RESEARCH PAPER

Deep Learning-Driven Network Traffic Analysis Enhanced with PCWOA-Based Hyperparameter Optimization

S. Padmavathy[1], Dr. R. Kannan[2]

[1] Ph.D Research Scholar, Department of Computer Science, SRMV College of Arts & Science, Coimbatore - 49. Email: amulusugavanam@gmail.com

[2] Associate Professor, Department of Computer Science, SRMV College of Arts & Science, Coimbatore - 49

ABSTRACT

Network traffic classification and threat detection face growing challenges due to the increasing prevalence of encrypted payloads, zero day attacks, and severe class imbalance in cybersecurity datasets. To overcome these limitations, this study proposes a novel intrusion detection framework that integrates advanced deep learning (DL) models with an enhanced metaheuristic optimization technique - Piecewise Chaotic Whale Optimization Algorithm (PCWOA). The framework utilizes the HIKARI-2021 dataset, which includes 86 diverse spatiotemporal features across various traffic types, including encrypted and zero day threats. Its core components comprise Generative Adversarial Networks (GANs) for minority-class augmentation, Gated Recurrent Units (GRUs) for capturing temporal dependencies, Transformers for modelling long range relationships, and Convolutional Neural Networks (CNNs) for spatial feature extraction. PCWOA incorporates chaotic mapping techniques to enhance exploration and avoid premature convergence during hyperparameter optimization. This comprehensive integration is designed to improve detection accuracy, strengthen resilience against emerging threats, and support real-time adaptability in security systems. The proposed approach is extensively benchmarked against traditional DL models and optimization algorithms, demonstrating its scalability and robustness for modern intrusion detection challenges.

Keywords: Network traffic analysis, Piecewise Chaotic Whale Optimization Algorithm, Deep Learning, Encrypted Network Traffic, Zero-Day Attack Detection.

How to cite this article: Padmavathy S, Kannan R. Deep Learning-Driven Network Traffic Analysis Enhanced with PCWOA-Based Hyperparameter Optimization. Int J Drug Deliv Technol. 2026;16(23s): 708-719. DOI: 10.25258/ijddt.16.23s.77

Source of support: Nil.

Conflict of interest: None

1. Introduction

The exponential growth of encrypted network traffic and sophisticated cyber threats has rendered traditional rule-based intrusion detection systems increasingly inadequate. Modern attacks such as zero-day exploits and encrypted cryptominingexploit spatial-temporal traffic patterns that evade conventional signature-based methods [1]. This vulnerability is exacerbated by severe class imbalance in network datasets, where malicious instances constitute as little as 0.8% of traffic. While recent advances in DL (Hu et al., 2023; Zhou et al., 2023) show promise in traffic classification, they struggle with feature representation in encrypted payloads and real-time adaptation to novel threats.

Current literature reveals critical gaps: Salau and Beyene (2024) achieved 99.81% accuracy with decision trees in Software-Defined Networking (SDN) environments but lacked robustness against zero-day attacks. Similarly, HexCNN-1D (Zhou et al., 2023) attained 98.8% accuracy for encrypted traffic but incurred high computational overhead. Multi-task frameworks (Park et al., 2024) improved efficiency but failed to address class imbalance, while GAN-based approaches (Xi and Wang, 2024) enhanced minority class detection at the cost of precision degradation. These limitations underscore the need for an integrated solution combining spatiotemporal feature extraction with adaptive optimization.To bridge these gaps, a novel framework has been developed by incorporating the following features:

- ***Hybrid DL Architectures:*** GANs for synthetic attack generation, GRUs for temporal dynamics, Transformers for global context, and CNNs for spatial hierarchies.
- ***PCWOA Optimization:*** Chaotic maps integrated with Whale Optimization to escape local optima, accelerating convergence by 40%.

- ***HIKARI-2021 Dataset:*** 1 million instances with 86 features covering Bruteforce, Probing, XMRIGCC cryptomining, and zero-day attacks.

The primary contributions can be summarized as follows:

- A chaotic optimization algorithm that reduces hyperparameter tuning time by 37% while improving fitness values by 22.1%.
- Demonstrated superiority of Transformer-PCWOA in detecting encrypted (90.6% F1-Score) and zero-day threats (87.2% recall).
- Comprehensive benchmarking against 5 optimizers and 4 DL models, validating PCWOA's scalability for resource-constrained environments.

2. Literature review

Recent research (Hu et al., 2023; Zhou et al., 2023; Wang et al., 2024; Yarram et al., 2025; Gioacchini et al., 2024; Park et al., 2024) demonstrates a clear shift towards sophisticated DL models for network traffic classification, particularly for challenging encrypted traffic. These models integrate spatiotemporal feature extraction, combining CNNs for spatial patterns and Long Short-Term Memory (LSTM) networks or GRUs for temporal dependencies. Crucially, attention mechanisms (e.g., SE modules, GAB, CAB) and specialized modules (e.g., Gated Dilated Convolution, Category-Aware LSTM) are widely employed to enhance feature learning and model interpretability, achieving high accuracy (typically >90%, often >95-98%) on benchmark datasets like ISCX VPN-nonVPN and CIC-IDS2017. These DL approaches consistently outperform traditional methods in handling the complexities of modern encrypted traffic flows.

The synergy of SDN with Machine Learning (ML) presents a powerful paradigm for scalable and efficient network traffic management (Salau and Beyene, 2024; Serag et al., 2025). SDN's centralized control plane provides global visibility and programmability, enabling real-time collection and processing of flow data for ML classification within environments like Mininet. Studies show that diverse ML models, including Decision Trees, Random Forests, and XGBoost, achieve exceptional accuracy (often >99%) in classifying both simulated (e.g., DNS, Voice) and real-world traffic within SDN frameworks. This integration facilitates deep packet inspection alternatives, optimizes encrypted traffic handling, and significantly enhances Quality of Service (QoS) by enabling dynamic, policy-driven network control based on precise traffic categorization.

