International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

A NEW APPROACH TO INTERNET TRAFFIC CLASSIFICATION: ARTIFICIAL BEE CLONING ALGORITHM - ONLINE SEQUENTIAL ANALYTICAL MACHINE-WAVELET (OSELM- WAVELET)

Thulfiqar Mahmood Tawfeeq[1], Mohsen Nickray[1]

1. Information Technology Department ,faculty of engineering , Qom University ,Qom ,Iran
Corresponding author,s Email : mahtho66@gmail.com

Abstract

This article discusses an important problem in network management and security—accurate and efficient classification of internet traffic. The need for real-time processing, as well as the complexity and everchanging nature of the network traffic, calls for both high accuracy and computational efficiency. To meet these demands, this study proposes a new approach that incorporates the Online Sequential Extreme Learning Machine (OSELM) and Wavelet Transform, which has been optimized by the Artificial Bee Cloning Algorithm. The proposed OSELM-Wavelet method uses Daubechies 4 wavelet transform to capture relevant frequency-domain features and, at the same time, preserve raw time-domain signals. This method improves the feature set by capturing time and frequency domain features simultaneously, boosting the input data by 800 features. Classification is carried out via the OSELM framework which supports efficient online training and inference and is thus suitable for real-time applications. The Artificial Bee Clony Algorithm is used to optimize the OSELM model parameters and improve the classification accuracy. This algorithm is inspired by the intelligent foraging behavior of bees and can effectively explore and exploit the parameter space for optimal solutions. The experiments bee algorithm-optimized OSELM-Wavelet conducted for internet traffic classification tasks demonstrated high accuracy and robustness.It outperforms conventional statistical and machine learning methods, particularly in situations demanding rapid adaptation and online learning. To summarize, the article combines OSELM-Wavelet with Artificial Bee Cloning Algorithm for Internet traffic classification, presenting a new solution. It aids in precisely and swiftly classifying traffic with minimal computation, thereby improving network security and management.

Keywords: Internet Traffic Classification, OSELM-Wavelet, Artificial Bee Cloning Algorithm

1. Introductiom

2. The swift development of the internet and the growing adoption of Virtual Private Networks (VPNs) have created the need for precise and efficient network traffic classification. For purposes such as traffic control, security evaluation, and anomaly detection, distinguishing

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

between VPN and Non-VPN traffic is essential. The task is complex because the VPN traffic is encrypted, and traditional methods of feature extraction and classification that rely on the contents of the packets cannot be used. Many existing methods rely on handcrafted features or pre-processed and trimmed sequences as a means to streamline input data. These techniques are unable to cope with the overwhelming dynamics of real network environments, where the need to process variable-length packets is combined with the need for accuracy and efficiency. Adding to the difficulty, modern network environments are susceptible to adversarial conditions that can occur due to even minor changes, such as packet drop, congestion, or malicious attempts to disrupt the network. Models that are trained on 'clean' traffic data tend to struggle in these settings, leading to unreliable performance and increased misclassification [1]. This underscores the need for strong classification models capable of dealing with encrypted VPN traffic while preserving accuracy in both normal operations and during adversarial scenarios. Therefore, these challenges demand a solution that integrates sophisticated deep learning approaches with robust defenses against adversarial manipulation. There has been substantial research aimed at optimizing network traffic classification performance under differing circumstances. In reference [2], an all-encompassing review of network traffic classification methodologies is presented, including port-based methods, deep packet inspection, and the application of statistical features alongside machine learning and deep learning techniques. Also addressed are the issues and prospects in this area. In [3], the authors propose a new approach to the early detection of anomalous traffic using an unsupervised deep learning model consisting of a CNN and an Autoencoder. This approach is said to achieve high accuracy. Nevertheless, the D-PACK model proposed in this paper only analyzes a small number of packets at the beginning of each flow, which limits the analysis to a subset of the data. This poses problems when the patterns of attacks are stealthily camouflaged in the unseen packets. A new approach based on deep Convolutional Recurrent Neural Networks (CRNN) to extract spatio-temporal features is presented in [4]. The paper shows how capturing sophisticated features leads to a marked increase in classification accuracy when compared to traditional methods. One drawback of this approach is its reliance on predefined statistical parameters such as packet counts and inter-arrival times, which can ignore crucial packet-level details. Publication [5] presents an enhanced algorithm of Support Vector Machine(SVM) called CMSVM which resolves the problem of data imbalance in classifying network traffic. The addition of active learning in this technique improves the performance and accuracy metrics. However, the authors believe that better performance could be achieved by feature dataset enhancement, or by applying CMSVM to other machine learning algorithms. In [6], a multi-scale feature attention method with CNNs for network traffic classification is introduced. This method demonstrates higher accuracy than existing techniques by analyzing the initial packets of network flows. It requires only one packet per flow for network traffic classification.

