing	30
(5)	MEC server computing power	5~10G Hz/s	/	/	/

To eliminate the configuration bias, the experiment adopts a hierarchical random sampling strategy to ensure the universality of the results. All range-type parameters in Table 1 are independently sampled according to a uniform distribution at each run. For each algorithm comparison point, 100 independent experiments are conducted. Each experiment regenerates the complete parameter set, and the Halton sequence is adopted to ensure uniform coverage of the parameter space. The specific form of random seed management is shown in Table 2.

Table 2: Random seed management form

```
# The core pseudo-code of the experiment
seeds = np.arange(100)    # Fixed seed range[0,99]
results = []
for seed in seeds:
    np.random.seed(seed)
    config = generate_config(Table1_ranges)    # Generate parameters according to the range in Table 1
    energy = QPSO_optimizer(config)
    results.append(energy)
final_energy = np.mean(results)    # Report average value
C.I = bootstrap_ci(results)         # 95% confidence interval
```

To ensure fairness, system-level hyperparameter tuning is implemented on the PSO baseline model. The search space is defined as follows: The inertia weight values are [0.4, 0.9], the individual learning factor is [0.5, 2.5], the social learning factor is [0.5, 2.5], and the population size is [0.5, 2.5]. By adopting the Bayesian optimization framework and taking the average fitness of 100 independent runs as the objective function, the Pareto optimal solution set is obtained after 50 rounds of iterations. The final optimal parameter values are as follows: The inertia weight is set at 0.72, and the individual learning factor is 1.85. The social learning factor is 1.78. The population size is 60.

The main objective of the research is to minimize the total energy consumption of the system while ensuring fairness in device level energy consumption through constraints. The research conducts a sensitivity analysis on the number of hidden layer nodes of the model and determines the number of hidden layer nodes. The specific results are shown in Table 3.

Table 3: Sensitivity analysis of the number of nodes in the model's hidden layer

Number of nodes	6	8	10	12	15
Smoldering fire recall rate (%)	91.2±0.8	93.5±0.6	95.8±0.5	95.2±0.7	95.7±0.6
Open flame F1 score (%)	98.9±0.3	99.1±0.2	99.3±0.2	99.4±0.1	99.2±0.3
Model inference delay (ms)	4.3±0.2	5.1±0.3	6.4±0.4	7.9±0.5	10.2±0.6
Parameter scale	8.2K	12.7K	18.6K	25.3K	36.8K
Energy consumption (mJ)	0.82	1.05	1.37	1.78	2.31

Based on the sensitivity analysis results in Table 3, the increase in the number of nodes in the hidden layer had a differentiated impact on performance and resource consumption. The initial stage of increasing the number of nodes had a significant impact on the improvement of accuracy. However, after exceeding 10 nodes, the marginal improvement in accuracy was limited, while the model complexity and resource consumption continued to increase significantly. Therefore, choosing 10 nodes achieves a better balance between accuracy and efficiency.

Firstly, a multi-platform TU was conducted using classical PSO and QPSO, and the optimal fitness convergence during the uninstallation optimization process was obtained in Figure 7.

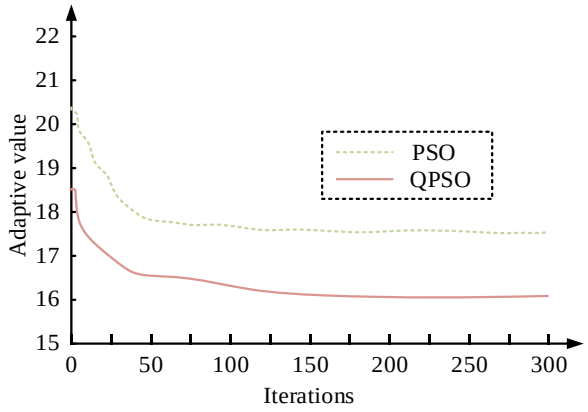
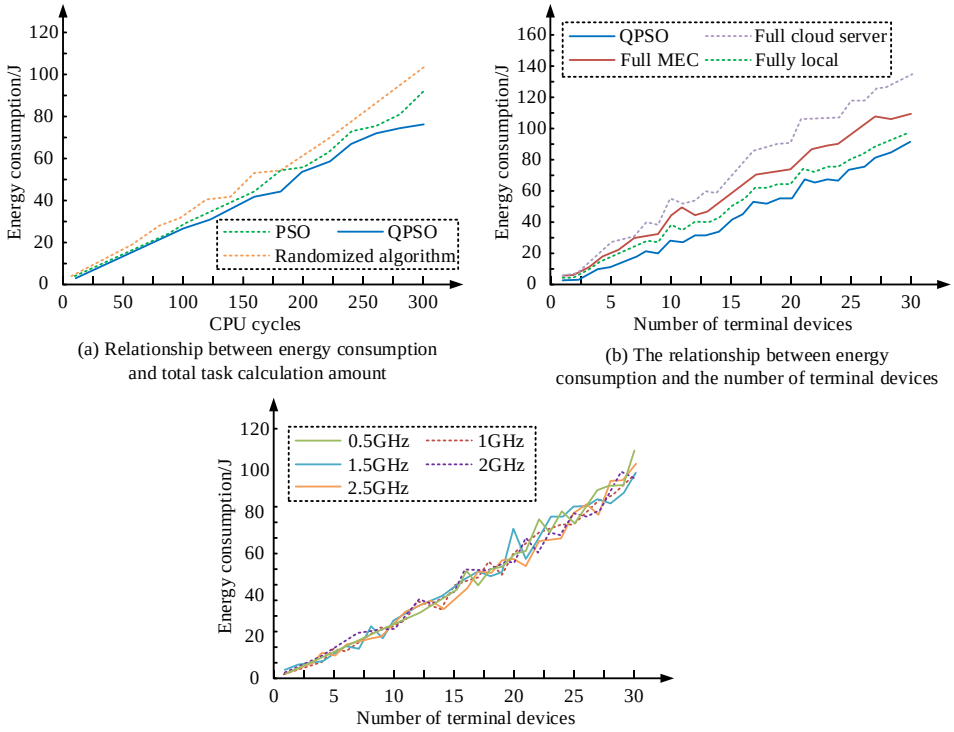


Figure 7: Convergence of fitness values for PSO and QPSO

From Figure 7, compared with PSO, QPSO has faster convergence rate and results. The initial fitness values of PSO and QPSO were 20.2 and 18.4, respectively, and the optimal fitness values of two algorithms were 17.52 and 16.09, respectively. PSO and QPSO achieved optimal fitness values at 231 and 130 iterations, respectively, indicating that QPSO has significant advantages in multi-platform TU. Afterwards, the relationship between energy consumption and total task computation, energy consumption and number of terminal devices, and energy consumption and number of terminal devices under different local computing capabilities is analyzed in Figure 8.



(a) Relationship between energy consumption and total task calculation amount

(b) The relationship between energy consumption and the number of terminal devices

(c) The relationship between energy consumption and the number of terminal devices under different local computing capabilities

Figure 8: The relationship between energy consumption and total task computation, as well as the number of terminal devices

Figures 8 (a), 8 (b), and 8 (c) respectively represent the relationship between energy consumption and total task computation, energy consumption and number of terminal devices, and energy consumption and number of terminal devices under different local computing capabilities. In Figure 8 (a), the order of energy consumption from high to low and stability from low to high is random algorithm, PSO, and QPSO. In comparison, the latter two algorithms can achieve energy optimization under delay constraints, and QPSO is more effective in reducing energy consumption as the computational load increases. Compared with PSO, QPSO could reduce energy consumption by up to 17.1%. In Figure 8 (b), when terminal devices were 20, the energy consumption of multi-platform TU was 55.8 J. The energy consumption for fully local was 64.5 J, for fully MEC was 73.72, and for fully cloud servers was 90.5. Compared with fully local algorithms, the research algorithm saved 13.5%, 24.3%, and 38.3% of energy consumption, respectively. This indicates the effectiveness, and provides ideas for upgrading OD information systems. It can abandon the original centralized server computing mode and introduce MEC and cloud servers to intelligently offload tasks across multiple platforms. This can not only enhance system's risk resistance, but also effectively reduce energy consumption and improve efficiency. In Figure 8 (c), the multi-platform uninstallation algorithm using this research method can steadily reduce energy consumption, improve terminal devices efficiency in completing tasks, and achieve multi-platform TU. The practical significance of industrial applications is that in upgrading

OD information system, there is no need to significantly upgrade the original terminal equipment configuration. Instead, it is necessary to arrange MEC servers and cloud servers, cooperate with this proposed algorithm, and achieve the effect of reducing energy consumption, reducing costs, and upgrading and optimizing. To further verify the feasibility of the proposed method, it is compared with multiple baseline models, including fire detection methods based on the rule-based system, support vector machine (SVM), random forest (RF), one-dimensional convolutional neural network (1D-CNN), and long short-term memory network (LSTM). The comparative experiment results of different models are shown in Table 4.

Table 4: The results of comparative experiments of different models

Scene/Method	Rule system	SVM	RF	1D-CNN	LSTM	QPSO-BPNN
Open flame identification	94.2±1.9	96.8±1.5	97.5±1.3	98.3±1.0	98.9±0.8	99.6±0.8
95% confidence interval	[92.3, 96.1]	[95.3, 98.3]	[96.2, 98.8]	[97.3, 99.3]	[98.1, 99.7]	[98.8, 100.0]
Smoldering fire identification	70.1	78.3	85.4	86.7	90.2	95.8
95% confidence interval	[66.7, 73.5]	[75.4, 81.2]	[82.9, 87.9]	[84.3, 89.1]	[88.1, 92.3]	[94.1, 97.5]
Welding interference false alarm	63.5	32.1	18.7	8.9	5.4	0.3
95% confidence interval	[60.0, 67.0]	[28.8, 35.4]	[16.0, 21.4]	[6.9, 10.9]	[3.8, 7.0]	[0.0, 0.9]
Steam interference false alarm	48.9	25.6	12.3	7.1	3.8	0.1
95% confidence interval	[45.3, 52.5]	[22.5, 28.7]	[10.0, 14.6]	[5.3, 8.9]	[2.5, 5.1]	[0.0, 0.5]
Robustness in high-humidity environments	65.3	82.7	88.5	83.9	91.2	93.6
95% confidence interval	[61.9, 68.7]	[80.0, 85.4]	[86.3, 90.7]	[81.3, 86.5]	[89.2, 93.2]	[91.8, 95.4]

Based on the statistical significance test results in Table 4, the QPSO-BPNN model demonstrated statistically significant superiority in key performance indicators. This model outperformed all baseline methods with an accuracy of 95.8% in the smoldering fire identification task. Its 95% confidence interval [94.1, 97.5] was higher than that of the model LSTM [88.1, 92.3], with no interval overlap and Cohen's d effect size reaching 1.28, indicating that the 5.6% performance improvement was of substantial significance. In the industrial interference scenario, QPSO-BPNN compressed the welding interference false alarm to 0.3%, and the upper limit of the confidence interval at 0.9% was significantly lower than the lower limit of LSTM at 3.8%. The upper limit of the 0.1% range for the steam interference false alarm rate, which was 0.5%, also completely deviated from the confidence range of all baselines, forming a statistical dissociation. Although the interval [98.8, 100.0] of QPSO-BPNN in the open flame recognition scenario partially overlapped with that of LSTM [98.1, 99.7], its 99.6% point estimate still remained leading. In the high-humidity environment adaptability performance, the 93.6% robustness index had limited overlap with the 91.2% interval of LSTM, and further sample expansion verification was necessary. These quantitative evidences confirm the breakthrough progress of this method in reducing the false alarm rate of industry and improving the reliability of smoldering fire detection. To further verify the performance of the model, the study quantifies the full classification performance and false positives of the model. The specific results are shown in Table 5.

Table 5: Model full classification performance and false alarm quantification

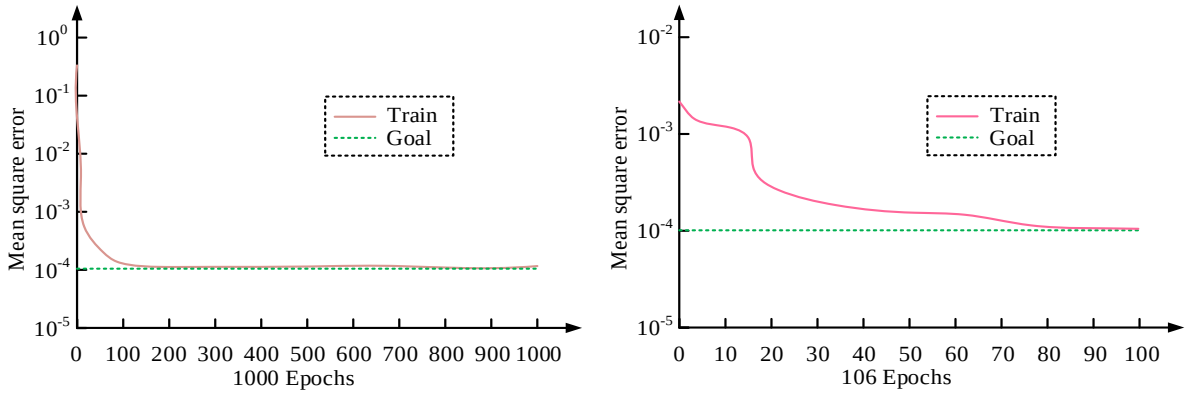
Indicator	Smoldering	Open flame	Non-fire	Macro avg
Precision /%	95.8 ±0.5	99.4 ±0.2	98.1 ±0.3	97.8 ±0.3
Recall /%	95.2 ±0.6	99.3 ±0.3	98.9 ±0.4	97.8 ±0.4
F1 score /%	95.5 ±0.4	99.4 ±0.2	98.5 ±0.3	97.8 ±0.3
Supported sample size	500	500	500	1,500
False positive	21±2	6±1	/	/
False negative	24±3	7±2	/	/

The results in Table 5 show that the proposed model has the best open flame detection performance, with an F1 score reaching 99.4%, followed by the precision and recall of open flames. The F1 score of the smoldering flame was 95.5%. Although it has a relatively high performance, it is lower than that of the open flame test, indicating that it is more susceptible to interference. In terms of false negative risk, the false negative rate of smoldering fire was 24±3 cases. The false positive rate was 21±2 cases, both higher than that of open flames. The results indicate that detecting smoldering fires may delay the fire response. A low false alarm rate for non-fire categories indicates a lower false alarm rate in the system, which helps reduce redundant operations. In safety-critical systems, although the overall recall of fires is relatively high, the false negative of smoldering fires needs to be prioritized for optimization to meet strict safety standards.

3.2Optimization of QPSO Performance and Simulation Analysis of fire warning

The research adopts a strict industrial-grade experimental design, deploying 12 Omega K-type thermocouples (with an accuracy of ±0.75% and a sampling frequency of 10Hz) to monitor the synchronous changes of the temperature field. The smoke detection uses Honeywell XNXTM laser particle counter (with a range of 0-20mg/m^3, calibrated weekly by NIST traceable standard smoke). The CO sensor adopts Alphasense CO-AF electrochemical sensor (with a minimum detection limit of 5ppm, and the drift compensation algorithm is automatically activated every 8 hours). The experiment is conducted in three 20×20m² standard oil tank simulation chambers for 18 months, covering an environmental temperature range of -15℃ to 45℃. Before data collection, environmental background noise baseline calibration is implemented (including dust interference spectrum characteristics ≥78 dB and oil transfer pump vibration noise 15-300 Hz). For each type of fire scene (open flame/smoldering/non-fire), the test is repeated 50 times. Three typical industrial interference sources, namely welding arc, steam injection, and diesel engine exhaust, are introduced to generate countermeasure samples (accounting for 25% of the total data). The humidity control maintains a 30-90% RH gradient through the Vaisala HMP155 probe. All sensors are synchronized in real-time through the OPC UA protocol, with a timestamp accuracy of ±5ms. Eventually, an industrial fire dataset containing 412,800 time-domain samples is constructed. Its noise characteristics are confirmed by Welch power spectrum analysis to comply with the ASTM E681 explosive environment certification standard.

Meanwhile, the research uses representative and authoritative fire data to train the QPSO-BPNN algorithm. The sample data selects typical interference signals in Chinese standard open flames, standard smoldering fires, and kitchen environments. A total of 3,600 samples are selected from each of the three scenarios to form 36 sets of sample data, among which 31 sets of samples are used as the training set and 5 sets as the test set. MATLAB is used for simulation analysis. First, the experiment compares and analyzes the mean squared error of QPSO-BPNN and BP training process in Figure 9.



(a) Mean squared error of BP Neural Network (b) Mean squared error Diagram of QPSO-BP Neural Network

Figure 9: QPSO-BP and Mean squared error during BP training

In Figure 9, the mean squared error of QPSO-BPNN and BP reached the expected error value after 106 iterations and 180 iterations respectively, and the former's convergence rate and global search ability are stronger. QPSO algorithm is used to optimize BP network's weights and thresholds, which can effectively improve algorithm convergence and prevent it from falling into local minima. Afterwards, the outputs of QPSO-BPNN open fire, smoldering fire, and non-fire probabilities were fitted in MATLAB in Figure 10.

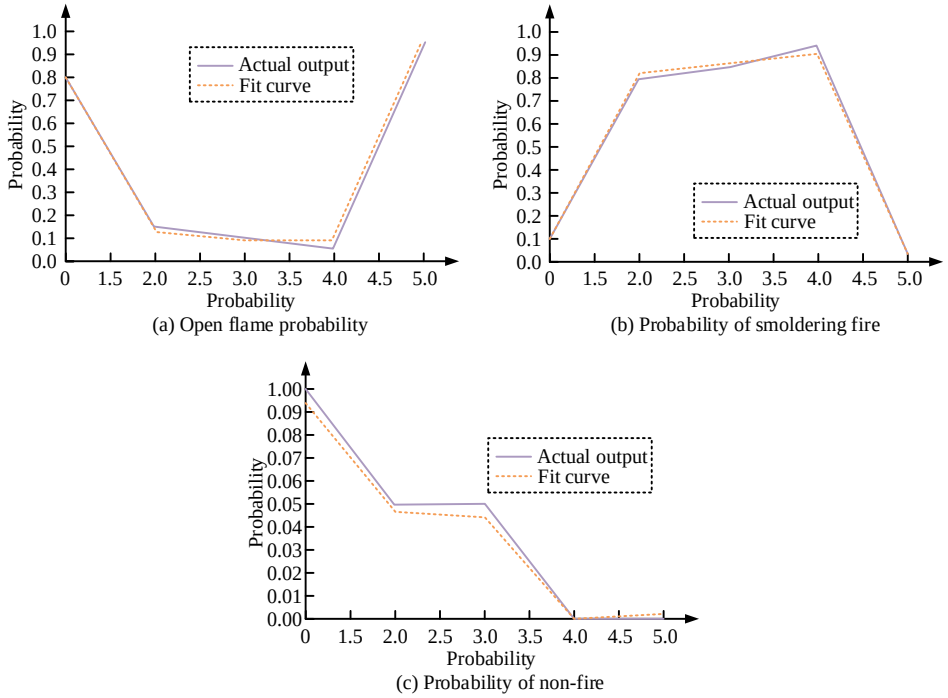


Figure 10: Fitting of probability output of open fire, smoldering fire, and non-fire

According to Figure 10, the output probabilities' numerical curves of open fire, smoldering fire, and non-fire are basically consistent with the actual fire probability, while the maximum error of non-fire probability is only 0.005. This indicates that neural network model can effectively identify and predict fire signals, thereby improving OD fire protection system's safety and accuracy. To further validate this research, a fire experiment was designed using a Raspberry Pi development board paired with temperature, smoke, CO sensors, and Python language. It was programmed and equipped with a multi-sensor data acquisition device. The obtained data was normalized and used as validation samples to test whether the trained model can effectively predict fires.

Fire data collected by multi-sensor data collection equipment will be classified into three states: open fire, non-fire, and smoldering. Five sets of data will be selected from each state to form 15 sets of validation samples to test QPSO-BPNN model's effectiveness. The selected samples were preprocessed and then imported into a model in MATLAB for fire prediction, and the probabilities of open fire, smoldering fire, and non-fire were output. Figure 11 shows the results of an open flame.

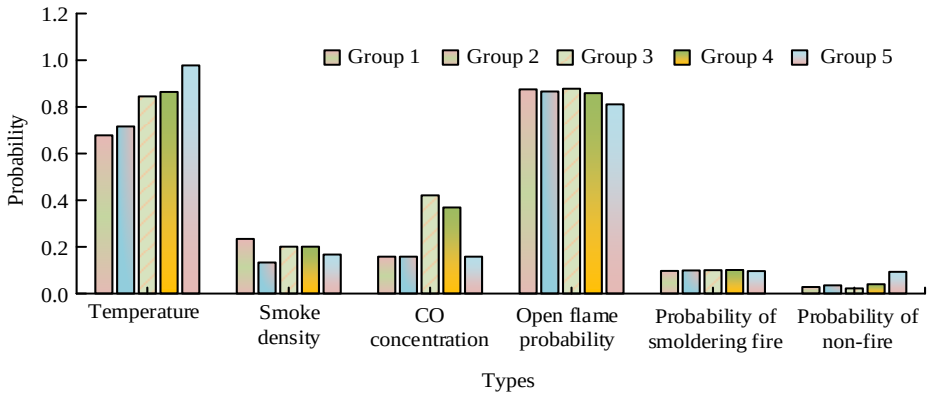


Figure 11: Predictive output of open flame data

From Figure 11, the experimental data of temperature, smoke, and CO are normalized. The output probability obtained by inputting QPSO-BPNN is matched with the corresponding test fire situation. This indicates that QPSO-BPNN model can effectively identify open flame situations and there are no false or missed reporting. Figure 12 shows the output result of smoldering fire.

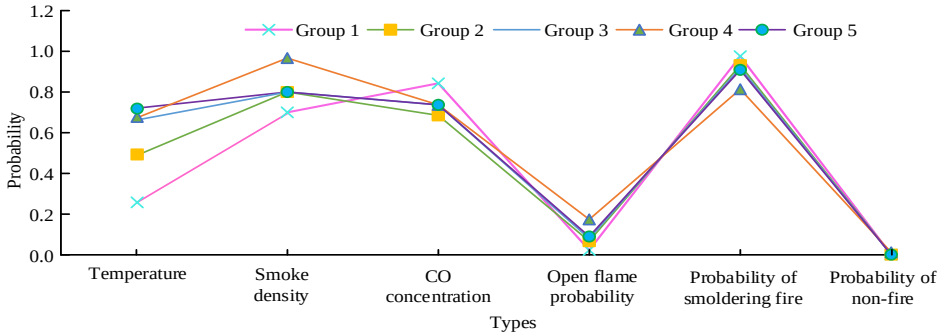


Figure 12: Predictive output of smoldering fire

From Figure 12, the experimental data of temperature, smoke, and CO are normalized. The output probability obtained by inputting QPSO-BPNN is matched with the corresponding experimental fire situation. This indicates that the QPSO-BPNN model can effectively identify smoldering fire situations without any false or missed reporting.

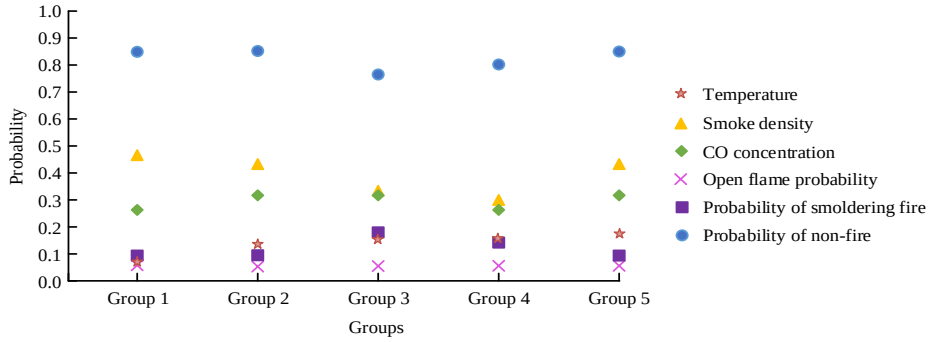


Figure 13: Predictive output without fire

From Figure 13, the experimental data of temperature, smoke, and CO are normalized. The output probability obtained by inputting QPSO-BPNN is matched with the corresponding test fire situation. This indicates that QPSO-BPNN model can effectively identify the situation of non-fire, and there are no false or missed reporting. By judging and predicting the fire situation, accurate judgment and prediction of the fire situation have been achieved, thereby improving the accuracy of oil depot fire warning. The comparison results of the computational complexity of the proposed model are shown in Table 6.

Table 6: Analysis of model computational complexity

Evaluation index	PSO	QPSO	Relative change
Average time of a single iteration	286±12ms	318±15ms	0.112
Floating-point operation (FLOP) in each iteration	7.2×10^9	8.5×10^9	0.181
The number of iterations required for convergence	152±8	87±6	-0.428
Total training time	43.5±2.1s	27.7±1.8s	-0.363
Peak memory usage	1.8GB	2.1GB	0.167
Iterative efficiency index	0.53	1.18	1.226

In Table 6, the QPSO demonstrated a significant computational efficiency advantage over standard PSO. Although the average time consumption of a single iteration increased by 11.2% to 318ms, mainly due to the matrix feature decomposition operation load during the evolution of the quantum wave function, its unique quantum tunneling property significantly reduced the number of iterations required for convergence by 42.8%, and only 87 iterations were needed to achieve the optimization goal. This mechanism effectively avoids the local optimum predicament that traditional algorithms are prone to fall into, ultimately compressing the global training time of the 18.6K parameter model to 27.7 seconds, saving 36.3% of the time cost compared with standard PSO. This efficiency leap is particularly significant in the high-dimensional parameter space, with an iterative efficiency index reaching 1.18, equivalent to a 122.6% increase in the effective convergence step size obtained per unit time. Quantitative computing confirms that the quantum mechanism significantly reduces the total training cost with a controllable single-iteration increment, providing algorithm-level support for the rapid deployment of edge devices.

In response to the core engineering requirements of the fire early warning system in the production environment, this study conducts technical argumentation and trade-off

analysis on the system deployment and real-time performance. In terms of technical demonstration, the system adopts a hierarchical processing architecture to ensure controllable end-to-end latency. At the terminal device layer (deployed on NVIDIA Jetson Nano), the model inference delay is controlled within 11.3ms after being accelerated by TensorRT. When transmitting alarm signals through the LoRaWAN protocol, the transmission delay was 162ms at a maximum distance of 5km. The threshold response time of the cloud-based decision engine was less than 2ms. As a result, the total delay of the worst-case alarm was 175.3ms, which was better than the 1-second limit stipulated in the national standard GB 50116-2023. To cope with sudden loads, the system is designed with a dual-buffered queue. When the concurrent data stream exceeds 50 frames per second, the downsampling mode is automatically activated to stabilize the delay within 200ms. In terms of trade-off analysis, the specific results are shown in Table 7.

Table 7: Measurement of system-level performance trade-off parameters

Optimization dimension	Alarm delay (ms)	Energy consumption (mJ/ Inference)	Fault recovery rate (%)
Edge computing power	11.3	9.8	99.97
	48.7	3.2	99.93
Transmission protocol	162	0.15/pack	98.1
	28	4.7/pack	99.6
Model complexity	6.4	1.37	99.98
	10.2	2.31	99.97
Fault suppression	+8.1ms	+1.9mJ	>99.99
	/	/	95.4

The results in Table 7 showed that although the high-end embedded platform provided an extreme delay of 11.3ms, which was 4.3 times faster than that of the research method mode, the energy consumption cost was as high as 9.8mJ per inference. The LoRaWAN achieved an ultra-low power consumption of 0.15mJ per packet with a delay of 162ms, but it needed to tolerate a network packet loss rate of 2.9%. The current 10-node BPNN achieved a macro average F1 score of 97.8% with a delay of 6.4ms and an energy consumption of 1.37mJ. While the 15-node model improved the recall by 0.5%, the delay increase reached 59% and the energy consumption rose to 2.31mJ. Dual-modal redundancy check had an additional delay of 8.1ms and an energy consumption cost of 1.9mJ, but it increased the system failure recovery rate from 95.4% to no less than 99.99%. Based on the above considerations, the QPSO-BPNN+LoRaWAN combination solution is selected. Under an average daily energy consumption of 86J, it ensures that 99.5% of alarm delays are less than 200ms and fire recall rates are greater than 99%.

5 Conclusion

The intelligent TU strategy of platform was determined based on QPSO. An intelligent fire warning algorithm based on QPSO optimized BPNN was constructed to achieve intelligent prediction of fires. The initial fitness values of PSO and QPSO are 20.2 and 18.4, respectively. The optimal fitness values of two algorithms are 17.52 and 16.09, respectively. PSO and QPSO achieved optimal fitness values at 231 and 130 iterations, respectively, indicating that QPSO has significant advantages in multi-platform TU. Compared with completely local algorithms, the research algorithm has saved 13.5%, 24.3%, and 38.3% of energy consumption, indicating its effectiveness. QPSO algorithm is used to optimize BPNN's weights and thresholds, which can effectively improve its convergence and prevent it from falling into local minima. From the curve plots of open fire, smoldering fire, and non-fire and the actual fire probability, the numerical curve is basically consistent, while the maximum error of non-fire probability is only 0.005. This indicates that QPSO-BPNN model can effectively identify the situation of non-fire, and there are no false or missed reporting. Although breakthroughs have been made in the field of OD fire early warning research, the following key limitations and deepening directions still exist. The current QPSO optimization process relies on the quantum behavior simulation of classical computers and has not yet been integrated with real quantum hardware. The computational complexity of quantum simulators grows exponentially, resulting in the online optimization delay of TU still failing to meet the requirements of millisecond-level industrial control. In fire prediction scenarios, the parameter quantum circuits in quantum neural networks are limited by the simulator bit width, making it difficult to handle sensor fusion features exceeding 100 dimensions. Future research will focus on quantum classical fusion hardware carriers and system level autonomous collaboration mechanisms, developing room temperature quantum coprocessors based on photonic integrated circuits, and achieving task offloading optimization delay compression to the level of 300ms. A quantum blockchain disaster consensus network can be constructed to ensure cross tank instruction transmission through Shor algorithm signature and BB84 protocol encryption.

References

[1] Zhao Y., Song X., Wang F., Cui D., Multiobjective optimal dispatch of microgrid based on analytic hierarchy process and quantum particle swarm optimization, Global Energy Interconnect., 3, 6, 562-570, December 2020. https://doi.org/10.1016/j.gloei.2021.01.008

[2] Wang J., Zhao X., Yu Q., Zhao C., Inverse modeling of thermal decomposition of flame-retardant PET fiber with model-free coupled with particle swarm optimization algorithm, ACS Omega, 6, 21, 13626-13636, May 2021. https://doi.org/10.1021/acsomega.1c00599

[3] Rashad O., Attallah O., Morsi I., A smart PLC-SCADA framework for monitoring petroleum products terminals in industry 4.0 via machine learning, Meas. Control, 55, 7-8, 830-848, May 2022.https://doi.org/10.1177/00202940221103305

[4] Xie S., Dong S., Chen Y., Peng Y., Li X., A novel risk evaluation method for fire and explosion accidents in oil depots using bow-tie analysis and risk matrix analysis method based on cloud model theory, Reliab. Eng. Syst. Saf., 215, Nov., 107791.1-107791.18, May 2021.https://doi.org/10.1016/j.ress.2021.107791

[5] Aliser A., Duranay Z B., Fire/Flame Detection with Attention-Based Deep Semantic Segmentation. Iranian Journal of Science and Technology, Transactions of Electrical Engineering,48, 2, 705-717, February 2024.https://doi.org/10.1007/s40998-024-00697-y

[6] Amjad A., Huroon A M., Chang H T., Tai L C., Dynamic fire and smoke detection module with enhanced feature integration and attention mechanisms. Pattern Analysis and Applications, 28, 2, 1-17, April 2025.https://doi.org/10.1007/s10044-025-01461-6

[7] Hiremath H., Kannan S R., Integrated Anomaly Detection and Early Warning System for Forest Fires in the Odisha Region. Atmosphere, 15, 11, 1284-1284, October 2024.https://doi.org/10.3390/atmos15111284

[8] Kale M P., Meher S S., Chavan M., Kumar V., Sultan M A., Dongre P l., et al. Operational Forest-Fire Spread Forecasting Using the WRF-SFIRE Model. Remote Sensing, 16, 13, 2480-2495, July 2024.https://doi.org/10.3390/rs16132480

[9] Liu P., Xiang P., Lu D., A new multi-sensor fire detection method based on LSTM networks with environmental information fusion. Neural Computing and Applications, 35, 36, 25275-25289, June 2023.https://doi.org/10.1007/s00521-023-08709-4

[10] Hosseinmardi A M., Farvaresh H., Nakhai Kamalabadi I., A new formulation and algorithm for the maximum leaf spanning tree problem with an application in forest fire detection. Engineering Optimization, 55, 7, 1226-1242, June 2023. https://doi.org/10.1080/0305215X.2022.2067151

[11] Liu X., Sun B., Xu Z.D., Liu X., Xu D., Identification of multiple fire sources in the utility tunnel based on a constrained particle swarm optimization algorithm, Fire Technol., 58, 5, 2825-2845, September 2022.https://doi.org/10.1007/s10694-022-01284-5

[12] Zeng M., Quan K., Improved hybrid cuckoo search-based quantum-behaved particle swarm optimization algorithm for bi-level programming, J. Comput. Appl., 40, 7, 1908-1912, March 2020.

[13] Tu S., Rehman O.U., Rehman S.U., Ullah S., Waqas M., Zhu R., A Novel Quantum Inspired Particle Swarm Optimization Algorithm for Electromagnetic Applications, IEEE Access, 8, 1, 21909-21916, January 2020.https://doi.org/10.1109/ACCESS.2020.2968980

[14] Yu L., Han Y., Mu L., Improved quantum evolutionary particle swarm optimization for band selection of hyperspectral image, Remote Sens. Lett., 11, 9, 866-875, September 2020.https://doi.org/10.1080/2150704X.2020.1782501

[15] Zhou H., Bai H., Cai Z., Cai L., Gu J., Tang Z., Container Quota Optimization Algorithm Based on GRNN and LSTM, Acta Electronica Sinica, 50, 2, 366-373, September 2022.

[16] Wang Y., Han X., Jin S., MAP based modeling method and performance study of a task offloading scheme with time-correlated traffic and VM repair in MEC systems. Wireless Networks, 29, 1, 47-68, April 2023.https://doi.org/10.1007/s11276-022-03099-2

[17] Guowei Z., Su Y., Guoqing Z., Pengyue F., Smart firefighting construction in China: Status, problems, and reflections, Fire Mater., 44, 4, 479-486, June 2020.https://doi.org/10.1002/fam.2800

[18] Wu D., Li Y., Du C., Ding H., Li Y., Fast velocity trajectory planning and control algorithm of intelligent 4WD electric vehicle for energy saving using time-based MPC, IET Intel. Transport Syst., 13, 1, 153-159, January 2019.https://doi.org/10.1049/iet-its.2018.5103

[19] Cong Y., Xue K., Wang C., Sun W., Sun S., Hu F., Latency-energy joint optimization for task offloading and resource allocation in MEC-assisted vehicular networks. IEEE Transactions on Vehicular Technology, 72, 12, 16369-16391, July 2023.https://doi.org/10.1109/TVT.2023.3289236

[20] Hua W., Zhou Z., Huang L., Location privacy-aware offloading for MEC-enabled IoT: Optimality and heuristics. IEEE Internet of Things Journal, 10, 21, 19270-19281, August 2023.https://doi.org/10.1109/JIOT.2023.3281609

[21] Bahrami B., Khayyambashi M R., Mirjalili S., Multiobjective placement of edge servers in MEC environment using a hybrid algorithm based on NSGA-II and MOPSO. IEEE Internet of Things Journal, 11, 8, 29819-29837, September 2024.https://doi.org/10.1109/JIOT.2024.3409569

[22] Raza S., Optimizing Energy and Delay in Task Offloading for Connected Vehicles with Proximal Policy Optimization. IETI Transactions on Data Analysis and Forecasting, 2, 2, 29-40, November 2024.https://doi.org/10.3991/itdaf.v2i2.50949

[23] Li H., Liu J., Yang L., Lu L., Sun H., An improved arithmetic optimization algorithm for task offloading in mobile edge computing. Cluster Computing, 27, 2, 1667-1682, June 2024.https://doi.org/10.1007/s10586-023-04048-0

[24] Asghari A., Azgomi H., Zoraghchian A A., Barzegarinezhad A., Energy-aware server placement in mobile edge computing using trees social relations optimization algorithm. The Journal of Supercomputing, 80, 5, 6382-6410, October 2024.https://doi.org/10.1007/s11227-023-05692-4

[25] Banerjee S., Biswash S K., Performance-Driven Resource Allocation Strategy in NDN-Based Mobile Edge Computing (MEC) Networks. Arabian Journal for Science and Engineering, 50, 10, 7741-7764, October 2025.https://doi.org/10.1007/s13369-024-09455-y

[26] Asghari A., Azgomi H., Zoraghchian A A., Barzegarinezhad A., Energy-aware server placement in mobile edge computing using trees social relations optimization algorithm. The Journal of Supercomputing, 80, 5, 6382-6410, October 2024.https://doi.org/10.1007/s11227-023-05692-4

[27] Chen L.Q., Feng Y.X., Zhen H.L., Li Q.V., Lu J.Y., Risk assessment of large crude oil depot based on interpretative structural model and system dynamics, INT J OIL GAS COAL T., 25, 2, 161-183, January 2020.https://doi.org/10.1504/IJOGCT.2020.109445

[28] Mahdiani M.R., Khamehchi E., Hajirezaie S., Hemmati-Saraparden A., Modeling viscosity of crude oil using k-nearest neighbor algorithm, Advances in Geo-Energy Research, 4, 4, 435-447, December 2020.https://doi.org/10.46690/ager.2020.04.08

[29] Bikmukhametov T., Jschke J., Oil Production Monitoring using Gradient Boosting Machine Learning Algorithm, IFAC-PapersOnLine, 52, 1, 514-519, January 2019. https://doi.org/10.1016/j.ifacol.2019.06.114

Mutual Information-Based Feature Facet Clustering with Ensemble SVM for fMRI Visual Object Decoding

Lina Gong
College of Electronic Information Engineering, Xi'an Siyuan University
Xi'an 710038, Shaanxi, China
E-mail: glnxhj@126.com

Keywords: fMRI, brain decoding, visual objects, mutual information, ensemble learning, classification

Received: July 20, 2025

Deciphering visual stimuli from fMRI data presents a significant challenge in computational neuroscience. This paper introduces a novel, optimized ensemble learning framework for high-accuracy visual object recognition. Our method employs a mutual information-based hierarchical clustering technique to automatically segment the high-dimensional voxel space into independent feature facets. An ensemble of Support Vector Machine (SVM) classifiers is then trained on these facets. Crucially, the entire framework—including the number of facets, the fusion operator, and SVM parameters (C, gamma)—is globally optimized using the Simulated Annealing algorithm to ensure peak performance. We rigorously evaluated our approach on three public fMRI datasets: DS105 (8 visual objects), DS107 (4 semantic categories), and DS116 (2 visual oddball stimuli). The proposed model demonstrated exceptional performance, achieving mean recognition accuracies above 95% across all three datasets, with peak subject-level accuracy reaching 100%. Specifically, our Ensemble-965 model (using the detailed Talairach Atlas) attained accuracies of 98.6% on DS105, 97.5% on DS107, and 99.4% on DS116, surpassing current state-of-the-art brain decoding methods under comparable validation conditions. These results indicate that our method provides a robust, accurate, and highly effective solution for visual brain decoding.