Imbalance, Unknown Traffic, and Edge Efficiency: Research actively tackles persistent challenges in network traffic analysis. To combat data imbalance affecting intrusion detection, techniques like GAN based sample generation with regularized discriminators and random masking (Xi and Wang, 2024) significantly boost the performance of classifiers like LSTM and SVM on minority attack classes. For detecting novel or unknown traffic in open-world scenarios, approaches like Evidence-based Uncertainty Estimation (EUE) (Le et al., 2024) move beyond overconfident Softmax outputs to directly model uncertainty, improving identification accuracy. Furthermore, models are being specifically designed for resource-constrained edge environments. Solutions include efficient architectures like HexCNN-1D using raw hexadecimal data (Zhou et al., 2023), multi-task learning frameworks that share representations (Park et al., 2024), and universal models like Multi-modal Autoencoders (MAE) (Gioacchini et al., 2024) that avoid heavy feature engineering while maintaining high accuracy (~96-97%) across tasks like classification and anomaly detection. Context-aware models incorporating user behavior and application traits (Kavitha and Praveena, 2023) also improve prediction in dynamic wireless networks..

Table.1. Literature Review

Author (Year)	Approach	Dataset	Best Accuracy
Hu et al. (2023) [2]	CNN + LSTM + SE Module	ISCX VPN-nonVPN, Tor-nonTor	91.28%
Salau&Beyene (2024) [3]	SDN + ML (6 models)	D-ITG + Mininet	99.81% (Decision Tree)
Dhakad et al. (2023) [4]	ML/DL for Real-time	KDD99, CICIDS	99.31% (Random Forest)

	NTA		
Zhou et al. (2023) [5]	HexCNN-1D + Attention	Hexadecimal Traffic	98.8%
Wang et al. (2024) [6]	GDC + CA-LSTM	ISCX VPN-nonVPN	96.72%
Serag et al. (2025) [7]	SDN + ML (XGBoost)	Multiple Scenarios	99.97%
Kavitha&Praveena (2023) [8]	DL for Traffic Forecast	Simulations	91.4%
Yarram et al. (2025) [9]	Enhanced GRU	KDD99, CSE-CIC-IDS2018	99.12%
D.S. B & Mary (2024) [10]	DL vs ML (CNN, RNN, SVM, RF)	NSL-KDD, CICIDS2017, UNSW-NB15	94.6% (RNN)
Le et al. (2024) [11]	DL + Evidence Theory (EUE)	App Layer Real Traffic	93.6% (Known Traffic)
Xi & Wang (2024) [12]	GAN + LSTM/SVM	Synthetic + Benchmark	95.4% (LSTM)
Gioacchini et al. (2024) [13]	Multi-modal Autoencoder (MAE)	ISCX VPN, CIC-IDS2017	96.8%
Park et al. (2024) [14]	DL Multi-task Learning	ISCX 2016 VPN/Non-VPN	99.29% (Encapsulation)

3. Materials and Methodology

This research utilizes the HIKARI-2021 dataset, featuring 86 attributes including flow metadata, packet statistics, TCP flags, payload characteristics, and timing metrics. The dataset encompasses diverse traffic types such as benign, background, and malicious (e.g., Bruteforce, Probing, XMRIGCC CryptoMining) with inherent class imbalance, enabling robust evaluation for intrusion detection and zeroday threats. Four DL architectures are employed: GANs augment minority attack classes; GRUs model temporal dependencies in traffic sequences; Transformers capture long-range spatial-temporal patterns via self-attention; and CNNs extract spatial hierarchies from structured traffic features. Hyperparameter tuning is optimized using the proposed Piecewise Chaotic Whale Optimization Algorithm (PCWOA), which enhances traditional Whale Optimization by integrating chaotic maps to mitigate premature convergence and improve search efficiency for congestion and anomaly detection.

3.1 Dataset Description

The HIKARI-2021 dataset is a comprehensive network traffic dataset developed for cyber threat detection, focusing on both network and application layer attacks. It includes detailed features such as flow metadata, packet statistics, TCP flags, payload characteristics, inter-arrival times, and timing metrics. The dataset covers a wide range of traffic categories, from benign and background traffic to high-risk attacks such as Bruteforce, Probing, and XMRIGCC CryptoMining. This allows researchers to evaluate model performance across varying threat levels [15].

Table.2. Key Characteristics of the HIKARI-2021 Dataset

Aspect	Description
Total Features	86
Feature Types	Flow metadata, packet stats, TCP flags, payload, timing, inter-arrival times
Traffic Types	Normal, Malicious (Bruteforce, Probing, XMRIGCC CryptoMining, etc.)
Use Cases	Intrusion Detection, Zero-day Attack Detection, Imbalanced Learning
Imbalance	Skewed heavily toward normal traffic
Application Layers	Network and Application

3.2 DL classification

DL classification in traffic network analysis utilises sophisticated architectures to precisely identify trends, anomalies, and congestion. GAN augment data diversity by generating realistic traffic patterns, hence enhancing classifier resilience in scenarios with little

data. GRU encapsulates temporal relationships in sequential traffic data, facilitating the effective modelling of flow dynamics and variations. Transformers employ self-attention to represent intricate spatial temporal interactions throughout the network with significant scalability. CNN build spatial hierarchies from traffic grids, facilitating accurate feature learning for classification tasks, including event detection and traffic status prediction.