A new system is described in [7] for the classification of encrypted traffic, which uses network flow features and applies machine learning algorithms like XGBoost and LightGBM, obtaining good results. An exhaustive analysis of network traffic classification methods is presented in

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[8], where they are classified into five groups: classification by statistics, correlation based, behavior based, content based, and port based. The paper also proposes a set of criteria to evaluate these methods. In [9], the use of deep learning for traffic classification is studied, with particular attention paid to the detection of encrypted traffic and the shortcomings of traditional port and content-based methods.

In [10], the author conducted a systematic review on the application of data mining and machine learning for traffic classification in smart cities. They focused on the issues of data intricacy and feature selection for classification purposes. In reference [11], a hybrid neural network model, Deep Multiscale Hybrid Neural Network (DM-HNN), is proposed which analyzes both flow-level and packet-level features. This dual-feature extraction approach outperforms single-mode models and enhances traffic classification performance. A new approach to network traffic classification, utilizing timing features from network packets, is introduced in reference [12]. This method accurately differentiates encrypted traffic from unencrypted, marking it as promising for real-time traffic evaluation. While the approach is effective in rapidly classifying VPN traffic, its accuracy of 95.02% is not sufficient without further refinement. In addition, all the works mentioned so far ignore the existence of adversarial data, as well as artifacts that are commonplace in network traffic data. In [13] a multi-task learning approach named “Multi-task Transformer (MTT)” is proposed.

This model identifies applications and classifies traffic at the same time using a single model, achieving better results and speed than other methods. Different from the other approaches, this paper uses packets as the sequence of bytes for input. However, it does not work with variable-length packet sizes in the flows. A deep learning technique termed as “Deep Packet” which combines a CNN and an Autoencoder to classify VPN versus non-VPN traffic was presented in [14] and this technique outperformed other methodologies in recognizing encrypted traffic. This approach also uses fixed-size packet inputs of 1480 bytes. Reference [15] investigates traffic classification techniques within software-defined networks (SDN) implementing machine learning techniques including SVM, nearest centroid, and Naïve Bayes, obtaining reasonable accuracy levels. The approach proposed in [15] does, however, derive some statistical properties from the input packet flow.

In the reviewed studies that focused on internet traffic classification, particularly encrypted VPN traffic, several challenges were observed. One of the main issues in these methods is the heavy reliance on data preprocessing and the use of fixed-length inputs, which leads to loss of important information and reduced accuracy in real network conditions. Additionally, many of these approaches concentrate only on the initial packets of data flows and are unable to examine all packets completely, which may result in failure to detect malicious patterns hidden in later packets. Moreover, most models lack robustness against varying conditions and adversarial data, and their performance significantly degrades when confronted with malicious data or network noise.