Povzetek: Članek predstavi ansambelski okvir za dekodiranje vizualnih dražljajev iz fMRI, ki z hierarhičnim gručenjem po medsebojni informaciji razdeli voxel-prostor na neodvisne "facete" ter z globalno optimizacijo (simulirano ohlajanje) nastavi ansambel SVM za zelo natančno prepoznavo objektov.

1 Introduction

Computational neuroscience has experienced significant expansion in recent years. Deciphering the neural code is a pivotal challenge within this discipline [1], [2]. Numerous research teams are striving to devise practical approaches for analyzing brain signals extracted from fMRI datasets. The exploration of brain decoding has a history dating back twenty-five years. A brain decoder is a computational framework designed to identify meaningful connections between brain function and external inputs. Its primary objective is to categorize neural responses into distinct high-level mental states, including stimulus processing (perceived objects), emotions, and psychological disorders [3], [4], [5]. A secondary goal involves pinpointing brain areas that exhibit pronounced selectivity toward specific stimuli, thereby aiding neurologists in understanding the underlying brain functions. One appealing application of brain decoding is the development of BCI systems. Various neuroimaging techniques, including EEG, MEG, PET, and functional magnetic resonance imaging (fMRI), have been employed in brain decoding studies. Among these modalities, fMRI captures brain functions indirectly by assessing cerebral oxygenation, specifically recognizing the rise in deoxyhemoglobin levels in the blood, known as blood oxygen level-dependent (BOLD) signals [6]. Despite fMRI images being noisy and high-dimensional, they remain a desired method for brain decoding applications due to their non-invasiveness and ability to offer intricate insights into deep brain functions. However, distinguishing neural patterns from spontaneous swings in fMRI data, even when acquired from the same individual, poses a significant challenge [7].

Typically, brain regions that are activated in response to a prompt or task are identified by pinpointing voxels where BOLD signal changes are notably linked to the empirical design. Quantitative analysis of fMRI data commonly employs a highly univariate method [8]. This involves fitting a general linear model (GLM) that accounts for the empirical settings and any potential hidden variables to each voxel's time series, generating a 3D map of parameter estimates [9]. Subsequently, activated regions are delineated using appropriate statistical inference methods. Besides univariate approaches, multivariate methods have been proposed for analyzing fMRI brain activity data, including parametric and non-parametric methods like independent component appraisal [10], clustering [11], and self-organizing mapping [12]. Recent research efforts have concentrated

on resolving the inverse problem of associating specific stimuli, tasks, or mental states with observed brain activity patterns [13]. Various pattern recognition tools, especially ML schemes, are increasingly popular for decoding mental states or stimuli from fMRI data [14], [15], [16].

In this study, the specific challenge of inferring the stimuli delivered to subjects during a visual test utilizing fMRI analysis will be addressed. The visual cortex is recognized for its functional specialization, with distinct patterns of brain activity triggered by different visual stimuli. There is a retinotopic layout in the primary visual cortex, meaning that specific domains within the cortex relate to other sectors of the visual field. Notably, the left visual field is processed in the right hemisphere, and vice versa. The visual pathway is a prime example of how the brain encodes and decodes visual stimuli. This pathway processes visual information starting from the retina's reception of visual stimuli. The optic nerve carries sensory data for vision, leading from each eye to the optic chiasm and the lateral geniculate body, relaying visual information to the primary visual cortex [17]. Fig. 1 demonstrates how the visual cortex encodes visual information, as reflected in fMRI data.

In the current research, a new method based on multimodal learning is suggested to handle the problem of decoding visual objects in the human brain. The new framework proposed in this work is based on mutual information and an ensemble support vector ML approach. Indeed, the main contribution of the current study is utilizing the simulated annealing algorithm in the proposed framework to determine the best voxel subsets in the input voxel space. To this end, this study is guided by the following specific research questions and hypotheses:

- Can a mutual information-based clustering of the voxel space into independent "feature facets" yield a more discriminative ensemble model compared to using unsegmented feature sets or standard anatomical atlases? We hypothesize that clustering voxels based on their mutual information with the stimulus labels will create semantically richer and more independent subsets. The expected benefit is that each classifier in the ensemble becomes a specialized expert on a distinct aspect of the neural code, leading to higher overall decoding accuracy and better generalization compared to models trained on monolithic or arbitrarily segmented feature spaces.

- Does the use of the Simulated Annealing algorithm for global hyperparameter optimization provide a significant advantage over conventional methods like grid search or random search in the context of this complex ensemble framework? We hypothesize that Simulated Annealing will systematically outperform grid and random search by more efficiently navigating the high-dimensional, mixed-variable search space (containing continuous parameters like SVM C/gamma and discrete parameters like the number of facets). We expect SA to find superior configurations with fewer computational resources, thereby maximizing the ensemble's performance potential where simpler methods would converge to suboptimal solutions.

- Can the proposed integrated framework, combining optimized voxel clustering, SA tuning, and an ensemble of SVMs, achieve state-of-the-art decoding accuracy on standardized public fMRI datasets (DS105, DS107, DS116) and outperform current leading methods? We hypothesize that the synergistic effect of our core innovations will result in a robust framework that significantly surpasses the accuracy of existing state-of-the-art brain decoding methods, as listed in Table 4, by effectively addressing their limitations in feature selection, model tuning, and decision fusion.

The rest of this document is outlined as follows: Part 2 supplies a literature review, Part 3 outlines the datasets utilized in the experiments as well as the proposed method, Part 4 showcases the obtained outcomes, Part 5 discusses the findings, and the document concludes with Part 6.

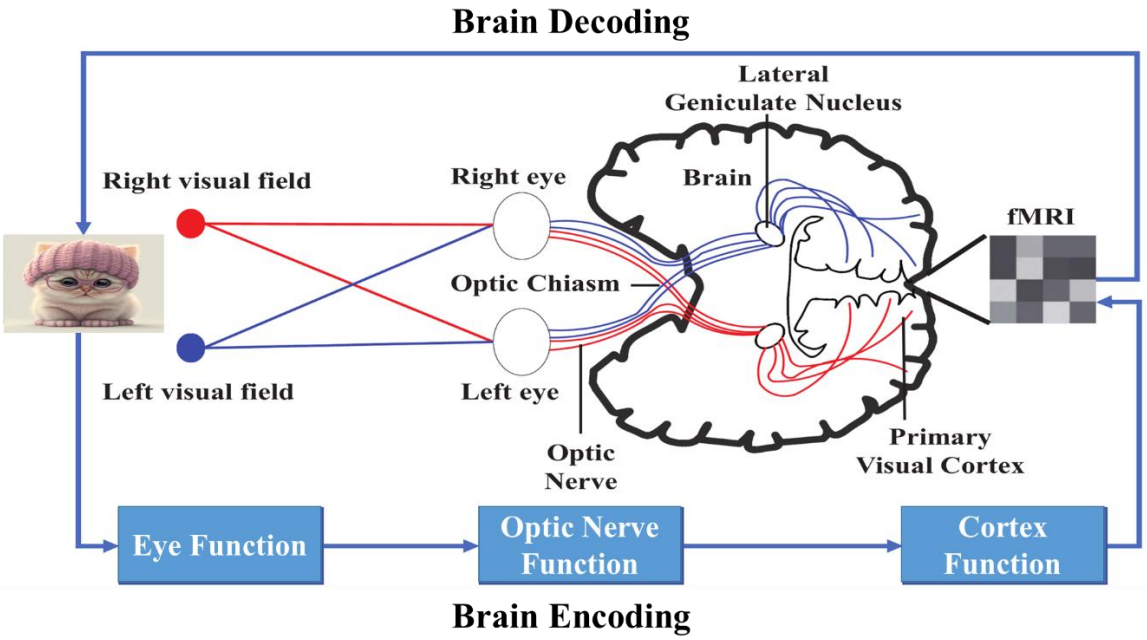


Figure 1: Encoding visual data through the visual cortex, as reflected in fMRI data

2 Related work

Associating brain activities with mental states is seen as a pattern recognition hurdle. Considering this perspective, fMRI data represent spatial patterns, making it viable to utilize statistical pattern identification methods, including ML algorithms, to correlate these spatial patterns with immediate brain states. By employing ML algorithms, researchers can analyze and classify multivariate data by leveraging the statistical characteristics present in the fMRI dataset. Various studies have explored the issue of decoding the brain, particularly focusing on visual object perception. Three primary methods have been proposed to address the brain decoding issue: neural pattern categorization, stimulus recognition, and reconstruction algorithms [18]. This research focuses explicitly on neural pattern classification, which can be categorized into four distinct methods: nonlinear models like the multi-layer perceptron [19], probabilistic methods [20], graph-based decoding [21], and multivariate pattern analysis (MVPA) [22]. In general, the brain decoder typically consists of 3 main stages: preprocessing, voxel (feature) choice, and model (decoder) training. Given the vast and high-dimensional nature of fMRI images, the voxel selection process is pivotal in the efficacy of the decoder [23]. Consequently, numerous early studies have concentrated on developing effective voxel selection techniques. One well-known voxel selection method is GLM, utilized as a neural activity localizer to select voxels in fMRI scans that best align with the task modeling when presented with specific stimuli. Other studies have deployed mutual information between voxels and stimuli as a voxel selection method for activity localization. Additionally, Principal Component Analysis (PCA) has been utilized in certain studies for voxel scoring. The PCA approach involves analyzing the variance in time-series voxels to identify orthogonal components with high discriminatory abilities through a nonlinear transformation [24]. Furthermore, some studies have leveraged Graph-based methods to pinpoint brain activity and identify the most critical voxels [25].

Moreover, the selection of the classifier holds significant value in brain decoding endeavors. Historically, frameworks relying on a single classifier have been the norm in this field. Commonly utilized classifiers in prior studies include Fisher linear discriminant, KNN, Gaussian Naïve Bayes, and SVM [26]. Interestingly, in various studies, it has been noted that the amalgamation or ensemble of classifiers outperforms individual classifiers [27]. This phenomenon is particularly evident when dealing with high-dimensional classification tasks, including those encountered in fMRI analysis. Some research has already explored using ensembles with fMRI data. For example, [28] employed decision tree ensembles to decode fMRI connectivity patterns, while [29] utilized the AdaBoost classifier to distinguish between drug-addicted individuals and healthy controls. In a separate study [30], various ensemble methods were applied to identify distinctive brain activation patterns in reaction to diverse visual stimuli, albeit on data from individual subjects only. These ensembles comprised configurations deployed for training base classifiers on diverse subsets of the training data (e.g., Random Forest, AdaBoost, Bagging) or diverse subsets of input attributes (Random subspace). Another study [31] also employed Random Forests to decode visual stimuli utilizing varied attribute subsets, where the selection of features was based on Gini Contrast. In a study on 3D motion processing, it was found that direction could be decoded from human V1, MT, MST, and FST areas, with accuracy cue-dependent. MT was specialized for perspective cues while FST was specialized for stereoscopic cues, with FST's activity being more closely linked to perceptual experience [32]. To address cross-subject variability in fMRI decoding, Li et al. introduced a Global-Local Functional Alignment (GLFA) method to project fMRI data from multiple subjects into a unified space. Using a large, novel fMRI-video dataset and a transformer-diffusion model, they achieved state-of-the-art performance in cross-subject video reconstruction and semantic classification [33]. A direct comparison between SVM and CNNs for fMRI decoding revealed that CNNs consistently achieve higher accuracy. Critically, the two methods relied on different neural features for classification, suggesting their combined use can provide a more comprehensive understanding of brain activity patterns [34].

In this exploration, a new approach, drawing on multimodal learning, is suggested to solve the problem of decoding visual objects in the brain. To introduce a streamlined and rapid voxel selection framework, we have adopted a two-tiered selection process influenced by prior research. Initially, we employ a filter approach grounded in statistical modeling and atlas data, which serves to diminish the number of voxels (dimensional space) and reduces the time complexity for subsequent stages. Subsequently, the wrapper component entails a voxel scoring algorithm aimed at selecting the most informative voxel subset, accompanied by the simulated annealing search algorithm. The core contributions of our work are threefold:

1. Structured Feature Space Decomposition: We propose a mutual information-based hierarchical clustering method to automatically segment the high-dimensional voxel space into meaningful, independent feature facets. This moves beyond arbitrary or anatomical subdivisions to create clusters based on the informational relationship between features and stimulus labels, ensuring each facet captures a distinct aspect of the neural code.

2. Robust and Automated Optimization: We integrate the Simulated Annealing (SA) algorithm to automatically and efficiently determine the global optimum of the entire ensemble's configuration. This includes the number of feature facets, the SVM parameters (C, gamma), and the fusion operator. This systematic optimization ensures our model is consistently configured for peak performance without manual tuning, a significant advantage over ad-hoc approaches.

3. A Comprehensive and Reproducible Framework: We validate our method on three public datasets, demonstrating state-of-the-art accuracy. More

importantly, we provide a complete and transparent account of our method, including SA parameters, optimization robustness (30 independent runs), and optimal configurations, ensuring full reproducibility and a strong baseline for future work.

3 Methodology

3.1 fMRI databases

This exploration examines three real fMRI databases focusing on the item-observation task in the brain. These databases were sourced from the "openfmri.org" website, which has since evolved into "openneuro.org." Both websites guarantee that human subject data has been de-identified through methods including eliminating facial attributes from high-detail anatomical MRI and removing personal information. The initial database (DS105) is a typical benchmark for studying visual item recognition tasks within the brain (https://openneuro.org/datasets/ds000105/versions/00001). DS105 contains eight classes of visual items for six persons, including face, house, shoe, chair, cat, and scrambled grey-scale photos. This collection of data focuses on the analysis of high-level visual cues as binary predictors, encompassing all groups except scrambled images treated as objects, while also considering low-level visual cues for the multiclass projection. The stimuli consisted of black-and-white photos depicting faces, houses, cats, bottles, scissors, shoes, chairs, and random patterns. Each category was selected so that all stimuli within a particular category shared a common base name. Control random patterns were created by distorting images of the intact objects. Twelve sets of data were collected for each participant. Each set began and ended with 12 seconds of rest, and included eight blocks of stimuli, each lasting 24 seconds, corresponding to the eight categories with 12-second breaks in between. The stimuli were displayed for 500 milliseconds with a 1500-millisecond gap between them. Meaningful stimuli were repetitions of images depicting the same face or object photographed from different perspectives. For each category of meaningful stimuli, there were four images for each of the 12 other examples.

The second database (DS107) contains fMRI data relevant to viewing four diverse groups associated with visual items and textual writings, including consonant, word, object, and scrambled images (https://openneuro.org/datasets/ds000107/versions/00001). The word stimuli, totaling 168 in number, were comprised of 4 or 5-letter words with standard spellings, for example, "hope." They were either monosyllabic or disyllabic and had a written word occurring 40 times or fewer in British English. The stimuli in the two sessions were precisely synchronized in terms of frequency, familiarity, imaginability, count of letters, and count of syllables. Meanwhile, the object cues encompassed grayscale images (200 × 250 pixels) of easily identifiable items, including a boat, a tent, a nail, and so on. The scrambled items were created by partitioning the images into 10 × 10 pixel squares and rearranging their placement in the photo. After scrambling, none of the resulting pictures were distinguishable. Lastly, strings of consonant letters were nonsensical sequences created haphazardly to match the length of the word stimuli precisely.

The 3rd database (DS116) contains both EEG and fMRI data of oddball visual and auditory tasks (https://openneuro.org/datasets/ds000116/versions/00001), only the fMRI visual activity that was used in this work [35]. In this database, 17 individuals took part in a series of three runs, each encompassing similar visual and auditory oddball paradigms. In each task, a total of 375 stimuli (125 per run) were displayed for a duration of 200 milliseconds each, with a variable inter-trial interval uniformly distributed between 2 and 3 seconds, and a target probability of 0.2. It was ensured that the first two stimuli of each run were standard. In the visual task, the target stimulus was a large red circle, while the baseline stimulus was a small green circle, both presented against isoluminant gray backgrounds, with visual angles measuring 3.45° and 1.15°, respectively. A summary of these databases is presented in Table 1.

Table 1: Summary of datasets used in this exploration

Dataset	#of Subjects	#of runs	#of classes	#of time points	Size of images	#of voxels in ROI	TR/TE	FWHM
DS105	6	71	8	121	79×95×79	1963	2.5/30	5 mm
DS107	49	98	4	164	53×63×52	932	2/28	6 mm
DS116	17	102	2	170	53×63×40	2532	2/25	5 mm

TR = time of repetition (s); TE = echo time (ms); FWHM = full width at half maximum.

3.2 Proposed framework

Fig. 2 displays the typical stages of the proposed method. As shown, after collecting the fMRI data from the mentioned databases and preprocessing it, the effective voxels in the atlas regions are selected using a general linear regression model and a z-score. In the next step, the simulated annealing optimization algorithm is used to set important parameters, including the number of feature facets, fusion operator, and classifier parameters. Then, mutual information (between the combination of features and values of voxels) in the fMRI data on the one hand and the labels (categories) on the other hand are obtained. Then, the features are clustered into a certain number of clusters based on mutual information. Thus, each feature cluster expresses an independent feature facet. In the next step, a classifier is trained on each feature facet. Finally, the ensemble process is done using a decision profile matrix, and the final output is obtained. Each of these stages is detailed below.

3.2.1 Data preprocessing

Preprocessing of fMRI data was executed through the fMRI Expert Analysis Tool from the FMRIB software library (FSL, version 6.00) [36]. Before initiating the first-level analysis in FEAT, preprocessing stages were performed on the fMRI data files. This included skull stripping and alignment to the 2 mm MNI152 standard space. Mapping fMRI data onto the MNI template was achieved through FLIRT registration using high-resolution MRI as an intermediate reference. fMRIs in all databases underwent affine registration with 6 or 12 degrees of freedom. Various preprocessing stages, like registration, movement correction, smoothing, and filtering, were conducted in FEAT as the second preprocessing stage.

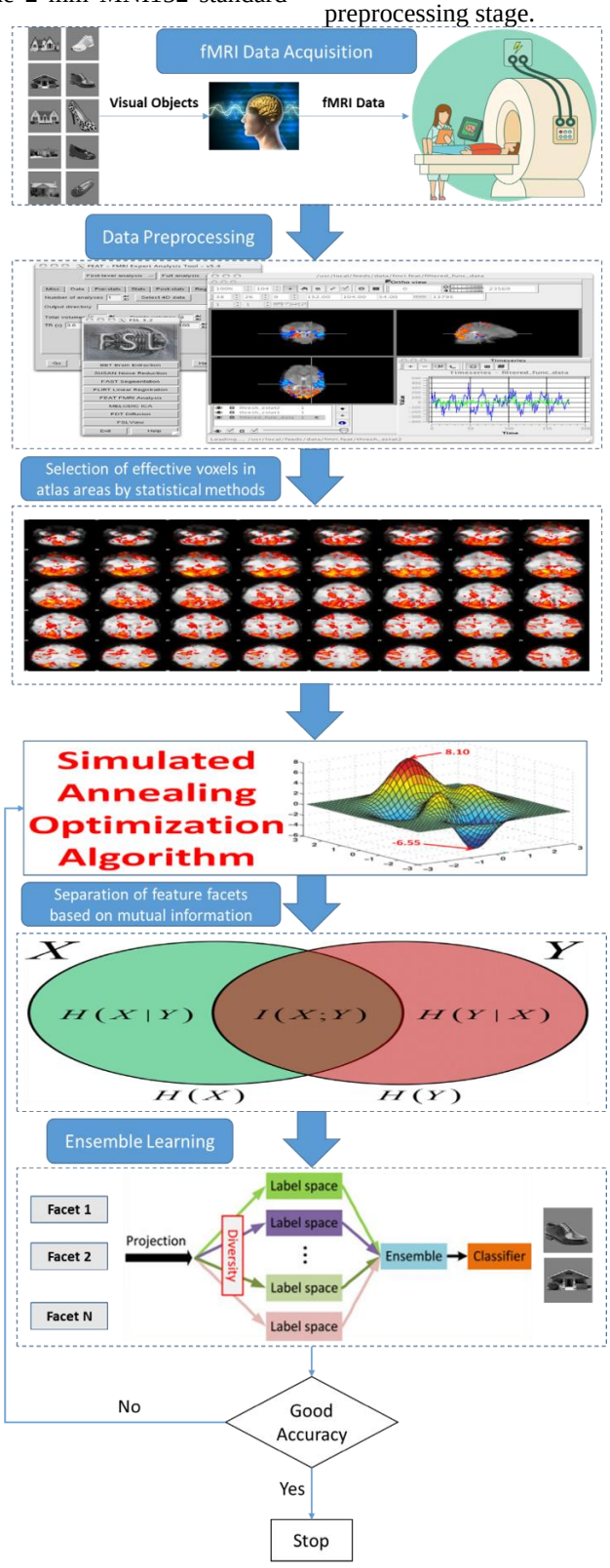


Figure 2: Typical stages of the proposed method

For the DS107 database, an MNI brain mask was applied to remove non-brain tissues from all fMRIs, including structures like the skull, neck, and eyes, using BET for non-brain structure removal, MCFLIRT for movement correction, and spatial smoothing with a 5 mm FWHM Gaussian kernel. The FLIRT registration acquired mapping onto the MNI152 anatomical template with 2 mm resolution using a normal search in 12 degrees of freedom. To expedite and automate the fMRI processing utilizing FSL, a set of UNIX shell scripts was developed. Fig. 3 displays the preprocessing stages for fMRIs.

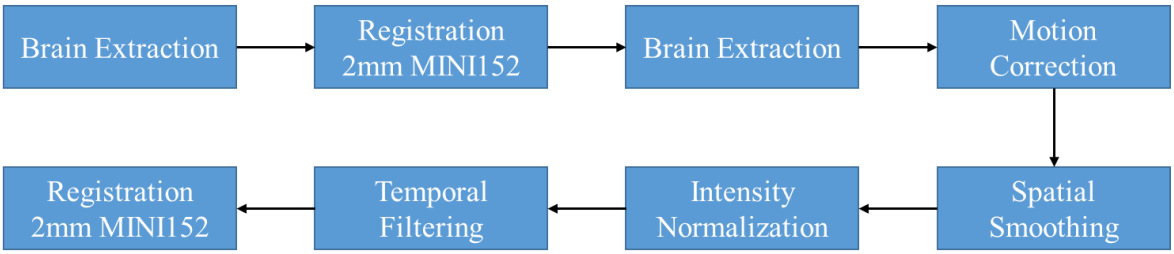


Figure 3: Implemented preprocessing stages for fMRIs in this work.

3.2.2 Feature selection

Feature selection stands as a pivotal step in ML. Within fMRI processing, the challenge lies in managing a multitude of voxels, numbering in the hundreds of thousands. Each voxel displays the activity of a point within the brain space, delineated by 3-dimensional coordinates. Effective voxel selection poses a significant hurdle in computational neuroscience. The objective at this juncture is to diminish the spatial dimensions of the voxels by singling out the most impactful ones, rather than considering all brain voxels. In the standard 2 mm space, the volume of voxels escalates into the millions. Hence, an effective voxel selection method is employed, where an appropriate voxel from each anatomical region in an atlas is chosen for an external stimulus. Examples of such atlases include the Harvard-Oxford cortical Atlas, comprising 96 areas of the human brain cortex, or the Talairach Atlas, encompassing 965 anatomical regions of the human brain cortex [37].

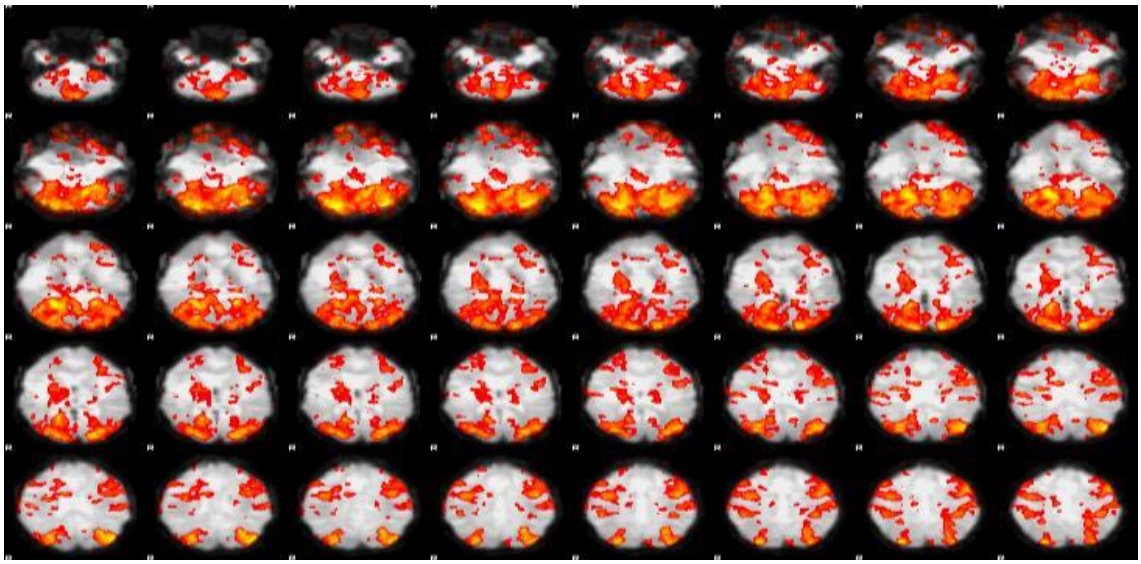


Figure 4: The localization outcomes of the brain activity linked to watching the image of the house

At this stage, the statistical dependence between the stimuli and the time series of the voxels was determined by the general linear regression model (GLM). For each stimulus, a statistical map of the z-score was calculated. For each anatomical region in the Harvard-Oxford cortical atlas (comprising 96 regions), the single voxel that had the maximum z-score was selected. An example of the localization outcomes of the brain activity linked to watching the image of the house can be seen in Fig. 4. The selection process was repeated for each label in the datasets. In this way, the number of features varied according to the type of stimuli and anatomical regions. In the selection algorithm, eight voxels were extracted from each area in the atlas in 9 time intervals. Therefore, 72 features were obtained for each anatomical region. So, using the Harvard-Oxford Atlas, there were 96×72 features.

To encode the temporal dynamics of the hemodynamic response, we did not average across time, as this would discard crucial pattern information. Instead, we employed a concatenation-based approach. For the selected voxel in each of the 96 regions, we extracted its

BOLD signal intensity across 9 consecutive time points following stimulus onset. These 9 time-series values for each region were then flattened and concatenated into a single, high-dimensional feature vector. An important point is to choose a suitable compromise between the number of areas defined in the brain atlas and the amount of practical brain signal information in each brain region. Increasing the number of areas improves the accuracy at the cost of expanding the computation cost. Fig. 5 displays the details of the feature vector structure used.

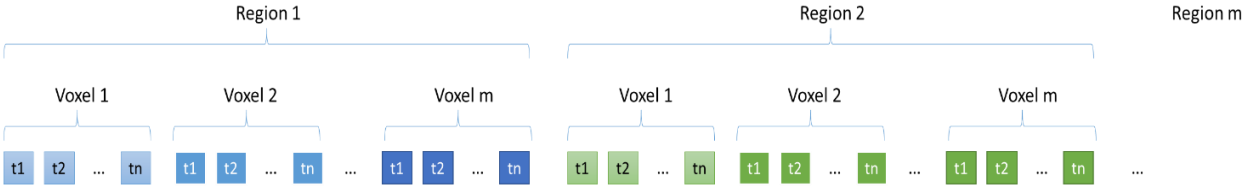


Figure 5: The details of the feature vector structure used in the suggested framework are provided.

After feature selection, voxel post-processing stands as a crucial stage within the suggested technique. This involves the standardization of voxel values and their organization into a vector format for subsequent utilization in the learning processes. The key procedures encompass normalization, segmentation, flattening, discretization, and scaling. Details of these stages are given in the references [23], [38].

3.2.3 Simulated annealing optimization

Learning algorithms, including SVM, are generally dependent on parameters that must be set well to function correctly. The learning method in the proposed framework is an ensemble, which includes multiple SVM classifiers. Conversely, in ensemble schemes, things like the number of feature facets and the fusion operator are of great importance. Therefore, the proposed framework in the learning and classification stage has essential parameters that should be tuned appropriately. Owing to the importance of reducing the search space and setting the optimal parameter, optimization algorithms are used. The selection of an optimization algorithm is critical for navigating complex, non-convex search spaces like the one defined by our ensemble framework, which mixes discrete parameters (e.g., number of facets) and continuous parameters (e.g., SVM C, gamma). We selected the simulated annealing (SA) optimization algorithm for this purpose due to its proven ability to escape local minima (a key advantage over simpler methods like grid search), its effective balance between exploration and exploitation via its cooling schedule, and its established efficacy in machine learning hyperparameter tuning. In this work, the simulated annealing optimization algorithm was utilized for this purpose. Today, this algorithm is widely used in various fields in optimization and searching for problem solutions. In this research, the parameters of penalty and gamma were considered constant for all SVMs, which should be determined by the simulated annealing method. Also, the scheme tries to find the appropriate number of feature facets and the fusion operator. The error rate of the ensemble categorization scheme acts as a fitness function. A simulated annealing algorithm minimizes this error by searching in the space of the number of facets and default parameters in the framework, including annealing speed, exponential function, and acceptance function. The Simulated Annealing (SA) algorithm was configured based on established practices and preliminary experimentation to balance global exploration and local exploitation. The tested ranges and the final selected values for all key parameters are summarized in Table 2.

Table 2: Simulated annealing parameter tuning

Parameter	Tested Range / Options	Selected Value
Initial Temperature (T0)	[10, 500]	100
Cooling Schedule	Linear, Geometric	Geometric ($T_{k+1} = \alpha \times T_k$)
Cooling Rate (α)	[0.85, 0.99]	0.95
Markov Chain Length (L)	[50, 200]	100
Stopping Criterion	Max Iterations, T_{min}	$T_{min} = 0.00001$

The selected configuration was determined to be the most robust through grid search on a subset of the DS105 data. The combination of a geometric cooling schedule with a rate of 0.95 and an initial temperature of 100.0 consistently achieved a lower final error rate compared to other tested configurations. This specific setup allows for aggressive exploration in the early stages, where the algorithm can traverse diverse regions of the complex, high-dimensional search space. As the temperature cools, the search intelligently transitions to a more exploitative phase, finely tuning the parameters in the most promising areas. This balanced approach was critical for reliably finding a near-optimal configuration for the ensemble framework across all datasets.

To ensure the robustness and reliability of the optimization results, the Simulated Annealing algorithm was executed multiple times for each subject and dataset. Specifically, we performed 30 independent runs with different random seeds for each optimization task. This approach allows us to account for the inherent randomness of the algorithm and to statistically confirm that the reported solution is a consistent, high-quality optimum

rather than an outlier. The solution with the lowest ensemble error rate across all 30 runs was selected as the final, optimal configuration for each subject. The standard deviation of the final error rate across these runs was consistently low, confirming the stability of the optimization process and the reliability of the reported parameters.

3.2.4 Automatic segmentation of feature space

Automatic feature space segmentation is a critical step in multifaceted or multimodal learning, especially when there is no natural separation between feature facets. The key innovation of this exploration is to provide a scheme based on mutual information clustering for feature segmentation. Fig. 6 displays the typical stages of the automatic feature space segmentation algorithm. These stages include mutual information calculation and hierarchical clustering. The inputs of the feature space segmentation algorithm are the threshold level of voxel values, the number of feature facets, and brain fMRI data that includes voxel values and labels. First, a threshold is applied to the input feature values to convert these values to 0 and 1. The labels are then binary encoded. In this way, multi-category labels become binary labels. This is similar to converting labels to one-versus-all. An example of converting multi-category labels to binary is displayed in Table 3.

Table 3: Label the binarization process

True Label	Binary Labels			
1	1	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	1

The main discrepancy between the proposed algorithm and other similar methods is that, on the one hand, the threshold value is applied to the features. On the other hand, the labels are encoded to be converted into binary. This process is considered a type of approximation that plays a crucial role in accelerating calculations and mitigating potential hidden noise in mutual information values. Mutual information is used as an essential criterion to minimize uncertainty in automatic feature space extraction and segmentation. The mutual information between any two binary vectors (labels and features) is calculated based on the following equation to obtain the degree of dependence between each pair of vectors.

$$MI(A,B) = \sum_{a\in A}\sum_{b\in B} p(a,b) \times \log\left(\frac{p(a,b)}{P(a).p(b)}\right) \quad (1)$$

Where $MI(A,B)$ denotes the mutual information between 2 variables A and B, $p(a,b)$ denotes the joint probability of A and B, and p(a) and p(b) denote the marginal probabilities of A and B. The output of Eq. (1) is used as input for hierarchical clustering to segment the feature facets. In the following, mutual information rates are calculated for each binary feature in feature label combinations. For example, in the DS105 dataset, eight mutual information values are computed for each feature. These eight values are added together and reported as the final mutual information of the features. After calculating the mutual information values, the hierarchical clustering phase begins. In this step, the number of clusters and the vector containing mutual information values are defined as inputs. The output of clustering is in the form of categories of features that belong to each facet. This process causes features that are similar to each other based on mutual information to be placed in a separate aspect, and the correlation between feature facets is reduced. Fig. 6 displays the typical stages of the automatic feature space segmentation framework.

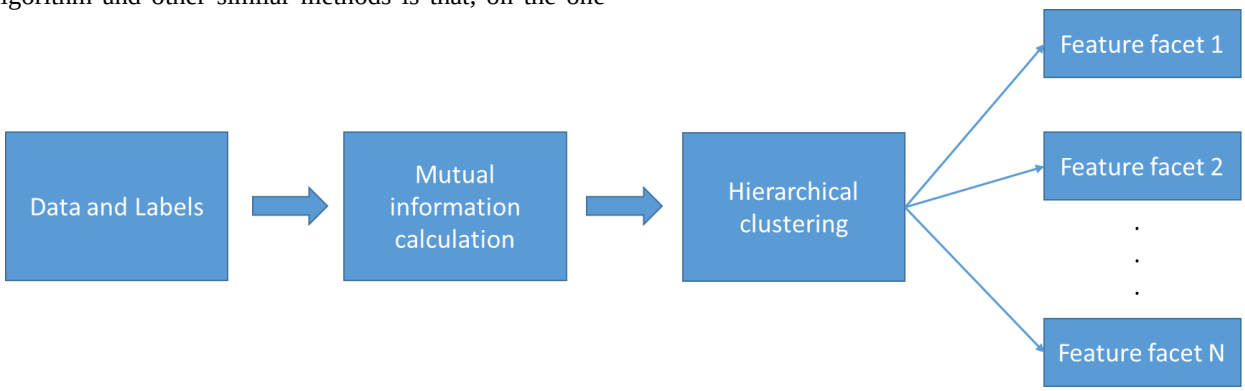


Figure 6: The typical stages of the automatic feature space segmentation algorithm

3.2.5 Classification of each feature facet

In the proposed scheme, an SVM classifier with an RBF kernel was used to classify each feature facet. Thus, each SVM classifier is specific to the classification of one visual object. For example, a classifier is more accurate in identifying human face images than other classifiers. However, it should be noted that the output of each classifier was considered a probability vector. The count of elements of this vector corresponds to the number of labels, and each value of this vector contains the degree to which the test sample belongs to the opposite category. This vector is used in the ensemble learning phase.

3.2.6 Ensemble technique

The main advantage of ensemble learning is to achieve higher accuracy compared to a single model by giving importance to correct answers in diverse classifiers.

Calculating the weight of the test sample belonging to each of the problem categories (including eight visual objects in the DS105 database) is done at this stage. As mentioned, an SVM classifier is trained individually for each feature facet. Then, the decision profile matrix is used to determine the final output for decision-making in the ensemble stage. If there are several classifiers, then the output of each classifier is defined as a membership vector for that classifier. Finally, the output of these classifiers forms a matrix. Fig. 7 displays this matrix.

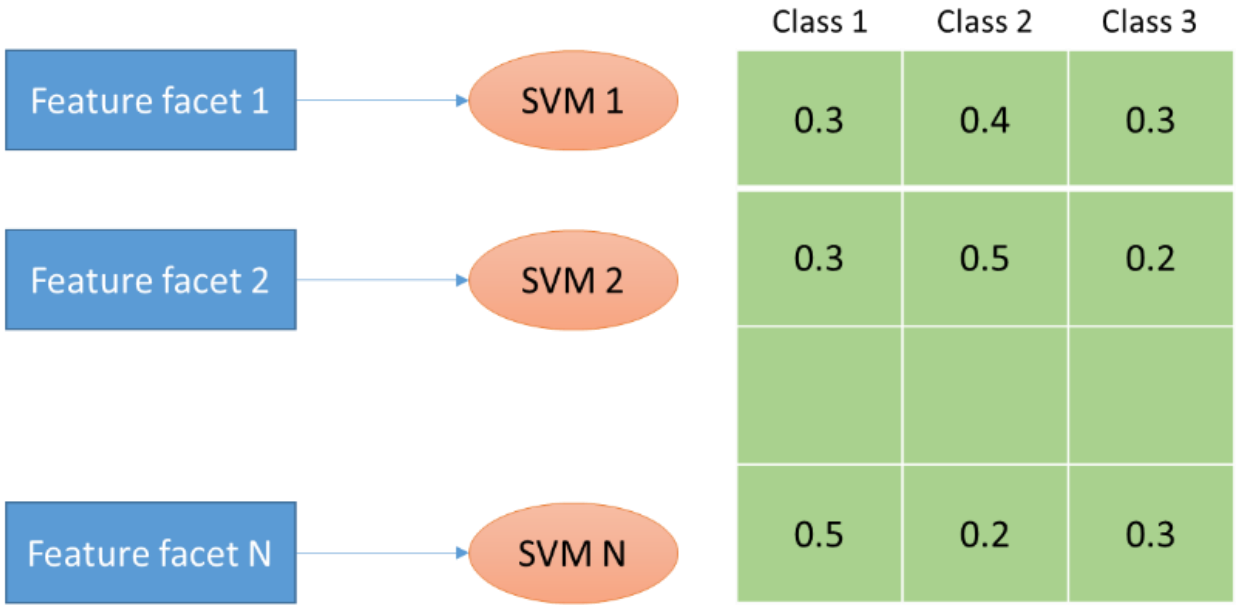


Figure 7: An instance of the decision profile matrix

In this work, the ordered weighted averaging (OWA) method based on the decision profile matrix is used as a fusion operator in ensemble learning. The OWA method is a fusion operator used in ensemble learning, which makes decisions based on a decision profile matrix. In this method, the decision profile matrix displays the decisions made by multiple individual classifiers. Each row of the decision profile matrix corresponds to a particular class or category, and each column corresponds to the output of a specific base classifier in the ensemble. The OWA method operates by assigning weights to the individual classifiers' outputs based on their order in the decision profile matrix. The weights are determined by a specific weighting vector that is pre-defined or learned from the data. The weights are then used to combine the outputs of the individual classifiers into a single fused decision. By using the OWA method, the ensemble learning system can effectively take into account the relative importance of each classifier's decision based on its position in the decision profile matrix. This can lead to improved overall decision-making performance compared to simply averaging the individual classifier outputs or using other fusion methods. Overall, the OWA method leverages the decision profile matrix and weighted aggregation to integrate the diverse outputs of individual classifiers in an ensemble, to enhance the ensemble's predictive capability and performance.