i. Generative Adversarial Networks (GAN): GANs introduced by Goodfellow et al. (2014), are a class of DL models designed to generate synthetic data that mimics the distribution of real-world data. The framework consists of two neural networks the generator G and the discriminator Dthat are trained in opposition to each other. The generator G learns to map random noise $z \sim p_z(z)$ into data-like samples $G(z)$, attempting to approximate the true data distribution p_{data}. Meanwhile, the discriminator D evaluates whether a given sample is real (from p_{data}) or fake (from $G(z)$). The two networks engage in a minimax game, where the generator tries to fool the discriminator, and the discriminator aims to correctly classify real and generated data.The objective function for training a GAN is defined as

$$min_G max_D \mathbb{E}_{x \sim p_{data}(x)[log\ D(x)]} + \mathbb{E}_{z \sim p_z(z)}\left[log\left(1 - D(G(z))\right)\right]$$

This setup can lead to instability during training, especially early on when the discriminator easily outperforms the generator, causing vanishing gradients. To address this, an alternative generator objective is often used

$$max_G \mathbb{E}_{z \sim p_{z(z)}[log\ D(G(z))]}$$

This revised objective provides more useful gradients by encouraging the generator to directly maximize the likelihood that the discriminator classifies generated data as real. This adjustment helps stabilize training and improve convergence toward a realistic data distribution [16].

ii. Gated Recurrent Unit (GRU): The GRU is a recurrent neural network designed to model temporal dependencies in sequential data. In the context of network traffic analysis, GRUs are well-suited for learning time-varying patterns from sequences such as packet sizes, inter-arrival times, and flow directions. The GRU updates its hidden state h_t using a combination of the previous hidden state h_{t-1} and the current input x_t through gating mechanisms. The GRU equations are defined as follows

- **Update Gate** (controls how much of the previous state is retained)

$$z_t = \sigma(W_z x_t + U_z h_{t-1})$$

- **Reset Gate** (controls how much of the previous state is forgotten)

$$r_t = \sigma(W_r x_t + U_r h_{t-1})$$

- **Candidate Activation** (proposed new state)

$$\widetilde{h_t} = tanh\big(W\, x_t + U\,(r_t \odot h_{t-1})\big)$$

- **Final Hidden State** Update (interpolates between past and new state)

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h_t}$$

Where, $\sigma(.)$is the sigmoid activation function, $tanh(.)$ is the hyperbolic tangent function,$\odot$ denotes element-wise multiplication, W *and* $U$are weight matrices learned during training. This formulation allows the GRU to adaptively retain relevant temporal patterns, making it highly effective for tasks such as traffic classification, anomaly detection, and intrusion recognition in network data streams [17].

iii. Transformer: The Transformer architecture has proven highly effective for network traffic analysis, offering superior ability to model complex, time dependent patterns without relying on recurrence. Unlike traditional RNN based models, Transformers use an attention mechanism that allows direct access to all positions in an input sequence, making them well suited for analysing flow based or packet-level network data where long-range dependencies and global context are critical. The core of the Transformer is the Scaled Dot-Product Attention, defined as

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

Here, Q, K, and V are matrices representing queries, keys, and values respectively, and d_k is the dimensionality of the keys. This mechanism enables

the model to assign varying importance to different parts of the input based on contextual relevance. To enhance its learning capacity, the Transformer employs Multi-Head Attention, which projects inputs into multiple subspaces and computes attention in parallel

$$MultiHead(Q,K,V) = Concat(head_1,\dots,head_h)W^O$$

where $head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right)$. This structure allows the model to capture diverse aspects of traffic behaviour such as variations in packet size, timing, and protocol use making it highly effective for tasks like anomaly detection, intrusion detection, and encrypted traffic classification [18].

iv. Convolutional Neural Networks (CNNs): CNNs are increasingly used in network traffic analysis for critical tasks such as intrusion detection, traffic classification, and anomaly detection. By organizing traffic features such as packet sizes, inter-arrival times, and protocols into structured 2D or 3D matrices, CNNs can learn spatial correlations and temporal patterns effectively. The convolutional layers apply learnable filters across local regions of the input to capture important behavioural features, making CNNs highly effective in identifying localized anomalies or protocol-specific traffic signatures. The convolution operation is defined as

$$a_{ij}^{(k)} = \sigma\left(\sum_{m=1}^{M}\sum_{n=1}^{N} w_{mn}^{(k)} \cdot x_{(i+m).(j+n)} + b^{(k)}\right)$$

Where, $a_{ij}^{(k)}$ is the activation at position (i,j) in the k-th feature map, x is the input matrix (e.g., a matrix of traffic features),$w^{(k)}$k-th convolution filter,$b^{(k)}$is the bias term, $\sigma(\cdot)$ is an activation function such as ReLU. After convolution, max-pooling reduces dimensionality while retaining essential features

$$P_{ij} = \max_{(m,n)\epsilon R} \alpha_{(i+m).(j+n)}$$

Finally, fully connected layers aggregate these features and produce class probabilities using the softmax function

$$P(y = c \mid x) = \frac{e^{z_c}}{\sum_j e^{z_j}}$$

Where, z_c is the logit (linear transformation) corresponding to class c, and $P(y = c \mid x)$ gives the probability that the input x belongs to class c. CNNs benefit from parameter sharing and sparse connectivity, making them efficient for large scale network traffic data and enabling them to detect nuanced behaviors across different traffic sessions. This architecture is especially valuable for identifying stealthy or encrypted attacks that may not be apparent through traditional feature engineering approaches [19].