In response to these challenges, our proposed method is based on the combination of the Artificial Bee Cloning Algorithm and the Online Sequential Extreme Learning Machine with

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Wavelet transform (OSELM-Wavelet). By leveraging multi-domain feature extraction and the optimization capabilities of the bee-inspired algorithm, this method can classify internet traffic accurately and rapidly in dynamic networks with variable-length data. Unlike previous models, it does not require complex preprocessing or data clipping and has the ability to adapt to real-world data and varying conditions. Additionally, the use of OSELM enables online learning and fast model updating, which is essential for dynamic network environments. Overall, our approach maintains high accuracy while offering significant robustness against adversarial conditions and network noise, making it highly suitable for security-critical applications and network traffic management.

The structure of this paper is organized as follows: Section 2 presents the fundamental concepts and background necessary for understanding the proposed method, including an overview of wavelet transforms, OSELM, and the Artificial Bee clony Algorithm. Section 3 describes the proposed methodology in detail, including the process of multi-domain feature extraction using wavelet analysis, the classification mechanism based on OSELM, and the optimization strategy using the bee-inspired algorithm. Section 4 introduces the dataset used in this study, along with its characteristics and preprocessing steps. Section 5 defines the evaluation metrics employed to assess the performance of the proposed approach. Section 6 presents the experimental results, showcasing the model's classification accuracy, robustness to network variability, and efficiency. Section 7 provides a comparative analysis between the proposed method and existing techniques in the literature. Finally, Section 8 concludes the paper with a summary of key findings and suggests directions for future research to further enhance encrypted traffic classification systems

2. Basic concepts

In this section, a detailed overview of the fundamental principles and key concepts required to understand the proposed method is presented.

2.1 Overview of Online Sequential Extreme Learning Machine (OSELM)

OSELM modifies the ELM model to accommodate the learning framework for sequential and online data streams. OSELM is different because it requires data to be presented sequentially, one observation at a time or in small mini-batches, with the model being updated stepwise. This is ideal for domains with continuous data streams like sensor networks, financial markets, or real-time applications. OSELM introduces the concept of sequential learning which means how the model can learn from additional data points over time without the need to re train the base model from scratch or keep all the previous data [16].

The general process in OSELM consists of the following steps [17]:

1. **Input Data**: The model gets a fresh data sample or a mini batch.
2. **Mapping to Hidden Layer**: The fresh data is mapped to the hidden layer using weights which are initialized randomly.

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3. **Update Output Weights**: The output weights are updated incrementally through a recursive least squares method.
4. **Prediction**: The modified model is used to make predictions with the new data.

This method ensures that OSELM manages large-scale and time-varying data with a low memory footprint and requires less computation, all while being efficient in how it's done.

- **Mathematical Formulation of OSELM**

Like ELM, the OSELM approach retains the same framework but changes it in one significant way. It updates the output weights in an online manner. The most important procedures in OSELM are summarized in the list below.

Sequential Mapping

At each time step t, the model receives a new input vector x_t and computes the corresponding hidden layer output:

$$H_t = G(W_t x_t + b_t) \quad (1)$$

Where:

- x_t is the new input data at time step t,
- W_t is the weight matrix between the input layer and hidden layer for the t-th data point,
- b_t is the bias term,
- G is the activation function.

The hidden layer output H_t is then used to update the output weights β_t

Incremental Update of Output Weights

The most significant benefit of OSELM is in the incremental changes made to output weights. Rather than performing a complete retraining of the model, OSELM uses a recursive least squares method to update the output weights. This enhances the model's ability to accommodate new data points.

The output weight update can be expressed recursively as:

$$\beta_{t+1} = \beta_t + \Delta\beta_t \quad (2)$$

Where the update term $\Delta\beta_t$ is computed as:

$$\Delta\beta_t = \frac{H_t^T (I + H_t H_t^T)^{-1}}{1 + H_t^T (I + H_t H_t^T)^{-1} H_t} \quad (3)$$

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Here, I is the identity matrix, and $\Delta\beta_t$ represents the incremental change in the output weights based on the new data point x_t This formula ensures that the model adapts incrementally as new data arrives, avoiding the need to store and process the entire dataset.