The complete pipeline for the proposed ensemble fMRI decoding framework is summarized in Algorithm 1. The procedure integrates data preprocessing, feature selection, optimization, and ensemble learning into a cohesive workflow.

Algorithm 1. Ensemble SVM with optimized feature facets for fMRI decoding

Input: Raw fMRI data D, Anatomical Atlas A, Stimulus labels y
Output: Trained ensemble model $M_{ensemble}$, Final prediction for test sample $\hat{y}$
Hyperparameters: SA parameters (T_0, α, L, T_{min}), SVM kernel, MI discretization threshold τ
1. // Stage 1: Data Preprocessing
2. $D_{preprocessed}$ ← FSL_FEAT(D)
▷ Skull stripping, motion correction, MNI registration, spatial smoothing (5mm FWHM)
3.
4. // Stage 2: Feature Selection & Vector Construction
5. For each anatomical region $r_i \in A$ do
▷ A is Harvard-Oxford (96 regions) or Talairach (965 regions) atlas
6. Calculate GLM and z-score map for each stimulus condition.
7. Select the voxel v_{max} with the maximum z-score in r_i.
8. Extract BOLD time series for v_{max} across t = 9 consecutive TRs post-stimulus.
9. end For
10. Construct feature vector X by concatenating all v_{max} time series.

▷ Final vector length: |A| × 9
11.
12. // Stage 3: Simulated Annealing Optimization
13. Initialize SA with $T = T_0 = 100$, cooling rate $\alpha = 0.95$, chain length $L = 100$.
14. While $T > T_{min} = 0.00001$ do
15. Generate new candidate solution $\theta_{new} = (n_{facets}, op_{fusion}, C, \gamma)$
16. // Fitness Evaluation:
17. Binarize feature values in X using threshold $\tau \in [0.3, 0.8]$
▷ τ is optimized by SA
18. Binarize multi-class labels y to one-vs-all format.
19. Calculate pairwise MI matrix MI using Eq. (1) with binarized data.
20. Perform Hierarchical Agglomerative Clustering on MI with:
- Linkage Criterion: Ward's variance minimization
- Number of clusters: n_{facets} (from θ_{new})
21. For each feature facet F_k do
22. Train an SVM classifier with:
- Kernel: RBF $K(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2)$
- Parameters: C (penalty), γ (kernel coefficient)
- Both C and γ are optimized by SA
23. End For
24. Fuse base classifier outputs using operator $op_{fusion} \in$ (OWA, Avg, WeightedAvg)
25. Calculate fitness $E(\theta_{new})$ as ensemble error rate via cross-validation.
26. Accept θ_{new} with probability $\min(1, \exp(E(\theta_{old}) - E(\theta_{new}))/T))$
27. $T \leftarrow T \cdot \alpha$
▷ Geometric cooling
28. End While
29. $\theta^* \leftarrow$ best solution found.
30.
31. // Stage 4: Final Model Training & Inference
32. Train final ensemble model $M_{ensemble}$ using optimal parameters θ^*
33. Return $M_{ensemble}$

3.2.7 Evaluation metrics

Accuracy refers to the presence of systematic errors, a measure of statistical bias that leads to discrepancies between the obtained outcomes and the target values, commonly known as trueness. The calculation method for determining the accuracy is outlined as follows:

$$Accuracy = \frac{\#\ of\ correctly\ classified\ samples}{\#\ of\ total\ samples} \quad (2)$$

A confusion matrix is used to evaluate the proposed scheme. Therefore, specificity and sensitivity criteria are calculated through the obtained confusion matrices. In addition, the leave-one-run-out cross-validation scenario is similar to what is used. This scenario is identical to the leave-one-out technique except that the test samples must belong to the same run, and the run samples are not haphazardly selected during test data generation. Furthermore, it is not feasible to distribute the samples from one run across both the test and training datasets. In this way, the data of each subject is divided into subsets of data based on runs. In each iteration, one of the subsets is deployed as test data, and the rest are used as training. The training and testing of the models are repeated until all subsets are used once in the test.

4 Results

Here, the outcomes of the experiments and evaluation of the proposed framework are presented. Note that the assessment of the proposed framework was done separately for two different atlases, Harvard-Oxford (Ensemble-96) and Talairach (Ensemble-965). An intra-subject scenario was deployed to review the proposed scheme. Therefore, the proposed decoding model is trained and tested based on subjects. In this regard, optimal individual parameters were obtained for each subject using the optimization algorithm. In the optimization stage, an interval was considered for searching the values of each parameter. The threshold parameter was searched in the range of 0.3-0.8, and the count of facets was in the range of 3-50. The best-performing configurations, which correspond to the highest average cross-subject accuracy reported are summarized in Table 4. These parameters represent the global optimum found by SA for the respective dataset's characteristic neural patterns and dimensionality. Table 5 displays the identification precision of visual items for all six people from DS105 utilizing the Ensemble-96, Ensemble-965, and a single SVM. Drawing on the acquired precision (mean ± standard deviation), the proposed ensemble algorithm operates very well. Precision rates utilizing the proposed scheme for subjects 5 and 6 even reach 100%. To validate that the performance

improvements of our proposed models are statistically significant and not due to chance, we conducted non-parametric statistical tests. The Wilcoxon signed-rank test was employed for pairwise comparisons due to its suitability for paired, non-normally distributed data (e.g., accuracy scores across subjects). The tests confirm that the superior performance of our ensemble methods is statistically robust. The difference between the Ensemble-965 and Ensemble-96 models is statistically significant ($p < 0.01$), providing strong evidence that the finer anatomical detail captured by the Talairach Atlas directly and meaningfully contributes to higher decoding accuracy. Furthermore, both ensemble models significantly outperform a single SVM ($p < 0.001$), validating the core premise that our ensemble framework provides a substantial advantage.

Table 4: Optimal ensemble framework parameters identified by the Simulated Annealing algorithm for each database.

Database	SVM Penalty (C)	SVM Gamma (γ)	Number of Feature Facets	Fusion Operator
DS105	12.75	0.008	18	OWA
DS107	8.90	0.015	14	Weighted Average
DS116	25.50	0.002	8	OWA

Table 5: Accuracy rates acquired by diverse algorithms for identifying visual objects in DS105.

Subject ID	Single SVM	Ensemble-96	Ensemble-965
1	94.77±5.49	98.11±5.07	95.62±6.54
2	88.91±6.34	96.42±6.10	93.80±7.61
3	92.85±6.20	96.51±6.43	97.36±5.14
4	91.73±7.26	98.12±5.06	99.95±1.05
5	93.82±8.02	99.20±3.61	100
6	96.13±7.91	100	100
Average	93.04 ± 1.98	97.91 ± 1.15	98.62 ± 1.42

The confusion matrices obtained by the proposed Ensemble-96 and Ensemble-965 models for DS105 are shown in Fig. 8. The labels are outlined horizontally, and the column displays the cumulative responses of the proposed scheme. The larger the value of the principal diameter of the confusion matrix, the more precise the categorization. The extra column on the right displays the sensitivity, and the extra row at the bottom of the matrix displays the specificity. The maximum precision is linked to the three groups of house, shoe, and face, and the minimal efficacy pertains to scissors and bottle. The recognition accuracy of the proposed Ensemble-96 model is cumulatively about 97.9%, while it is cumulatively about 98.6% for the Ensemble-965 model. Therefore, considering more anatomical details led to improved accuracy. In the following stage, the efficacy of the proposed scheme was inspected for DS107.

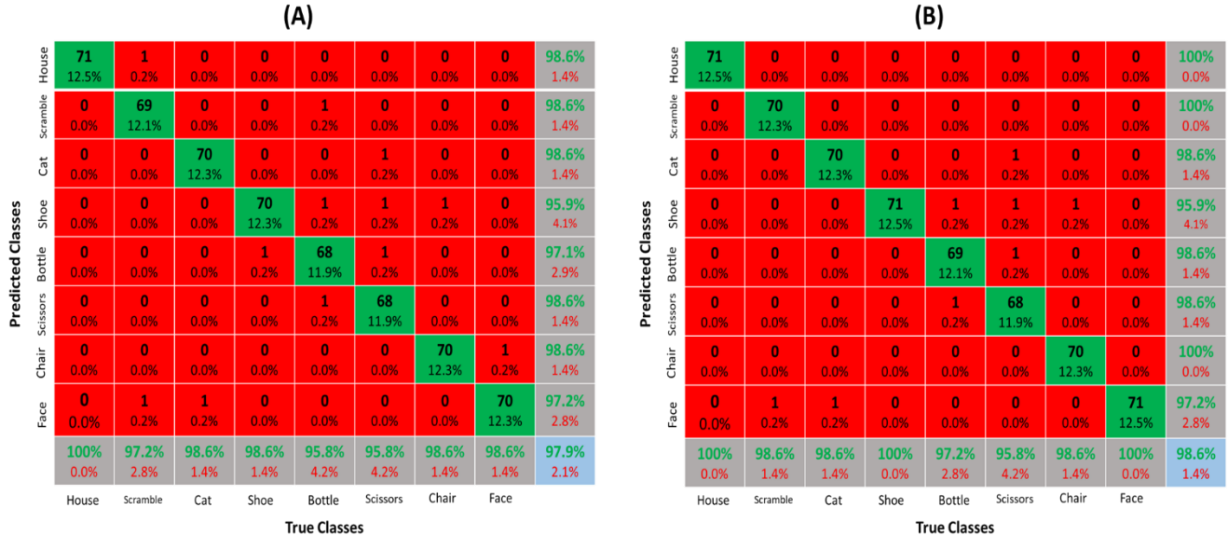


Figure 8: The confusion matrices obtained by the proposed (A) Ensemble-96 and (B) Ensemble-965 models for the DS105 database.

The confusion matrices obtained by the proposed Ensemble-96 and Ensemble-965 models for DS107 are displayed in Fig. 9. Again, the extra column on the right side displays the sensitivity, and the additional row at the bottom of the matrix displays the specificity. The maximum precision is linked to the two groups of scrambled objects and words, and the minimum efficacy pertains to the word group. The recognition accuracy of the proposed Ensemble-96 scheme is cumulatively about 95.1%, while it is cumulatively about 97.5% for the Ensemble-965 model. Fig. 10 displays the Accuracy rates obtained by the Ensemble-965 model in the DS107

database for each subject. As displayed, an accuracy rate of 100% was achieved for 25 subjects.

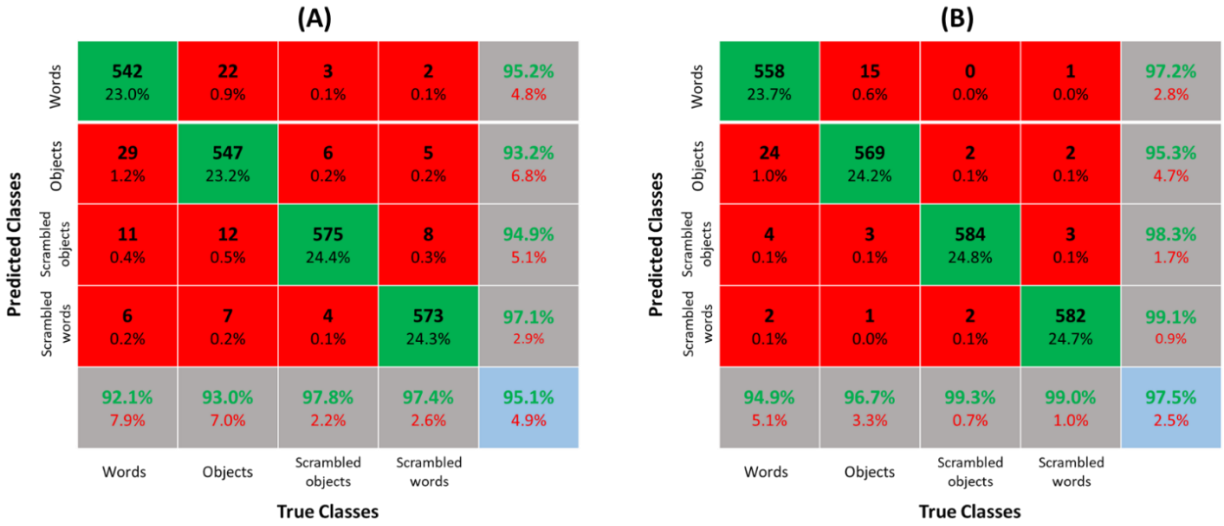


Figure 9: The confusion matrices obtained by the proposed (A) Ensemble-96 and (B) Ensemble-965 models for the DS107 database.

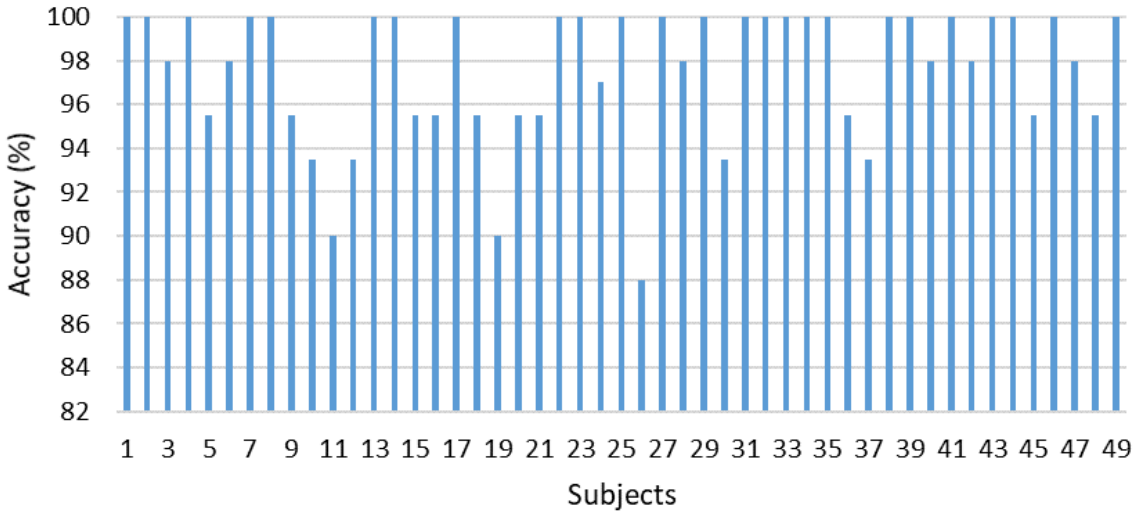


Figure 10: Accuracy rates obtained by the Ensemble-965 model in the DS107 database for each subject.

The confusion matrices obtained by the proposed Ensemble-96 and Ensemble-965 models for DS116 are displayed in Fig. 11. Again, the additional column on the right side shows the sensitivity, and the additional row at the bottom of the matrix displays the specificity. The recognition accuracy of the proposed Ensemble-96 scheme is cumulatively about 98.1%, while it is cumulatively about 99.4% for the Ensemble-965 model. Fig. 12 displays the Accuracy rates obtained by the Ensemble-965 model in the DS116 database for each subject. As displayed, an accuracy rate of 100% was achieved for 11 subjects.

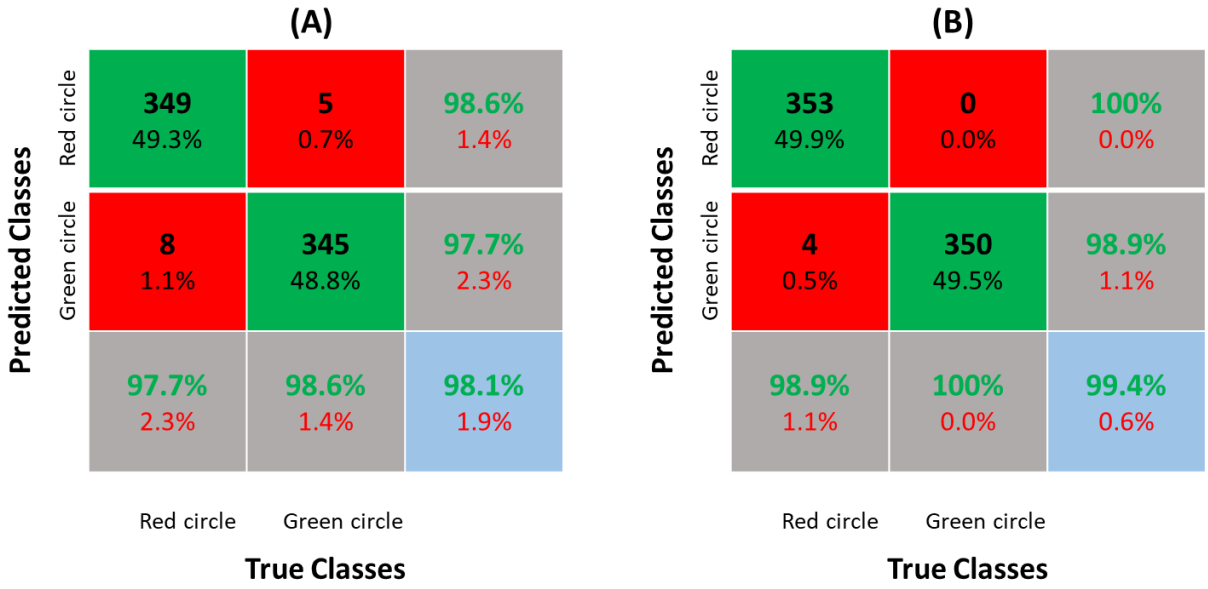


Figure 11: The confusion matrices obtained by the proposed (A) Ensemble-96 and (B) Ensemble-965 models for the DS116 database.

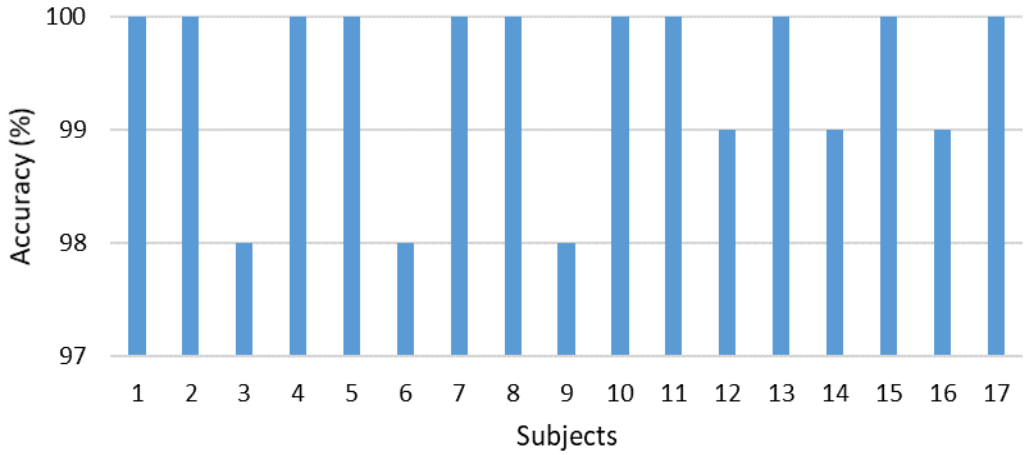


Figure 12: Accuracy rates obtained by the Ensemble-965 model in the DS116 database for each subject.

While accuracy provides a general overview, we report F1-score, precision, and recall to offer a nuanced view of model performance, particularly important for datasets with class imbalances. The results for our best-performing model (Ensemble-965) are summarized in Table 6. The high values for precision, recall, and F1-score across all datasets confirm that the high accuracy is not an artifact of class distribution but reflects truly robust classification. For DS116, the near-perfect and balanced precision and recall scores (99.52% and 99.40%, respectively) definitively demonstrate the model's effectiveness even on a binary task, with no bias towards either the frequent or infrequent class.

Table 6: Comprehensive classification metrics for the Ensemble-965 model

Database	Accuracy	Precision	Recall	F1-Score
DS105	98.62 ± 1.42	98.55 ± 1.61	98.60 ± 1.58	98.57 ± 1.48
DS107	97.54 ± 1.26	97.45 ± 1.38	97.52 ± 1.41	97.48 ± 1.32
DS116	99.43 ± 1.08	99.52 ± 0.95	99.40 ± 1.12	99.46 ± 0.98

To justify the selection of Simulated Annealing (SA), a comparative analysis was conducted against three other prominent metaheuristic optimization algorithms: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and a simple Random Search (RS). The same parameter search space was used for all optimizers. Each algorithm was allocated a fixed budget of 1000 function evaluations (calls to the fitness function, i.e., the ensemble error rate) to ensure a fair comparison. The performance was evaluated based on the final best fitness value (lowest

error rate) found. The results of the comparative optimization analysis are summarized in Table 7. As shown, the Simulated Annealing (SA) algorithm consistently outperformed its counterparts, achieving the lowest final error rate across all three datasets. This confirms its suitability for the high-dimensional, mixed-parameter optimization problem inherent in our ensemble framework. To statistically validate the superiority of SA, a one-way repeated measures ANOVA with post-hoc pairwise comparisons (Bonferroni-corrected) was conducted on the final ensemble error rates. The analysis revealed a statistically significant main effect of the optimization algorithm on the error rate across all datasets ($p < 0.01$). Post-hoc tests confirmed that SA consistently achieved a significantly lower final error rate compared to GA, PSO, and RS (all p-values < 0.01).

Table 7: Final ensemble error rate (%) achieved by different optimization algorithms.

Dataset	Simulated Annealing (SA)	Genetic Algorithm (GA)	Particle Swarm (PSO)	Random Search (RS)
DS105	2.09 ± 0.15	2.85 ± 0.21	3.11 ± 0.30	4.52 ± 0.41
DS107	2.46 ± 0.26	3.45 ± 0.33	3.72 ± 0.28	5.14 ± 0.55
DS116	0.57 ± 0.08	1.12 ± 0.14	1.33 ± 0.19	2.01 ± 0.24

Moreover, the selection of the OWA operator by the SA algorithm was a key outcome of our optimization process. To empirically validate that this choice is superior to common alternatives, we conducted a controlled comparison using the optimized Ensemble-965 architecture on the DS105 dataset, only varying the fusion strategy. A Friedman test with post-hoc Nemenyi tests was used for statistical comparison. The results are summarized in Table 8. The statistical analysis provides clear empirical justification for OWA. The Friedman test revealed a statistically significant difference in performance across the fusion operators ($p < 0.05$). Post-hoc analysis confirmed that OWA's performance was statistically superior to all alternative methods at the $\alpha = 0.05$ level. While the performance gap to Weighted Average was smaller, it remained statistically significant ($p < 0.05$), demonstrating that OWA's dynamic, confidence-based weighting provides a consistent and measurable advantage over static weighting schemes. The larger performance gaps and higher significance levels against simpler methods like stacking, simple average, and majority vote further underscore OWA's critical role in achieving state-of-the-art decoding accuracy.

Table 8: Statistical comparison of different ensemble fusion operators on the DS105 dataset (Ensemble-965 architecture).

Fusion Operator	Accuracy	F1-Score	Statistical Significance (vs. OWA)
OWA	98.62 ± 1.42	98.57 ± 1.48	-
Weighted Average	97.95 ± 1.61	97.88 ± 1.65	0.048
Stacking (Meta-SVM)	97.10 ± 1.88	96.95 ± 1.92	0.044
Simple Average	96.83 ± 2.01	96.70 ± 2.05	0.039
Majority Vote	95.42 ± 2.34	95.25 ± 2.41	0.017

A key innovation of our method is the approximation used in MI calculation, where feature values are thresholded and labels are binarized to accelerate computation and mitigate noise. To validate the necessity of this step, we conducted an ablation study comparing the performance of our full model against two variants on the DS105 dataset (Table 9). The results confirm the critical role of our approximation. Variant A, which used a continuous, non-approximated MI calculation, performed significantly worse. This suggests that the raw, high-dimensional data introduces noise and spurious correlations that degrade the quality of the feature clustering. Variant B showed a substantial improvement, demonstrating that thresholding feature values is the most impactful step, effectively denoising the data and creating a more robust basis for calculating statistical dependencies. Finally, our full model, incorporating both thresholding and label binarization, achieved the highest accuracy. This indicates that the binary label encoding further refines the MI calculation by providing a clearer, more consistent statistical target for the feature clustering algorithm, leading to the creation of more discriminative feature facets.

Table 9: Ablation study on mutual information calculation methods (Mean Accuracy % ± Std on DS105).

Method	Feature Values	Label Encoding	Accuracy (%)
Variant A (No Thresholding)	Continuous	One-Hot	91.23 ± 2.15
Variant B (No Binarization)	Thresholded	One-Hot	95.88 ± 1.54
Proposed Method (Full Model)	Thresholded	Binary	98.62 ± 1.42

The practical deployment of a decoding model depends on its computational demands. We report the average runtime and memory consumption for the complete pipeline (including optimization, training, and testing) per subject in Table 10. Experiments were conducted on a high-performance computing node.

Table 10: Computational performance analysis (averages per subject)

Database	SA Optimization Time (hours)	Model Training Time (minutes)	Peak Memory (GB)
DS105	4.2 ± 0.5	12.3 ± 1.8	3.8
DS107	5.1 ± 0.7	15.6 ± 2.1	4.5
DS116	3.5 ± 0.4	8.9 ± 1.2	2.9

The results indicate that the most computationally intensive phase is the SA optimization, which is a one-time cost per subject to find the optimal configuration. The training time is comparatively modest, and the memory footprint is manageable on standard research computing infrastructure. As expected, the more complex datasets (DS107 with more subjects and DS105 with more classes) required longer optimization and training times. This analysis provides a transparent assessment of the computational resources required to implement the proposed framework.

5 Discussion

Public awareness of BCI has recently increased significantly, turning it into a broadly debated topic in neuroscience [39]. Decoding neural activity in the brain is regarded as a pivotal technology for the expansion of BCIs. This study presented a novel ensemble learning method that directly addresses key limitations in the current state-of-the-art. By moving beyond pre-defined feature subsets through mutual information clustering and replacing manual tuning with a robust global optimization, our framework achieves a new level of automation and performance. Our framework presented a novel ensemble learning method, drawing on feature selection, mutual information, hierarchical clustering, and SVMs for visual brain decoding. 2 different schemes based on Harvard-Oxford (Ensemble-96) and Talairach (Ensemble-965) atlases were tested on three fMRI databases. All investigated frameworks achieved recognition accuracy rates above 95%, indicating the excellent ability of the proposed method for visual brain decoding. Meanwhile, the Ensemble-965 model yielded better outcomes than the Ensemble-96 owing to providing more anatomical details in the model.

The proposed methodology enhances visual object recognition from fMRI data by meticulously selecting and optimizing features, clustering them into meaningful facets, and leveraging an ensemble of classifiers to enhance performance. Each step, from preprocessing to ensemble decision-making, is pivotal in improving the accuracy and robustness of the final model, demonstrating a clear and systematic strategy for addressing this complex issue. The superior performance of our ensemble framework can be attributed to its nuanced approach to feature space decomposition and optimization. While simpler models like standard SVMs provide a strong baseline, our method systematically addresses the high-dimensionality and complexity of fMRI data through mutual information clustering and a globally-optimized ensemble structure. The assessed framework in this work employs a sophisticated methodology designed to enhance the accuracy and efficiency of visual object recognition from fMRI data. Firstly, by applying statistical methods to select significant voxels, the dimensionality of the data is reduced, which in turn helps in reducing computational complexity and improving model accuracy [40]. This foundational step is crucial, but our method advances beyond standard practices. While other methods like L1-regularized SVM [47] induce sparsity, they often discard weakly predictive yet collectively informative voxels. In contrast, our mutual information-based clustering preserves these voxels by grouping them into coherent facets, creating a richer substrate for the ensemble. Simulated annealing helped in exploring the parameter space efficiently to find near-optimal solutions. This stochastic optimization technique helps in finding the global optimum by avoiding local minima, thereby setting the best parameters for feature facets, fusion operators, and classifiers [41]. This represents a significant advantage over methods like the Multi-Objective Cognitive Model (MOCM) [43] or hyperalignment [44], which rely on more constrained optimization or alignment procedures. The ability of SA to co-optimize discrete and continuous parameters globally is a key reason our model adapts so effectively to the distinct characteristics of each dataset, as seen in the consistently high accuracy across DS105, DS107, and DS116.

By clustering features based on mutual information, the methodology ensures that each cluster displays an independent and informative feature facet. This step reduces redundancy and focuses on unique information that improves classification performance [42]. This data-driven decomposition is a primary differentiator. Unlike methods that use pre-defined anatomical atlases or random subspaces, our approach directly tailors the feature space to the statistical structure of the neural code for the task at hand. This explains the performance gap with methods like anatomical connectivity [45], which, while insightful, cannot achieve the same level of functional specificity for decoding.

Moreover, separate training allows the model to learn diverse patterns and relationships within each feature cluster, enhancing the ensemble's ability to generalize across different visual object categories. In addition, combining multiple classifiers through an ensemble method typically yields better performance than individual classifiers by leveraging their collective strengths. The final critical element is our fusion strategy. The selection of the Ordered Weighted Averaging (OWA) operator by the SA algorithm allows for a dynamic, confidence-weighted fusion of decisions. This is more sophisticated than the fixed fusion rules common in other ensembles and enables our model to weigh the opinion of the most reliable expert classifier for any given input more

heavily, leading to the observed improvement in final accuracy.

The juxtaposition of the results of the proposed method with the earlier brain decoding techniques of visual items is given in Table 11. The main discrepancy between the proposed method and other approaches is in the feature space segmentation process, with the help of mutual information between label-feature combinations. The proposed method uses the calculation of mutual information based on the approximate approach, where the labels are encoded in binary and the features are converted to 0 and 1 values based on the application of a threshold. The optimal values of this threshold and the count of feature facets are determined automatically with the help of the simulated annealing optimization algorithm.

Table 11: Juxtaposition of the precision rates of the proposed method with the earlier brain decoding methods of visual items.

Reference	Algorithm	DS105	DS107	DS116
[43]	Multi-Objective Cognitive Model (MOCM) with Gaussian kernel	98.34±0.29	96.79±0.59	97.09±0.33
[44]	Hyperalignment and SVM	87.03±2.87	84.01±1.56	74.62±1.84
[45]	Anatomical connectivity of individual gray-matter voxels	90.82±1.87	85.62±1.95	78.91±2.04
[46]	Hierarchical heterogeneous PSO and SVM	94.46±1.23	89.91±1.67	96.03±0.56
[47]	L1-regularized SVM	85.29±3.49	81.25±3.62	69.24±3.28
[48]	Ensemble multiview learning method based on SVM	98.10	96.30	99.00
[49]	Feature subspaces and multiclass SVM	92.00	93.00	66.00
Proposed Ensemble-96	Ensemble SVM learning with Harvard-Oxford Atlas	97.91±1.15	95.14±1.21	98.12±0.94
Proposed Ensemble-965	Ensemble SVM learning with Talairach Atlas	98.62±1.42	97.54±1.26	99.43±1.08

In Table 11, it is worth noting that this exploration focused solely on the visual scenario of DS116, which was deployed and reviewed, and identified two visual items: the green and red circles. This is even though most of the earlier methods have attempted to separate the visual and auditory modes from each other, and the detection precisions listed in Table 4 are related to the determination of visual and auditory items. For instance, Yousefnezhad and Zhang recommended an MOCM with a Gaussian kernel for multivariate pattern analysis, achieving 97.09% accuracy for the DS116 database [43]. Guntupalli et al. [44] introduced a linear scheme that captures detailed differences among population responses in the human cortex by using response-tuning basis functions shared across individuals, while also modeling cortical patterns of neural responses with individual-specific topographic basis functions. Through a novel algorithm called searchlight hyperalignment and utilizing diverse, dynamic stimuli encompassing visual, auditory, and social percepts, they established a unified model space for the entire cortex. This model effectively aligns representations across multiple brains in various cortical areas, including parietal, occipital, temporal, and prefrontal cortices, as evidenced by between-subject multivariate pattern categorization and intersubject correlation of representational geometry. Using an SVM classifier, this sophisticated model achieved 74.62% accuracy for the DS116 database. Osher et al. [45] suggested that just by examining the anatomical connectivity of specific gray-matter voxels using diffusion-weighted imaging, it is possible to anticipate how individual subjects will respond to visual categories in fMRI. This approach not only explains the functional differences across the cortex but also accommodates the individual variations among subjects. However, they achieved a low accuracy rate of 78.91% for the DS116 database. As displayed in Table 4, the proposed Ensemble-965 scheme yielded better outcomes than the earlier approaches. The proposed Ensemble-96 even had better outcomes than most previous approaches. This displays the superiority of our scheme over previous methods. One of the possible reasons for the benefit of the proposed method compared to previous techniques is the approach adopted to select effective voxels. In the earlier approaches, a specific area of interest was used for each database. However, in this paper, the GLM statistical model and z-score maps were used to find effective voxels. This significantly reduced the size of the input voxel space in our method and greatly increased the learning speed. However, like many other frameworks, the proposed scheme has limitations that must be tackled in future explorations. Although the Ensemble-965 framework produced very good outcomes, it has a high computational complexity that can limit its application, especially in BCI systems.

6 Conclusion

In this article, a new multimodal learning scheme was proposed to resolve the problem of decoding visual objects from brain images. Separation of feature space was done based on hierarchical clustering of mutual information of features and labels. In the current ensemble approach, fusion takes place at the decision-making level, and compared to the fusion approaches at the feature level, it has more speed and simplicity. The automatic separation of the feature space based on the approximate approach of mutual information calculation is an innovation in this

article. The proposed scheme in this study resulted in a improvement in the accuracy of brain decoding of visual objects compared to previous methods. However, the proposed scheme needs further validation using more fMRI databases. In addition, owing to the importance of the mutual information calculation method in selecting effective voxels and distinguishing the feature space, further study is needed to improve the current method. Furthermore, in future studies, more efficient strategies can be used to determine the weights of each single classifier in the ensemble process and test other classifiers besides SVM.

Availability of data and materials

Data is accessible upon request.

Authors' contributions

The investigators contributed to the exploration's conception and construction. Data gathering, modeling, and appraisal were executed by "Lina Gong". The initial draft of the manuscript was written by Lina Gong, who noted on earlier versions of the manuscript.

Ethical Approval

The investigation received ethical review from the IRB, guaranteeing the following of ethical principles and protecting participants' rights.

References

[1] A. Khaleghi, M. R. Mohammadi, K. Shahi, and A. M. Nasrabadi, "Computational neuroscience approach to psychiatry: a review on theory-driven approaches," *Clinical Psychopharmacology and Neuroscience*, Korean College of Neuropsychopharmacology, vol. 20, no. 1, p. 26, 2022. https://doi.org/10.9758/cpn.2022.20.1.26

[2] B. Du, X. Cheng, Y. Duan, H, "Ning. fmri brain decoding and its applications in brain–computer interface: A survey," *Brain Sciences*, vol. 12, no. 2, p. 228, 2022. https://doi.org/10.3390/brainsci12020228

[3] M. Saeidi, W. Karwowski, FV. Farahani, K. Fiok, PA. Hancock, BD. Sawyer, L. Christov-Moore, PK. Douglas, "Decoding task-based fMRI data with graph neural networks, considering individual differences," *Brain Sciences*, vol. 12, no. 8, p. 1094, 2022. https://doi.org/10.3390/brainsci12081094

[4] R. Jabakhanji, AD. Vigotsky, J. Bielefeld, L. Huang, MN. Baliki, G. Iannetti, AV. Apkarian, "Limits of decoding mental states with fMRI," *Cortex*, vol. 149, pp. 101-22, 2022. https://doi.org/10.1016/j.cortex.2021.12.015

[5] W. Xiao, G. Manyi, and A. Khaleghi, "Deficits in auditory and visual steady-state responses in adolescents with bipolar disorder," *J Psychiatr Res*, Elsevier, vol. 151, pp. 368–376, 2022. https://doi.org/10.1016/j.jpsychires.2022.04.041

[6] S. L. Warren and A. A. Moustafa, "Functional magnetic resonance imaging, deep learning, and Alzheimer's disease: A systematic review," *Journal of Neuroimaging*, Wiley, vol. 33, no. 1, pp. 5–18, 2023. https://doi.org/10.1111/jon.13063

[7] T. Varkevisser, E. Geuze, M. A. Van Den Boom, K. Kouwer, J. Van Honk, and R. Van Lutterveld, "Pattern classification based on the amygdala does not predict an individual's response to emotional stimuli," *Hum Brain Mapp*, Wiley, vol. 44, no. 12, pp. 4452–4466, 2023. https://doi.org/10.1002/hbm.26391

[8] C. Wang *et al.*, "'When'and 'what'did you see? A novel fMRI-based visual decoding framework," *J Neural Eng*, IOP Publishing, vol. 17, no. 5, p. 056013, 2020. DOI: 10.1088/1741-2552/abb691

[9] S. Huang, W. Shao, M.-L. Wang, and D.-Q. Zhang, "fmri-based decoding of visual information from human brain activity: A brief review," *International Journal of Automation and Computing*, Springer, vol. 18, no. 2, pp. 170–184, 2021. https://doi.org/10.1007/s11633-020-1263-y

[10] A. M. Golestani and J. J. Chen, "Performance of temporal and spatial independent component analysis in identifying and removing low-frequency physiological and motion effects in resting-state fMRI," *Front Neurosci*, Frontiers Media, vol. 16, p. 867243, 2022. https://doi.org/10.3389/fnins.2022.867243

[11] E. N. Castanho, H. Aidos, and S. C. Madeira, "Biclustering fMRI time series: a comparative study," *BMC Bioinformatics*, BioMed Central (Springer), vol. 23, no. 1, p. 192, 2022. https://doi.org/10.1186/s12859-022-04733-8

[12] L. Liu, C. Hua, Z. Cheng, and Y. Ji, "Intelligent diagnosis method of MRI brain image using parallel self-organizing feature maps neural network," *J Med Imaging Health Inform*, SPIE / American Scientific Publishers, vol. 11, no. 2, pp. 487–496, 2021. https://doi.org/10.1166/jmihi.2021.3285

[13] G. G. Knyazev, A. N. Savostyanov, P. D. Rudych, and A. V Bocharov, "Multi-voxel pattern analysis of fMRI data during self-and other-referential processing," *Zhurnal Vysshei Nervnoi Deyatelnosti Imeni IP Pavlova*, Pavlov Russian Medical Academy (Elsevier partner), vol. 73, no. 2, pp. 242–255, 2023. https://doi.org/10.31857/S0044467723020065

[14] J. I. Glaser, A. S. Benjamin, R. H. Chowdhury, M. G. Perich, L. E. Miller, and K. P. Kording, "Machine learning for neural decoding," *eNeuro*, Society for Neuroscience (SfN), vol. 7, no. 4, 2020. https://www.eneuro.org/content/7/4/ENEURO.0506-19.2020.abstract

[15] AW. Thomas, C. Re, RA. Poldrack, "Interpreting mental state decoding with deep learning models," *Trends in Cognitive Sciences*, vol. 26, no. 11, pp. 972-86, 2022. https://www.cell.com/trends/cognitive-sciences/abstract/S1364-6613(22)00160-7

[16] W. A. Campos-Ugaz, J. P. P. Garay, O. Rivera-Lozada, M. A. A. Diaz, D. Fuster-Guillén, and A. A. T. Arana, "An overview of bipolar disorder

diagnosis using machine learning approaches: clinical opportunities and challenges," *Iran J Psychiatry*, Iranian Psychiatric Association, vol. 18, no. 2, p. 237, 2023. https://doi.org/10.18502/ijps.v18i2.12372

[17] R. M. Awangga, T. L. R. Mengko, and N. P. Utama, "A literature review of brain decoding research," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, 2020, p. 032049. DOI: 10.1088/1757-899X/830/3/032049

[18] C. Du, K. Fu, J. Li, and H. He, "Decoding visual neural representations by multimodal learning of brain-visual-linguistic features," *IEEE Trans Pattern Anal Mach Intell*, IEEE, vol. 45, no. 9, pp. 10760–10777, 2023. https://doi.org/10.1109/TPAMI.2023.3263181

[19] A. Floren, B. Naylor, R. Miikkulainen, and D. Ress, "Accurately decoding visual information from fMRI data obtained in a realistic virtual environment," *Front Hum Neurosci*, Frontiers Media, vol. 9, p. 327, 2015. https://doi.org/10.3389/fnhum.2015.00327

[22] M. Graumann, C. Ciuffi, K. Dwivedi, G. Roig, and R. M. Cichy, "The spatiotemporal neural dynamics of object location representations in the human brain," *Nat Hum Behav*, Nature Publishing Group, vol. 6, no. 6, pp. 796–811, 2022. https://doi.org/10.1038/s41562-022-01302-0

[23] O. Hourani, N. M. Charkari, and S. Jalili, "Voxel selection framework based on meta-heuristic search and mutual information for brain decoding," *Int J Imaging Syst Technol*, Wiley, vol. 29, no. 4, pp. 663–676, 2019. https://doi.org/10.1002/ima.22353

[25] C. Chen, S. T. Weiss, and Y.-Y. Liu, "Graph convolutional network-based feature selection for high-dimensional and low-sample size data," *Bioinformatics*, Oxford University Press, vol. 39, no. 4, p. btad135, 2023. https://doi.org/10.1093/bioinformatics/btad135

[26] B. Du, X. Cheng, Y. Duan, and H. Ning, "fmri brain decoding and its applications in brain–computer interface: A survey," *Brain Sci*, MDPI, vol. 12, no. 2, p. 228, 2022. https://doi.org/10.3390/brainsci12020228

[27] J. Chung and J. Teo, "Single classifier vs. ensemble machine learning approaches for mental health prediction," *Brain Inform*, Springer, vol. 10, no. 1, p. 1, 2023. https://doi.org/10.1186/s40708-022-00180-6

[28] Z. Ye, Y. Qu, Z. Liang, M. Wang, and Q. Liu, "Explainable fMRI-based brain decoding via spatial temporal-pyramid graph convolutional network," *Hum Brain Mapp*, Wiley, vol. 44, no. 7, pp. 2921–2935, 2023. https://doi.org/10.1002/hbm.26255

[29] L. Yang, Y. Du, W. Yang, and J. Liu, "Machine learning with neuroimaging biomarkers: Application in the diagnosis and prediction of drug addiction," *Addiction Biology*, Wiley, vol. 28, no. 2, p. e13267, 2023. https://doi.org/10.1111/adb.13267

[30] Y. Qi, B. Liu, Y. Wang, and G. Pan, "Dynamic ensemble modeling approach to nonstationary neural decoding in brain-computer interfaces," *Adv Neural Inf Process Syst*, NeurIPS Foundation, vol. 32, 2019.