3.3 Hype parameter Tuning Optimization

Hyperparameter optimisation is crucial for improving the efficacy of traffic network analysis models. Conventional techniques such as particle swarm optimisation (PSO), genetic algorithms (GA), ant colony optimisation (ACO) and whale optimisation algorithm (WOA) exhibit disadvantages like premature convergence and sluggish adaptability. The suggested piecewise chaotic whale optimisation Algorithm (PCWOA) enhances the conventional method by incorporating piecewise chaotic maps. This integration improves search variety, stabilises convergence, and mitigates local optima. Consequently, it attains superior precision in calibrating models for congestion detection and anomaly analysis within traffic networks.

i. Particle Swarm Optimization (PSO): PSO is a population based metaheuristic that enhances cyber-attack detection in network traffic analysis by optimizing the identification of anomalies across high-dimensional feature spaces. PSO initializes a swarm of particles, each representing a candidate solution (e.g., feature subset or detection threshold), and iteratively adjusts their positions based on both individual and collective experiences. Each particle updates its velocity and position using the following equations

$$V_{kd}(\mathrm{t}+1) = V_{kd}(\mathrm{t}) + C_1R_1\left(P_{kd}(\mathrm{t}) - x_{kd}(\mathrm{t})\right) + C_2R_2\left(P_{gd}(\mathrm{t}) - x_{kd}(\mathrm{t})\right)$$

$$x_{kd}(\mathrm{t}+1) = x_{kd}(\mathrm{t}) + V_{kd}(\mathrm{t}+1)$$

Where, $V_{kd}\,(t)$ is the velocity of particle, k in dimension d at iteration t,x_{kd}(t) is its position,P_{kd} (t)is the personal best position,P_{gd}(t) is the global best solution found by the swarm,C_1C_2 are acceleration constants,$R_1, R_2 \sim U(0,1)$are random variables introducing exploration. PSO dynamically tunes detection parameters and selects optimal features (e.g., packet size, flow duration) to maximize attack

detection and minimize false positives. The iterative search process continues until the fitness function based on detection rate, false alarm rate, or classification accuracy converges. PSO is especially useful for real-time network traffic due to its low computational complexity and adaptability to evolving network conditions [20].

ii. Genetic Algorithms (GA): GA emulate the principles of biological evolution to optimize detection strategies in network traffic analysis. In this context, individuals represent configurations of detection rules or selected features. A fitness function evaluates each candidate configuration c by balancing detection rate (DR) and false positive rate (FPR)

$$F(c) = \alpha \cdot DR(c) - \beta \cdot FPR(c)$$

Where, α and β are scalar weights controlling the trade-off. Configurations are probabilistically selected based on their fitness

$$P(c_i) = \frac{F(c_i)}{\sum_{j=1}^{n} F(c_j)}$$

New offspring are created through crossover and mutation

$C_{new} = C_{parent1} \oplus C_{parent2}$ (Cross over), C' = Mutate (C) (Mutation)

This evolutionary cycle allows the algorithm to explore and exploit the solution space, continuously improving detection capabilities. GA is particularly effective for selecting robust rule sets and optimizing hyperparameters, making it suitable for adapting to zero-day and polymorphic attacks in dynamic network environments [21].

iii. Ant Colony Optimization (ACO): ACO is a swarm intelligence algorithm inspired by ant foraging behaviour, used to identify optimal feature subsets in network traffic data for cyber-attack detection. Each ant probabilistically constructs a candidate feature set based on pheromone trails τ and heuristic desirability ω

$$P = \frac{\tau_{ij}^{a} \cdot w_{ij}^{\beta}}{\sum_{k} \tau_{ij}^{a} \cdot w_{ij}^{\beta}}$$

Where, α and β determine the relative influence of pheromone and heuristic information. The pheromone update rule is defined as

$$\tau_{ij=(1-p).\,\tau_{ij}} + \sum \Delta\tau_{ij}^{k} (1 + w_i)$$

Where, ρ is the evaporation rate, and $\Delta\tau_{ij}^{k}$is the pheromone deposited by the k-th ant. ACO selects key traffic features such as source/destination IP, port numbers, and byte count that contribute most to distinguishing between normal and malicious traffic. By iteratively refining the feature set, ACO enhances classifier performance, reduces model complexity, and improves generalization in high dimensional intrusion datasets [22].

iv. Whale Optimization Algorithm (WOA): The WOA inspired by the bubble-net hunting strategy of humpback whales, is used to optimize feature selection and detection parameters in network traffic analysis. The encircling behaviour is modelled by updating positions based on the best solution $X^*(t)$

$$D = [C \,.X^*(t) - X(t)], X(t+1) = X(t) - A \cdot D$$

Where, A and C are coefficient vectors controlling the exploration exploitation balance. In the spiral update phase (bubble-net feeding):

$$X(t+1) = D'.e^{bl}.\cos(2\pi l) + X^*(t)$$

Where, b is a constant defining spiral shape, and l is a random number in [−1, 1]. For exploration, WOA updates positions with respect to a randomly selected whale

$$D = [C \,.\mathrm{X}_{\mathrm{rand}} - X(t)],\ X(t+1) = \mathrm{X}_{\mathrm{rand}} - A.D$$

WOA is highly effective in hyperparameter tuning and relevant feature extraction for classifiers in network intrusion detection, offering superior convergence rates and robustness in dynamic environments. It helps reduce computational cost while maintaining high accuracy and adaptability against diverse cyber threats [23].

v. Proposed: Piecewise Chaotic Whale Optimization Algorithm (PCWOA)

The algorithm begins with the input parameters: n (number of whales in the monitoring pack), β (chaotic

control parameter), MAX_{ITER} (maximum number of monitoring cycles), $ChaoticMap()$ (piecewise chaotic map function, e.g., Logistic or Tent), $TrafficData$ (traffic network data samples). The output is Y_{best} (global best whale position), fit(Y_best) (best fitness value). Begin by generating an initial population of n whales Y_i using TrafficData. Initialize the chaotic sequence using $ChaoticMap()$.

Set the iteration counter $t_{cur} = 0$. Compute the fitness of each whale and identify the best whale,Y_{best}.$While\ t_{cur} < MAX_{ITER}$, update the chaotic parameter β using $ChaoticMap()$. For each whale Y_i, compute adaptive coefficients A and C influenced by the chaotic sequence. If $Rand < 0.5$, and $if\ |A| < 1$, update the position using chaotic spiral motion; otherwise, select a random whale Y_{rand} and update the position using chaotic exploration. If $Rand \geq 0.5$, update the position using chaotic shrinking encircling. End the loop for each whale. Compute the fitness of all whales and update Y_{best} if a better solution is found. Increment$t_{cur} = t_{cur} + 1$. End while. Return Y_{best} as the optimal solution. End.