Prediction

Once the output weights have been updated, the OSELM can make predictions for new data:

$$y_t = H_t\beta_t \quad (4)$$

Where y_t is the predicted output for the new data point x_t, and β_t is the updated output weights.

- **Advantages of OSELM**

OSELM offers several advantages over traditional ELM and other online learning algorithms:

- **Real-time Learning**: OSELM's ability to learn in real time makes it ideal for scenarios where data is continuously streaming because there is no need for batch processing.
- **Low Memory Usage**: Compared to batch methods, OSELM's memory requirements are lower because it only needs to store the output weights and the hidden layer outputs.
- **Computational Efficiency**: Learning is efficient with OSELM since output weights are updated with recursive least squares, eliminating the need for full model retraining.
- **Adaptability**: It is well suited for non-stationary contexts where data characteristics shift over time and require continuous adaptation is OSELM's flexible ability to adjust to changes in data distribution.
- **Scalability**: OSELM can handle large-scale datasets efficiently, as it processes data incrementally and does not require storing all past data.

- **Applications of OSELM**

OSELM is especially helpful in areas where information is produced incessantly or where the setting is subject to change, such as :

• **Real-Time Predictive Analytics:** OSELM is capable of performing real-time forecasting activities including predicting stock market trends, forecasting energy consumption, or predicting demand.

• **Sensor Networks**: OSELM can be utilized in IoT systems where there is constant generation of sensor data which requires real-time processing.

• **Autonomous Systems:** OSELM can be incorporated in robotics and self-driving cars, where the systems are required to learn and adapt from new sensory information as it comes in.

• **Medical Diagnostics:** OSELM can be used to monitor medical data like vital signs or monitor disease progression, where real-time decisions are essential.

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

• **Online Learning and Data Streams:** Tasks like spam detection, classification of web pages, and recommendation systems where data is generated over time are best handled by OSELM.

The OSELM uses online learning tasks to train a powerful model. It offers an effective blend of Extreme Learning Machines and Sequential Learning, making OSELM scalable and efficient in memory and computation while learning from large-scale, real-time data streams. OSELM's capability for incremental model updates makes it ideal for dynamic, non-stationary environments, allowing for broad applications ranging from real-time predictions and sensor networks to autonomous systems [18].

2.2 Basics of the Wavelet Transform

The Wavelet Transform serves as a robust mathematical tool for the analysis of signals and data in the time-frequency domain. While the Fourier Transform provides frequency information under the assumption of a stationary signal, the Wavelet Transform provides simultaneous localization in time and frequency. This feature makes it ideal for the analysis of non-stationary or complex signals with changing time characteristics. Consequently, the Wavelet Transform is widely used in biomedical signal processing, finance, geophysics, image and audio analysis, as well as in machine learning [19].

Wavelets are energy-finite oscillatory functions which are localized as a result of transformations like dilation, compression, shifting. A wavelet serves as the mother function from which these wavelets are derived. The basis functions are particularly useful for capturing local signal features, including sharp changes, discontinuities, and structures that span multiple scales. In wavelet transforms, a signal is broken down into different components which, in turn, represent the signal's structure on various scales and at different positions. This allows one to retrieve both global features and local details [20]

Mathematically, the Continuous Wavelet Transform (CWT) of a signal x(t) is defined as:

$$W(a,b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{+\infty} x(t)\psi^*\left(\frac{t-b}{a}\right) dt \quad (5)$$

Where:

- ψ(t) is the mother wavelet function,
- a is the scale parameter, which controls frequency resolution,
- b is the translation parameter, which controls time localization,
- ψ∗ denotes the complex conjugate of the mother wavelet.

For practical cases of working with digital signals, the Discrete Wavelet Transform (DWT) is preferred. DWT techniques involve a signal decomposition based on a filter bank organized hierarchically using high-pass and low-pass filters. The hierarchical decomposition is referred to as Multiresolution Analysis (MRA), which provides a systematic representation of the signal at different levels of resolution.