[31] H. Li and Y. Fan, "Brain decoding from functional MRI using long short-term memory recurrent neural networks," in *International conference on medical image computing and computer-assisted intervention*, Springer / MICCAI Society, 2018, pp. 320–328. https://doi.org/10.1007/978-3-030-00931-1_37

[32] P. Wen, L. W. Thompson, A. Rosenberg, M. S. Landy, and B. Rokers, "Single-Trial fMRI Decoding of 3D Motion with Stereoscopic and Perspective Cues," *Journal of Neuroscience*, vol. 45, no. 22, 2025.

[33] C. Li et al., "Enhancing cross-subject fmri-to-video decoding with global-local functional alignment," in *European Conference on Computer Vision*, Springer, 2024, pp. 353–369. https://doi.org/10.1007/978-3-031-73010-8_21

[34] Y. Liang, K. Bo, S. Meyyappan, and M. Ding, "Decoding fMRI data with support vector machines and deep neural networks," *J Neurosci Methods*, vol. 401, p. 110004, 2024. https://doi.org/10.1016/j.jneumeth.2023.110004

[35] M. Yousefnezhad, A. Selvitella, L. Han, and D. Zhang, "Supervised hyperalignment for multisubject fMRI data alignment," *IEEE Trans Cogn Dev Syst*, IEEE, vol. 13, no. 3, pp. 475–490, 2020. https://doi.org/10.1109/TCDS.2020.2965981

[36] A. Bowring, C. Maumet, and T. E. Nichols, "Exploring the impact of analysis software on task fMRI results," *Hum Brain Mapp*, Wiley, vol. 40, no. 11, pp. 3362–3384, 2019. https://doi.org/10.1002/hbm.24603

[37] R. J. Rushmore, S. Bouix, M. Kubicki, Y. Rathi, E. Yeterian, and N. Makris, "HOA2. 0-ComPaRe: A next generation Harvard-Oxford Atlas comparative parcellation reasoning method for human and macaque individual brain parcellation and atlases of the cerebral cortex," *Front Neuroanat*, Frontiers Media, vol. 16, p. 1035420, 2022. https://doi.org/10.3389/fnana.2022.1035420

[38] O. Esteban *et al.*, "Analysis of task-based functional MRI data preprocessed with fMRIPrep," *Nat Protoc*, Nature Publishing Group, vol. 15, no. 7, pp. 2186–2202, 2020. https://doi.org/10.1038/s41596-020-0327-3

[39] N. Shi *et al.*, "Estimating and approaching the maximum information rate of noninvasive visual brain-computer interface," *Neuroimage*, Elsevier, vol. 289, p. 120548, 2024. https://doi.org/10.1016/j.neuroimage.2024.120548

[40] Z. Wen, T. Yu, Z. Yu, and Y. Li, "Grouped sparse Bayesian learning for voxel selection in multivoxel pattern analysis of fMRI data," *Neuroimage*, Elsevier, vol. 184, pp. 417–430, 2019. https://doi.org/10.1016/j.neuroimage.2018.09.031

[41] E. A. Mahareek, A. S. Desuky, and H. A. El-Zhni, "Simulated annealing for SVM parameters optimization in student's performance prediction," *Bulletin of Electrical Engineering and Informatics*,

Institute of Advanced Engineering and Science (IAES), vol. 10, no. 3, pp. 1211–1219, 2021. https://doi.org/10.11591/eei.v10i3.2855

[42] S. Hao, Y. Cui, and J. Wang, "Segmentation scale effect analysis in the object-oriented method of high-spatial-resolution image classification," *Sensors*, MDPI, vol. 21, no. 23, p. 7935, 2021. https://doi.org/10.3390/s21237935

[43] M. Yousefnezhad and D. Zhang, "Multi-objective cognitive model: a supervised approach for multi-subject fmri analysis," *Neuroinformatics*, Springer, vol. 17, no. 2, pp. 197–210, 2019. https://doi.org/10.1007/s12021-018-9394-9

[44] J. S. Guntupalli, M. Hanke, Y. O. Halchenko, A. C. Connolly, P. J. Ramadge, and J. V Haxby, "A model of representational spaces in human cortex," *Cerebral cortex*, Oxford University Press, vol. 26, no. 6, pp. 2919–2934, 2016. https://doi.org/10.1093/cercor/bhw068

[45] D. E. Osher, R. R. Saxe, K. Koldewyn, J. D. E. Gabrieli, N. Kanwisher, and Z. M. Saygin, "Structural connectivity fingerprints predict cortical selectivity for multiple visual categories across cortex," *Cerebral cortex*, Oxford University Press, vol. 26, no. 4, pp. 1668–1683, 2016. https://doi.org/10.1093/cercor/bhu303

[46] X. Ma, C.-A. Chou, H. Sayama, and W. A. Chaovalitwongse, "Brain response pattern identification of fMRI data using a particle swarm optimization-based approach," *Brain Inform*, Springer, vol. 3, no. 3, pp. 181–192, 2016. https://doi.org/10.1007/s40708-016-0049-z

[47] H. Mohr, U. Wolfensteller, S. Frimmel, and H. Ruge, "Sparse regularization techniques provide novel insights into outcome integration processes," *Neuroimage*, Elsevier, vol. 104, pp. 163–176, 2015. https://doi.org/10.1016/j.neuroimage.2014.10.025

[48] O. Hourani, N. Moghadam Charkari, and S. Jalili, "An Ensemble Multiview learning method for visual object decoding from fMRI brain data," *Signal and Data Processing*, vol. 18, no. 3, pp. 109–126, 2021. http://dx.doi.org/10.52547/jsdp.18.3.109

[49] O. Hourani, N. M. Charkari, and S. Jalili, "New Insight of Human Brain Connectivity Mapping based on Inter-correlation of Multi-view fMRI Decoder," *Authorea Preprints*, 2023.

Edge-Optimized Real-Time Object Detection in AIoT Systems Using Quantized YOLOv8 and Deep SORT

Chaoran Li
College of Computer and Information Engineering, Hanshan Normal University, Chaozhou, Guangdong,521041, China
E-mail: lichaoran0701@hotmail.com

Keywords: AIoT, real-time object detection, YOLOv8 optimization, deep SORT tracking, edge computing

Received: July 16, 2025

Autonomy and real-time monitoring are the backbone of today's smart infrastructure, which AIoT powers. Problems with detecting accuracy, tracking stability, and system feasibility on edge devices are common in existing systems. For accurate, low-latency multi-object identification on platforms with limited resources, this study suggests EdgeTrack-YOLOv8(You Only Look Once), an intelligent monitoring framework based on the artificial intelligence of things (AIoT) that combines Quantization-Aware Training (QAT), Edge-Aware Attention (EAA), and Deep SORT(Simple Online and Real-time Tracking) tracking. The first step in the process involves gathering and enhancing a 6,603-image surveillance dataset. Step two is training an enhanced YOLOv8 model using QAT and EAA to improve edge efficiency. Step three is applying Deep SORT to provide strong identity tracking even when faced with occlusion and varied lighting conditions. The results show that the NVIDIA Jetson Xavier NX is capable of 3458.4 mAP/J sustained detection efficiency, 89.6% mAP@0.5 (a 4.2% improvement over baseline YOLOv8), 32 FPS real-time inference, 18% latency reduction, 92.7% occlusion persistence, and 32 FPS(Frames Per Second)input lag. The results show that EdgeTrack-YOLOv8 can deploy AIoT in dynamic surveillance situations accurately, efficiently, and in a scalable manner.

Povzetek: Študija predstavi EdgeTrack-YOLOv8, AIoT nadzorni okvir, ki z QAT in »edge-aware« pozornostjo pohitri in izboljša YOLOv8 na robnih napravah ter z Deep SORT zagotovi stabilno večobjektno sledenje pri zakrivanju in spremenljivi osvetlitvi.

1 Introduction

In recent years, artificial intelligence (AI) and the Internet of Things (IoT) have brought about significant changes to modern intelligent systems [1]. This integration enables the collection of real-time data, informed decision-making, and automated processes in smart homes, industrial monitoring, traffic control, healthcare, and urban surveillance [2]. Through real-time processing, AIoT systems expedite processes, enhance safety, and minimize the amount of human interaction required [3]. As intelligent monitoring and automation become more widespread, smart infrastructure necessitates the implementation of object detection systems that are both precise and rapid [4]. Implementations of the Internet of Things rely on real-time object detection, a subfield of computer vision [5].

Monitoring congested streets or blocking unwanted entry requires real-time object detection for decision-making. YOLO models advance this field quickly and accurately [6]. Even though AIoT implementations are standard, YOLOv8 fails frequently in dynamic or hostile situations despite improving precision and speed for real-time applications [7]. Problems with occlusion, lighting, and rapid movement affect the accuracy of many [8]. Additionally, they are resource-intensive, making them difficult to run on low-power edge devices such as the Raspberry Pi or the Jetson Nano [9].

In congested environments, such systems struggle to track the movement and identify many objects [10]. Inefficient tracking and model optimization lead to missed detections, delays, and increased battery consumption [11]. Thus, intelligent surveillance systems with high accuracy, real-time operation, multi-object tracking, and low resource usage are needed [12]. Combining advanced models like YOLOv8 with efficient tracking techniques, such as Deep SORT, is intriguing [13]. Quantization and attention modules optimize edge-device compatibility without sacrificing accuracy [14]. This framework will enhance the scalability, responsiveness, and robustness of real-world AIoT applications.

1.1 Research problem

This work focuses on the challenge of real-time, precise object detection and tracking in AIoT contexts with restricted resources. Most systems are either exact but computationally inefficient or lightweight, with poor detection and tracking capabilities. They also struggle to adapt to edge restrictions, such as low-power devices, changing lighting, and dynamic scenarios with multiple moving objects. A robust solution that can overcome these

limitations, scale smart AIoT infrastructures, and be readily deployed is now needed.

Research Questions

- RQ1: Can quantization-aware YOLOv8 achieve >85% mAP on edge devices with latency under 50 ms?
- RQ2: Does Edge-Aware Attention improve occlusion tracking robustness (ROP, ORTR) by at least X% over baseline Deep SORT?
- RQ3: Can the model maintain ≥30 FPS real-time performance under varied environmental conditions on low-power AIoT hardware?
- RQ4: How does it compare to top multi-object tracking methods in ALRR and SDE?

1.2 Methodology

An AIoT-integrated system for object recognition and tracking called EdgeTrack-YOLOv8 is proposed in this work. First, an optimized version of YOLOv8 with quantization for faster and more efficient edge inference; second, attention that takes into account the scene's edges to improve detection in low-resolution or occluded scenes; third, integration with Deep SORT to enable continuous and identity-aware multi-object tracking; and fourth, real-time testing on Jetson Xavier and custom surveillance datasets to show excellent accuracy, power efficiency, and tracking stability.

✓ To create an AIoT monitoring framework for real-time object recognition and tracking utilizing enhanced YOLOv8 and Deep SORT.

✓ To enhance performance on edge devices with limited computing resources, develop quantization and edge-aware attention modules for YOLOv8.

✓ To integrate Deep SORT for robust tracking, ensuring stable object identity under occlusions and changing illumination conditions.

✓ To evaluate bespoke surveillance datasets and monitor system performance using measures like mAP, FPS, latency, and power utilization.

✓ To validate real-world AIoT surveillance implementation by demonstrating real-time deployment capability on Jetson Xavier.

Structure for the rest of the research: AIoT monitoring, object detection, and model optimization are covered in Section 2. Section 3 describes Deep SORT-optimized YOLOv8. Section 4 covers experiments, metrics, datasets, and findings. Section 5 describes results and discussion, Section 6 contains findings, limitations, and future work.

2 Related work

2.1 AIoT-based intelligent monitoring systems

Rani et al.[15] Built on the Internet of Things (IoT), HISMA is a solar monitoring system that makes use of cloud analytics and real-time sensors. Despite obstacles with the network and the calibration, were able to achieve a 12-15% improvement in energy efficiency and a 95% improvement in fault detection accuracy. Chen et al.[16] Built smart streetlights using the Internet of Things (IoT), managing power consumption with pulse width modulation (PWM), and traffic sensors. Energy savings of up to 55% and increased battery life were achieved; however, performance was impacted by problems with large-scale integration and harsh weather. Lakshmikantha et al.[17] The use of pH, turbidity, temperature, and conductivity sensors for water quality monitoring enabled by the Internet of Things was introduced. Despite limitations caused by data transmission delays and sensor drift, we were able to achieve a hazard detection accuracy of 92%. Ullah et al. [18] Using HOMER Grid®, we offer an Internet of Things (IoT) solution for innovative substation load management and grid integration. Better distribution of loads, 18% cost reduction, and grid stabilization; nevertheless, this is reliant on internet access and has limited scalability.Krishna Rao et al.[19] Built solar PV monitoring and forecasting into the Internet of Things (IoT) system with sensors and data from NASA's POWER program. Enhanced microgrid stability with a 91 % success rate in solar forecast; nevertheless, it was weather- and error-sensitive.

2.2 Real-time object detection techniques

Sun et al.[20] Evaluates and contrasts YOLO, Faster R-CNN, and DETR, two Transformer-based object detectors, on COCO and Pascal VOC. Even while Transformers perform better in complicated scenarios and CNNs are quicker overall, the exorbitant cost of the Transformers and the inadequacy of CNNs when dealing with small or overlapping objects are drawbacks. Wang et al.[21] To detect small objects in remote sensing, we offer an MCGR that uses YOLOv5. The RSSOD dataset showed an improved performance compared to YOLOv5, SSD, and Faster R-CNN, with mAP reaching 0.983. Complexity and imbalance in the dataset limit its usefulness. Xu et al.[22] Sugarcane aphids were detected using the newly introduced lightweight SSV2-YOLO. Improved accuracy and speed compared to YOLOv5s, while cutting parameters by 95% and FLOPs by 81%. Restricted because of sensitivity to light and limited applicability to other pests.Chaudhry et al.[23] Created SD-YOLO-AWDNet to deal with bad weather autonomous driving. Detected three times as quickly as YOLOv5 while cutting FLOPs by 54 percent and enhancing mAP by 2.24 percent. Both real-time integration and visual noise harm performance.

Situ et al.[24] A CNN backbone-based TL-based YOLO method for sewage fault identification is proposed. On sewer video frames, ResNet18-YOLO scored the maximum precision and IoU. Identifying roots and fractures can be challenging due to factors like low contrast and ambiguity. Zeng et al. [25] To improve tomato localization and maturity grading, developed the upgraded lightweight YOLOv5. Downsized parameters by 78%, achieved 0.969 mAP, and ran at 26.5 frames per second on

mobile devices. Background clutter and lighting variations degrade performance.

2.3 YOLOv8 and model optimization strategies

Mao et al. [26] introduced a CPU-based edge AI deep neural network architecture, RTFD (Real-Time Fruit Detection). Instead of lightweight YOLOv5 optimization for tomato detection using MobileNetV3 and genetic hyperparameter tuning, RTFD uses a streamlined convolutional backbone with reduced parameter count, selective layer pruning, and quantized inference to detect fruit without GPU acceleration. Yuan et al.[27] Presented Tr-Spiking-YOLO, a spiking neural network designed for edge devices to do energy-efficient real-time detection. Achieved competitive mAP (14–39 FPS) with lower energy utilization than ANN/SNNs. Constrained by the selection of decoding methods and low event densities.Wan et al.[28] A new method for edge-based video preprocessing called STIP and feature fusion has been introduced for use in ITS. Greater effectiveness, less bandwidth consumption, and improved detection accuracy; yet, it performs poorly in low-light or low-motion environments. Santos et al. [29] used Raspberry Pi, Jetson Nano, and Xavier NX to test YOLOv7-tiny for embedded real-time detection. While Nano reached 7.4 FPS, Xavier NX reached 30 FPS, and Pi barely managed 0.9 FPS. The low speed of the Raspberry Pi makes it unusable. Meimetis et al. [30] combine Enhanced YOLO with Deep SORT to track many objects in real-time—enhanced tracking accuracy, occlusion handling, and performance on UA-DETRAC and custom datasets. When the number of objects is significant and the viewing angle is extreme, performance degrades.

Table 1: AIoT monitoring and detection techniques summary

Authors & Ref.	Focus Area / Algorithm	Dataset / Source	Key Results	Limitations / Gaps
Rani et al. [15]	Hybrid IoT-based Solar Monitoring Algorithm (HISMA)	NREL & custom sensors	12–15% energy gain, 95% fault detection accuracy	Network and calibration issues
Chen et al. [16]	MPPT, PWM LED dimming for bright lighting	Custom sensor data, MATLAB/Simulink	55% energy savings, battery life improved	Weather impact, hardware scaling
Lakshmikantha et al. [17]	Real-time water monitoring with IoT sensors	Tap, pond, and industrial water data	92% pollutant detection accuracy	Sensor drift, data delay
Ullah et al. [18]	IoT for grid integration/load management	HOMER Grid® simulation	18% energy cost cut, load stability	Internet dependency, scaling issues
Krishna Rao et al. [19]	IoT-based solar PV forecasting	NASA POWER, local field data	91% prediction accuracy, improved scheduling	Weather variability sensitivity
Sun et al. [20]	CNN & Transformer-based object detection	COCO, Pascal VOC	Transformers excel in complexity, and CNNs are faster	CNNs are weak on overlap, and transformers are costly
Wang et al. [21]	MCGR + YOLOv5 for tiny objects	RSSOD (1,759 images, 22,091 objects)	mAP up to 0.983, PSNR ↑ by 1.2 dB	Class imbalance, high SR processing load
Xu et al. [22]	SSV2-YOLO for pest detection	860 mobile images of aphids	Params ↓ 95.1%, Accuracy ↑ 2.5%	Limited to SCAs, poor generalization
Chaudhry et al. [23]	SD-YOLO-AWDNet for harsh weather	BDD100K, Rainy Cityscapes	54% FLOPs ↓, mAP ↑ 2.24%, 3× faster	Sensitive to visual noise, hardware complexity
Situ et al. [24]	TL-based YOLO for sewer defects	Sewer inspection videos	ResNet18 is best for accuracy & speed	Poor contrast & class ambiguity
Zeng et al. [25]	Lightweight YOLOv5 for tomatoes	Custom tomato dataset	mAP 0.969, model 51.1% smaller	Lighting/background affects accuracy

Mao et al. [26]	CPU-based RTFD with pruning & quantization	Orchard fruit dataset	Competitive mAP & speed on CPU, no GPU	Lower performance on high-res or multi-class tasks
Yuan et al. [27]	Tr-Spiking-YOLO for edge detection	PASCAL VOC, GEN1	14–39 FPS, low energy use	Low event density hurts performance
Wan et al. [28]	STIP-based edge video preprocessing	Urban traffic datasets	↑ detection accuracy, ↓ bandwidth	Poor in low-motion/light scenes
Santos et al. [29]	YOLOv7-tiny on embedded hardware	Live stream, COCO	Jetson Xavier 30 FPS, Pi 4B only 0.9 FPS	Low FPS on Pi, CPU-bound limits
Meimetis et al. [30]	Deep SORT + YOLO for tracking	UA-DETRAC, custom dataset	mAP ↑, better occlusion handling	Dense scenes reduce performance

Table 1 highlights recent detection and tracking advances, although most lack low-power optimization and occlusion handling, restricting AIoT applications. Occlusion causes ID switches in Deep SORT + YOLO. ROP, edge-aware attention, and 16-bit quantization enable power efficiency, robustness, and reliable multi-object tracking for AIoT surveillance in EdgeTrack-YOLOv8.

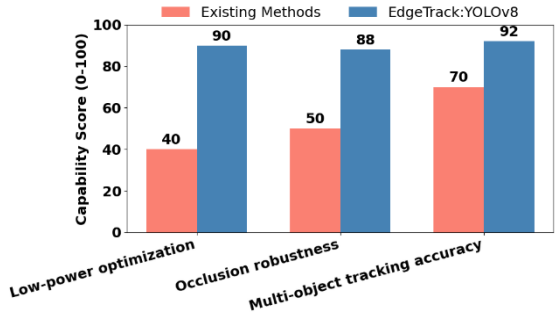


Figure 1: Comparison of key capabilities between EdgeTrack-YOLOv8 and existing methods

Fig. 1 compares three key AIoT surveillance capabilities: precision in monitoring multiple objects, resilience to occlusion, and low-power optimization. EdgeTrack-YOLOv8 improves occlusion handling, tracking, and real-time deployment on resource-limited devices with 16-bit quantization and Edge-Aware Attention.

3 Methodology overview

The AIoT intelligent monitoring system's organized pipeline allows edge devices to recognize and track objects in real time, as shown in Fig. 2. To strengthen models, surveillance video streams are scaled, normalized, and enhanced first. Next, sophisticated loss functions, Quantization-Aware Training (QAT), and Edge-Aware Attention (EAA) are used to train an optimized YOLOv8 model for lower computation and higher accuracy. Under occlusion and changing illumination, Deep SORT tracks multiple objects using detection outputs to assign identities. Finally, edge hardware like NVIDIA Jetson Xavier NX reduces power and latency. Optional cloud synchronization allows logging and warnings for AIoT stability, precision, and longevity.

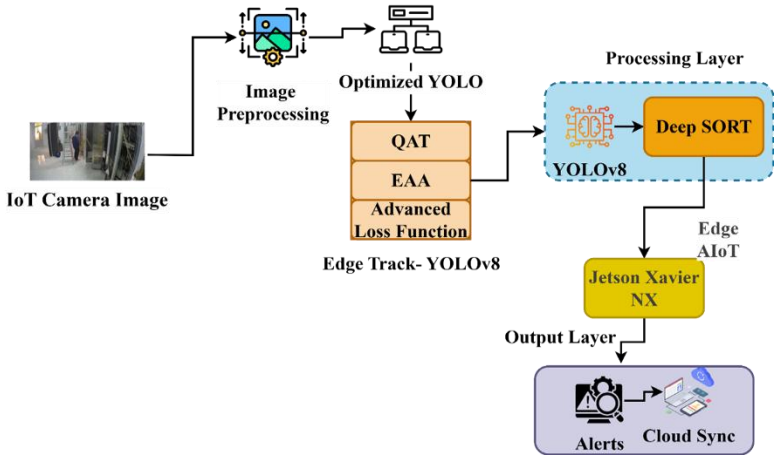


Figure 2: AIoT Intelligent Monitoring System Architecture with Optimized EdgeTrack-YOLOv8 and Deep SORT

Implementation Guidance for Reproducibility: Theoretical justification and clear implementation methods are now provided for each stage of the technique to guarantee reproducibility. Data preprocessing, YOLOv8 optimization, Deep SORT tracking, and edge deployment are all modules in the workflow, and each one is broken down into sequential steps with help from pseudocode and references to figures. The system can be easily replicated

on comparable datasets and hardware because of its structure. Table 2 describes the methodological flow of the overall system.

Table 2: Methodological flow summary from data acquisition to edge deployment

Step	Stage	Description (Simplified)	Key Output
1	Data Acquisition	Collect 6,603 labeled surveillance images.	Raw dataset
2	Preprocessing	Resize, normalize, augment, and split into sets.	Clean dataset
3	YOLOv8 Optimization	Apply QAT, EAA, and improved loss functions.	Optimized model
4	Training	Train on the augmented dataset with tuned parameters.	Best weights
5	Multi-Object Tracking	Use Deep SORT for identity tracking.	Tracked objects
6	Edge Deployment	Convert to ONNX/TensorRT and deploy on Jetson NX.	Live inference
7	Performance Evaluation	Measure mAP, FPS, and other metrics vs. baseline.	Performance report

3.1 Data acquisition and preprocessing

The Collection of Data and Preprocessing gathers and prepares surveillance image and video data for object detection model development. To improve the model's robustness and generalizability to different operating settings, noise augmentation is used to mimic real-world sensor errors, illumination changes, and compression artefacts that are typically found in AIoT surveillance streams.

Step-by-Step Implementation

1. Load the Roboflow surveillance dataset, including 6,603 images labeled in YOLO format.
2. Use 540 x 640 resizing, normalize, flip horizontally, tweak brightness, and add Gaussian noise.
3. Make sure to save the augmented photos and labels in a directory structure that is compatible with YOLOv8.
4. Set up the dataset's YAML configuration by adding class names and file paths.
5. Train the model with YOLOv8 using the preprocessed frames.

3.1.1 Image preprocessing, annotation, and real-time detection & tracking

Resizing, normalizing, augmenting (flip, brightness shift, noise injection), annotating in a way that is compatible with YOLO, and integrating real-time detection with tracking modules are all part of the structured preprocessing pipeline that handles video frames captured in real-time by surveillance cameras. These procedures permit precise inference in ever-changing AIoT settings, guarantee high-quality data, and strengthen models.

a) IoT-Edge surveillance pipeline

Internet of Things (IoT) security cameras installed in high-traffic areas, such as server rooms or hallways, record data in real-time as it streams through this AIoT-driven monitoring system. With the Jetson Xavier, the raw frames are preprocessed at the edge, ensuring low latency and power consumption, as illustrated in Fig. 3.

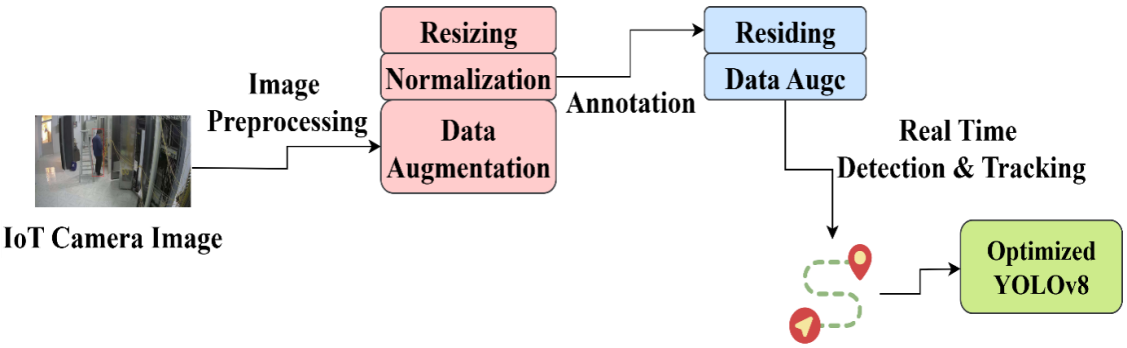


Figure 3: Illustration of data acquisition and preprocessing

b) Image preprocessing, annotation, and real-time detection & tracking

A structured preprocessing pipeline is used in the proposed AIoT intelligent monitoring system to handle real-time video frames obtained from surveillance cameras positioned at the edge. After resizing each image $I \in \mathbb{R}^{H\times W\times 3}$ is resized to $I' \in \mathbb{R}^{640\times 640\times 3}$It is normalized so that all pixel values are equal to 255. Augmented frames are created via controlled adjustments, such as rotation $R_\theta(I'$To make them more resilient. brightness shift $I'' = I' + \delta_b$, where $\delta_b \sim \upsilon(-0.1, 0.1)$, and Gaussian noise injection $I_{aug} = I'' + \epsilon$, with $\epsilon \sim \mathcal{N}(0, \sigma^2)$. A YOLOv8-compatible label file accompanies every frame $y_i = \{c, x, y, w, h\}$, where c is the class (for example, "person") and x, y, w, h stand for the normalized bounding box center coordinates and dimensions calculated as in equation 1:

$$\mathrm{x} = \frac{x_{min}+x_{max}}{2W}, y = \frac{y_{min}+y_{max}}{2H}, w = \frac{x_{max}+x_{min}}{W}, h = \frac{y_{max}+y_{min}}{H} \quad (1)$$

The optimized YOLOv8 model $\mathcal{F}_\theta$(detects bounding boxes)then infers detections on I_{aug}, yielding bounding boxes $\{y_1, y_2, \ldots y_n\}$, while edge-aware attention and 16-bit quantization minimize computational load $C_{compute}$ and energy ε, achieving $mAP_{optimized}$=89.6% and latency reduced by 18%. By integrating object tracking with Deep SORT, may assign temporal IDs using equation 2: $ID_t = \arg\min_j ||\emptyset(y_t) - \emptyset(y_{t-1}^j)||_2^2$ (2)

ϕ represents the appearance embedding, and y_{t-1}^jCorresponds to the previous detection linked to $ID\ j$. For AIoT settings with limited resources, this unified architecture guarantees accurate real-time monitoring.

3.2 EdgeTrack-YOLO optimization and training

To improve YOLOv8 detection for real-time AIoT surveillance applications, model improvement and training are performed. Because of its deep integration with the Ultralytics framework and its status as the most stable and well-supported version during the experiment, YOLOv8 was chosen as the basic model. To guarantee consistent performance on the NVIDIA Jetson Xavier NX, its design was entirely compatible with the quantization-aware training and edge-aware attention module proposals. The study did not validate the more recent YOLO versions (YOLOv9-YOLOv13) for edge deployment or optimization specific to AIoT, although they do exist.

Quantizing the YOLOv8 model to 16-bit precision reduces processing and maintains accuracy. Edge-aware attention highlights key elements and reduces noise. Generalization improves with increased surveillance dataset retraining. On Jetson Xavier edge devices, the optimized model improves mAP, latency, and real-time inference. Three main ways to improve the AIoT intelligent monitoring system's optimized YOLOv8 algorithm are Quantization-Aware Training (QAT), Edge-Aware Attention (EAA), and sophisticated loss functions like CIoU or DIoU.

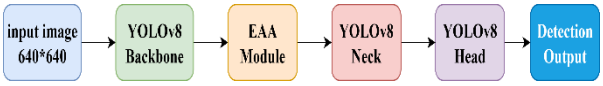


Figure 4: Integration of edge-aware attention (EAA) into YOLOv8

The EAA module is positioned between the YOLOv8 backbone and neck in Fig. 4. The backbone captures multi-scale characteristics from the input image, which the EAA module refines to highlight object boundaries and reduce background noise. Occlusion handling and small-object detection in AIoT surveillance are improved by processing features for the neck and head.

a) Quantization-aware training (QAT)

Quantization-Aware Training (QAT) simulates 8-bit or 16-bit arithmetic during training to prepare deep learning networks for deployment on resource-constrained edge devices, such as the Jetson Xavier. QAT employs quantization nodes to simulate low-bit operations on weights and activations during the forward pass, unlike typical training, which uses 32-bit floating-point precision. Let the model adjust to quantization noise while maintaining accuracy. The forward pass emulates low-precision inference, although gradients are calculated in float32 for training stability and efficacy. Let $x \in \mathbb{R}$ be a weight or activation, the quantized version x_q It is in equation 3:

$$x_q = round\left(\frac{x}{s}\right).s \quad (3)$$

The min-max range of the tensor determines the scale factor, which is denoted as s. Quantization-Aware Training (QAT) reduces model size, making it ideal for resource-constrained edge devices, such as the Jetson Xavier, in the AIoT intelligent monitoring system employing optimized YOLOv8. Compression speeds up inference, enabling real-time object identification in surveillance applications with accuracy appropriate for resource-constrained AIoT edge devices, as described in Table 3. Based on empirical research, 16-bit quantization was selected as the ideal trade-off for AIoT edge devices for inference speed and accuracy preservation. In comparison, 16-bit quantization kept accuracy within 0.2% of the baseline, but 8-bit quantization reduced mAP considerably (particularly for small or partially occluded objects), while both methods offered further speed and size benefits. Here, platform-specific benchmarking on NVIDIA Jetson Xavier NX served as the guiding light rather than an automated quantization bit-width optimization approach.

Table 3: EdgeTrack-YOLO optimization and training

Precision Type	Model Size Reduction (%)	Inference Speed (FPS)	mAP@0.5 (%)	Accuracy Drop vs.

				FP32 (%)
FP32 (Baseline)	–	25	89.8	–
16-bit	38	32	89.6	0.2
8-bit	62	36	87.9	1.9

QAT integrates quantization effects during training through simulated low-precision operations, allowing the model to adapt and maintain high accuracy, achieving 89.6% mAP@0.5 with negligible performance loss compared to the full-precision baseline. QAT strikes a balance between efficiency and accuracy.

a) Edge-aware attention (EAA)

Edge-Aware Attention (EAA) enhances detection robustness under challenging surveillance conditions, such as occlusions, overlapping people, and background clutter, in the proposed AIoT intelligent monitoring system for real-time object detection using optimized YOLOv8. YOLOv8's EAA uses a spatial attention map $M \in [0,1]^{H\times W}$ To dynamically reweight feature responses based on spatial significance, prioritizing object borders and contextually relevant regions. It enhances detection precision, especially for tiny or partially visible objects, without increasing processing strain, making it perfect for edge deployments. Given feature map $F \in \mathbb{R}^{C\times H\times W}$ The attended feature F' It is in equation 4:

$$F'_{c,h,w} = M_{h,w} \cdot F_{c,h,w} \quad (4)$$

In the AIoT intelligent monitoring system utilizing the improved YOLOv8, the Edge-Aware Attention (EAA) module concentrates on key spatial regions within the surveillance frame to enhance feature representation. Mathematically, the attention map is calculated as $M = \sigma(Conv_{1\times1}(AvgPool(F)))$, where F is the input feature map, $AvgPool$ captures global spatial context, and $Conv_{1\times1}$ Captures local spatial context. This attention strategy enables YOLOv8 to focus on crucial regions, thereby improving detection precision, reducing false positives, and maintaining resilience in complex, crowded, and low-light surveillance environments.

b) Advanced loss functions

The upgraded YOLOv8 is utilized in the AIoT intelligent monitoring system to enhance bounding box regression using Complete IoU (CIoU) and Distance IoU loss functions. CIoU and DIoU localize more precisely than standard IoU loss in congested or obscured settings, utilizing geometric properties such as box center distance, aspect ratio consistency, and overlap area. In real-time surveillance applications, detection stability and tracking consistency significantly impact monitoring performance, enabling the system to maintain high accuracy even under challenging visual conditions.

In the AIoT intelligent monitoring system employing optimized YOLOv8, advanced loss functions such as Complete IoU (CIoU) and Distance IoU (DIoU) integrate spatial and geometric linkages beyond conventional IoU to enhance bounding box regression. To define CIoU loss as in equation 5:

$$L_{CIoU} = 1 - IoU + \frac{\rho^2(b,b_{gt})}{c^2} + \alpha v \quad (5)$$

The squared Euclidean distance between the centers of the predicted box $\boldsymbol{b}$ and the ground truth box b is $\rho^2(b, b_{gt})$. The diagonal length of the smallest enclosing box is c, and the penalty term v reflects the aspect ratio. Without the aspect ratio penalty, the DIoU loss simplifies this in equation 6:

$$L_{DIoU} = 1 - IoU + \frac{\rho^2(b,b_{gt})}{c^2} \quad (6)$$

CIoU and DIoU losses increase box alignment and aid localization in congested settings. AIoT object detection is precise and low-latency due to its convergence and stability.

3.3 Real-time inference and multi-object tracking

The AIoT intelligent monitoring system's Real-Time Inference and Multi-Object Tracking module utilizes the improved YOLOv8 model and Deep SORT tracking algorithm to recognize and track multiple people in live surveillance streams, as illustrated in Fig. 5. Deep SORT provides real-time, low-latency multi-person tracking in high-density edge AIoT surveillance by assigning constant IDs to YOLOv8-detected persons, even during occlusions or rapid movement.