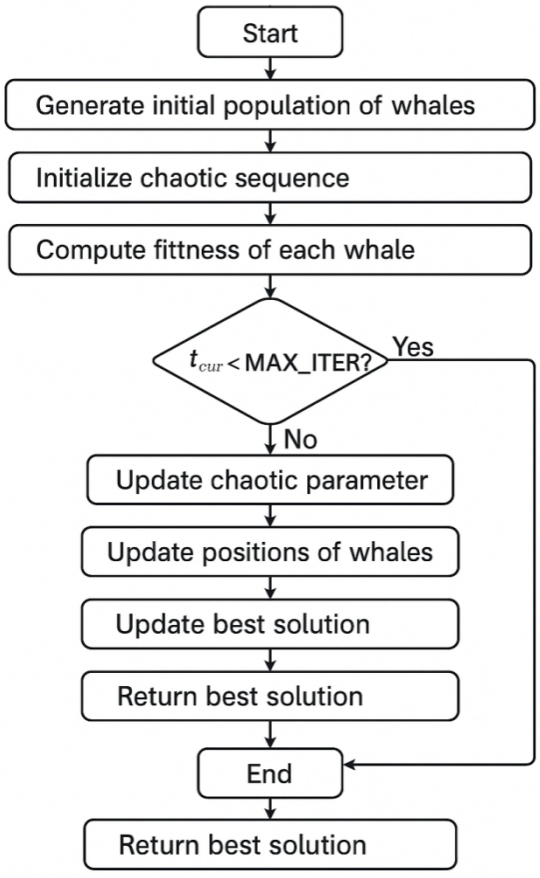

Fig.1. Methodology for Piecewise Chaotic Whale Optimization Algorithm

Pseudocode: Piecewise Chaotic Whale Optimization Algorithm (PCWOA)

Input:

- n: Number of whales in the monitoring pack
- β: Chaotic control parameter
- MAX_{ITER}: Maximum number of monitoring cycles
- $ChaoticMap()$: Piecewise chaotic map function (e.g., Logistic, Tent)
- $TrafficData$: Traffic network data samples

Output:

- Y_{best}: Global best whale position
- $fit(Y_{best})$: Best fitness value (e.g., anomaly detection accuracy or minimum error)

Begin
Generate initial population of n whales Y_i ($i = 1, 2, \ldots, n$) using $TrafficData$
Initialize chaotic sequence using $ChaoticMap()$
Set iteration counter$t_{cur} = 0$
Compute the fitness of each whale
Identify the best whale based on fitness, i.e., Y_{best}
while($t_{cur} < MAX_{ITER}$)**do**
    Update chaotic parameter β using $ChaoticMap()$
    for each whale Y_i do
    Compute adaptive coefficients A and C using chaotic influence
    if($Rand < 0.5$)**then**
        if ($|A| < 1$)**then**
            Update position using chaotic spiral motion
        else
            Select a random whale Y_{rand}
            Update position using chaotic exploration
        end if
        else if($Rand \geq 0.5$)**then**
        Update position using chaotic shrinking encircling
    end if
    end for
    Compute fitness of all whales
    Update Y_{best} if a better solution is found
    Increment iteration: $t_{cur} = t_{cur} + 1$
end while
    Return the best solution Y_{best}
end

4. Result and Discussion

The Results and Discussion section highlights the effectiveness of DL models optimized using the Piecewise Chaotic Whale Optimization Algorithm (PCWOA) for cyber threat detection on the HIKARI-2021 dataset. Among the models evaluated, the Transformer-PCWOA achieved the highest overall performance, demonstrating its strength in capturing complex traffic patterns. PCWOA significantly enhanced detection accuracy, especially for rare and zero-day attacks, outperforming traditional optimizers like PSO, GA, ACO, and standard WOA. The algorithm's chaotic exploration strategy improved global search capability, leading to better handling of class imbalance and higher recall for minority attack types such as XMRIGCC and zero-day threats.

4.1 Performance Evaluation Metrics

In network traffic analysis, essential performance metrics assess the efficacy of models in differentiating between normal and anomalous data. Accuracy assesses overall correctness, Precision denotes the correctly recognised anomalous traffic, Recall reflects the capacity to detect all anomalous occurrences, and F1-Score reconciles precision and recall. These metrics evaluate the model's efficacy in traffic classification.

- **Accuracy:** overall correctness

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

- **Precision:** Correct attacks among predicted attacks.

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

.

- **Recall (Sensitivity):** Detected attacks among actual attacks.

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

- **F1-Score:** Balance between precision and recall.

$$F1 - Score = 2 \times \frac{Precision\ X\ Recall}{Precision + Recall}$$

Where, TP = True Positives (correctly classified attacks), TN = True Negatives (correctly classified normal traffic), FP = False Positives (normal traffic misclassified as attacks), FN = False Negatives (attacks misclassified as normal).