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Each level of decomposition in DWT results in two components:

• Approximation coefficients are obtained through the low-pass filter, capturing the signal's slow, gradual changes or coarse features.

• Detail coefficients are the result of high-pass filtering and capture rapid changes or high-frequency components.

An essential feature of the Wavelet Transform is its ability to joint time-frequency localization. At low scales, wavelets provide high time resolution and low frequency resolution, which is useful for the analysis of rapidly changing features. At high scales, they provide high frequency resolution and low time resolution, which is suitable for the analysis of slowly changing phenomena. This resolving power which is dependent on the scale is more flexible and adaptive as compared to the use of fixed-window techniques such as Short-Time Fourier Transform (STFT) [21].

Wavelet families have been designed for particular purposes, with some examples being

• **Haar wavelet:** the simplest to compute, but not very smooth.

• **Daubechies wavelets:** Smooth and complex signals are easier to work with because of these wavelets' compact support and high number of vanishing moments.

• **Symlets, Coiflets, and Biorthogonal wavelets:** These are designed to balance the symmetry and smoothness along with the computational efficiency.

The Wavelet Transform is an associate, adaptable, and effective method of analyzing signals and data. It overcomes the limitations of time-domain and frequency-domain techniques by providing multi-resolution localized analysis slicing at different scales. This property makes it one of the essential methods used in the feature extraction, denoising, compression, and even pattern recognition in diverse scientific and engineering fields [22].

2.3Artificial Bee Algorithm

This model of an artificial bee colony includes three bee types: employed bees, onlooker bees, and scout bees. Employed bees focus on gathering food from a designated source. Onlooker bees periodically assess if the food source is still worthwhile while moving among employed bees, and finally, scout bees look for new food sources. In bee algorithms, a solution in the search space is a food source and the initial number of food sources is equal to the number of bees in the hive. The objective function's value at that position (fitness value) determines a bee's source's quality [23]. A bee colony has the capability to exploit food sources over large distances in diverse directions. Areas with abundant nectar and pollen are visited by a large number of bees because these areas can be collected with minimal effort, while regions with less nectar and pollen attract fewer bees [24].

Food Search

In a colony, the food search procedure begins with the dispatching of scout bees to look for flower patches. Scout bees journey from one flower patch to another. The colony further

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

extends its search during the flowering season by retaining a portion of its population as scouts. After every flower patch has been surveyed, each scout bee engages in a specific dance above the flower patch that holds a dependable nectar and pollen resource. This dance, called the waggle dance, reveals the flower patch's, relative to the hive, direction, distance, and its quality. This information dispatches additional and follower bees to the flower patch.

Most follower bees choose the flower patches which are most likely to offer pollen and nectar. IFocus in one area causes the bees to scatter across the flower patch, within the scope of their dance, so that not just one, but all the best flowers within it are eventually reached [25].

In bee algorithm optimization, all possible solutions are represented as food sources within a parameter space. To explore the solution space, scout bees simplify it and evaluate the position using a fitness function. The evaluated solutions are ranked, and other bees, termed new forces, search the vicinity of the pylons which contain the highest ranks to get the best solutions—which are termed flower patches. The goal of the selective search is to optimize the chosen solution towards the maximal value of the fitness function [26].

Main Components of the Algorithm

1. **Employed Bees:**

   What an employed bee does mainly consists of collecting food from a designated location, and workers will always deplete the resource fully. In practice, this behavior corresponds to creating new positions around where the employed bees are working and checking if the new position offers better food. Employed bees always recall the location of the best food sources until it is depleted.

2. **Onlooker Bees:**

   Onlooker bees supervise the work of foraging bees. They hover over the hive, watching the activities of the employed bees, and determine which ones have been successful in food collection. Onlooker bees always go for the best performers, employing a probabilistic method called the "meeting place" where other bees are also directed to the identified successful location to maximize food collection.