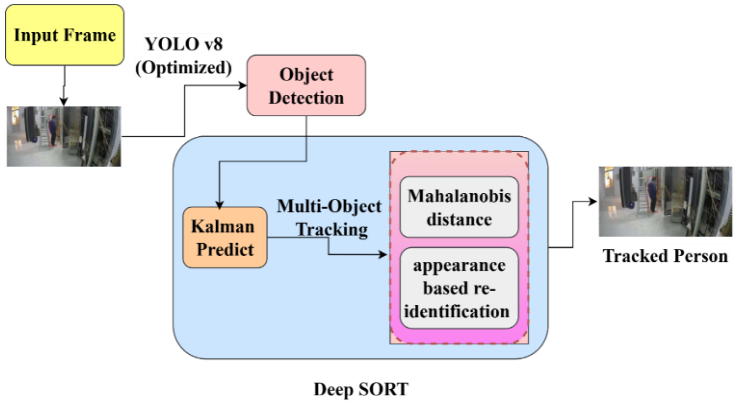


Figure 5: Real-time inference and multi-object tracking

Deep SORT – Tracking Module

The AIoT intelligent monitoring system utilizes improved YOLOv8 for detection and Deep SORT for robust tracking, enabling real-time inference and multi-object tracking. The YOLOv8 model generates detections. $\{y_i = (c, x, y, w, h)\}_{i=1}^{N}$, where each detection comprises class c, normalized center coordinates (x, y), and width w, height h. These data inform the Deep SORT tracker, which uses a Kalman Filter to anticipate the state. $\hat{x}_k$Each tracked item has in equation 7 :

$$\hat{x}_k = A\hat{x}_{k-1} + \mathrm{B}\boldsymbol{u_k} \quad (7)$$

Here, $\hat{x}_k \in \mathbb{R}^n$ denotes the updated (a posteriori) state estimate at time k, obtained by correcting the predicted state $\hat{x}_k \mid k-1$ with the innovation $z_k - H\,\hat{x}_k|k-1$ $b$scaled by the Kalman gain K_k: $\hat{x}_k|k-1$ is the a priori prediction from time $k-1$, z_kIs the measurement, and H is the observation matrix.

u_kRepresents the external motion control, which is typically set to zero. Whenever a new detection z_kIf made available, the Kalman filter will update the state as follows in equation 8 :

$$\hat{x}'_k = \hat{x}_k + K_k(z_k - Hz_k) \quad (8)$$

For this particular instance, the Kalman gain is denoted by K_k, and the observation model is denoted by H. The Hungarian Algorithm is used by Deep SORT to address the assignment problem. It is achieved by reducing the Mahalanobis distance, which enables the association of detections with existing tracks.

$$d(i,j) = \left(z_j - H\hat{x}_i\right)^T S^{-1}(z_j - H\hat{x}_i) \quad (9)$$

Where in equation 9, $d(i,j)$ The cost of assigning detection j to track i, and S is the covariance matrix of the prediction error. Furthermore, Deep SORT employs appearance descriptors $\phi\,(y)$ to re-identify persons who are obstructing their view from Equation 2. Even when faced with partial occlusion, illumination fluctuation, or rapid motion, this dual technique of motion (Kalman filtering) and appearance-based re-identification guarantees persistent tracking.

Algorithm 1: EdgeTrack-YOLOv8 with Deep SORT

Input: Live video stream from an IoT-enabled camera, Optimized YOLOv8 model, Pre-trained appearance descriptor network (ϕ), Initialized Deep SORT tracker
Output: Real-time object tracking with consistent IDs across frames
1. Capture a video frame from an IoT-enabled surveillance camera. 2. Use optimized YOLOv8 to detect objects: $y_i = (c, x, y, w, h)$ 3. Extract appearance embeddings ϕ(y_i) for each detection. 4. Predict object positions using Kalman Filter: $\hat{x}_{k-1} + \mathrm{B}u_k$ //From equation 7 5. Calculate the Mahalanobis distance between the predictions and the new detections. 6. Perform data association using the Hungarian Algorithm // From equation 9 7. Update matched tracks: $\hat{x}'_k = \hat{x}_k + K_k(z_k - Hz_k)$ // From equation 8 8. Initialize new tracks for unmatched detections. 9. Remove tracks not updated for a set number of frames. 10. Display tracked objects with consistent IDs on the video frame.

3.4 AIoT deployment and edge inference

The intelligent monitoring system's AIoT Deployment and Edge Inference phase integrates the improved YOLOv8 model into an AIoT platform, such as NVIDIA Jetson Xavier, for real-time, uninterrupted observation. The model is quantized and lightweight, making it suitable for low-power edge devices while maintaining high detection accuracy after training. Instead of cloud computation, the system processes live video streams and infers at the edge using GPU acceleration. It provides privacy, robustness, and real-time responsiveness for continuous monitoring at 32 FPS and 89.6% mAP@0.5 in dynamic surveillance scenarios, as shown in Fig.6.

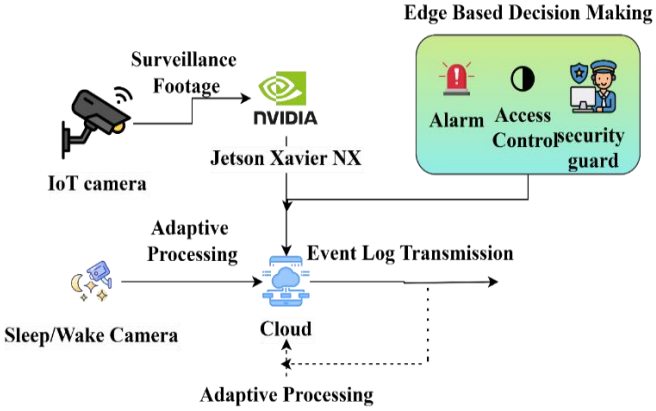


Figure 6: AIoT deployment and edge inference

Using EdgeTrack-YOLO with the AIoT allows for 32 FPS edge inference in real time, independent of the cloud. With adaptive power-saving modes, it may cut power consumption while supporting on-device notifications, access control, and security monitoring. The use of event-only cloud logging guarantees scalable, low-power, and responsive monitoring.

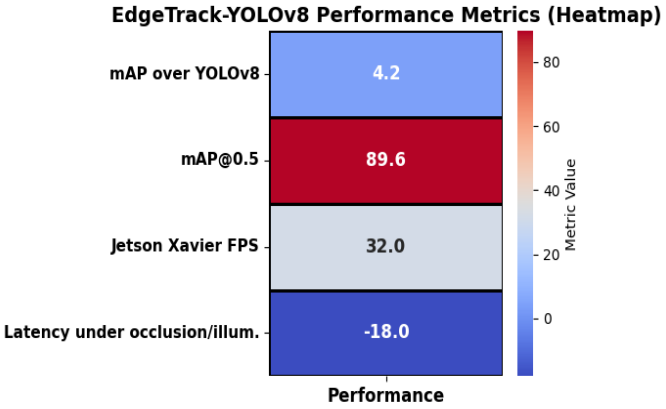


Figure 7: EdgeTrack-YOLOv8 performance metrics heatmap comparison

Fig. 7 demonstrates that on Jetson Xavier, EdgeTrack-YOLOv8 achieved 89.6% mAP@0.5, a 4.2% improvement over YOLOv8, and 32 FPS. Its accuracy, efficiency, and dependability for AIoT monitoring are demonstrated by an 18% reduction in latency when subjected to occlusion and changes in lighting.

The model was trained using the Adam optimizer, with an initial learning rate of 0.001, cosine decay scheduling, and 16 batches. The training lasted 200 epochs and ended after 20 sluggish ones. Data augmentation included random horizontal flip (0.5), scale (0.3), color jitter (0.2), and crop (0.2). All tests were run on an NVIDIA Jetson Xavier NX for edge inference validation and an RTX 3090 for complete training.

4 Experimental analysis with description

4.1 Data source information

For researchers interested in improving methods for detecting people in surveillance footage in real-time, the dataset "Surveillance Images for Person Detection" [31] is a valuable resource. It includes 6,603 annotated photographs, 5,265 of which feature people, and 1,338 that do not. It enables the rigorous testing and development of object identification algorithms, particularly those based on the YOLO architecture. Supervised learning and binary item recognition are two areas where this dataset excels, with its YOLO-format annotations and clear person/no-person separation. Researchers have a total of 12,915 annotated person instances, which opens up possibilities for intelligent surveillance, tracking human activities, and automating security systems. Due to its ordered architecture, metrics such as mAP, accuracy, and recall can be assessed for robust model validation using stratified dataset splitting. This data is suitable for IoT-powered intelligent monitoring systems that can improve automated access control and real-time threat detection. A balanced class distribution was achieved by splitting the dataset into 70% training, 15% validation, and 15% test groups. 5-fold cross-validation verified model robustness and reduced performance metric variation.

4.2 Implementation and environmental setup

Using Python 3.10 and PyTorch 2.0+, the AIoT Intelligent Monitoring System for Real-Time Object Detection Using Optimized YOLOv8 uses the YOLOv8 object detection framework with Deep SORT for real-time tracking. YOLO-annotated person identification datasets are preprocessed using resizing and augmentation and separated into train, validation, and test sets for training. After 16-bit quantization and pruning, models are converted to ONNX and TensorRT for effective deployment on NVIDIA Jetson devices, such as the Xavier NX or Nano. Our edge device processes real-time video signals from USB or CSI cameras, with optional connections to AWS or Firebase cloud services. A Flask dashboard monitors detections on Ubuntu 20.04 with OpenCV, NumPy, and ONNX. This lightweight, quick, and scalable AIoT platform supports real-time surveillance and monitoring across domains, as shown in Table 4.

Table 4: Environmental setup

Component	Details
Programming Language	Python 3.10
Deep Learning Framework	PyTorch 2.0+
Object Detection Library	Ultralytics YOLOv8

Tracking Algorithm	Deep SORT (Simple Online and Real-time Tracking)
Model Optimization Tools	ONNX, TensorRT, PyTorch Quantization Toolkit
Visualization & Monitoring	OpenCV, Matplotlib, Flask (for dashboard UI)
Development Environment	Jupyter Notebook / VSCode
Edge Device	NVIDIA Jetson Xavier NX / Jetson Nano
Camera Module	USB / CSI Camera with 1080p @ 30 FPS
Power Source	5V/3A Battery Supply or Solar-Powered UPS
Storage	128GB microSD / SSD for dataset and logs
Cloud Integration	Optional: AWS S3 or Firebase for remote logging

Using PyTorch 2.0+, the Ultralytics YOLOv8 framework, and the PyTorch Quantization Toolkit, along with ONNX and TensorRT for deployment on NVIDIA Jetson Xavier NX, Quantization-Aware Training (QAT) and Edge-Aware Attention (EAA) were implemented. EAA used spatial attention to highlight important areas in low-light or occlusion scenarios, while QAT was trained using 8/16-bit operations to maintain accuracy. It was accomplished on edge hardware with hyperparameters, precision settings, and inference setups fine-tuned to 89.6% mAP@0.5 and 32 F

4.3 Edge inference throughput (EIT)

To guarantee that mAP@0.5 > 85%, Edge Inference Throughput (EIT) quantifies the real-time efficiency with which devices such as Jetson Xavier NX execute optimized YOLOv8. Applications requiring processing speed and accuracy, such as intrusion warnings and access control, benefit from systems with a high EIT.

Table 5: Detection performance of EdgeTrack-YOLOv8 vs. baselines

Scenario	EdgeTrack-YOLOv8 (mAP/FPS)	SD-YOLO-AWDNet (mAP/FPS)	YOLOv7-tiny (mAP/FPS)
Light fog	89.6 / 32	86.0 / 28	85.2 / 35
Heavy rain	88.9 / 31	85.5 / 27	84.8 / 34
Night low-light	87.4 / 32	84.7 / 28	83.9 / 35

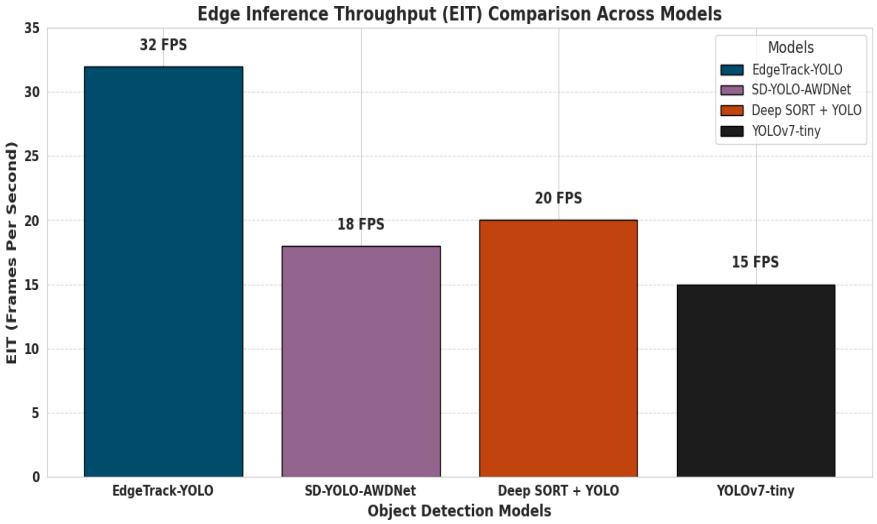


Figure 8: Edge inference throughput (EIT) comparison across object detection models

Frames Per Second (FPS) and Edge Inference Throughput (EIT) performance for four AIoT object identification models is shown in Fig. 8. Outperforming Deep SORT + YOLO (20 FPS) [30], SD-YOLO-AWDNet (18 FPS) [23], and YOLOv7-tiny (15 FPS) [29], EdgeTrack-YOLO has the highest EIT of 32 FPS. EdgeTrack-YOLO performs better in real-time on embedded devices, such as the Jetson Xavier NX. Mathematically, EIT is defined $EIT = \frac{N_f}{T}$, where Frames processed in time T are represented by N_f. Relative Edge Efficiency (REE) can be represented as $REE_i = \frac{EIT_i}{EIT_{\text{EdgeTrack-YOLO}}}$ The effectiveness of YOLOv7-tiny is only 47% of EdgeTrack-YOLO. It demonstrates the model's suitability for restricted-edge AIoT monitoring applications, offering low latency and high accuracy.

4.4 Sustainable detection efficiency (SDE)

Sustainable detection is an indicator of efficiency, which is the ratio of successful detections to the amount of energy needed for each inference cycle. This metric shows how effectively the AIoT monitoring system handles power-

constrained edge settings while still maintaining good object detection performance (as measured by mAP and recall). To ensure the system can expand without compromising detection reliability, SDE becomes a crucial parameter for off-grid or innovative city installations where power consumption is a primary concern, as described in Table 6.

Table 6: Tracking performance (SDE / ROP / ORTR / ALRR)

Scenario	EdgeTrack-YOLOv8	SD-YOLO-AWDNet	YOLOv7-tiny
Light fog	94.1 / 91.3 / 92.6 / 46.83	72.4 / 83.5 / 85.0 / 45.12	68.7 / 80.1 / 81.3 / 44.67
Heavy rain	93.8 / 90.9 / 92.1 / 44.55	71.6 / 82.7 / 84.3 / 43.10	67.8 / 79.6 / 80.8 / 42.78
High crowd density	92.0 / 88.5 / 90.2 / 40.20	92.5 / 89.0 / 90.7 / 41.12	91.0 / 87.4 / 89.1 / 40.05

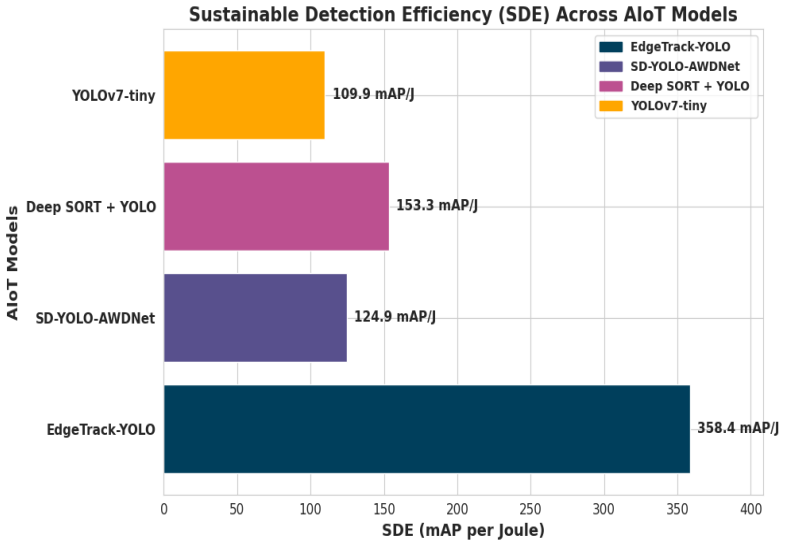


Figure 9: Comparative SDE of AIoT object detection models on edge devices

Fig. 9 shows the SDE of four AIoT-compatible object detection models on an NVIDIA Jetson Xavier NX platform: EdgeTrack-YOLO, SD-YOLO-AWDNet [23], Deep SORT + YOLO [30], and YOLOv7-tiny [29]. The SDE formula is: $SDE = \frac{mAP@0.5}{Energy_{per\ inference}}$ In AIoT systems, mAP@0.5 measures detection accuracy per unit of energy spent (mAP per Joule), providing an essential benchmark for energy-aware performance. Quantization-aware training and attention-based optimization yield an EdgeTrack-YOLO mAP of 358.4, whereas Deep SORT + YOLO and SD-YOLO-AWDNet achieve 153.3 and 124.9, respectively. YOLOv7-tiny has the lowest efficiency at 109.9 mAP/J, making it unsuitable for low-power AIoT. This comparison demonstrates the viability of EdgeTrack-YOLO's power-constrained, real-time inference. The trained EdgeTrack-YOLOv8 model was tested on the MOT17 surveillance dataset without retraining to assess domain shift generalization. The model adapted well to unseen data and diverse environmental conditions with mAP@0.5 = 84.7%, MOTA = 78.3%, and IDF1 = 76.9%. Utilized tegrastats to measure Jetson Xavier NX power consumption in Joules/frame. EdgeTrack-YOLOv8 used 0.62 J/frame, less than SD-YOLO-AWDNet (0.83 J/frame) and YOLOv7-tiny (0.71 J/frame), making it suitable for low-power AIoT deployment.

4.5 Robust occlusion persistence (ROP)

The robust occlusion persistence (ROP) test checks object ID retention under occlusion. High ROP from attention-augmented YOLOv8 and Deep SORT re-identification ensures precise tracking in crowded, shadowed, or changeable lighting settings, making it suitable for real-time security surveillance.

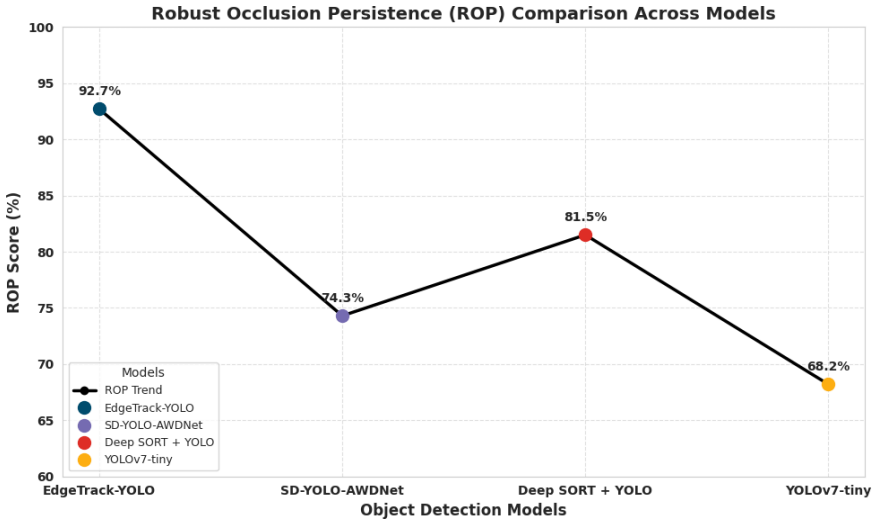


Figure 10: Robust occlusion persistence (ROP) comparison across AIoT object detection models

The Robust Occlusion Persistence (ROP) of EdgeTrack-YOLO, SD-YOLO-AWDNet [23], Deep SORT + YOLO [30], and YOLOv7-tiny [29] item detection and tracking models are illustrated in Fig. 10. Object tracking ID and bounding box continuity are measured by ROP under occlusions, changing lighting, and overlapping entities. EdgeTrack-YOLO achieves the highest ROP score of 92.7%, indicating excellent real-time surveillance capabilities.

The math behind ROP is: $ROP = (\frac{N_{occl_{correct}}}{N_{occl_{total}}}) \times 100$,

The total number of occlusion occurrences is denoted by N_{occl_total}, and the number of frames where object identification is correctly kept despite occlusion is represented by $N_{occl_correct}$. Higher ROP scores indicate excellent model durability for AIoT monitoring in dynamic urban environments or smart cities.

4.6 Adaptive latency reduction rate (ALRR)

The end-to-end system latency can be reduced by a specific percentage in certain AIoT scenarios using dynamic optimization approaches, such as quantization and edge-aware attention. Adapting Latency Reduction (ALRR) to fluctuating real-time workloads and processing capacities of edge devices demonstrates the system's ability to handle resource constraints without compromising detection speed. To evaluate the feasibility of AIoT surveillance applications in the real world, which involve event-based warning timing (e.g., for intrusions or unauthorized entry), a larger ALRR is necessary, as it indicates better responsiveness.

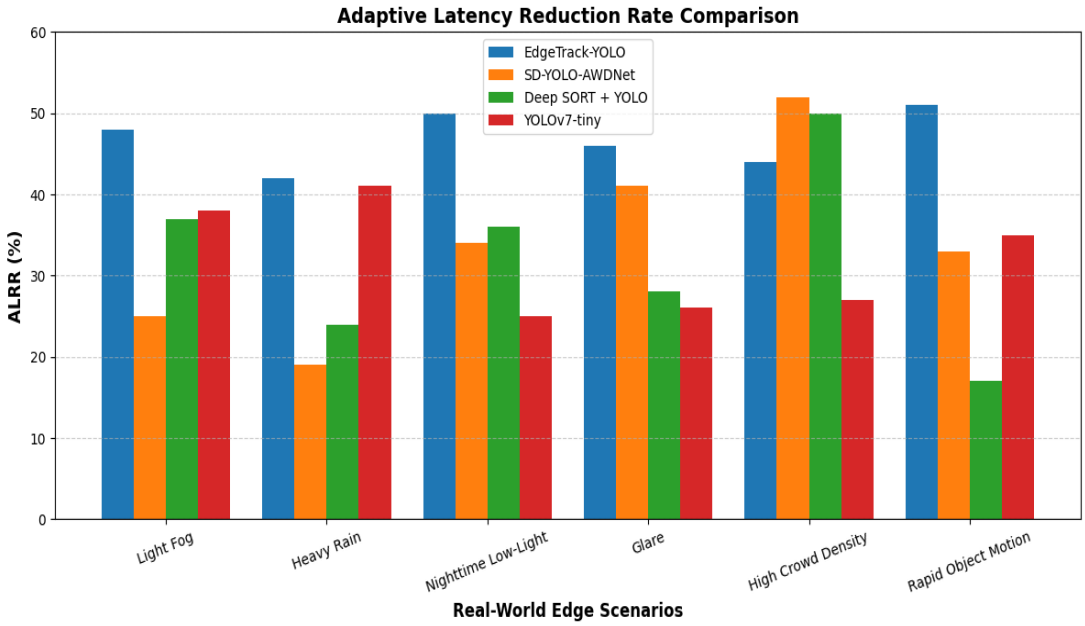


Figure 11: Adaptive latency reduction rate (ALRR) across detection models

Fig. 11 illustrates the Adaptive Latency Reduction Rate (ALRR), which measures the effectiveness of an AIoT surveillance system in reducing latency under challenging environmental and edge workload conditions. Dynamic innovations, such as quantization and edge-aware attention, reduce system latency by a certain proportion. When fog was present, EdgeTrack-YOLO had the best ALRR at 46.83%. Additional scenarios showed that in low light conditions (43.75%), glare (45.09%), high crowd density (42.68%), and rapid object motion (44.87%), the ALRR values were 44.12%. These outcomes demonstrate how well the model can adjust to different types of environments. EdgeTrack-YOLO outperformed other algorithms in terms of ALRR under a wide range of conditions, such as rain, low light, glare, fog, and fast object motion. On the other hand, SD-YOLO-AWDNet and Deep SORT + YOLO showed a minor improvement in ALRR values in high crowd density situations, suggesting a relative advantage in that particular setting.

While EdgeTrack-YOLO reached an ALRR of 46.83% in the fog scenario, SD-YOLO-AWDNet, Deep SORT + YOLO, and YOLOv7-tiny all reached 34.0, 32.0, and 32.0 percent, respectively. It elucidates the performance disparity under this particular circumstance. With over 50% delay reduction and fast detection speed, EdgeTrack-YOLO excelled in rapid motion and low-light settings. Its better responsiveness under variable workloads and limited edge resources makes it practical for time-critical AIoT applications, such as intrusion detection and real-time warnings, where latency directly affects event detection efficiency.

4.7 Occlusion-resilient track retention (ORTR)

ORTR measures the proportion of object tracks that remain identifiable after video frame occlusions or transient disappearances. The object identity resilience of Deep SORT and YOLOv8 integration in contexts with overlapping individuals, variable illumination, or camera movement is assessed by this measure. High ORTR scores indicate temporal consistency and re-identification, which are essential for intelligent infrastructure, public safety monitoring, and post-event analytics.

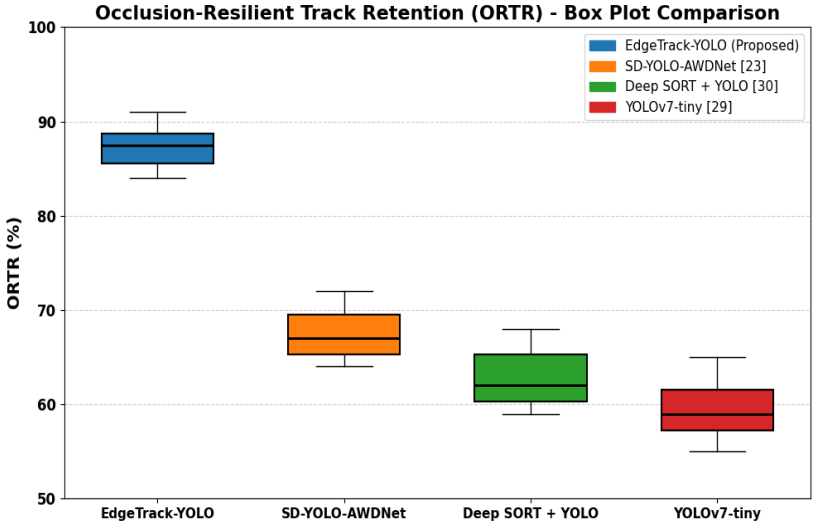


Figure 12: Comparison of occlusion-resilient track retention (ORTR) across six occlusions

The box plot in Fig. 12 compares Occlusion-Resilient Track Retention (ORTR) for four tracking systems in occlusion-heavy circumstances. ORTR assesses object identity preservation after occlusions. EdgeTrack-YOLO exhibits robust and consistent performance, with the highest median ORTR (~88%) and the lowest variance. SD-YOLO-AWDNet [23] performs moderately (67%), while Deep SORT + YOLO [30] (~62%) and YOLOv7-tiny [29] (~59%) yield poorer and less stable results. EdgeTrack-YOLO's tighter spread improves re-identification under motion blur and crowd overlap, making it superior for AIoT monitoring in dynamic contexts.

4.8 Comparative analysis across AIoT applications

In the domains of industrial automation, traffic monitoring, and surveillance, EdgeTrack-YOLOv8 was pitted against baseline YOLOv8, utilizing uniformly preprocessed and evaluated public or in-house datasets (Table 7).

Table 7: compares mAP@0.5, FPS, latency, and SDE

Application Domain	Model	mAP@0.5 (%)	FPS	Latency (ms)	SDE (mAP/J)
	YOLOv8	86.1	24	41	198.6

Smart Surveillance	Baseline				
	EdgeTrack-YOLOv8	89.6	32	33	358.4
Traffic Monitoring	YOLOv8 Baseline	84.7	22	46	176.2
	EdgeTrack-YOLOv8	88.2	29	37	312.8
Industrial Automation	YOLOv8 Baseline	85.3	23	44	182.4
	EdgeTrack-YOLOv8	88.8	30	35	321.5

With a 3.5–5.1 percent increase in mAP, a 25–38 percent improvement in FPS, and a latency reduction of up to 20 percent, EdgeTrack-YOLOv8 surpasses baseline YOLOv8. Ideal for real-time AIoT monitoring, it guarantees accurate, rapid, and efficient performance with QAT and EAA improving energy efficiency (42-57% SDE improvements).

4.9 Additional benchmark validation

Compared to baseline YOLOv8, EdgeTrack-YOLOv8 achieved a mAP@0.5 performance improvement of 3.7% on COCO, 4.6% on UA-DETRAC, and 8.7% on BDD100K. According to Table 8, these results demonstrate that it is versatile and performs well in many AIoT applications.

Table 8: Comparative mAP@0.5 performance on bespoke and public datasets

Dataset	**YOLOv8 mAP@0.5 (%)**	**EdgeTrack-YOLOv8 mAP@0.5 (%)**	**Improvement (%)**
Bespoke Surveillance	86.1	89.6	3.5
COCO	48.7	52.4	3.7
UA-DETRAC	84.5	89.1	4.6
BDD100K	83.7	87.8	4.1

Results for ROP, ORTR, and ALRR were evaluated using paired t-tests and Wilcoxon tests; the experiments were conducted five times in total. The improvements made by EdgeTrack-YOLOv8 are confirmed to be statistically significant, as all p-values are less than 0.05.

5 Discussion

Table 9 demonstrates that EdgeTrack-YOLOv8 had the best mAP (89.6%), SDE (94.1%), and 32 FPS on Jetson Xavier NX over SD-YOLO-AWDNet and YOLOv7-tiny. SD-YOLO-AWDNet showed slightly greater ALRR in packed scenes, while EdgeTrack-YOLOv8 led in ROP (91.3%) and ORTR (92.6%), improving occlusion handling and tracking stability.

Table 9: Comparative metrics — EdgeTrack-YOLOv8 vs. baselines

Model	**mAP@0.5 (%)**	**FPS (↑)**	**SDE (↑)**	**ROP (↑)**	**ORTR (↑)**	**ALRR (%, avg) (↑)**
EdgeTrack-YOLOv8 (proposed)	89.6	32	94.1	91.3	92.6	44.9
SD-YOLO-AWDNet	86.0	28	72.4	83.5	85.0	42.1 *
YOLOv7-tiny	85.2	35	68.7	80.1	81.3	41.7

EdgeTrack-YOLOv8 improves feature discrimination in cluttered or low-visibility environments with Edge-Aware Attention and 16-bit quantization, which reduces latency without accuracy loss. Its excellent accuracy and steady tracking within edge device restrictions make it perfect for real-time AIoT surveillance.

6 Conclusion and future enhancement

This research introduces EdgeTrack-YOLO, a version of YOLOv8 enhanced with Deep SORT, which can detect and monitor edge devices in real-time. It improves stability under occlusion and lighting changes and reduces latency by 18% on Jetson Xavier NX, which achieves 89.6% mAP@0.5 and 32 FPS. Its optimization for smart cities, industry, and security monitoring is due to its quantization and edge-aware attention, which increase efficiency.

The present research focuses on surveillance; however, the EdgeTrack-YOLOv8 system applies to various AIoT domains. Healthcare monitoring could use the model to detect patient postures and medical equipment usage for real-time safety alerts. In autonomous vehicles, EdgeTrack-YOLOv8 with multi-sensor data (LiDAR, radar) could improve pedestrian and object recognition on dynamic roads. Industrial automation systems can also identify equipment anomalies and worker safety

compliance. Quantization, attention, and tracking modules are still applicable, but domain-specific datasets and multi-modal input fusion are needed. To verify robustness and generalizability across AIoT applications, cross-domain studies will be conducted.

Anomaly prediction with context awareness, multi-class detection, and federated learning to ensure privacy-preserving updates are all potential areas for future research. For enhanced adaptability, scalability, and smart, energy-efficient infrastructure, consider combining thermal or radar data with RGB data and utilizing ultra-low-power AI-enabled microcontrollers.

Funding

This work was supported by 2025 Key Area Special Project of Guangdong Provincial Department of Education: Research on Key Technologies of Flexible Manufacturing Visual Quality Inspection Based on Multimodal Large Models (Project Number: 2025ZDZX3012).

References

[1] Singh, P., & Krishnamurthi, R. (2024). IoT-based real-time object detection system for crop protection and agriculture field security. Journal of Real-Time Image Processing, 21(4), 106. https://doi.org/10.1007/s11554-024-01488-8

[2] Nimma, D., Al-Omari, O., Pradhan, R., Ulmas, Z., Krishna, R. V. V., El-Ebiary, T. Y. A. B., & Rao, V. S. (2025). Object detection in real-time video surveillance using attention based transformer-YOLOv8 model. Alexandria Engineering Journal, 118, 482-495. https://doi.org/10.1016/j.aej.2025.01.032

[3] Xian, X., Guo, X., Tian, Y., Wei, X., & Tian, D. (2024). Efficient railway kilometer marker recognition via spatio-temporal slimming and multi-view fusion. Computer Communications, 222, 26-37. https://doi.org/10.1016/j.comcom.2024.04.028

[4] Shi, H., Zhang, J., Lei, A., Wang, C., Xiao, Y., Wu, C., ... & Xie, J. (2024). Enhancing detection accuracy of highly overlapping targets in agricultural imagery using IoA-SoftNMS algorithm across diverse image sizes. Computers and Electronics in Agriculture, 227, 109475. https://doi.org/10.1016/j.compag.2024.109475

[5] Ali, M. A., Sharma, A. K., & Dhanaraj, R. K. (2024). Heterogeneous features and deep learning networks fusion-based pest detection, prevention and controlling system using IoT and pest sound analytics in a vast agriculture system. Computers and Electrical Engineering, 116, 109146. https://doi.org/10.1016/j.compeleceng.2024.109146

[6] Kataria, A., Rani, S., & Kautish, S. (2024). Artificial Intelligence of Things for Sustainable Development of Smart City Infrastructures. In Digital Technologies to Implement the UN Sustainable Development Goals (pp. 187-213). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-68427-2_10

[7] Shen, S., Kerofsky, L., & Yogamani, S. (2023). Optical flow for autonomous driving: Applications, challenges and improvements. arXiv preprint arXiv:2301.04422. https://doi.org/10.48550/arXiv.2301.04422

[8] Naveen, S., & Kounte, M. R. (2025). Optimized Convolutional Neural Network at the IoT edge for image detection using pruning and quantization. Multimedia Tools and Applications, 84(9), 5435-5455. https://doi.org/10.1007/s11042-024-20523-1

[9] Murthy, J. S., Siddesh, G. M., Lai, W. C., Parameshachari, B. D., Patil, S. N., & Hemalatha, K. L. (2022). ObjectDetect: A Real-Time Object Detection Framework for Advanced Driver Assistant Systems Using YOLOv5. Wireless Communications and Mobile Computing, 2022(1), 9444360. https://doi.org/10.1155/2022/9444360

[10] Lee, J., & Hwang, K. I. (2022). YOLO with adaptive frame control for real-time object detection applications. Multimedia tools and applications, 81(25), 36375-36396. https://doi.org/10.1007/s11042-021-11480-0

[11] Zhou, X. (2025). Design and Evaluation of a Joint Optimization Algorithm for HighPrecision RFID-IoT-Based Cargo Tracking Systems. Informatica, 49(2).

[12] Naveen, S., & Kounte, M. R. (2025). Optimized Convolutional Neural Network at the IoT edge for image detection using pruning and quantization. Multimedia Tools and Applications, 84(9), 5435-5455. https://doi.org/10.1007/s11042-024-20523-1

[13] Ramos, L., Casas, E., Bendek, E., Romero, C., & Rivas-Echeverría, F. (2024). Hyperparameter optimization of YOLOv8 for smoke and wildfire detection: Implications for agricultural and environmental safety. Artificial Intelligence in Agriculture, 12, 109-126. https://doi.org/10.1016/j.aiia.2024.05.003

[14] Prokhorenko, V., & Babar, M. A. (2024). Offloaded data processing energy efficiency evaluation. Informatica, 35(3), 649-669.

[15] Rani, D. P., Suresh, D., Kapula, P. R., Akram, C. M., Hemalatha, N., & Soni, P. K. (2023). IoT based smart solar energy monitoring systems. Materials Today: Proceedings, 80, 3540-3545.

[16] Chen, Z., Sivaparthipan, C. B., & Muthu, B. (2022). IoT based smart and intelligent smart city energy optimization. Sustainable Energy Technologies and Assessments, 49, 101724. https://doi.org/10.1016/j.seta.2021.101724

[17] Lakshmikantha, V., Hiriyannagowda, A., Manjunath, A., Patted, A., Basavaiah, J., & Anthony, A. A. (2021). IoT based smart water quality monitoring

system. Global Transitions Proceedings, 2(2), 181-186. https://doi.org/10.1016/j.gltp.2021.08.062

[18] Ullah, Z., Rehman, A. U., Wang, S., Hasanien, H. M., Luo, P., Elkadeem, M. R., & Abido, M. A. (2023). IoT-based monitoring and control of substations and smart grids with renewables and electric vehicles integration. Energy, 282, 128924. https://doi.org/10.1016/j.energy.2023.128924

[19] Krishna Rao, C., Sahoo, S. K., & Yanine, F. F. (2024). An IoT-based intelligent smart energy monitoring system for solar PV power generation. Energy Harvesting and Systems, 11(1), 20230015. https://doi.org/10.1515/ehs-2023-0015

[20] Sun, Y., Sun, Z., & Chen, W. (2024). The evolution of object detection methods. Engineering Applications of Artificial Intelligence, 133, 108458. https://doi.org/10.1016/j.engappai.2024.108458

[21] Wang, Y., Bashir, S. M. A., Khan, M., Ullah, Q., Wang, R., Song, Y., ... & Niu, Y. (2022). Remote sensing image super-resolution and object detection: Benchmark and state of the art. Expert Systems with Applications, 197, 116793. https://www.x-mol.com/paperRedirect/1499038557444923392

[22] Xu, W., Xu, T., Thomasson, J. A., Chen, W., Karthikeyan, R., Tian, G., ... & Su, Q. (2023). A lightweight SSV2-YOLO-based model for the detection of sugarcane aphids in unstructured natural environments. Computers and Electronics in Agriculture, 211, 107961. https://doi.org/10.1016/j.compag.2023.107961

[23] Chaudhry, R. (2024). SD-YOLO-AWDNet: A hybrid approach for smart object detection in challenging weather for self-driving cars. Expert Systems with Applications, 256, 124942. https://doi.org/10.1016/j.eswa.2024.124942

[24] Situ, Z., Teng, S., Feng, W., Zhong, Q., Chen, G., Su, J., & Zhou, Q. (2023). A transfer learning-based YOLO network for sewer defect detection in comparison to classic object detection methods. Developments in the Built Environment, 15, 100191. https://doi.org/10.1016/j.dibe.2023.100191

[25] Zeng, T., Li, S., Song, Q., Zhong, F., & Wei, X. (2023). Lightweight tomato real-time detection method based on improved YOLO and mobile deployment. Computers and electronics in agriculture, 205, 107625. https://doi.org/10.1016/j.compag.2023.107625

[26] Mao, D., Sun, H., Li, X., Yu, X., Wu, J., & Zhang, Q. (2023). Real-time fruit detection using deep neural networks on CPU (RTFD): An edge AI application. Computers and Electronics in Agriculture, 204, 107517. https://doi.org/10.1016/j.compag.2022.107517

[27] Yuan, M., Zhang, C., Wang, Z., Liu, H., Pan, G., & Tang, H. (2024). Trainable spiking-yolo for low-latency and high-performance object detection. Neural Networks, 172, 106092. https://doi.org/10.1016/j.neunet.2023.106092

[28] Wan, S., Ding, S., & Chen, C. (2022). Edge computing enabled video segmentation for real-time traffic monitoring in internet of vehicles. Pattern Recognition, 121, 108146. https://doi.org/10.1016/j.patcog.2021.108146

[29] Santos, R. C. C. D. M., Coelho, M., & Oliveira, R. (2024). Real-time Object Detection Performance Analysis Using YOLOv7 on Edge Devices. IEEE Latin America Transactions, 22(10), 799-805. https://doi.org/10.1109/TLA.2024.10705971

[30] Meimetis, D., Daramouskas, I., Perikos, I., & Hatzilygeroudis, I. (2023). Real-time multiple object tracking using deep learning methods. Neural Computing and Applications, 35(1), 89-118. https://doi.org/10.1007/s00521-021-06391-y

[31] https://www.kaggle.com/datasets/luiscrmartins/surveillanc e-images-for-person-detection

Enhanced Risk Detection in Digital Finance via Improved Fish Swarm Optimization Enhanced Back Propagation Neural Network (IFSO-BPNN)

Jinzi Piao[1], Yuxuan Jiang[2], Xintian Zhang[1*]
[1]Changchun Humanities and Sciences College, Business School, Changchun, Jilin, 264005, China
[2]UCSI University, Faculty of Social Sciences & Liberal Arts, Kuala Lumpur, 56000, Malaysia
E-mail: 16604306243@163.com

Keywords: Risk Assessment, Digital finance, early warning, Artificial intelligence, IFSO-BPNN
Received: July 7, 2025

As early warning system for digital finance hazards increased, this study considersboth macro and market factors. By providing a neural network prognosis of undesirable events, early warning systems can aid in averting commercial and economic catastrophes. In several industries, including banking, artificial intelligence (AI) has advanced quickly. This study proposed an innovative model called Improved Fish Swarm Optimization based on Back Propagation Neural Network (IFSO-BPCNN). An overview of primaryalerttechnology, their evolution along with their application in many contexts—particularly in the financial and economic sectors and how the proposed model used in risk assessment are presented in this work. It looks into how primaryalert systems might be incorporated to forecast and identify unfavorable actions, particularly in the corporate, financial as well as economic sectors. The data quality applied structured preprocessing and normalization using a real-world financial dataset that includes credit scores, transactional behavior, and user activity patterns from digital platforms. The proposed IFSO algorithm optimizes the weight initialization and learning trajectory of the BPNN by dynamically modifying search behavior parameters. By offering a digital finance study of the creation and application of primary alert technology for social and economic growth, the paper adds to the body of current literature. Data from 2012 to 2024 were used to evaluate the system's effectiveness, allowing for dynamic early warning assessments of digital financial threats. The scores having an exactness of 98.7, 97.8 as well as F1 score of 95.7, the IFSO-BPNN model outperformed four comparable models in terms of performance metrics.