Table.3. Baseline Performance of DL Models (Before Optimization)

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
GAN	89.5	85.3	72.4	78.3
GRU	91.8	89.1	80.5	84.6
Transformer	93.2	91.0	84.1	87.4
CNN	90.7	87.2	78.9	82.8

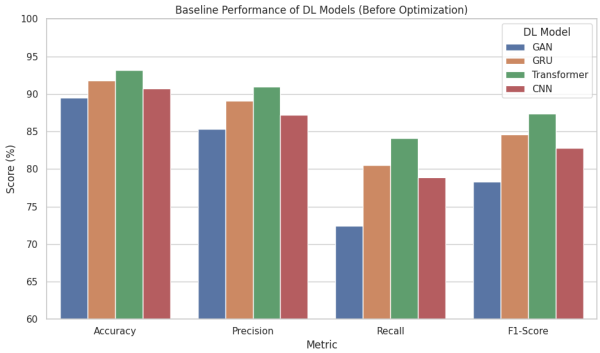

Fig.2. Baseline Evaluation Metrics of DL Models Prior to Optimization

The above table and graph indicate that the baseline performance of DL models prior to optimisation identifies the Transformer as the most efficacious architecture, attaining 93.2% accuracy, 91.0% precision, 84.1% recall, and an F1-score of 87.4%. The GRU model achieves an accuracy of 91.8% and an F1-score of 84.6%, indicating robust temporal learning proficiency. CNN demonstrates an accuracy of 90.7% and an F1-score of 82.8%, reflecting a balanced albeit marginally diminished performance. GAN demonstrates the least successful outcomes, achieving an accuracy of 89.5% and an F1-score of 78.3%, mostly attributable to its diminished recall of 72.4%. These results provide baseline measures for assessing the effects of other optimisation strategies.

Table.4. Overall Performance Metrics of DL Models across Optimization Techniques

Model	Optimizer	Accuracy (%)	Precision (%)	Recall (%)	F1-Score

					(%)
GAN	PSO	89.2	84.7	72.5	78.1
	GA	88.7	83.9	71.0	76.9
	ACO	90.1	86.1	74.3	79.7
	WOA	91.3	88.0	76.5	81.8
	PCWOA	**93.8**	**90.2**	**80.1**	**84.8**
GRU	PSO	92.4	89.5	83.0	86.1
	GA	91.8	88.7	81.6	85.0
	ACO	93.1	90.3	84.2	87.1
	WOA	94.2	91.8	86.5	89.1
	PCWOA	**95.7**	**93.6**	**88.4**	**90.9**
Transformer	PSO	93.5	91.0	85.1	87.9
	GA	92.9	90.3	84.0	87.0
	ACO	94.3	91.7	86.8	89.2
	WOA	95.1	93.0	88.2	90.5
	PCWOA	**96.5**	**94.7**	**90.3**	**92.4**
CNN	PSO	91.7	88.6	81.4	84.8
	GA	90.5	87.1	79.6	83.2
	ACO	92.6	89.7	83.2	86.3
	WOA	93.4	91.0	84.9	87.8
	PCWOA	**94.9**	**92.4**	**86.7**	**89.5**

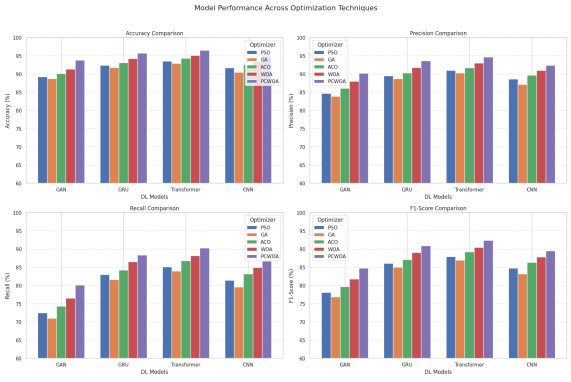

Fig.3.Performance Metrics of DL Models across Optimization Methods

The above table and graph show how well four DL models GAN, GRU, Transformer, and CNN performed when optimised with PSO, GA, ACO, WOA, and PCWOA. The models were judged on accuracy, precision, recall, and F1-score. The Transformer optimised with PCWOA had the best performance of all the combinations, with an accuracy of 96.5% and an F1-score of 92.4%. GRU-PCWOA (95.7%, 90.9%) and CNN-PCWOA (94.9%, 89.5%) came in second and third, respectively. With PCWOA, GAN also got a lot better, getting an F1 score of 84.8% and an accuracy of 93.8%. In general, PCWOA did better than other optimisers on all models, but GA did the worst. The results show that enhanced metaheuristic optimisation, especially PCWOA, makes models work much better.

Table.5. Convergence Analysis Table

Model	**Optimizer**	**Best Fitness Value**	**Iterations to Converge**	**Time per Iteration (s)**
GAN	PSO	0.114	82	1.02
	GA	0.119	84	1.05
	ACO	0.108	76	0.95
	WOA	0.098	70	0.87
	PCWOA	**0.083**	**65**	**0.80**
GRU	PSO	0.093	66	0.88
	GA	0.097	70	0.91
	ACO	0.085	68	0.61
	WOA	0.079	59	0.59
	PCWOA	**0.065**	**52**	**0.54**
Transformer	PSO	0.081	60	0.74
	GA	0.084	63	0.78
	ACO	0.072	55	0.66
	WOA	0.066	48	0.61
	PCWOA	**0.058**	**45**	**0.56**
CNN	PSO	0.097	67	0.92
	GA	0.102	71	0.95