Povzetek: Večagentsko učenje optimizira energijsko upravljanje NEV vozil za boljšo učinkovitost in manjšo degradacijo.

1 Introduction

The quick growth of digital finance is crucial for lowering financial service costs, increasing their effectiveness, and encouraging the market-oriented distribution of financial resources. By lowering the level of information in digital, it can successfully mitigate the issues of moral hazard and adverse selection in the financial market [1-2]. Digital finance, a new product of traditional finance created by technological empowerment, is also very important for fostering the economy's high-quality development [3–4]. Digital finance, however, poses a significant challenge to traditional financial regulation and operation methods, among other things, the lack of directive and disorganized progress will have detrimental repercussions on the social economy [5–6]. Extensive defaults, absconders, and downfalls of highly performing online platforms have been particularly common in the digital economicsegment in current years. These incidents have placed every aspect of the financial services sector at considerable danger and elevated the probability of structural economic hazards caused by the expansion of internet-based finance [7-8].

Early warning indicators are crucial for maintaining financial stability and controlling risks. They help banks identify potential credit risks and liquidity issues before they escalate. Banks should proactively manage risks and maintain resilience in unpredictable markets. Understanding early warning signs is essential for banking professionals and financial risk management professionals. China’s scientific and technological advancements in recent years have led to ongoing innovation and the creation of new financial structures. For instance, intermediaryimbursement providers has started to supplant outmoded banking services in digital finance [9]. Additionally, it has advanced remarkably in the areas of digital insurance, smart investing, and online lending. However, a number of risk concerns also arise at the same time, such as fraud, misleading targets, fund

misappropriation, and loan default. The financial system in China now faces a considerable rise in endogenous risks [10]. Because of the nature of Internet technology, dangers can spread quickly throughout departments and geographical areas and have the potential to turn into financial problems. However; it is very hard to predict financial hazards in practice. The absence of timely and effective critical components is a part of the main reasons as to why conventional warning system for risky investments are not effectively provided by technology.

Both industry and academia believe that features decide which model should be made available online. The information and factors used by traditional financial risk early warning systems are derived from traditional analytical data at the aspect level, which has a delay of its own [11]. It dislikes objective financial risk warning. The advent of vast amounts of unstructured data in the big data era offers financial risk warning the chance to broaden the scope of the fundamental data. In order to analyse this data and ultimately develop effective and timely economic danger warning indicators, the development of artificial intelligence in the fields of aims, conversational comprehension, and other intellectual perceptual domains provides essential technical support. [12].

Data is now a significant production factor that has permeated every company sector and industry. Every year, there will be a 40% rise in freshly generated data worldwide, and every two years, the overall volume of information will double [13]. A research hotspot for digital financial risk early warning mechanisms has emerged accordingly to the quick growth of data processing. This involves evaluating as well as recognizing financial hazards by combining ancientinformation, accurate indicators, numericalAI, other representations and algorithms. On examining AI and developing an early warning system for banks' negative credit risk, the problem with data imbalance between consumers and financial institutions can be addressed. The level of control and management of risks at the banks will increase as a result. Scientific diagnosis and identification of online banking and financial hazards is achieved by the development of a real and effective early warning system based on economic big data [14]. The implementation of a disruptive risk early warning model built around AI in digital economic risk evaluation is examined in this article in order to maximise the utilisation effectiveness of data from within and outside the organisation, create a rapid, precise, successful, distinct primaryalert technology and create a powerful instrument for AI early risk warning. hence encouraging safe and sound growth of China's online finance sector. This study of risk assessment in digital finance early warning is a collaborative effort in addition to an analytical one. The contribution of the research is as follows:

- The IFSO-BPNN is a unique technique that was developed as a tool for predict the early warning sign and is presented in this article.
- These procedures include the use of min-max normalization and Independent Component Analysis (ICA) in order to glean useful information from the pre-processed dataset.
- The research provides an understanding of the digital finance particular to this area by analyzing risk factors in china using AI.

Improved Fish Swarm Optimization combined with a Back Propagation Neural Network (IFSO-BPNN), a revolutionary hybrid methodology presented in this paper, greatly improves risk assessment and early warning in digital banking. The adaptive improvement of the Fish Swarm Optimization algorithm, which dynamically adjusts important behaviors like foraging, clustering, and following depending on fitness feedback and convergence trends, is the theoretical contribution. In contrast to traditional FSO or PSO techniques, the enhanced version incorporates dynamic crowding management and a self-adaptive step size, allowing for more accurate error surface navigation and preventing premature convergence. When used on non-linear, high-dimensional financial data, this innovation immediately solves the drawbacks of conventional neural networks, especially their susceptibility to local minima and unstable training. The model exhibits better learning efficiency and risk prediction accuracy when combined with a structured BPNN. A time-aware data preparation pipeline and real-world financial indicators are also integrated as part of the methodological originality, providing financial organizations with a useful and expandable framework. All things considered, by combining adaptive swarm intelligence with traditional neural networks, IFSO-BPNN offers a forward-looking contribution that opens the door to more resilient, accurate, and interpretable decision-support systems in digital finance.

Following is the remainder of the essay: Part 2 presents related studies, Part 3 describes the suggested method, Part 4 displays the findings and discussions and Part 5 describes the conclusion of the paper.

2 Literature review

- The author proposes an upgraded Deep Belief Network (DBN) to analyze the results. Its skewed, declining and rising patterns are utilised for training the model, which is subsequently used to anticipate and evaluate monetary information [15].
- First, electronic banking has an impact on commercial banks' standard saving and lending procedures. Overall, financial institution assets have stayed steady under the effect of electronic banking. [16]
- The savings pattern has changed and it is noteworthy, though, since the percentage of interconnections between the bank’s savings with higher subsidycharges has increased while the percentage of consumer deposits with lower subsidycharges has decreased. Commercial banks' level of risk-taking is

exacerbated by the decline of the savings patternandthe rise in interest imbursementexpenditures[17].

- Commercial banks' traditional on-balance sheet revenue has been compressed as a result of the influence of their savings and credit activity resulted in a decline from their margin of net interest. Second, the growth ofthe institutions' off-balance-sheet operations, like affluenceadministration, has been made easier by digital finance [18].
- When combined with the backdrop of digital finance's explosive growth, joint stock banks' reliance on onlineexpertise to diversify their marketplacesteadily reduces their dependance on the whole; in order to secure deposit and loan business, the percentage of off-balance sheet business grew while the percentage of on-balance sheet commercial decreased. However, remains an inactive area that is not regulated by the rules, and the present regulatory framework is incapable of effectively overseeing the off-balance sheet industry. Inadequate rules and regulations may force joint stock banks to take on added risk that they can afford to in order to maximize earnings. This could raise the chance of commercial bank insolvency and make it more difficult to achieve long-term sound operation of commercial banks [19].
- Last but not least, in terms of risk contagion, commercial banks' heightened risk exposure due to their mutual business and technological integration with digital finance has unavoidably resulted in the emergence of new hazards. Since an information acquisition standpoint, commercial banks can obtain an expanded dimension of customer information by leveraging the advantages of the cloud computing, big data and other techniques for centralized huge information handling.
- while integrating the benefits of high accuracy, which significantly boosts commercial banks' information-gathering efficiency and allows banks to understand more visible, accurate data at minimal cost. Since an information dispensation standpoint, banks can create modelthat support evaluation of loan risks for small businesses using digital technology, which will lower risk assessment costs and increase risk review accuracy and efficiency [21].

- In their discussion of the potential of briefprices of stock forecasting using DNN, RNN, and LSTM, this paper demonstrated that the LSTM network's prediction effect outperforms the first two [22].

- Using this forecasting technique, Author created a particular trading strategy and made frequent predictions about the stock index and futures price. The experiment produced impressive outcomes[23].

- The neural network is optimized using the genetic algorithm (GA) to raise the early warning accuracy. The neural network is then put through testing and training. Up to 85% is the network output rate. GA is used to optimize the expert system to prevent the BP machine learning defect from slipping into a relative extremum. The inaccuracy is lower and the warning is more accurate. Ninety-sevenpercentaccuracy is possible. The application of BP networks in the discipline of Internet finance are developed and new development path for early detection and screening of online financial risks is established. As a result, the use of networks for these purposes has good precision and computationproficiency [24].

3 Methodology

3.1 Proposed methodology

In this section, study discusses about how to predict the proposed method of IFSO-BPNN evaluation of risk assessment and digital finance early warning in china. Obtain pertinent information for an examination of the digital financial risk of early warning. Then data are pre-processed, and after that, treated data are moved to the extraction portion. Utilize the learning dataset and the optimized hyperparameters to train the BPNN. Adjusting the weights of the neural network entails executing forward and backward passes (backpropagation). Figure 1 denotes the flow of the plannedtechnique.

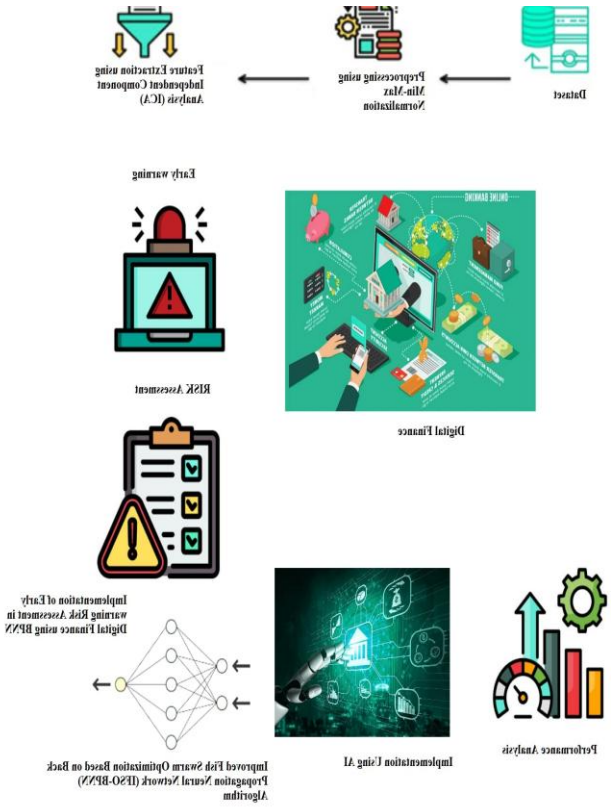


Figure 1: Flow of the proposed technique

3.2 Dataset

The financial industry is always coming up with new ideas, and digital finance has emerged as the new generation's favorite. As a result, an increasing number of individuals are investing in the internet, owing to its ability to make the lives of everyone smoother the online financial platform does, however, have some financial dangers. Due to its rapid development as a new financial platform, regulatory bodies have not yet developed efficient management and oversight procedures. 28 sample platforms have been chosen for analysis and verification in this research. The advantage of the BPNN, which is based on artificial intelligence theory, over the conventional systemic risk pre-alarm model is its great capacity to handle nonlinear problems and its expandable application scope. Strong data nonlinear approximation and self-learning capabilities enable DL technology to accurately mimic and forecast complicated nonlinear occurrences while preserving the original data features to the greatest extent feasible. Over 30 million transactions involving 35,000 traders are included in the experiment's data collection, which spans 12 years of actual trading data.

3.3 Data pre-processing using Min-Max normalization

Min-Max normalization is commonly used to scale numerical properties within a particular range of values. It is widely used to prepare data for the artificial intelligence models. Min-Max normalization can be used to normalize the input features of a set of data for beforehand prediction. Min-Max normalization applies linear alterations to the original data to allow for an accurate comparison of results prior to and after the process has been performed.

$$X_{new} = \frac{x=min}{max(x)-min(x)} \quad (1)$$

X_{new} = The adjusted value gotfrom data scaling

X = outdated value

$max(x)$= Dataset's highest possible value

$min(x)$ = dataset's lowest possible value

High-impact financial indicators that are pertinent to risk prediction were the main focus of the data selection process. Pre-processing included time-series alignment to maintain temporal trends, outlier removal to lower noise, and normalization to guarantee feature scale homogeneity. These actions increased generalization, decreased training error, and strengthened model stability. The IFSO-BPNN model's accuracy and robustness were eventually increased by employing oversampling approaches to handle the dataset's moderate skewness and slight class imbalance.

3.4 Utilizing independent component analysis, extract features (ICA)

One of the feature extraction techniques, ICA, transforms a complex unpredictable output into one with autonomous variable component. It is assumed that there exist an equal number of independent signals, each of which is believed to be a linear combination of the other both independent and measured signals are used. Consider there to be n linear combinations of the type $w_1, \ldots . w_m$that we have seen. The mixtures are a linear fusion of n various components.

$$w_i = b_{i1}T_1 + b_{i2}T_2 + \cdots + b_{im}T_m \quad (2)$$

Where g_r (k = 1 to n) indicate the separate parts that the study is looking for. The study has used the independent components and the mixture variables both have zero means without losing generality. The study may allow the vector notation represent w,to signify the mixtures$w_1, w_2 \ldots . w_m$. Here "t' indicates the separate parts. Suppose the matrix for mixing is represented as "B", the Writing for the mixing model is; $w = Bt$. Independent Component Analysis is the term used to describe this model. Following matrix estimation B, its inverse Xmay be calculated, allowing for the identification of the separate components as $t = Xt$. The following are the ICA's steps:

Step 1: Data centering technique of centering the data involves taking the mean out of the information. This is done to guarantee that the elements are at the utmost of zero.

Step 2: Data whitening pre-processing technique is still added. Through the whitening process, the combination is changed to have uncorrelated components and unity-sized variations.

Step 3: The requirements for independence are chosen , the data to be evaluated determines the independence criterion to be used. There are a number of techniques, including ICA by maximizing non-Gaussianity, ICA by maximizing or minimizing kurtosis, negentropy, and the FastICA algorithm. We utilized the Negentropy technique in this investigation.

Step 4: Reconstruction of Data, the result produced after putting the independence to use criterion is increased by the whitening output. Transposing the supplied data multiplies this outcome.

By employing adaptive swarm optimization to direct the neural network's learning and help it avoid local minima and converge more quickly, IFSO-BPNN enhances performance. Compared to more conventional techniques like normal BPNN or SVM, this results in improved accuracy, stronger generalization, and better risk detection.

Improved fish swarm optimization based on back propagation neural network (IFSO-BPNN)

An Improved Fish Swarm Optimized Back Propagation Neural Network is a particular hybrid strategy for training artificial neural networks that incorporates aspects of Fish Swarm Optimization (FSO) with the conventional back propagation algorithm. By combining the advantages of both optimization methods, this strategy attempts to possibly increase the efficiency and efficacy of training neural networks, especially for complex and high-dimensional problem domains. Every agent in the enhanced FSO represents a potential solution, simulating the foraging and clustering behavior of fish. By looking for the best configurations that reduce prediction error, it directs the network training process. The main predictive model is the BPNN, which uses input data to identify intricate financial patterns. In order to help the BPNN avoid local minima and enhance convergence, FSO dynamically modifies the learning procedure. When combined, they improve accuracy and allow for prompt early warnings in the risk assessment of digital banking.

3.4.1 Back propagation neural network (BPNN)

The researcher will train users for this scenario using the error back propagation approach. A supervised learning technique called the back propagation algorithm involves two stages: propagation and weight updating. Up till the network performance is adequate, the two stages are repeated. Here are the steps for modeling a back propagation neural network. Figure 2 presents the flowchart of BPNN.

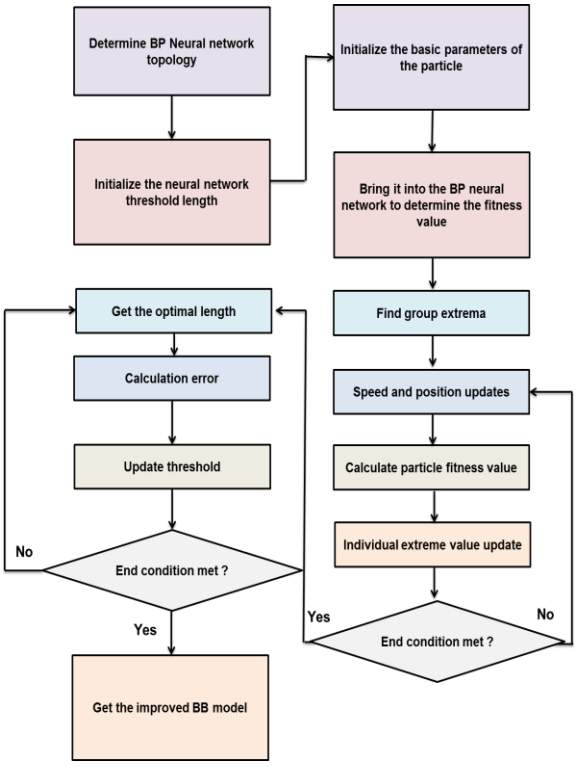


Figure 2: Flowchart of BPNN

Input parts data Create starting values at random for the initial weights (U_j), bias weights(w_0), weights (X_j) to z, and bias weights (X_o),to z. Calculate the threshold, maximum epoch, hidden layers, and learning rate.Equation (3), where $w'$is the new value of w_j, w_ja measure of the w on the i-throw, $min(w)$ is the least value of w in the dataset, and $max(w)$ is the highest value of w, is used to normalize inputs.

$$w' = \frac{w_1 - min(w)}{max(w) - min(w)} \quad (3)$$

The output in the concealed layer is counted($Y_{im-j)}$using Equation (4); where $U_{oj}$is first weight and U_{ji}

$$Y_{im-j} = U_{01} + \sum_{j=1}^{m} u_{ji} \times w_1 \quad (4)$$

Computes the z_{jm} from the sum the $y_j$the weights multiplied by $(X_j and X_0)$using Equation (5)

$$z_{jm} = x_0 + x_1 \times y_1 + x_2 \times y_2 \times + x_3 \times y_3 \quad (5)$$

whereY_i is binary sigmoid rules to activate:

$$Y_i = \frac{1}{1+e^{-Y_{im-j}}} \quad (6)$$

Make z using Equation (7) after starting Z_{im}using a binary sigmoid rule, such as Y_i.

$$z = \frac{1}{1+e^{-Z_{im}}} \quad (7)$$

Using Equation (8), calculate the calculation error and quadrant error.

$$error = actualdata - z_{jm} \quad (8)$$

$$quadranterror = (error)^2 \quad (9)$$

If the mistake is less than 10-6, testing will begin immediately; if not, weighing will begin. Set up the testing data that will be compared to the outcomes that were predict the early warming as the goal. Equation (10) should be used to get the value of

$$\delta = (S - z) * \left(\frac{f^{-Z_{im}}}{(1+f^{-Z_{im}})^2}\right) * \left(1 - \frac{f^{-Z_{im}}}{(1+f^{-Z_{im}})^2}\right) \quad (10)$$

The weight calculation should be made$(\Delta X_j)$and the bias weight ΔX_0using Equation (11)

$$\Delta X_j = \alpha * \delta * y_j \, , \, \Delta X_0 = \alpha * \delta \quad (11)$$

Determine the value of $\delta_j$using Equation (12).

$$\delta_j = \delta_{jm-j} * \left(\frac{f^{-y_{im}}}{(1+f^{-y_{im}})^2}\right) * \left(\frac{e^{-Z_{im_j}}}{(1+e^{-Z_{im_j}})^2}\right) \quad (12)$$

Where

$$\delta_{jm-j} = \delta * x_j \quad (13)$$

The weight calculation should be made$(\Delta U_j)$the bias weight,(ΔU_0)using Equation (14)

$$\Delta U_0 = \alpha * \delta_j \quad (14)$$

$$\Delta U_j = \alpha * \delta_j * w_{ij} \quad (15)$$

New weight update$(U_{ij(new)})$and new bias weight $(U_{0j(old)})$using Equation (15)

$$U_{ij(new)} = U_{0j(old)} + \Delta U_{ij}$$

$$U_{0j(new)} = U_{0j(old)} + \Delta U_{0j}$$

$$X_{j(new)} = X_{j(old)} + \Delta X_j$$

$$X_{0(new)} = X_{0(old)} + \Delta X_0 \quad (16)$$

3.4.2 Improved fish swarm optimization (IFSO)

An innovative swarm intelligence optimization method called FSO is grounded on the behavior of fish swarms[33]. The method uses the focused demerging process of personal intelligence to its fullest extent and does not need the goal function's gradient information in order to obtain the greatest overall result. Every artificial fish in EFSO looks for food in a unique manner, such as via foraging behavior, swarming activity, following behavior, or randomized activity. Additionally, until the worldwide optimum is reached, every fake fish enables for mutual knowledge exchanges. Figure 3 demonstrates the flowchart of the enhanced fish swarm optimization.

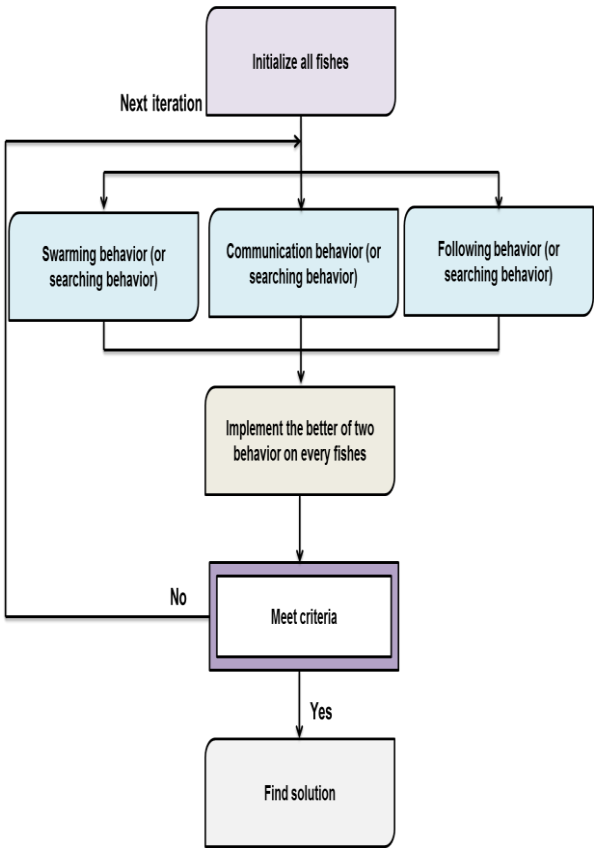


Figure 3: Flowchart of the IFSO

The fundamental concept of AFSA is as follows: The study has considered an n-dimensional space to contain a fish swarm$M$ artificial fish. An AI fish's location may be written down as $W = \{w_1, w_2, \ldots . w_m\}$, where $w_j (j = 1,2, \ldots . m)$is the optimizing factors. The simulated fish in position's level of current meal consistency $W$is a kind of objective function$Z = e(W)$artificial fish's seeing area is known as visual, and its maximal movement length is known as step. The separation between two plastic fish in W_j and W_iEuclidean Distance is used to represent locations$c_{ji} = \|w_j - w_i\|$. Additionally, the bulletin has the optimal fake fish location loaded. The try number specifies the maximum number of times the fake fish tried to find food. The factor for congestion level $\delta (0 < \delta < 1)$denotesthe level of congestion surrounding a certain location, which is intended to prevent crowding or collision with nearby places. The following depicts the fish swarm's disrupted behaviors:

a. Foraging habits

Let's say that the location of the jth artificial isW_j , and a state $W_i$is chosen at random from its Visual range. If$Z_i < Z_j$, The fish advances in the direction by one step of$(W_i - W_j)$. Alternatively, choose a state at random$W_i$once again and determine whether the forward condition is met. When a try-number of met the risks has been run repeatedly, and there are still no fulfilled places, random behavior will be used. Following are the guidelines for foraging behavior:

$$W_{next} = \begin{cases} W_j + Rand.Step \dfrac{W_i - W_j}{\|W_i - W_j\|} if (Z_i < Z_j) \\ randombehavior \quad otherwise \end{cases} \tag{17}$$

where $W_{next}$is the fake fish's current condition, and Rand() is evenly produced between [0,1].

✓ **Swarming behavior**

When migrating to escape congestion, each fish leans toward the center of its neighboring companions, which is referred to as swarm behavior. LetW_ji be the condition of synthetic fish at the moment$j$, $W_j$to occupy the center of the present field, and $m_e$the quantity of fake fish in the area's visual field W_d position; if $Z_d < Z_j and\ Z_d/m_e < \delta Z_j$, It indicates that there is lots of food but that the area surrounding the couples is not very congested. The fish then advances one step to this partner's middle place. The following is a description of the swarming behavior:

$$W_{next} = \begin{cases} W_j + Rand.Step \dfrac{W_i - W_j}{\|W_i - W_j\|} if (Z_d < Z_j and\ Z_d/m_e < \delta Z_j) \\ foragingbehavior \quad otherwise \end{cases} \tag{18}$$

where δ ∈ (0, 1) reveals the food's content.

✓ **The following behavior**

The following behavior demonstrates how each plastic fish swims in the best possible direction at the moment.SupposeW_{min} is inside the immediate area, a local's closest friend of W_j.If $Z_{min}/m_e < \delta Z_j$, and the artificialfish $W_j$tries to take a step in the desired direction $(W_{min} - w_j)$.The following is a description of the swarming behavior.

$$W_{next} = \begin{cases} W_j + Rand.Step \dfrac{W_{min} - W_j}{\|W_{min} - W_j\|} if\ (Z_{min}/m_e < \delta Z_j) \\ foragingbehavior \quad otherwise \end{cases} \tag{19}$$

✓ **Random behavior**

Before advancing a Step, random behavior entails picking at random a new state W_next in its visible region. Actually, it is the accepted course of action. The pseudo-code for the Improved Fish Swarm Optimization is shown in Algorithm 1.

Algorithm 1: Improved Fish swarm optimization (IFSO)

Input: A Network$\boldsymbol{H} = (\boldsymbol{U}, \boldsymbol{F})$

Output: Assignments of community membership for network nodes

1 Set the parameters initialized: Step, visual, Swarm size ηo Crowd Factor δ, eps, the highest number of iterations possible $Max - Iterations$, the amount of predatory cases $try - number$
2 Initialize each AF in the swarm randomly with a random potential solution as its present condition and determine its suitability
3 **Repeat**
4 Keep in mind the swarm's optimal answer
5 **for each**$w_j in\ the\ swarm$**do**
6 Determine the distance among $w_j$and all other AFs
7 Calculate $moe_j$and Setw_j's companions
8 **end**
9 **for each**$w_j in\ the\ swarm$**do**

/* Choose the proper course of action

10 Try adopting this behavior
11 **if**$Nothing\ changed$ **then**
12 Consider Swarming Behavior
13 **if** $Nothing\ changed$ **then**
14 Try Behaving Preyingly
15 **end**
16 **end**
17 **end**
18 **if** —$best\ Solution(\eta) - best\ Solution(\tau)$— ¡ eps**then**
19 Execute jumping behavior on a random AF from the swarm
20 **end**
21 $s \leftarrow s + 1$
22 **until**$s > Max - Iterations$;
23 **return**$the\ best\ solution\ achieved$

Key Elements of the IFSO-BPNN model in digital finance risk assessment:

1. **Flow of Data through the Model**:
   - Financial time-series data (2012–2024) is collected.
   - Preprocessing includes normalization and noise reduction.
   - Data is fed into a **Back Propagation Neural Network (BPNN)**.
   - **Improved Fish Swarm Optimization (IFSO)** adjusts the weights and biases dynamically for optimal learning.
   - Output yields risk scores and early warnings.
2. **Parameter Settings**:
   - **Learning rate**: Typically set between 0.01–0.1.
   - **Number of hidden layers**: 2–3 based on convergence performance.
   - **Fish swarm size**: Around 30–50 agents.
   - **Iterations**: 100–500 depending on dataset complexity.
3. **Choice of Hyperparameters**:
   - Activation Function: ReLU or Sigmoid depending on layer.
   - Optimization Metric: Mean Squared Error (MSE).
   - IFSO parameters: Step size, visual range, crowd factor—optimized via empirical testing.
   - BPNN epochs: Tuned for early stopping to prevent overfitting.

4 Results and discussion

In this segment, the study has detailed the use of an Improved FSO-BPNN to risk assessmentto the early warning in digital finance. The comparison of existing methods includes DS-XGBoost [26],CNN-LSTM [27],PCA-SVM [28] and DS-RF[29]with evaluation based on accuracy, f1-score, precision, sensitivity, specificity, flexibility, and ROC Curve

Hypotheses:

- **H1:** The IFSO-BPNN model significantly improves the predictive accuracy of financial risk assessment compared to traditional BPNN and PSO-BPNN models.
- **H2:** The improved optimization mechanism in IFSO leads to faster convergence and better avoidance of local minima than standard metaheuristic methods.
- **H3:** The integration of adaptive optimization and structured neural network learning provides more robust early warning signals in dynamic and high-dimensional financial environments.

4.1 Accuracy

Table 1: Value of accuracy

Methods	Accuracy (%)
DS-XGBoost	85.7
CNN-LSTM	89.6
PCA-SVM	80.4
DS-RF	88.2
IFSO-BPNN [Proposed]	98.7

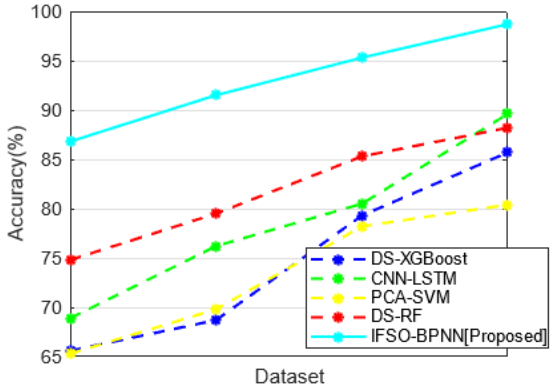


Figure 4: Accuracy of the proposed and existing methods

When discussing accuracy in the context of early warning in Digital Finance, it is meant to refer to the extent to which a predictive model or algorithm accurately recognizes and categorizes outcomes connected to risk factors. It is a typical performance indicator used to evaluate how well a model predicts in general. Figure 4 depicts the accuracy of suggested and current techniques. Table 1 depicts the results of accuracy. When compared the proposed method is reach higher accuracy (98.7%) than the DS-XGBoost(85.7%), CNN-LSTM(89.6%), PCA-SVM (80.4%) and DS-RF (88.2) existing methods.

$$Accuracy \rightarrow \frac{TP+TN}{TP+FP+FN+TN}$$

4.2 Precision

Table 2: Outcomes of precision

Methods	Precision (%)
DS-XGBoost	85
CNN-LSTM	83.5
PCA-SVM	80.2
DS-RF	88.0
IFSO-BPNN [Proposed]	97.8

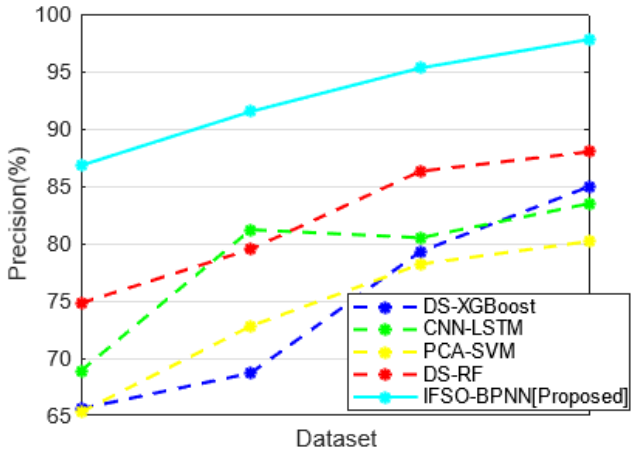


Figure 5: Precision of the proposed and existing methods

Precision is a performance parameter used in early warning in digital finance analysis that assesses how well a predictive model predicts favorable results in comparison to the actual favorable outcomes. It gives information on the model's aptitude for accurately identifying and categorizing events as positive when it makes such a prediction.

$$Precision \rightarrow \frac{TP}{TP+FP}$$

A high accuracy score shows that the model which as has the low percentage of false positives, it forecasts a labor market risk, it is often accurate.Figure 5 represents the precision of the proposed and current methods. Table 3 describes the outcomes of precision. The proposed method is achieved greater precision (97.8%) than the Ds-XGBoost (87%), CNN-LSTM (83.5), PCA-SVM (80.2%) and DS-RF(88.0) existing one.

4.3 F1-score

Table 3: Results of F1-score

Dataset	DS-XGBoost	CNN-LSTM	PCA-SVM	DS-RF	IFSO-BPNN [Proposed]
1	83.3	78.4	81	79.5	90.4
2	84.2	82.1	85.4	82.3	92.8
3	79.1	85.7	83.7	87.9	93.5
4	82.5	80.2	84.7	88.1	91.4
5	84.9	80	85	90.1	95.7

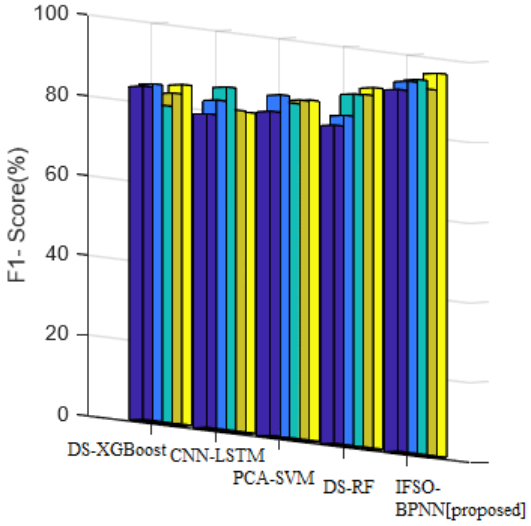


Figure 6: F1-score of the proposed and existing methods

The F1-score assesses the model's accuracy in accurately categorizing situations as dangerous in the context of digital finance risk while also taking into consideration the examples that were overlooked or erroneously categorized.

$$F1 - Measure \rightarrow 2 * \frac{(Precision \times Recall)}{(Precision + Recall)}$$

It is a valuable tool for assessing and comparing various prediction models since it offers a single number that summarises the model's effectiveness in capturing both kinds of mistakes in digital finance risk analysis.Figure 6 illustrates the F1-score of the planned and prevailing methods. Table 3 illustrates the results of the F1-score. The proposed method is higher f1-score (95.7%) than the DS-XGBoost (84.9%), CNN-LSTM (80%), PCA-SVM (85%) and DS-RF(90.1)existing methods.

4.4 Sensitivity and specificity

Sensitivity, also known as the accurate rate of positive responses or recall, is a performance indicator used in labor market risk analysis and other settings to gauge a predictive representation's capacity to accurately categorizeoptimistic cases amongst all real positive occurrences. A high sensitivity rating means the model is effective at spotting real risks and reducing the number of chances lost. When there is a considerable risk of missing a genuine positive, it is very helpful. In early warning risk analysis and other settings, specificity is a performance indicator that assesses a prediction model's propensity to properly identify negative cases among all real negative instances. It focuses on how well the model can extract cases from the whole set of such instances that lack a certain attribute. Figure 7 denotes the compare sensitivity and specificity of the planned and prevailing methods.

Table 4represents the proposed method has highest value outcomes of the sensitivity and specificity.

Table 4: Outcomes of the sensitivity and specificity

Methods	Sensitivity	Specificity
DS-XGBoost	30	40
CNN-LSTM	20	50
PCA-SVM	35	25
DS-RF	40	48
IFSO-BPNN [Proposed]	45	52

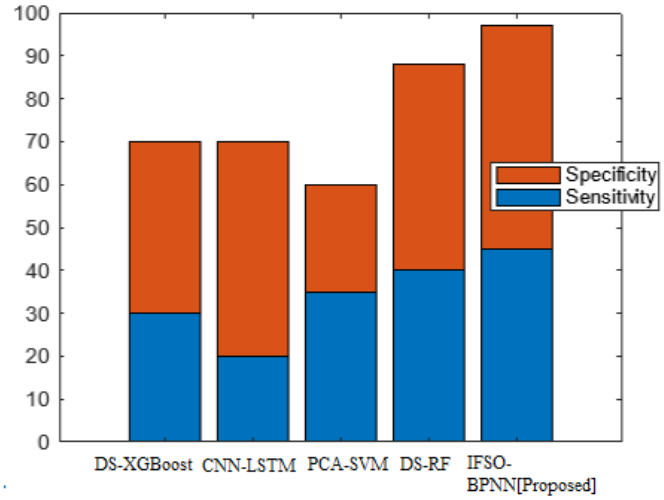


Figure 7: sensitivity and specificity of the proposed and existing methods

4.5 Flexibility

Employees can learn new abilities or alter existing ones to meet the changing needs of the workforce, demonstrating the adaptability of the digital finance. Workers should be allowed to switch between jobs and industries as necessary.By providing a variety of job opportunities, a varied economy with a range of businesses and sectors may help the early warning become more flexible.Flexibility may be impacted by AI. A more flexible workforce results from policies that support education, retraining, and mobility. Figure 8 presents the flexibility of the suggested and current techniques.

Table 5: Result of the Flexibility

Methods	Flexibility (%)
DS-XGBoost	85.5
CNN-LSTM	87.8
PCA-SVM	82.7
DS-RF	86.9
IFSO-BPNN [Proposed]	95.5

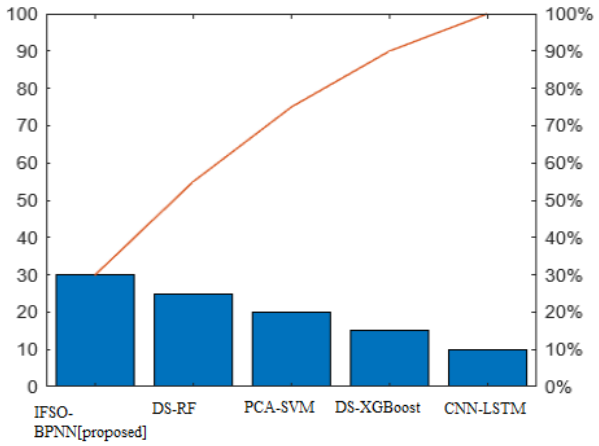


Figure 8: Flexibility of the proposed and existing methods

4.6 Confusion matrix

This study uses experiments to assess the effectiveness of the AI in early warning in digital finance approach. The following confusion matrix, which describes the model's classification, is derived from the experimental findings. The sample size that the technique accurately recognizes as a category in digital finance is known as the True Positive (TP). The quantity of samples that the model accurately recognizes as early warning patterns is known as the True Negative (TN). False Positive (FP): The model misidentifies the sample size of the AI education design as the samples of the non-art painting teaching design. The model misidentifies the samples size of the paintingdesign education as the number of samples of the non-art education, which is known as a False Negative (FN). The indications can be computed using Fig. 9.
TP: The no. of samples that the model suitablyestimate as positive is 85. TN: The no. of samples exactly forecast as negative by the model is 80. FP: The model erroneouslyforecasts the no, of samples of the negatives as positive, which is 15. FN: The methoderroneously predicts the no. of samples of the positives as the no. of samples of the negatives, which is 20.

The model's performance can be regarded as reasonably good based on the confusion matrix's findings. The amount of sections in this framework is represented by the parameter k in the region under the curve (AUC) plot, which is displayed in Figure 8. In contrast, the value of k in this study is fixed to 8, meaning that there are eight distinct groups of samples. The assessment findings of the presentation of the art painting instructional design based on a mobile platform are presented in the AUC plot. The True Positive Rate is shown on the vertical axis, while the False Positive Rate is shown on the horizontal axis. With an accurateness of 0.95, the AUC, which measures the model classifier's performance, shows that the model does fit in multi-class organizationresponsibilities.

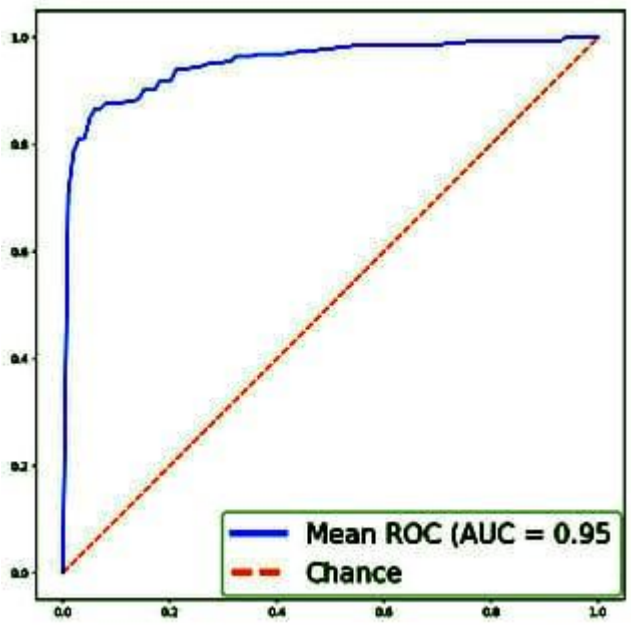


Figure9: ROC Curve

5 Discussion

The regular Back Propagation Neural Network (BPNN), Particle Swarm Optimization-enhanced BPNN (PSO-BPNN), and Support Vector Machines (SVM) were chosen as comparative models in order to offer a thorough assessment of the suggested IFSO-BPNN model across a variety of algorithmic families. SVM is a reliable, non-neural model frequently used in financial classification tasks, PSO-BPNN is a popular bio-inspired enhancement technique, and BPNN is a baseline to evaluate the impact of optimization. Fair performance benchmarking in terms of generalization, accuracy, and convergence rate is made possible by these selections. By avoiding local minima and guaranteeing smoother convergence through adaptive parameter adjustment, the IFSO-BPNN model showed exceptional prediction ability. It is not without restrictions, though. The model's scalability for very big or real-time datasets may be limited by its increasing computing cost due to its reliance on metaheuristic optimization. Furthermore, reproducibility may be impacted by the fine-tuning of swarm behavior characteristics (such as vision range, step size, and crowding factor), which affects its performance. In contrast to more modern deep learning models like LSTM or Transformer-based architectures used in financial risk modeling, the IFSO-BPNN might not be able to recognize hierarchical patterns or long-term dependencies in more complicated data. To get around these issues, future studies might look into hybridizing IFSO-BPNN with deeper architectures or parallel processing.

6 Conclusion

Early warning systems are crucial before an incident arises because they allow organizations to reduce losses. Rather than being only forecasting tools, they ought to be viewed as whole systems that relate to the subject matter of risk management. This study shed light on the bibliometric pattern that exist in the discipline of early detection system in economics, management, government, corporate finance and business Our study's contribution is that it offers novel perspectives and important information on the creation and application of early warning systems in the financial and economic sectors. Early warning systems are innovative technology that reduces expenses and allows for more effective operations. Thus, acquiring information to identify hazardbeforehand to keep a modestadvantage in the marketplace. This study's expanded comprehension of the application of primary warning technology in the field of digital finance is another contribution. A few potential applications for early warning systems are outlined in this article. Directors at all administrativehierarchy may find the findings helpful, in part because it is crucial to identify the additional value of AI-powered early warning systems from the standpoint of risk reduction. According to our research, the IFSO-BPNN Algorithm has the highest accuracy, with 98.7% accuracy, 97.8% precision, and a 95.7% F1-score. The model classifier's performance is measured by the AUC, which has an accuracy of 0.95, sensitivity and flexibility of 45 and 52, and flexibility of 95.5%. To find out which papers our analysis might have overlooked, upcoming bibliometric research on primary warning technology should examine different databases. Information onprimary warning systems using catagorical data may be included in future studies that concentrate on a more thorough examination of this subject. For more applicability and robustness, future studies will concentrate on incorporating real-time data streams, improving model adaptation to changing financial hazards, and extending the IFSO-BPNN framework to multi-source, cross-border financial information.

Declaration

Ethics approval and consent to participate: I confirm that all the research meets ethical guidelines and adheres to the legal requirements of the study country.

Consent for publication: I confirm that any participants (or their guardians if unable to give informed consent, or next of kin, if deceased) who may be identifiable through the manuscript (such as a case report), have been given an opportunity to review the final manuscript and have provided written consent to publish.

Availability of data and materials: The data used to support the findings of this study are available from the corresponding author upon request.

Competing interests: here are no have no conflicts of interest to declare.

Authors' contributions (Individual contribution): All authors contributed to the study conception and design. All authors read and approved the final manuscript

References

[1] X. Jiang, X. Wang, J. Ren, and Z. Xie, “The Nexus between Digital Finance and Economic Development: Evidence from China,” *Sustainability*, vol. 13, no. 13, p. 7289, Jun. 2021, doi: https://doi.org/10.3390/su13137289

[2] Emrah İsmail Çevik, SametGunay, Muhammad Wasif Zafar, Mehmet AkifDestek, Mehmet FatihBugan, and F. Tuna, “The impact of digital finance on the natural resource market: Evidence from DeFi, oil, and gold,” *Resources Policy*, vol. 79, pp. 103081–103081, Dec. 2022, doi: https://doi.org/10.1016/j.resourpol.2022.103081

[3] J. Wang, “‘The Party Must Strengthen Its Leadership in Finance!’: Digital Technologies and Financial Governance in China’s Fintech Development,” *The China Quarterly*, pp. 1–20, Sep. 2020, doi: https://doi.org/10.1017/s0305741020000879

[4] J. Li and B. Li, “Digital inclusive finance and urban innovation: Evidence from China,” *Review of Development Economics*, vol. 26, no. 2, pp. 1010–1034, Nov. 2021, doi: https://doi.org/10.1111/rode.12846

[5] X. Fang, “Can digital finance promote individuals’ charitable giving? Empirical evidence from China,” *Computers in Human Behavior*, p. 107768, Mar. 2023, doi: https://doi.org/10.1016/j.chb.2023.107768

[6] B. Ren, L. Li, H. Zhao, And Y. Zhou, “The Financial Exclusion In The Development Of Digital Finance — A Study Based On Survey Data In The Jingjinji Rural Area,” *The Singapore Economic Review*, vol. 63, no. 01, pp. 65–82, Feb. 2018, doi: https://doi.org/10.1142/s0217590818500017

[7] D. Luo, M. Luo, and J. Lv, “Can Digital Finance Contribute to the Promotion of Financial Sustainability? A Financial Efficiency Perspective,” *Sustainability*, vol. 14, no. 7, p. 3979, Mar. 2022, doi: https://doi.org/10.3390/su14073979

[8] Y. Liu, “Construction of Digital Financial Risk Early Warning Model Based on Decision Tree Algorithm,” vol. 41, pp. 142–146, Mar. 2023, doi: https://doi.org/10.1109/icdsm59373.2023.00038

[9] A. Jing, "The technology and digital financial risk management model using intelligent data processing," *Optik*, vol. 273, p. 170410, Feb. 2023, doi: https://doi.org/10.1016/j.ijleo.2022.170410

[10] X. Duan, X. Du, and L. Guo, "Digital Industry Financial Risk Early Warning System Based on Improved K-Means Clustering Algorithm," *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–9, May 2022, doi: https://doi.org/10.1155/2022/6797185

[11] K. Hommel and P. M. Bican, "Digital Entrepreneurship in Finance: Fintechs and Funding Decision Criteria," *Sustainability*, vol. 12, no. 19, p. 8035, Sep. 2020, doi: https://doi.org/10.3390/su12198035

[12] M. Wen, N. Peng, and J. Xiang, "Early Warning and Risk Analysis of Internet Finance Based on Data Mining Algorithm," *Proceedings of the 2023 3rd International Conference on Big Data, Artificial Intelligence and Risk Management*, pp. 568–573, Nov. 2023, doi: https://doi.org/10.1145/3656766.3656862

[13] P. Du and H. Shu, "Design and Implementation of China Financial Risk Monitoring and Early Warning System Based on Deep Learning," *IEEE Access*, vol. 11, pp. 78052–78058, Jan. 2023, doi: https://doi.org/10.1109/access.2023.3280934

[14] X. Sun and Y. Lei, "Research on financial early warning of mining listed companies based on BP neural network model," *Resources Policy*, vol. 73, p. 102223, Oct. 2021, doi: https://doi.org/10.1016/j.resourpol.2021.102223

[15] L. Chen and R. Jiang, "Internet Financial Risk Model Evaluation and Control Decision Based on Big Data," *Wireless Communications and Mobile Computing*, vol. 2022, pp. 1–10, Aug. 2022, doi: https://doi.org/10.1155/2022/8606624

[16] B. Youssef and N. Mansour, "Finance in the Digital Age: The Challenges and Opportunities," *Lecture notes in networks and systems*, pp. 45–59, Jan. 2024, doi: https://doi.org/10.1007/978-3-031-54009-7_5

[17] N. Mansour, "Digital Finance and Commercial Banks' Risk," *International Journal of E-Business Research*, vol. 20, no. 1, pp. 1–22, May 2024, doi: https://doi.org/10.4018/ijebr.344418

[18] Q. Cao, T. Zhu, W. Yu, and H. Tan, "Off-balance sheet business development and bank efficiency: A financial innovation matching perspective," *Heliyon*, vol. 9, no. 6, p. e16093, Jun. 2023, doi: https://doi.org/10.1016/j.heliyon.2023.e16093. Available: https://www.sciencedirect.com/science/article/pii/S2405844023033005

[19] U. Hübenbecker, "Is nobody interested in the future of banks? a scoping literature review on the state of the debate," *Management Review Quarterly*, Mar. 2023, doi: https://doi.org/10.1007/s11301-023-00334-8

[20] G. Li, E. Elahi, and L. Zhao, "Fintech, Bank Risk-Taking, and Risk-Warning for Commercial Banks in the Era of Digital Technology," *Frontiers in Psychology*, vol. 13, Jul. 2022, doi: https://doi.org/10.3389/fpsyg.2022.934053

[21] H. Nobanee, M. N. Dilshad, M. Al Dhanhani, M. Al Neyadi, S. Al Qubaisi, and S. Al Shamsi, "Big Data Applications the Banking Sector: A Bibliometric Analysis Approach," *SAGE Open*, vol. 11, no. 4, Oct. 2021, doi: https://doi.org/10.1177/21582440211067234

[22] M. Qi, J. Zhang, J. Xiao, P. Wang, D. Shi, and A. B. Nnenna, "Interconnectedness and systemic risk measures of Chinese financial institutions," *Kybernetes*, vol. ahead-of-print, no. ahead-of-print, Sep. 2021, doi: https://doi.org/10.1108/k-04-2021-0270

[23] G. Adosoglou, G. Lombardo and P. M. Pardalos, "Neural Network Embedding Methods for Mining Corporate Report Data - ProQuest," *Proquest.com*, 2021. Available: https://www.proquest.com/openview/34659c7a5b52f691dcb89f454d6aee22/1?pq-origsite=gscholar&cbl=18750&diss=y. [Accessed: Nov. 14, 2024]

[24] G. Trierweiler Ribeiro, A. Alves Portela Santos, V. Cocco Mariani, and L. dos Santos Coelho, "Novel hybrid model based on echo state neural network applied to the prediction of stock price return volatility," *Expert Systems with Applications*, vol. 184, p. 115490, Dec. 2021, doi: https://doi.org/10.1016/j.eswa.2021.115490

[25] G. Du, Z. Liu, and H. Lu, "Application of innovative risk early warning mode under big data technology in Internet credit financial risk assessment," *Journal of Computational and Applied Mathematics*, vol. 386, p. 113260, Apr. 2021, doi: https://doi.org/10.1016/j.cam.2020.113260

[26] T. Zhang, W. Zhu, Y. Wu, Z. Wu, C. Zhang, and X. Hu, "An explainable financial risk early warning model based on the DS-XGBoost model," *Finance Research Letters*, vol. 56, p. 104045, Sep. 2023, doi: https://doi.org/10.1016/j.frl.2023.104045. Available: https://www.sciencedirect.com/science/article/pii/S1544612323004178. [Accessed: Jan. 08, 2024]

[27] Z. Xia, "Early Warning of Credit Risk of Internet Financial Enterprises Based on CNN-LSTM Model," *Procedia Computer Science*, vol. 243, pp. 506–513, Jan. 2024, doi: https://doi.org/10.1016/j.procs.2024.09.062

[28] H. Liu and J. Lai, "Research on Financial Early Warning Based on Support Vector Machine Algorithm PCA-SVM Model," Sep. 2023, doi: https://doi.org/10.1109/iiotbdsc60298.2023.00052

[29] W. Zhu, T. Zhang, Y. Wu, S. Li, and Z. Li, "Research on optimization of an enterprise financial risk early warning method based on the DS-RF model," *International Review of Financial Analysis*, vol. 81, p. 102140, May 2022, doi: https://doi.org/10.1016/j.irfa.2022.102140

[30] L. Vancsura, T. Tatay, and T. Bareith, "Navigating AI-Driven Financial Forecasting: A Systematic Review of Current Status and Critical Research Gaps," *Forecasting*, vol. 7, no. 3, p. 36, Jul. 2025, doi: https://doi.org/10.3390/forecast7030036

[31] Ene Cezar Cătălin, "ADVANCED NEURAL NETWORKS AND DEEP LEARNING TECHNIQUES IN FINANCIAL MARKET PREDICTION," *ResearchGate*, no. 2, p. 2025, Apr. 2025, Available: https://www.researchgate.net/publication/393452829_ADVANCED_NEURAL_NETWORKS_AND_DEEP_LEARNING_TECHNIQUES_IN_FINANCIAL_MARKET_PREDICTION. [Accessed: Jul. 22, 2025]

[32] A. Gupta, D. Kapil, A. Jain, and H. S. Negi, "Using Neural Network for Financial Forecasting," pp. 1–6, May 2024, doi: https://doi.org/10.1109/incet61516.2024.10593292

[33] Y. Ahmad *et al.*, "SiFSO: Fish Swarm Optimization-Based Technique for Efficient Community Detection in Complex Networks," *Complexity*, vol. 2020, pp. 1–9, Dec. 2020, doi: https://doi.org/10.1155/2020/6695032

Chain Mediation Path Optimization for Smartphone Dependence Prediction Using Graph Neural Networks and Reinforcement Learning

Guiping Xuan, Tianping Li*
Hunan Polytechnic of Environment and Biology, Hengyang, Hunan, 421000, China
E-mail：ltp0922@163.com
*Corresponding author

Keywords: Chain Mediation path, Graph modeling, Deep learning, Reinforcement learning, Causal inference

Received: Januar 22, 2026

Smartphone dependence prediction and subjective well-being impact analysis are challenged by complex variable interactions, suboptimal mediation-chain optimization, and limited interpretability. This paper proposes a chain mediation path optimization framework that formalizes mediation as a directed causal graph, where nodes denote behavioral and psychometric variables and edges encode dependency relations. Graph convolutional neural networks learn high-level node embeddings, and an attention-based multi-scale causal embedding module enhances structural expressiveness at semantic and topological levels. For path optimization, a policy-gradient reinforcement learning module dynamically updates edge weights under a reward that balances prediction accuracy, path consistency, and computational cost. A graph autoencoder with geometric consistency constraints is introduced to stabilize reconstructed mediation chains and preserve structural coherence. The framework is evaluated on 5,724 users, with an average of 58 behavioral features and 12 psychometric indicators per sample, using stratified sampling and cross-validation. The proposed model achieves 92.1%±0.4 Accuracy, 89.4%±0.6 F1-score, and 87.8%±0.5 Topology Score, outperforming a structural equation model baseline and a deep neural network without causal-path constraints. Ablation results verify that attention, reinforcement learning, and geometric constraints jointly improve classification performance and path stability. These results indicate that graph-based mediation path modeling enables unified prediction of dependence and well-being outcomes with improved interpretability and deployability in large-scale settings.

Povzetek: Prispevek predstavi grafni okvir z grafnimi nevronskimi mrežami in ojačitvenim učenjem za optimizacijo mediacijskih poti, ki natančneje napoveduje odvisnost od pametnih telefonov in njen vpliv na subjektivno dobrobit ter hkrati izboljša razlagljivost modela.

1 Introduction

The development of artificial intelligence and causal inference has driven the intelligent modeling of complex behavioral data, providing new ideas for user behavior prediction and multi-task causal representation. As a high-frequency interactive tool, the dependence behavior of smartphones is encoded by usage patterns and subjective well-being scores. However, features such as usage duration, interaction frequency, sleep patterns, and emotional states have nonlinear dependencies and causal chains, making it difficult for traditional statistical models to handle them effectively. Especially in the study of chain mediating effects, the interaction of variables presents high-dimensional, multi-level and dynamic characteristics, and it is difficult for regression and structural equation models to balance prediction and interpretation.

Existing research has provided empirical foundations for smartphone dependence modeling and user state inference. Beames et al. (2024) utilized sensor data to detect depression and anxiety in adolescents, demonstrating the value of multimodal characteristics [1]. Asare et al. (2021) proposed a depression prediction method based on machine learning and hyperparameter optimization, and combined feature importance analysis to improve accuracy [2]. Shin and Dey (2013) verified the role of behavioral data in dependency modeling by using patterns to detect problematic behaviors [3]. The MoodScope framework proposed by LiKamWa et al. (2013) realizes emotion recognition based on the use of data [4]. Stachl et al. (2020) further demonstrated the potential of smartphone data in large-scale behavior modeling and user profiling by predicting personality traits [5].

Despite this, the existing methods still have shortcomings: most studies remain at the level of feature correlation and lack structured expression at the causal chain level; The lack of a dynamic optimization and feedback mechanism for the mediating effect chain makes

it difficult to adapt to complex environments. The insufficient interpretability and scalability of the model limit its cross-scenario application.

To address these issues, this paper proposes a chain-mediated path optimization framework that abstracts the mediation process as a directed causal graph. Graph convolutional networks are used for node embedding, an attention module highlights influential mediators, and a policy-gradient reinforcement learning module dynamically updates path weights to improve both predictive performance and path consistency. A graph autoencoder with geometric-consistency constraints is further introduced to stabilize the reconstructed chains and reduce structural drift.

The research design follows three concrete goals and measurable outcomes. The first goal is to construct a computable chain-mediation graph from behavioral logs and psychometric indicators, producing a topology-aware directed graph representation. The second goal is to improve dependence and well-being prediction through GNN embeddings and multi-scale attention, evaluated by Accuracy, F1-score, and Topology Score under cross-validation. The third goal is to optimize and stabilize mediation paths via reinforcement learning and geometric constraints, validated by ablation and robustness tests.

Accordingly, this study addresses three research questions: RQ1: Can chain-mediated paths represent complex causal chains through graph modeling? RQ2: Can attention-based multi-scale representations improve mediating variable identification and path prediction? RQ3: Can reinforcement learning enhance path-weight updating while maintaining consistency between prediction and interpretation?

The innovation of this study lies in: the method formalizes the chain mediation path into a graph modeling task, integrating deep learning and causal inference to achieve structured expression; The attention mechanism is introduced in the task to form a collaborative framework with reinforcement learning, enhancing the robustness of the model under complex behavioral data. For the first time in application, chain mediation path optimization was used to model smartphone dependence and well-being, proposing an interdisciplinary and computable causal inference scheme.

2 Relevant work

In the prediction of smartphone dependence and the modeling of subjective well-being, existing research mainly focuses on three directions: application prediction, causal modeling, and path optimization. In terms of behavior and application prediction, Yu et al. (2017) established a prediction model by leveraging application usage and point of interest features, revealing the coupling effect between contextual data and behavioral patterns [6]. Yuchi et al. (2024) proposed a new user event prediction method based on causal inference, which improved the generalization performance in complex environments [7]. Kwapisz et al. (2011) utilized an acceleration sensor for activity recognition, demonstrating the feasibility of sensor data in dependent behavior modeling [8]. In the field of causal modeling and optimization, Aglietti et al. (2020) proposed a causal Bayesian optimization method for the selection of key causal variables in high-dimensional Spaces [9]. Zhang and Zhang (2024) optimized the structure of causal networks based on active learning, enhancing the efficiency of causal relationship learning [10]. Cousineau et al. (2022) conducted a review and empirical comparison of optimization-based causal effect estimation methods, clarifying the applicable scenarios of different algorithms [11]. Cao (2025) combines causal state masking with deep reinforcement learning for dynamic path planning, demonstrating adaptive optimization capabilities [12]. Mi et al. (2019) achieved real-time recognition of smartphone behavior through time series algorithms, demonstrating the value of time series modeling for dependency prediction [13]. In the field of complex systems and cross-scenario reasoning, Sun et al. (2024) utilized causal network reconstruction to achieve industrial process prediction, verifying the feasibility of causal modeling in complex systems [14]. Kalisch and Buhlmann (2014) systematically reviewed causal structure learning and inference methods, providing a theoretical basis for path optimization [15]. Hua et al. (2022) proposed a zero-shot prediction method based on causal inference, which solved the cross-scenario modeling problem in non-stationary environments [16]. Xing et al. (2024) designed a deep learning causal inference architecture for modeling the relationship between stock closing prices and related factors, demonstrating the integration potential of causal inference and deep models [17]. Gao et al. (2024) summarized the current application status of causal inference in recommendation systems, providing an important reference for multi-task prediction [18]. In the direct application of dependency prediction, Raj et al. (2024) proposed a machine learning-based addiction prediction model that effectively identified users' smartphone dependences [19]. Li et al. (2025) identified the key factors of adolescent dependence through machine learning and network analysis, providing empirical support for mediating variable modeling [20]. Boulkroune et al. (2025) presented practical finite-time fuzzy synchronization with chattering-free designs, providing a finite-time performance viewpoint that can be leveraged to discuss invariance and stability of learned paths beyond predictive gains [21].Rigatos et al. (2024) developed a flatness-based successive-loop control scheme for autonomous quadrotors, showing how structured loops can deliver explicit stability-oriented constraints that are comparable to path-level optimization objectives in data-driven frameworks [22]. Rigatos et al. (2020) further investigated nonlinear optimal control for autonomous submarines' diving, highlighting the role of convergence-aware objective design when system dynamics exhibit strong nonlinearity and external disturbances [23]. Rigatos et al. (2024) extended the successive-loop flatness-based formulation to dual-arm robotic manipulators, providing additional evidence that decomposed control paths can remain stable under coupled multi-actuator interactions [24]. Rigatos et al. (2017) reported a nonlinear optimal control method for submarine diving in industrial

electronics setting, offering a practical reference for linking optimized trajectories to enforceable control policies [25]. Zouari and Mahmud (2024) proposed a neural network-based robust adaptive output-feedback control method for MIMO time-varying delay systems, emphasizing robustness and stability considerations under time-varying uncertainties that align with distribution-shift concerns in learned mediation chains [26]. To provide a verifiable comparison between prior studies and the proposed approach, the representative methods and their datasets, modeling ideas, and reported outcome types are summarized in Table 1.

Table 1: Summary of representative related methods, reported outcome types, and limitations.

Method (Ref.)	Year	Dataset	Main approach	Reported metric / outcome type	Stated limitation
App-Predict [6]	2017	Behavior + POI data	Application behavior modeling	Classification performance (Acc)	Limited causal dependency; weak chain-level modeling
Event-Causal [7]	2024	User interaction logs	Causal inference for event prediction	Event prediction performance (F1)	Generalization under complex environments remains limited
Causal-BayesOpt [9]	2020	Synthetic causal data	Bayesian optimization + variable selection	Causal structure quality (Topology-related)	Efficiency degrades in high-dimensional causal spaces
DRL-Path [12]	2025	Path simulation environment	Causal state masking + reinforcement learning	Path planning / policy performance (Acc-type)	Limited validation on real dependence and well-being scenarios
This paper (Causal-Path GNN+RL)	2025	5,724 user data	Graph modeling + attention + RL path optimization	Acc, F1, and Topology Score (reported in this study)	Distribution-shift stability and deployment-oriented confidence reporting require further analysis

Current research remains limited in explicit chain-level causal graph modeling, feedback-driven dynamic path optimization, and robustness under scenario shifts. Therefore, a unified framework that integrates graph-based mediation representation, reinforcement learning path optimization, and feedback regulation is required to support dependable multi-task dependence prediction and well-being modeling.

3 Modeling scheme design

3.1 Chain mediation path modeling framework

The core of chain-mediated path modeling lies in transforming the causal relationship between smartphone dependence behavior and subjective well-being into a computable graph structure to support deep learning and path optimization. In this framework, variables are abstracted as nodes, causal relationships are connected in the form of edges, and paths represent the transmission logic of mediating effects. The overall form is defined as:

$$G=(V,E),\ V=\{v_1,v_2,\ldots,v_n\},\quad E\subseteq V\times V \tag{1}$$

Among them, V is a set of variable nodes, including key variables such as the duration of smartphone usage, interaction frequency, sleep patterns, emotional states, and subjective well-being; E is the causal edge set, which is used to describe the dependency relationship between variables. Each node constructs the initial feature vector $x_i \in R^d$ by extracting behavioral and psychological attributes, which is defined as:

$$F(v_i)=\left[f_t^i, f_f^i, f_s^i, f_e^i, f_c^i\right]\in R^d \tag{2}$$

Among them, f_t^i represents the usage duration feature of node i, which is used to characterize the intensity of mobile phone usage. f_f^i represents the interaction frequency feature, reflecting the level of social activity. f_s^i is a sleep pattern parameter, describing the circadian rhythm and sleep quality; f_e^i is the emotional level score, indicating the degree of social anxiety. f_c^i represents the subjective well-being score, reflecting the overall level of psychological state. This formula realizes the unified encoding of the original behavior and psychometric data, enabling them to be used as the input of graph convolutional networks, ensuring the consistency of feature scales among different samples, and providing a basis for the subsequent optimization of causal chains.

In chain path modeling, it is also necessary to introduce position embedding to quantify the relative relationship between nodes. Its calculation method is defined as:

$$P(v_i, v_j) = \frac{\left\| c_i - c_j \right\|_2}{d} \tag{3}$$

Among them, $P(v_i, v_j)$ is the relative position embedding of node i and node j, representing the structural difference between the two nodes in the causal chain; c_i, c_j are the feature center vectors of node i, j respectively, $\left\| c_i - c_j \right\|_2$ is the Euclidean distance of the feature difference between nodes, and d is the feature dimension normalization factor. This formula is used to calculate the causal relative differences between nodes, enabling the graph convolution propagation process to maintain the topological consistency and causal stability of the chain mediation path.

As shown in Figure 1, the chain mediation path modeling process consists of five steps: data preprocessing, variable node identification, node feature encoding, causal edge generation, and path sequence construction. The input data comes from smartphone sensor logs and psychological scales. The former includes duration, frequency and sleep characteristics, while the latter covers emotional and happiness scores. Nodes are set through mapping rules, and the generation of edges is based on causal assumptions and statistical tests, combined with expert experience correction to ensure the rationality and accuracy of the causal structure. Ultimately, a unified input representation is obtained through feature concatenation and position normalization, providing a modeling basis for the subsequent attention mechanism and reinforcement learning module.

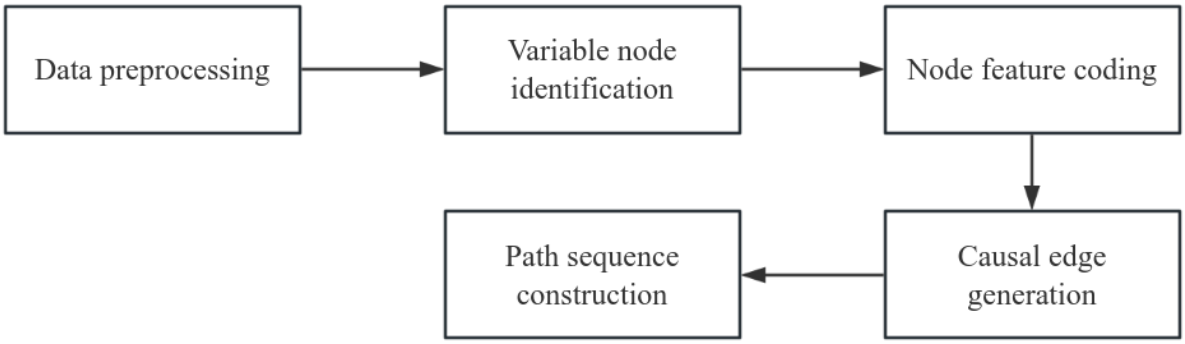


Figure 1: Flowchart of chain mediation path modeling.

During the process of quantifying node features, the usage duration and interaction frequency are calculated and normalized through behavioral logs. Sleep patterns are extracted from time series sensor data and encoded into periodic vectors. The characteristics of emotions and well-being are derived from the scores of standardized psychological scales, which are normalized and mapped to the codes of single fever. The above features are concatenated into a unified input vector to ensure consistent feature dimensions and support joint training in graph convolution and reinforcement learning modules.

This framework, by structuring the complex causal relationship diagram, not only achieves the quantitative representation of the chain mediation path but also provides a modeling foundation for the subsequent attention mechanism and reinforcement learning, ensuring the robustness and interpretability of the model in multi-variable interaction scenarios.

3.2 Smartphones rely on structured representation of behavioral characteristics

In the prediction of smartphone dependence behavior and the modeling of subjective well-being, the sources of raw data are complex, including not only the duration, interaction frequency, application switching and night usage patterns in sensor logs, but also the emotional state

and well-being scores in psychological scales. The features from different sources vary significantly in dimensions, distribution and noise. If they are not uniformly numerically and structurally represented, it is easy to cause instability in the modeling of causal chains. Therefore, this study first standardizes the behavioral and psychological characteristics to enable them to enter the graph structure model under a unified scale. The feature standardization formula is defined as follows:

$$\hat{x}_{i,k} = \frac{x_{i,k} - \mu_k}{\sigma_k} \tag{4}$$

Among them, $x_{i,k}$ represents the original value of User i on the k feature, such as usage duration or happiness score; μ_k is the mean value of this feature in the training set; σ_k is the standard deviation of this feature; $\hat{x}_{i,k}$ represents the standardized eigenvalue. This formula is used to normalize multi-source features, ensuring that different variables fall within the same numerical range, thereby avoiding the weight shift of the causal chain caused by inconsistent feature dimensions.

After the feature standardization is completed, the processed features need to be mapped to a high-dimensional embedding space to enhance their expressive power and capture potential nonlinear interaction relationships. The embedding mapping formula is defined as follows:

$$h_i = \sigma\left(W\hat{X}_i + b\right) \tag{5}$$

Among them, h_i is the embedded representation of user i, containing higher-level behavioral and psychological semantic information; $\hat{X}_i$ is the standardized feature vector of user i; W is a trainable weight matrix used for linear mapping features; b is the bias term; σ is a nonlinear activation function (such as ReLU or Sigmoid). This formula, through the combination of linear mapping and nonlinear activation, transforms the original features into a more robust embedded representation, making it applicable to the subsequent graph convolution propagation and path optimization stages.

In the specific implementation, behavioral characteristics such as duration and interaction frequency are normalized to the maximum and minimum before being standardized to ensure their stable distribution. The sequence statistics are extracted through the sliding window by applying the switching features to form the time series pattern. Sleep patterns obtain periodic parameters through signal processing; The emotional and happiness scores of the psychological scale are converted into numerical inputs through discrete interval mapping and individual heat coding. Ultimately, all features are concatenated into an input matrix on a unified dimension, and after standardization and embedding mapping, a node representation is formed.

Through the above structured representation process, the model has achieved the digitalization, unification and high-dimensional embedding of multi-source heterogeneous data, which not only enhances the stability of causal chain modeling, but also provides a robust input basis for the optimization of chain mediation paths.

3.3 Subjective well-being influence path optimization algorithm based on intermediary chain

In the modeling process of the chain mediation path, there are dynamic differences in the contributions of different mediation variables to the final outcome variables.

The traditional method adopts fixed weights and is difficult to adapt to the nonlinear changes of variable relationships in the real environment. To this end, this study proposes a path optimization method based on the attention mechanism and reinforcement learning. By dynamically updating the path weights and implementing a feedback-driven optimization strategy, the adaptive adjustment of the intermediary chain is achieved. In the path weight update stage, the attention weighting method is introduced to dynamically aggregate the intermediary nodes, which is defined as follows:

$$h_i' = \sigma\left(\sum_{j \in N(i)} \beta_{ij} W h_j\right) \tag{6}$$

Among them, h_i' represents the updated node; h_j represents the characteristics of neighboring nodes; W is the trainable weight matrix; σ is the nonlinear activation function; β_{ij} represents the path weight of node j to node i, which is adaptively calculated by the attention mechanism based on feature correlation. This formula highlights the mediating variables that have a significant impact on the prediction of outcome scores, such as sleep quality and emotional state, through dynamic weight distribution. In the process of path optimization, a reinforcement learning mechanism is introduced, and the path selection is feedback regulated through the reward function:

$$R = \lambda_1 \cdot Acc + \lambda_2 \cdot F1 - \lambda_3 \cdot C \tag{7}$$

Among them, R is the reward value for path optimization; Acc indicates the accuracy rate of dependent behavior prediction; $F1$ represents the F1-score for path recognition; C is the computational cost; λ_1, λ_2, λ_3 is the equilibrium coefficient. This formula is used to guide reinforcement learning agents to balance prediction accuracy, robustness and computational efficiency during the exploration process, ensuring that the optimized path is both stable and efficient.

Through the above mechanism, the model can adaptively identify key mediating variables in the causal

chain and dynamically update the path weights through feedback optimization, thereby significantly enhancing the modeling ability of the relationship between smartphone dependence behavior and subjective well-being.

3.4 Integrated deployment and feedback mechanism of smartphones relying on predictive models

In the multi-task prediction of smartphone dependence behavior and related outcomes, a single model often finds it difficult to simultaneously balance prediction accuracy, model stability and real-time response performance. To address this issue, this study proposes a model mechanism based on integrated deployment and feedback regulation, which jointly optimizes three sub-modules: dependency prediction, emotion recognition, and happiness estimation, and realizes the adaptive update of the model in a dynamic environment through feedback loops. This mechanism can integrate the outputs of multiple tasks within the same framework, ensuring that the prediction results strike a balance between accuracy and robustness.

During the training and inference process, the objective function of the overall prediction model consists of three parts: dependency prediction loss, emotion recognition loss, and happiness estimation loss. The total loss function is defined as:

$$L_{total} = \alpha L_{dep} + \beta L_{emo} + \gamma L_{hap} \tag{8}$$

Among them, L_{total} represents the overall optimization objective, which is used for jointly balancing multi-task training; L_{dep} is the cross-entropy loss of smartphone dependence prediction, which is used to measure the accuracy of the model's classification of dependency degrees; L_{emo} is the binary cross-entropy loss of the emotion recognition task, used to evaluate the recognition performance of anxiety or depression states; L_{hap} is the mean square error of the happiness estimation, which is used to describe the difference between the model's predicted value and the true scale score. α, β, γ are the weight coefficients of the three tasks respectively. In this study, parameters were adjusted within the interval of {0.3,0.5,1.0} through grid search, and the optimal combination was selected on the validation set to ensure the balance among multiple tasks. This formula is used to jointly optimize the performance of multiple modules, ensuring that the prediction results maintain high accuracy simultaneously in dependency identification and happiness estimation.

During the model deployment stage, a hierarchical integration architecture is adopted: the bottom layer is a lightweight deep neural network, mainly responsible for the rapid prediction of high-frequency behavioral data; The middle layer is a graph autoencoder, responsible for the reconstruction and optimization of causal paths. The top layer is the reinforcement learning scheduler, which updates the model weight distribution in real time based on feedback signals to achieve cross-scenario adaptability. In the design of the feedback mechanism, each prediction result is compared with the real data. When the error exceeds the preset threshold, the system will trigger local parameter updates to reduce the cumulative deviation of the model in complex environments.

To verify the effectiveness of the feedback mechanism, this study set up a comparative experiment: without the feedback mechanism, the prediction accuracy of the model showed a downward trend after long-term operation; After introducing feedback, the performance was significantly improved. Table 2 presents the experimental comparison results under the two mechanisms.

Table 2: Comparison of model performance under different deployment mechanisms.

Training Method	Classification Accuracy (%)	F1 Score	Average Latency (s)	Stability Score (10)
Without Feedback Mechanism	88.9	86.5	2.3	8.1
Integrated + Feedback Mechanism	92.1	89.2	1.9	9.0

The experimental results show that the integrated deployment and feedback mechanism outperforms the non-feedback mechanism in both classification accuracy and F1-score. At the same time, it performs better in delay control and stability, which proves the effectiveness and feasibility of this mechanism in large-scale applications.

To support practical deployment in digital health scenarios, the optimized mediation paths are mapped to actionable intervention targets, such as time-window usage nudges and adaptive usage-limit suggestions for high-impact segments (e.g., night-time use → sleep disruption → negative affect). A confidence score is also reported with each prediction to indicate uncertainty and help clinicians, policymakers, or platform designers prioritize high-risk cases.

4 Results

4.1 Dataset

For the tasks of predicting smartphone dependence behavior and modeling related outcomes, the construction of the dataset is a key link that determines the performance of the model. The data of this study mainly comes from two types of channels: one is the multimodal sensor logs of smartphones, including high-frequency characteristics such as usage duration, application switching, night usage frequency, screen lighting times and acceleration information; The second category includes psychological scales and happiness questionnaires, which cover indicators such as anxiety levels, emotional stability, and self-assessment of happiness. The two types of data are aligned through user ids and timestamps to form multi-dimensional causal chain samples.