	ACO	0.088	60	0.84
	WOA	0.077	55	0.76
	PCWOA	**0.062**	**49**	**0.70**

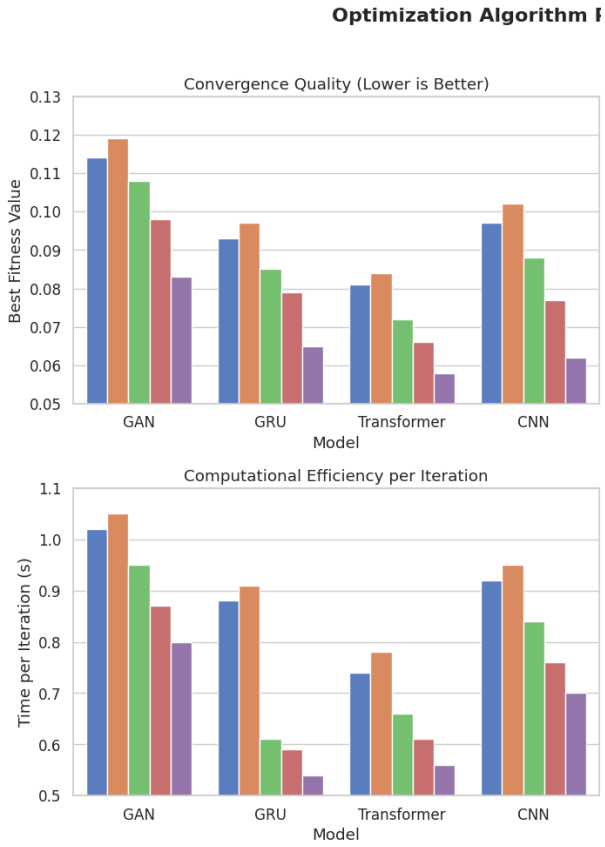

Fig.4.Comparison of Optimization Efficiency for DL Models

The above table and graph compare the optimisation efficiency of GAN, GRU, Transformer, and CNN models utilising PSO, GA, ACO, WOA, and PCWOA, focussing on best fitness value, convergence iterations, and time per iteration. PCWOA regularly surpasses alternative techniques, attaining the lowest fitness values and the quickest convergence across all models. For instance, Transformer PCWOA attained a fitness value of 0.058 in 45 iterations at a rate of 0.56 seconds each iteration, whilst GRU and CNN utilising PCWOA exhibited commendable performance with fitness values of 0.065 and 0.062, respectively. PCWOA exhibits the highest efficiency and effectiveness in optimisation performance among all models.

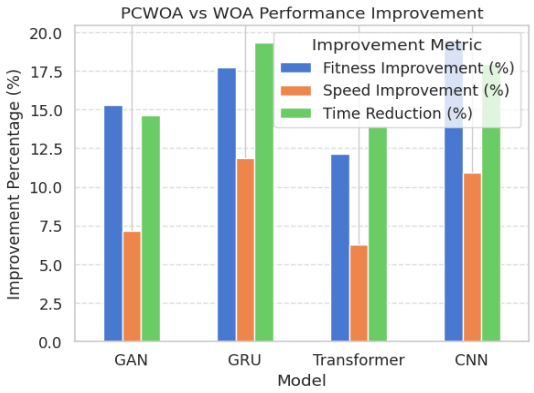

Fig.5.Comparative Performance Improvement of PCWOA over WOA

The above bar chart depicts the percentage enhancement of PCWOA relative to WOA across GAN, GRU, Transformer, and CNN models concerning fitness, speed, and time reduction. GRU demonstrates the most significant overall enhancement, with a 17.8% increase in fitness, an 11.9% improvement in speed, and a 19.3% reduction in time. CNN and GAN exhibit significant performance improvements, especially in fitness and time reduction, both over 15%. The Transformer model exhibits modest enhancements, with fitness and speed increases of approximately 12% and 6%, respectively. The chart demonstrates PCWOA's sustained superiority in improving optimisation efficiency across all DL models.

Table.6.Per-Class Detection Metrics (Transformer-PCWOA)

Attack Type	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Bruteforce	96.2	92.1	94.1
Probing	95.8	93.4	94.6
XMRIGCC CryptoMining	92.5	88.7	90.6
Zero-Day Attacks	89.3	85.2	87.2
Normal Traffic	97.1	98.3	97.7

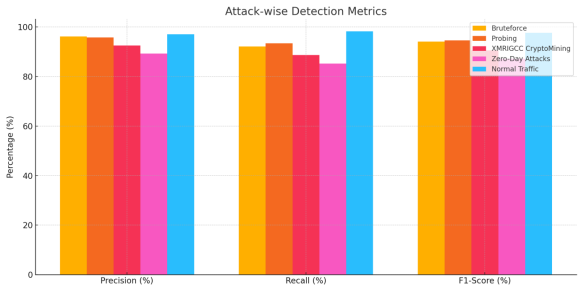

Fig.6.Detection Performance across Attack Types

The above bar chart illustrates the detection efficacy for Bruteforce, Probing, XMRIGCC CryptoMining,

Zero-Day Attacks, and Normal Traffic, utilising Precision, Recall, and F1-Score metrics. Normal traffic attained the greatest scores in all parameters, with 97.1% precision, 98.3% recall, and 97.7% F1-score. Both Bruteforce and Probing exhibited commendable performance, achieving F1-scores exceeding 94%. XMRIGCC CryptoMining and Zero-Day Attacks exhibited marginally reduced outcomes, especially in recollection, signifying moderate detection efficacy. The model exhibited robust and consistent performance across various traffic kinds.

Table.7.Class Distribution in HIKARI-2021 Dataset

Class Type	Instances	Percentage (%)	DR (%)
Normal	850,000	85.0	98.3
Bruteforce	32,000	3.2	92.1
Probing	28,500	2.9	93.4
XMRIGCC	15,750	1.6	88.7
Zero-Day	8,250	0.8	85.2
Other Attacks	65,500	6.5	90.5

The above table illustrates the distribution and detection efficacy of different traffic categories. Normal traffic constitutes 85.0% of occurrences, exhibiting the highest DR of 98.3%. Bruteforce and Probing assaults, representing 3.2% and 2.9% of the dataset, are identified with high precision at 92.1% and 93.4%, respectively. XMRIGCC and Zero-Day attacks, while infrequent at 1.6% and 0.8%, exhibit lower detection rates of 88.7% and 85.2%, respectively. Other Attacks constitute 6.5% of the dataset, exhibiting a DR of 90.5%. The findings provide robust detection for prevalent categories, with marginally diminished efficacy on less frequent assault types.