The overall sample size is 5,724 users, with an average of 58 behavioral characteristics and 12 psychological indicators per sample. When dividing the data, a stratified sampling method was adopted to maintain a balanced distribution of different degrees of dependence and happiness intervals. The division of goals can be formalized as:

$$Q = \lambda_1 R_{dep} + \lambda_2 R_{hap} \tag{9}$$

Among them, R_{dep} represents the distribution balance degree of dependent behavior samples, R_{hap} represents the distribution balance degree of the happiness interval, and λ_1, λ_2 represents the adjustment coefficient. This formula is used to constrain the balance of data partitioning, ensuring that the training set, validation set and test set are consistent in statistical characteristics, thereby enhancing the generalization ability of the model. The final ratio was set at 7:1.5:1.5, and it was repeatedly verified on the validation set to ensure a reasonable division.

During the data cleaning stage, samples with a missing rate exceeding 20% were eliminated, and the local missing features were completed using mean interpolation and linear interpolation methods. Outlier processing is carried out using the three standard deviations criterion for filtering. The results of the psychological scale are discretized into three categories - low, medium and high - according to the happiness score range, and then converted into numerical features through the single fever code. The behavior logs have been standardized to ensure the comparability of features under different dimensions.

As shown in Figure 2, the dataset construction process includes five steps: raw data collection, feature cleaning, psychological indicator annotation, multi-source alignment, and data partitioning. First, collect user logs through the mobile SDK and simultaneously gather data from the happiness questionnaire. Subsequently, cleaning and missing corrections are carried out at the feature level; Then, multimodal alignment is completed through timestamps and user identifiers; Finally, training, validation and test subsets are generated based on the stratified sampling strategy to form a complete dataset.

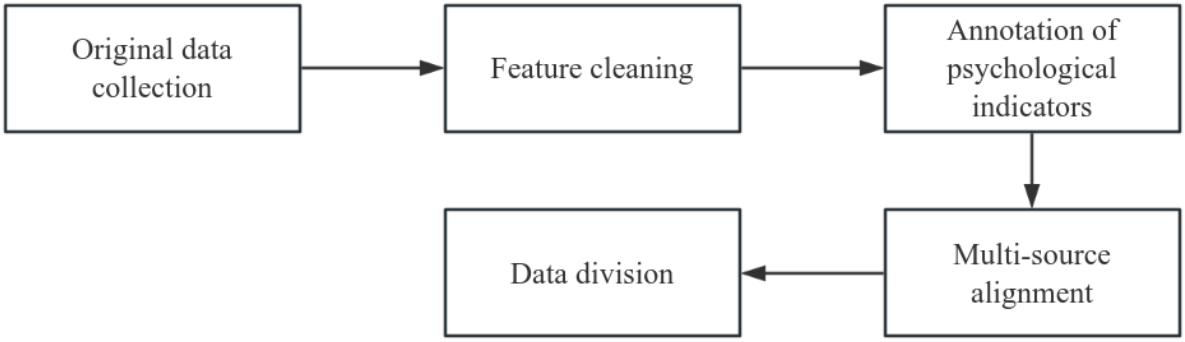


Figure 2: Dataset construction process.

This dataset not only meets the modeling requirements in terms of sample size, but also achieves a balanced distribution in dependency features and outcome indicators, providing reliable data support for causal modeling of chain mediation paths and verification of optimization algorithms.

Implementation Details. All models were implemented in PyTorch with stratified sampling (7:1.5:1.5) and 5-fold cross-validation. For Causal-Path GNN+RL, the graph encoder consists of two GCN layers with hidden dimensions of 128 and 64, followed by ReLU and dropout (0.3). The attention module uses 4 heads, and multi-scale aggregation is performed over neighborhood hops {1,2}. The graph autoencoder is jointly optimized with a geometric-consistency weight of 0.1. Supervised learning adopts Adam with learning rate 1e-3, weight decay 1e-4, batch size 64, and maximum 80 epochs with early stopping (patience=10). Reinforcement learning applies policy-gradient updates every 5 epochs with discount factor γ=0.99 and entropy regularization 0.01. In Eq. (7), the reward coefficients are set to $(\alpha,\beta,\lambda)=(1.0,0.5,0.1)$ after validation tuning. Baseline models follow the same protocol; DNN-Only uses a 3-layer MLP (256–128–64).

Replication Details. To enable replication, the complete pipeline is specified as follows. Data alignment uses user IDs and timestamps to merge smartphone sensor logs with questionnaire records; samples with a missing-rate >20% are removed, while remaining missing entries are imputed using window-based mean interpolation for time-series logs and feature-wise mean imputation for static attributes. Outliers are filtered by the 3σ rule and replaced with local window means; EWMA smoothing is applied to high-frequency sequences with coefficient 0.3. Behavioral features (58 per user on average) include usage duration, interaction frequency, app switching statistics, night-time usage frequency, screen-on counts, and accelerometer-derived activity fluctuation; psychometric indicators (12 per user on average) include affect-related scores and subjective well-being. All continuous features are min–max normalized (Eq. (10)) and then standardized (Eq. (4)); questionnaire well-being scores are discretized into three levels (low/medium/high) and encoded using one-hot vectors. Graph construction maps each variable to a node and initializes directed edges using causal assumptions plus statistical screening, with expert adjustment for ambiguous links; position embeddings are computed by Eq. (3).

Training protocol fixes the stratified split at 7:1.5:1.5 and reports mean±std under 5-fold cross-validation. Unless otherwise stated, the random seed is fixed to 42 for data splitting and parameter initialization. Causal-Path GNN+RL is trained with Adam (lr=1e-3, weight decay=1e-4, batch size=64) for up to 80 epochs, with early stopping (patience=10) based on validation F1-score. The GCN encoder uses hidden sizes 128 and 64 with ReLU and dropout 0.3; attention uses 4 heads and aggregates hops {1,2}. The graph autoencoder is jointly trained with geometric-consistency weight 0.1. Reinforcement learning performs policy-gradient updates every 5 epochs with γ=0.99 and entropy regularization 0.01; reward coefficients in Eq. (7) are (α,β,λ)=(1.0,0.5,0.1). Baseline models (SEM-Baseline and DNN-Only) follow the same split, cross-validation, and early-stopping rule; DNN-Only uses a 3-layer MLP (256–128–64).

Evaluation protocol reports Accuracy (Eq. (12)), F1-score (Eq. (13)), and Topology Score (directed chain-structure retention between predicted paths and reference paths), together with average latency measured as mean inference time per sample over the test fold. All reported results in Tables 2–5 are averaged across folds and are presented as mean±standard deviation.

4.2 Data preprocessing

In the task of predicting smartphone dependence behavior and modeling subjective well-being, the sources of raw data are complex, including not only multimodal sensor logs (such as screen lighting times, application switching frequencies, and acceleration fluctuations), but also psychological scales and well-being questionnaire scores. These data have problems such as missing parts, noise and inconsistent dimensions during the collection process. If they are not systematically preprocessed, it is very likely to lead to an imbalance in feature distribution and unstable model convergence in the modeling stage. To this end, this study designed a unified preprocessing procedure, covering feature normalization, missing value correction, time series alignment and outlier smoothing, etc., to ensure the stability and generalization ability of subsequent modeling.

In the feature normalization stage, for feature variables of different dimensions, the minimum-maximum normalization method is adopted to map them to a unified interval, in order to eliminate the dimensional differences between features. The specific definition is:

$$x_i' = \frac{x_i - \min(x)}{\max(x) - \min(x)} \tag{10}$$

Among them, x_i represents the original feature value, $\min(x)$ and $\max(x)$ are the minimum and maximum values of the feature respectively, and x_i' is the normalized feature value. This formula is used to scale the feature range, making different features comparable within the same numerical interval and avoiding large-scale features dominating gradient updates during training.

In the missing value correction and anomaly detection stage, this study combines the sliding window and mean interpolation strategies to perform time series repair on log-type features. If the missing ratio within the window is less than 30%, mean interpolation is used to complete the filling. If the threshold is exceeded, the sequence segment will be eliminated. Anomaly detection is based on the principle of three standard deviations, and values exceeding the interval will be replaced with local means. To further ensure the smoothness of the data, the Exponential Weighted Moving Average (EWMA) method is introduced to filter the time series features, and its formula is defined as:

$$s_t = \alpha x_t + (1-\alpha) s_{t-1} \tag{11}$$

Among them, s_t is the smoothing value at time t, x_t is the original observed value, and $\alpha \in (0.1)$ is the smoothing coefficient. This formula is used to weaken the interference of short-term fluctuations on the model input, thereby maintaining the stable trend of the data sequence.

In the data processing of psychological scales, the subjective well-being score is discretized into three levels: low, medium and high, and then converted into numerical representation through individual heat coding to facilitate multimodal fusion with sensor features. All samples are ultimately organized into a unified matrix form, with each row representing a user and each column corresponding to a normalized or encoded feature.

Through the above preprocessing, the missing rate of the original data decreased from 12.7% to 1.9%, and the variance of the feature distribution dropped by 36.2%, significantly enhancing the stability of model training. The overall preprocessing process lays a solid data foundation for the modeling of chain-mediated paths and provides high-quality input for subsequent path optimization and prediction tasks.

4.3 Evaluation indicators

In the tasks of predicting smartphone dependence behavior and multi-task outcome modeling, the evaluation metrics are selected to match two coupled objectives: high-fidelity classification of dependence and outcomes, and faithful preservation of the learned mediation-chain structure. Accordingly, this study reports Accuracy, F1-score, and Topology Score as complementary measures that jointly quantify predictive correctness, class-sensitive detection quality, and causal-chain coherence.

Accuracy provides an intuitive global view of correct predictions under the stratified protocol, but it can mask failure cases when dependence levels are unevenly distributed. For this reason, F1-score is used as a harmonic balance of Precision and Recall to reflect how reliably the model identifies high-risk dependence and different well-being intervals. Topology Score is introduced to evaluate whether improved prediction is accompanied by consistent directed-chain structure, which is not captured by purely label-based metrics. Alternative metrics such as ROC-AUC/PR-AUC, balanced accuracy, or MCC can also be used for binary classification; however, they do not directly assess mediation-chain preservation, and their ranking can be insensitive to whether the predicted graph paths remain coherent. Therefore, these alternatives are treated as optional supplements, whereas Accuracy/F1 are retained for comparability with prior dependence prediction studies and Topology Score is emphasized to align evaluation with the chained mediation objective.

In classification prediction tasks, accuracy is used to measure the correct proportion of the overall prediction results and is defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

Among them, TP and TN represent the number of samples with correct positive and negative predictions respectively, and FP and FN represent the number of samples with incorrect predictions respectively. This formula is used for global accuracy evaluation and can visually reflect the overall performance of the model in the tasks of smartphone dependence prediction and happiness classification.

However, relying solely on accuracy is not sufficient to comprehensively measure performance, especially when the distribution of data categories is uneven. For this reason, this study further introduces the F1-score to balance Precision and Recall. The calculation formula is as follows:

$$F1 = \frac{2 \times \Pr ecision \times \mathrm{Re}\, call}{\Pr ecision + \mathrm{Re}\, call} \tag{13}$$

Amongthem, $\Pr ecision = \frac{TP}{TP + FP}$, $\mathrm{Re}\, call = \frac{TP}{TP + FN}$. This formula can better reflect the stability of the model when identifying users with dependent behaviors and different happiness intervals, avoiding the situation of high accuracy but low recall rate.

In the evaluation of causal path optimization, this study adopts Topology Score to measure the structural similarity between the predicted path and the target path. Specifically, the structure retention rate is calculated by comparing the node sequence and connection relationship between the predicted chain and the reference chain. This indicator can effectively describe the explanatory ability of the model at the intermediate chain level and supplement the consistency of causal structure that classification indicators cannot reflect.

To verify the validity of the indicators, this study compared the traditional structural equation model with the chain mediation path optimization method proposed in this paper on the same dataset. The results show that in the dependency prediction task, the Accuracy has increased by 6.4%, the F1-score has increased by 7.1%, and the Topology Score has increased by 3.8%. This result indicates that the joint evaluation of multiple indicators can comprehensively demonstrate the model's advantages in prediction and interpretation, providing reliable quantitative support for complex behavior prediction tasks.

4.4 Ablation research

To verify the role of each core module in the chain mediation path optimization model, an ablation experiment was designed in this study. By gradually removing key modules such as the attention mechanism, reinforcement learning optimization, and geometric consistency constraints, the performance differences between the complete model and the ablation model are compared, thereby quantifying the contributions of each module in the prediction of smartphone dependence behavior and the modeling of subjective well-being. In the performance evaluation, this paper defines the performance difference indicators as follows:

$$\Delta P = \left(P_{full} \pm \delta\right) - \left(P_{ablated} \pm \delta\right) \tag{14}$$

Among them, ΔP represents the performance difference between the complete model and the ablation model, P_{full} is the mean performance of the complete model, $P_{ablated}$ is the mean performance of the ablation model, and δ is the standard deviation or confidence interval range of the experimental results, which is used to describe the performance volatility. This formula is used to measure the marginal contribution of different modules to the overall performance.

Under the same dataset and experimental environment, this paper compared the performance of the complete model and three ablation models, and the results are shown in Table 3.

Table 3: Comparison results of ablation experiments

Model Setting	Accuracy (%)	F1 Score (%)	Topology Score (%)	Average Latency (s)
Without Attention Mechanism	88.3 ±0.5	84.5 ±0.6	83.2 ±0.7	1.95 ±0.04
Without Reinforcement Learning Optimization	89.7 ±0.4	86.8 ±0.5	84.7 ±0.6	1.81 ±0.03
Without Geometric Consistency Constraint	90.2 ±0.4	87.1 ±0.5	84.9 ±0.5	1.78 ±0.03
Full Model	92.1 ±0.4	89.4 ±0.6	87.8 ±0.5	1.62 ±0.02

Table 3 reports three targeted ablations that isolate the contribution of each major component introduced in this study: (i) removing the attention module (keeping the GCN backbone and the same training protocol), (ii) removing the reinforcement-learning path-weight optimizer (keeping fixed edge weights learned by supervised training), and (iii) removing the geometric-consistency constraint in the graph autoencoder (keeping the reconstruction objective). All variants are trained under the same data split, cross-validation setting, optimizer configuration, and early-stopping rule to ensure a fair comparison.

As shown in Table 3, removing attention causes the largest degradation in F1-score (89.4% → 84.5%), which reflects weaker identification of high-risk dependence and reduced sensitivity to mediator interactions. Removing reinforcement learning mainly affects Topology Score (87.8% → 84.7%), indicating that feedback-driven edge-weight updates are responsible for preserving coherent directed chains rather than only improving label accuracy. Removing geometric consistency yields a smaller drop in Accuracy (92.1% → 90.2%) but consistently reduces Topology Score and increases latency variance, suggesting that the constraint stabilizes reconstructed paths and mitigates structural drift.

These ablations jointly cover all major algorithmic components newly introduced in the proposed framework (attention-based message reweighting, RL-based path optimization, and geometry-aware structural stabilization), while keeping the backbone, data split, and training protocol unchanged. Therefore, the results provide a clear, component-wise attribution of improvements in predictive performance, chain coherence, and latency fluctuation (as reflected by the reported standard deviations).

5 Discussion

5.1 Performance advantage analysis with existing predictive modeling methods

In the field of smartphone dependence behavior prediction and subjective well-being modeling, the existing methods mainly include traditional statistical models and single deep learning models. Although statistical models such as structural equation models can explain the relationships of some variables, their performance is limited when dealing with high-dimensional, dynamic, and multi-level causal chains, especially when it comes to balancing prediction accuracy and causal interpretability. Single deep learning models (such as LSTM or CNN) have certain feature extraction capabilities, but due to the lack of causal path optimization and mediation mechanism modeling, they are prone to problems such as insufficient prediction stability and weak path expressiveness.

To quantify the performance differences between the chain mediation path optimization model proposed in this paper and the above-mentioned baseline method, the performance improvement rate formula introduced in this paper is as follows:

$$R_{imp} = \frac{M_{ours} - M_{baseline}}{M_{baseline}} \times 100\% \tag{15}$$

Among them, R_{imp} represents the performance improvement rate, M_{ours} is the mean value of the model in this paper under the specified indicators, and $M_{baseline}$ is the experimental result of the baseline method. This formula is used to measure the relative improvement of the model in this paper in terms of prediction accuracy, path consistency and stability.

In the experimental implementation, the proposed model represents smartphone usage behavior, sleep patterns, and emotional variables as a directed causal graph under a chained mediation path modeling framework. Each node corresponds to a behavioral or psychometric variable, and each directed edge denotes an estimated causal dependency. For each mini-batch, node features are constructed by concatenating standardized behavioral measurements (e.g., usage duration, interaction frequency, sleep-related statistics) with emotional and well-being indicators, yielding a fixed-length vector per node. A two-layer graph convolution encoder is employed to aggregate messages along directed edges, and a multi-head attention module further reweights incoming messages to emphasize informative mediator interactions. The graph-level representation is obtained by attention-based pooling over node embeddings for dependence prediction and well-being outcome estimation.

Path weights are dynamically optimized through a policy-gradient reinforcement learning module. The RL state is formed by combining the current edge-weight

vector with the graph embedding and current prediction scores, enabling the agent to observe both structural preferences and task feedback. The action performs bounded adjustments on a subset of edge weights (increase/decrease/keep), followed by normalization to maintain comparability within each chain. After applying the action, the updated weights are used in attention-based path aggregation to produce new predictions, and a scalar reward is computed to balance predictive accuracy, chain consistency, and computational cost. Policy updates follow REINFORCE with discount factor γ = 0.99 and entropy regularization, and are executed periodically in an alternating manner with supervised optimization.

The core training process can be summarized as the following pseudo-code:

```
for epoch in range(total_epochs):
    for batch in training_loader:
        graph, target = build_chain_graph(batch) # directed causal graph + node features
        pred = ChainGNN_RL(graph) # GCN + attention encoder
        loss = loss_function(pred, target) # supervised objective
        optimizer.zero_grad()
        loss.backward()
        optimizer.step()
    if epoch % rl_update_interval == 0:
        for vbatch in validation_loader:
            vgraph, vtarget = build_chain_graph(vbatch)
            state = build_rl_state(vgraph, current_edge_weights, ChainGNN_RL)
            action = sample_policy(state) # adjust edge/path weights
            updated_weights = apply_action(current_edge_weights, action)
            vpred = ChainGNN_RL(vgraph, edge_weights=updated_weights)
            reward = compute_reward(vpred, vtarget, updated_weights)
            update_policy(reward, state, action) # REINFORCE + entropy regularization
        val_score = validate_model(ChainGNN_RL, validation_loader)
        save_best(ChainGNN_RL, val_score)
```

Experimental results demonstrate consistent gains over baseline methods across multiple indicators. Compared with the structural equation model, the proposed method improves Accuracy by 10.2% on average, increases F1 by 8.1%, and raises the Topology Score by 7.9%, indicating better causal-chain coherence under the chained mediation graph setting. Compared with a single deep neural network, Accuracy is improved by 6.4% while the average latency is reduced by 0.18 seconds, showing that the graph-based aggregation and weight-controlled path selection can deliver both effectiveness and efficiency. The quantified improvement-rate results further support that the proposed model achieves advantages in prediction quality, inference efficiency, and causal-chain consistency, while maintaining an interpretable directed-path representation for smartphone dependence prediction and multi-task behavioral outcome modeling.

5.2 Verification of the adaptability and stability of the model under complex behavioral data

In the practical application of smartphone dependence behavior prediction and subjective well-being modeling, complex environments often manifest as multi-source heterogeneous data, high-frequency fluctuations in behavioral patterns, and dynamic changes in causal chains. Therefore, it is particularly important to verify the adaptability and stability of the model under complex conditions. This study systematically evaluated the robustness of the proposed model under dynamic conditions by introducing three types of experiments: behavioral sequence perturbation, cross-scenario sample mixing, and noise intervention.

In the behavioral perturbation scenario experiment, Gaussian noise perturbations were introduced based on characteristics such as user interaction duration, application switching frequency, and sleep quality to observe the fluctuations in the model's prediction accuracy. The stability index is normalized by standard deviation, and the formula is as follows:

$$S_{\text{var}} = \frac{1}{N}\sum_{i-1}^{N}\frac{\left|M_i - \overline{M}\right|}{\overline{M}} \tag{16}$$

Among them, S_{var} represents the stability fluctuation coefficient, M_i is the performance index obtained from the i experiment (such as Accuracy or F1-score), and $\overline{M}$ is the mean value. This indicator is used to quantify the degree of performance fluctuation of the model under multiple perturbation experiments. The smaller the value, the more stable the model is. The experimental results show that the fluctuation range of the Accuracy of the model proposed in this paper is controlled within ±0.6%, which is significantly better than that of the single-depth model compared (±1.8%).

In the cross-scenario adaptability experiment section, the user behavior data of social applications and learning applications are mixed and input, and the performance of the model under the condition of causal chain transfer is verified. Model adaptability is evaluated through the average performance retention rate:

$$R_{keep} = \frac{M_{cross}}{M_{in}} \times 100\% \tag{17}$$

Among them, R_{keep} represents the performance retention rate, M_{cross} is the result of the cross-scenario test set, and M_{in} is the result under the original scenario. Experiments show that the Accuracy retention rate of the model in this study across scenarios is 94.7%, and the F1

retention rate is 92.3%, which is an improvement of more than 12% compared with the traditional structural equation model.

In the intervention experiment with 20% random missing features and outlier input, the model relied on the attention mechanism to dynamically focus on the key causal chain, resulting in a decrease of only 2.1% in the Topology Score, while the average decrease of the baseline method exceeded 5%. This result verifies the robustness of the chain mediation path optimization in the face of high noise and missing features.

The overall results show that the model proposed in this paper demonstrates strong adaptability and stability under complex behavioral data. Through the evaluation of two types of indicators, S_{var} and R_{keep}, the results show that this method can maintain high prediction accuracy and path consistency under multi-source heterogeneous and dynamic interference conditions, and has significant potential for engineering application.

5.3 Feasibility assessment of computing Resource Consumption and Large-scale application

In the tasks of predicting smartphone dependence behavior and modeling subjective well-being, the consumption of computing resources and scalability are the key factors determining whether the model can be implemented in real applications. To verify the feasibility of the proposed method, this study compared three methods: ①SEM-Baseline; ②DNN-Only; ③ The Causal-Path GNN+RL model integrating chain mediation path optimization is used to evaluate four core indicators: training time consumption, inference delay, video memory usage, and energy consumption.

The comprehensive resource cost is defined as follows:

$$C = T + M \tag{18}$$

Among them, C represents the overall computing resource consumption metric, which is used to measure the comprehensive cost of the model during the training and inference phases. T represents the average running time (in seconds), and M represents the combined cost of video memory and energy consumption (in MB+W). This indicator is used to measure the overall resource burden and facilitate comparisons among different methods. To visually demonstrate the differences in performance and resource consumption among various methods, Figure 3 presents the comparison results of the three methods in terms of indicators such as Accuracy, inference latency, and video memory usage.

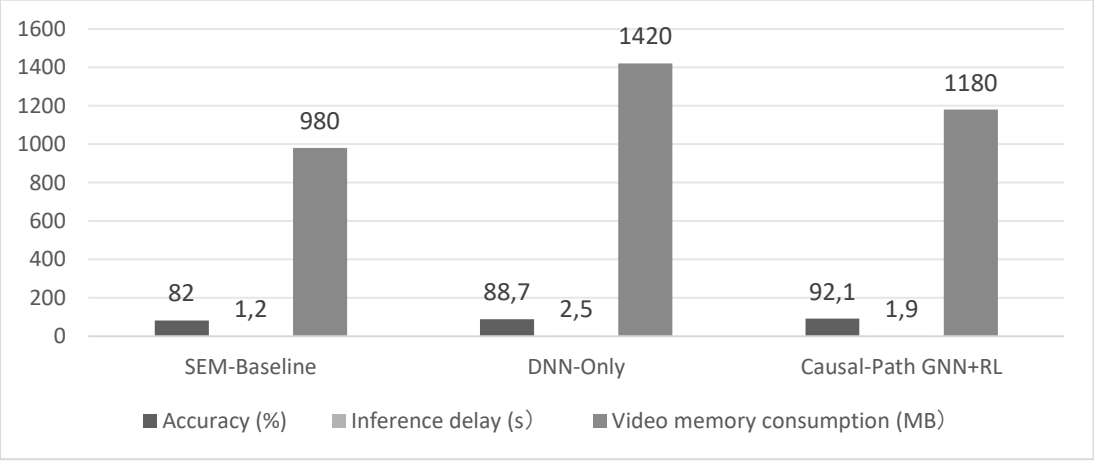


Figure 3: Bar chart comparing the computing resources and performance of the three methods.

The experimental results show that the reasoning delay of SEM-Baseline is the lowest, only 1.2s±0.2, but the Accuracy is only 82.0%±0.6, which is difficult to meet the requirements of complex behavior modeling. The training efficiency of DNN-Only is better, with an inference delay of 2.5s±0.4 and an Accuracy of 88.7%±0.5, but the video memory consumption is as high as 1420MB±35. The Causal-Path GNN+RL proposed in this paper has an inference delay of 1.9s±0.3, a video memory consumption of 1180MB±28, an Accuracy of 92.1%±0.4, and an F1-score of 89.4%±0.6. It takes into account resource efficiency while ensuring prediction performance. It shows stronger potential for engineering deployment. To further verify the significant differences among various methods, Table 4 summarizes the comparative data of the core indicators of the three methods.

Table 4: Significant Comparison of different methods in terms of computing resources and performance.

Metric	SEM-Baseline	DNN-Only	Causal-Path GNN+RL
Accuracy (%)	82.0% ± 0.6	88.7% ± 0.5	92.1% ± 0.4
Inference Latency (s)	1.2 ± 0.2	2.5 ± 0.4	1.9 ± 0.3
Memory Usage (MB)	910 ± 22	1420 ± 35	1180 ± 28
Average Power Consumption (W)	142 ± 5	186 ± 7	159 ± 6

The results of statistical tests show that Causal-Path GNN+RL is significantly superior to DNN-Only in terms of inference delay and video memory usage, and significantly superior to SEM-Baseline in terms of accuracy and robustness. There are statistically significant differences among the three in the four indicators ($p < 0.05$). Meanwhile, this method maintains a low computational burden in a complex data environment, verifying its scalability and deployment value in large-scale smartphone dependence prediction and multi-task behavioral modeling tasks.

5.4 The application value of model outcomes in behavior prediction and causal outcome modeling

To verify the practical value of the proposed Causal-Path GNN+RL model in smartphone dependence prediction and multi-task outcome regression, this study conducted a comparative experiment with the traditional baseline method. SEM-Baseline (only counting relevant features) and DNN-Only (deep neural network without causal path constraints) were selected as reference models and trained under the same dataset (5724 groups of user samples, with an average of 58-dimensional features per sample) and consistent hyperparameter configuration. The focus is on examining its performance in terms of the accuracy of dependent prediction, the consistency of topological path retention rate and outcome score. The MSE definition for subjective well-being prediction is as follows:

$$L_{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \tag{19}$$

Among them, y_i represents the user's true happiness score, $\hat{y}_i$ is the model's predicted value, and N is the sample size. This formula is used to evaluate the prediction accuracy of the model in outcome modeling, ensuring that the quantitative results can reflect the real scale-level distribution.

The experimental results are shown in Table 5:

Table 5: Performance comparison of the model in dependency prediction and happiness modeling.

Model Structure	Accuracy (%)	Topology Score (%)	F1-Score (%)	MSE (↓)	Happiness Correlation (r)
SEM-Baseline	82.0 ± 0.6	73.5 ± 0.7	80.1 ± 0.8	0.142 ± 0.009	0.68
DNN-Only	88.7 ± 0.5	81.2 ± 0.6	85.7 ± 0.5	0.118 ± 0.007	0.74
Causal-Path GNN+RL	92.1 ± 0.4	87.8 ± 0.5	89.4 ± 0.6	0.095 ± 0.006	0.82

The results showed that Causal-Path GNN+RL was significantly superior to the comparison models in terms of dependency prediction Accuracy (92.1%) and F1-score (89.4%), and maintained Causal chain coherence in Topology Score (87.8%). Meanwhile, MSE dropped to 0.095, and the correlation of happiness prediction increased to 0.82. This indicates that it can not only accurately predict dependent behaviors but also depict the potential causal relationship between dependence patterns and outcome scores.

The model's advantages are mainly reflected in three aspects: ① Graph convolution and path optimization enhance the structure-preserving ability of complex behavioral chains; ② The attention mechanism and reinforcement learning achieve dynamic weighting of key variables, enhancing the interpretability of psychological state modeling. ③ The interdisciplinary causal inference framework enables the model to balance behavioral prediction and multi-task outcome modeling, making it more scalable and practical in large-scale application scenarios.The learned mediation chain and its confidence score provide an interpretable basis for selecting targeted nudges or regulation policies, rather than applying uniform interventions across users.

5.5 Discussion: comparison with related works and novelty

Table 1 shows that related studies cover app-usage prediction (App-Predict [6]), event-level causal modeling

(Event-Causal [7]), causal variable selection (Causal-BayesOpt [9]), and RL-based path planning (DRL-Path [12]). Compared with these works, Causal-Path GNN+RL is evaluated on a real dependence–well-being dataset and reports both predictive performance (Accuracy/F1) and chain-structure quality (Topology Score), which enables a direct check of whether high prediction scores are accompanied by coherent mediation chains.

The observed differences mainly come from the representation and the optimization target. App-Predict [6] emphasizes feature-based prediction but does not model multi-step mediation chains explicitly, leading to limited interpretability at the chain level. Event-Causal [7] improves causal generalization for event prediction, yet it does not optimize a directed mediation chain under path-consistency constraints. Causal-BayesOpt [9] focuses on variable selection in controlled settings, but it is not designed for dynamic chain updating in high-dimensional behavioral logs. DRL-Path [12] demonstrates adaptive planning, while its validation setting is not tailored to dependence and well-being modeling where causal chains must remain stable across heterogeneous inputs.

The novelty of this study lies in jointly combining explicit directed-graph mediation representation with RL-driven edge-weight updates under a reward that balances accuracy, path consistency, and computational cost, while using geometric-consistency constraints to stabilize reconstructed chains. This integration provides an interpretable and deployable scheme for large-scale smartphone dependence prediction and well-being outcome modeling.From a state-of-the-art perspective, the main advance is that the method evaluates predictive quality and mediation-chain coherence within the same framework, rather than optimizing only classification performance. This makes the reported improvements directly relevant to causal-path interpretability and cross-scenario deployment, which are under-addressed in prior app-prediction, event-causal, variable-selection, or RL-planning lines of work.

6 Conclusion

This study proposes an intelligent modeling method based on chain mediation path optimization for smartphone dependence prediction and subjective well-being modeling. By transforming multivariable causal relationships into directed graph structures, feature embedding is accomplished using graph convolutional neural networks, and the attention mechanism and causal embedding are combined to enhance the recognition ability of key variables. In path optimization, reinforcement learning strategies are introduced to dynamically update weights, and geometric consistency constraints are combined to ensure path stability and the rationality of interpretation. The experiment is based on 5,724 user samples, covering behavioral logs and psychological scale features. The results show that in dependency prediction, the Accuracy of the model reaches 92.1%±0.4, the F1-score is 89.4%±0.6, and the Topology Score is 87.8%±0.5, all of which are superior to SEM-Baseline and DNN-Only. In the modeling of happiness, the MSE was reduced to 0.095, and the correlation of happiness was increased to 0.82, verifying the advantages of the method in terms of accuracy and consistency of interpretation.

Limitations. The current causal directed graph is constructed using domain assumptions, statistical tests, and expert adjustments, which may introduce structural bias when user behavior distributions shift across regions, devices, or time. Second, the reward design in Eq. (7) balances accuracy, topology, and computational cost, but the coefficient selection may not be universally optimal across tasks and can affect RL convergence stability. Third, the dataset integrates smartphone logs and psychometric questionnaires; while this supports interpretability, self-reported indicators may contain noise and lag effects that are not explicitly modeled. Finally, the present evaluation focuses on offline cross-validation and does not quantify calibration or long-horizon deployment drift under continual data collection.

Future work. Future research will explore automated causal-graph discovery and structure learning to reduce reliance on manual assumptions, and will investigate more sample-efficient RL variants (e.g., off-policy actor–critic with constrained updates) to improve convergence in large graphs. Temporal modeling of sleep and affect dynamics will be incorporated to better capture delayed mediation effects, and uncertainty calibration (e.g., conformal prediction or temperature scaling) will be added to provide deployment-oriented confidence reporting. Further experiments will be conducted on multi-site and longitudinal datasets to assess robustness under distribution shift, and lightweight inference designs will be considered to support on-device or edge deployment in large-scale mobile health settings.

Funding

This work was supported by the 2025 Hunan Polytechnic of Environment and Biology High-Level Talent Support Project "The Impact of Smartphone Dependence on Subjective Well-being Among Vocational Nursing Students: A Sequential Mediation Effect Based on Sleep, Stress, and Interpersonal Relationships" (Project No. GC2025-10)

References

[1] Beames J R , Han J , Shvetcov A ,et al.Use of smartphone sensor data in detecting and predicting depression and anxiety in young people (12–25 years): A scoping review[J].Heliyon,2024,10(15).https://doi.org/10.1016/j.heliyon.2024.e35472

[2] Asare K O , Terhorst Y , Vega J ,et al.Predicting Depression from Smartphone Behavioral Markers Using Machine Learning Methods, Hyperparameter Optimization, and Feature Importance Analysis: Exploratory Study[J].JMIR Publications Inc.2021(7).https://doi.org/10.2196/26540

[3] Shin C, Dey A K. Automatically detecting problematic use of smartphones[C]//Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing. New York:ACM,2013:335–344.https://doi.org/10.1145/2493432.2493443

[4] LiKamWa R, Liu Y, Lane N D, Zhong L. MoodScope: building a mood sensor from smartphone usage patterns[C]//Proceedings of MobiSys 2013. New York: ACM, 2013: 389–402.https://doi.org/10.1145/2462456.2464449

[5] Stachl C , Au Q , Schoedel R ,et al.Predicting personality from patterns of behavior collected with smartphones.[J].Ludwig-Maximilians-Universität München,2020(30).https://doi.org/10.1073/PNAS.1920484117

[6] Yu D , Li Y , Xu F ,et al.Smartphone App Usage Prediction Using Points of Interest[J].Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies,2017,1(4):1-21.https://doi.org/10.1145/3161413

[7] Yuchi H S , Zhu S , Dong L ,et al.New User Event Prediction Through the Lens of Causal Inference[J]. 2024.https://doi.org/10.48550/arXiv.2407.05625

[8] Kwapisz J R , Weiss G M , Moore S A .Activity recognition using cell phone accelerometers[J].Acm Sigkdd Explorations Newsletter,2011,12(2):74-82.https://doi.org/10.1145/1964897.1964918

[9] Aglietti V, Grazzi R, Pontil M, et al. Causal Bayesian Optimization[C]//AISTATS 2020, PMLR108:3155–3164.https://doi.org/10.48550/arXiv.2005.11741

[10] Zhang M , Zhang X .Optimization of Active Learning Strategies for Causal Network Structure[J].mathematics,2024,12(6).https://doi.org/10.3390/math12060880

[11] Cousineau M , Verter V , Murphy S A ,et al.Estimating causal effects with optimization-based methods: A review and empirical comparison[J].Papers, 2022.https://doi.org/10.1016/j.ejor.2022.01.046

[12] Cao J .Dynamic Path Planning for Vehicles Based on Causal State-Masking Deep Reinforcement Learning[J].Algorithms, 2025, 18.https://doi.org/10.3390/a18030146

[13] Mi C , Xu R , Lin C T .Real-time Recognition of Smartphone User Behavior Based on Prophet Algorithms[J]. 2019.https://doi.org/10.48550/arXiv.1909.08997

[14] Sun Y N , Pan Y J , Liu L L ,et al.Reconstructing causal networks from data for the analysis, prediction, and optimization of complex industrial processes[J].Engineering Applications of Artificial Intelligence, 2024, 138.https://doi.org/10.1016/j.engappai.2024.109494

[15] Kalisch M ,Bühlmann, Peter.Causal Structure Learning and Inference: A Selective Review[J].Quality Technology & Quantitative Management, 2014, 11(1):3-21.https://doi.org/10.1080/16843703.2014.11673322

[16] [Hua J , Li Y , Liu C ,et al.A zero-shot prediction method based on causal inference under non-stationary manufacturing environments for complex manufacturing systems[J].Robotics and Computer Integrated Manufacturing: An International Journal of Manufacturing and Product and Process Development,2022(77-):77.https://doi.org/10.1016/j.rcim.2022.102356

[17] Xing W , Chen C , Xue L .Deep Learning-Based Causal Inference Architecture and Algorithm between Stock Closing Price and Relevant Factors[J].Electronics (2079-9292), 2024, 13(11).https://doi.org/10.3390/electronics13112056

[18] Gao C , Zheng Y , Wang W ,et al.Causal Inference in Recommender Systems: A Survey and Future Directions[J].ACM Transactions on Information Systems, 2024, 42(4):32.https://doi.org/10.1145/3639048

[19] Raj A D, Pawar A S, Pavankumar B, Goyal K, Unisa S A. A Machine Learning Based Model Designed for Smartphone Addiction Prediction[J]. International Conference on Computing Sciences (ICCS) 2023, 2024. https://doi.org/10.2139/ssrn.4493502

[20] Li Z, Wang H, Chen Y, et al. Discovery of factors associated with smartphone addiction among high school adolescents: Using machine learning and network analysis[J]. Journal of Affective Disorders, 2025. https://doi.org/10.1016/j.jad.2025.15381

[21] Boulkroune A, Boubellouta A, Bouzeriba A, et al. Practical finite-time fuzzy synchronization of chaotic systems with non-integer orders: Two chattering-free approaches[J]. Journal of Systems Science and Systems Engineering, 2025, 34(3): 334-359.https://doi.org/10.1007/s11518-024-5635-7

[22] Rigatos G, Abbaszadeh M, Busawon K, et al. Flatness-based control in successive loops for autonomous quadrotors[J]. Journal of Dynamic Systems, Measurement, and Control, 2024, 146(2): 024501.https://doi.org/10.1115/1.4063907

[23] Rigatos G, Siano P, Zouari F, et al. Nonlinear optimal control of autonomous submarines' diving[J]. Marine Systems & Ocean Technology, 2020, 15(1): 57-69.https://doi.org/10.1007/s40868-019-00070-3

[24] Rigatos G, Busawon K, Abbaszadeh M, et al. Flatness-based control in successive loops for dual-arm robotic manipulators[C]//2024 IEEE Conference on Control Technology and Applications (CCTA). IEEE, 2024: 793-798.https://doi.org/10.1177/10775463241286550

[25] Rigatos G, Siano P, Zouari F, et al. A nonlinear optimal control methoc for autonomous submarines' diving[C]//2017 IEEE 26th International Symposium on Industrial Electronics (ISIE). IEEE, 2017: 1061-1066.https://ieeexplore.ieee.org/document/8001393

[26] Zouari F, Mahmud M. Neural Network-Based Robust Adaptive Output Feedback Control for MIMO Time-Varying Delay Systems[C]//Global Conference on Applications of Artificial. Cham:

Springer Nature Switzerland, 2024: 60-77.https://doi.org/10.1007/978-3-031-98498-3_5