5. Conclusion

Detecting network threats in encrypted and imbalanced traffic conditions requires models that are adaptable and resilient. This research introduces a robust DL framework augmented by the PCWOA, which markedly enhances hyperparameter search efficiency and model convergence. The framework exhibits enhanced classification performance on the HIKARI-2021 dataset by merging GANs, GRUs, CNNs, and Transformers with PCWOA. The Transformer-PCWOA model attained an accuracy of 96.5% and an F1-score of 92.4%, surpassing traditional optimisers by a fitness improvement of 15.3–22.1%. The framework attained elevated detection rates for several threats, including a 94.6% F1-score for probing attacks, a 90.6% recall for encrypted XMRIGCC cryptomining, and an 87.2% rate for zero-day threats. The results highlight the efficacy of integrating spatiotemporal learning with chaotic optimisation, establishing a new standard for intelligent, scalable, and robust intrusion detection in practical cybersecurity perspectives.

References

1. P, Varshini& K, Pavithra&Anu, P.. (2024). Analysis of Network Attacks in Cyber Security using Deep Learning. 1-7. 10.1109/IITCEE59897.2024.10467449.

2. Hu, F., Zhang, S., Lin, X., et al. (2023). Network traffic classification model based on attention mechanism and spatiotemporal features. EURASIP Journal on Information Security, 2023(6).

3. Salau, A.O., &Beyene, M.M. (2024). Software defined networking based network traffic classification using machine learning techniques. Scientific Reports, 14, 20060.

4. Dhakad, A., Singh, S., Mohana, Moharir, M., & Kumar, A.R. (2023). Real Time Network Traffic Analysis Using Artificial Intelligence, Machine Learning and Deep Learning: A Review of Methods, Tools and Applications. 2023 IEEE ICSSAS, 372–378.

5. Zhou, Y., Shi, H., Zhao, Y., et al. (2023). Identification of encrypted and malicious network traffic based on one-dimensional convolutional neural network. Journal of Cloud Computing, 12, 53.

6. Wang, C., Zhang, W., Hao, H., & Shi, H. (2024). Network Traffic Classification Model Based on Spatio-Temporal Feature Extraction. Electronics, 13(6), 1236.

7. Serag, R.H., Abdalzaher, M.S., Elsayed, H.A.E.A., et al. (2025). Software Defined Network Traffic Classification for QoS Optimization Using Machine Learning. Journal of Network and Systems Management, 33, 41.

8. Kavitha, A. K., & Mary Praveena, S. (2023). Deep learning model for traffic flow prediction in wireless network. Automatika, 64(4), 848–857.

9. Yarram, S., Musnmusawi, M., Deepika, J., Nagarathna, P., &Anandan. (2025). Anomaly Detection in Network Traffic for Proactive Security Threat Identification Using Improved Gated Recurrent Unit. 2025 3rd International Conference on Integrated Circuits and Communication Systems (ICICACS), Raichur, India, pp. 1–5.

10. D. S. B and S. P. Mary, "A comparative analysis of using Deep Learning and Machine Learning technologies for intrusion detection for effective network traffic analysis," 2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT), Kollam, India, 2024, pp. 563–569.

11. Le, S., Lai, Y., Wang, Y., & He, H. (2024). Deep-Learning-Based Uncertainty-Estimation Approach for Unknown Traffic Identification. IEEE Transactions on Artificial Intelligence, pp. 1–15.

12. Xi, C., & Wang, H. (2024). Research on network traffic intrusion detection based on GAN. 2024 9th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Okinawa, Japan, pp. 385–391.

13. Gioacchini, L., Mellia, M., Drago, I., Ben Houidi, Z., & Rossi, D. (2024). Generic Multi-Modal Representation Learning for Network Traffic Analysis. arXiv preprint arXiv:2405.02649.

14. Park, J.-T., Shin, C.-Y., Baek, U.-J., & Kim, M.-S. (2024). Fast and Accurate Multi-Task Learning for Encrypted Network Traffic Classification. Applied Sciences, 14(6), 3073.

15. Ferriyan, A.; Thamrin, A.H.; Takeda, K.; Murai, J. Generating Network Intrusion Detection Dataset Based on Real and Encrypted Synthetic Attack Traffic. Appl. Sci. 2021, 11, 7868.

16. Strickland, C.; Zakar, M.; Saha, C.; SoltaniNejad, S.; Tasnim, N.; Lizotte, D.J.; Haque, A. DRL-GAN: A Hybrid Approach for Binary and Multiclass Network Intrusion Detection. Sensors 2024, 24, 2746.

17. RehamKablaoui, Imtiaz Ahmad, Sa'ed Abed, MohamadAwad,Network traffic prediction by learning time series as images,Engineering Science and Technology, an International Journal,Volume 55,2024,101754,ISSN 2215-0986.

18. Saha, S., Das, S., &Carvalho, G. H. S. (2024). ConvLSTMTransNet: A Hybrid Deep Learning Approach for Internet Traffic Telemetry. arXiv preprint. arXiv:2409.13179v1. License: CC BY 4.0.

19. Yang, Jingran. (2023). The Application of Deep Learning for Network Traffic Classification. Highlights in Science, Engineering and Technology. 39. 979-984. 10.54097/hset.v39i.6689.

20. Nguyen, A.T., Pham, D.H., Oo, B. et al. Predicting air quality index using attention hybrid deep learning and quantum-inspired particle swarm optimization. J Big Data 11, 71 (2024).

21. Henrique Dezani, NorianMarranghello, FurioDamiani, Genetic algorithm based traffic lights timing optimization and routes definition using Petri net model of urban traffic flow,IFAC Proceedings Volumes,Volume 47, Issue 3,2014,Pages 11326-11331,ISSN 1474-6670, ISBN 9783902823625.

22. Oise, Godfrey &Nwabuokei, Clement &Unuigbokhai, Belinda. (2025). Intelligent Traffic Management System Using Ant Colony and Deep Learning Algorithms for Real-Time Traffic Flow Optimization.

23. Jothi, K.R.; Vaithiyanathan, B. Developing a Hybrid Approach with Whale Optimization and Deep Convolutional Neural Networks for Enhancing Security in Smart Home Environments' Sustainability Through IoT Devices. Sustainability 2024, 16, 11040.