3. **Scout Bees:**

   Once a food source has been overexploited and its quality does not improve any further, a bee assumes the role of a scout and randomly searches for new areas. This process can be likened to mutation in genetic algorithms, and is crucial in preventing stagnation in local optima.

Steps of the Bee Algorithm

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- The erratic intelligent behavior of bees can be consolidated into the following steps. Firstly, the bees try to search the environment randomly for food sources that are advantageous (fitness value).
- Once located, they become employed bees and start harvesting food from the specified source.
- The employed bee returns to the hive and unloads the nectar. After unloading, the bee can either return directly to the source or share the information on the source through a waggle dance in the dance area.
- If the food source is depleted, the employed bees become scouts and randomly look for new food sources.
- The onlooker bees remain in the hive and observe the employed bees as they gather at the food sources. They choose one food source from the gathered food sources that has the highest profit among them.
- The selection of the food sources is determined by the specific quality of the source (fitness value).

Mathematical Formulations of the Artificial Bee Colony (ABC) Algorithm

Let's assume we have an optimization problem which contains D decision variables and SN food sources (which is equal to the number of employed bees). The actions of the three types of bees in the ABC algorithm: employed bees, onlooker bees, and scout bees, guide the algorithm's workflows. The main equations that formulate the characteristics and activities of the ABC algorithm are given next [27].

1. Initialization of the Food Sources

At first, the population of food sources (candidate solutions) is created randomly in accordance to the specified parameters.

$$x_{ij} = x_j^{min} + rand(0,1).(x_j^{max} - x_j^{min}) \quad (6)$$

x_{ij} : The j-th parameter of the i-th food source.

x_j^{max}, x_j^{min}: The lower and upper bounds of the variable j.

rand(0,1): A uniformly distributed random number in the range [0, 1].

2. Employed Bee Phase (Position Update)

Every employed bee changes the existing solution by drawing upon the knowledge of another randomly chosen food source.

$$v_{ij} = x_{ij} + \phi_{ij}.(x_{i,j} - x_{kj}) \quad (7)$$

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

v_{ij}: The new position (solution) for dimension j of bee i.

x_{kj}: The j-th parameter of a randomly chosen food source k\ i.

ϕ_{ij}: A random number in the range [−1,1] usually drawn from a uniform distribution.

If v_{ij} exceeds the search boundaries, it is adjusted as:

$$v_{ij} = \min\left(\max\left(v_{ij}, v_j^{mim}\right), v_j^{max}\right) \quad (8)$$

3. Fitness Evaluation

The fitness value for each solution x_i, based on the objective function $f(x_i)$is calculated using the following scheme to ensure positivity:

$$fit_i = \begin{cases} \frac{1}{1+f_i} & if\ f_i \geq 0 \\ 1 + |f_i| & if\ f_i < 0 \end{cases} \quad (9)$$

4. Onlooker Bee Phase (Probabilistic Selection)

Based on the fitness values of different tolls, onlooker bees will select food sources based upon a probability proportional to their fitness values.

$$P_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j} \quad (10)$$

Each onlooker bee selects a food source x_i based on this probability and generates a new candidate solution using the same equation as in the employed bee phase.

5. Scout Bee Phase (Abandonment and Random Search)

If a food source x_i cannot be improved after a predefined number of trials (known as the **limit**), it is abandoned. The corresponding employed bee becomes a scout and generates a new random solution:

$$x_{ij} = x_j^{min} + rand(0,1).\left(x_j^{max}, x_j^{min}\right) \quad (11)$$

6. Memorizing the Best Solution

During each iteration, the most optimal solution so far obtained (in terms of having the least objective function value) is preserved.

$$x_{best} = \arg\min f(x_i) \quad (12)$$

7. Termination Condition

An algorithm will naturally stop when certain conditions are met, including:

- The maximum number of iterations is completed,
- The objective function value is at an acceptable level,

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- There has been no tangible progress over multiple iterations

3. Methodology

This research seeks to refine the precision of distinguishing between VPN and non-VPN traffic by utilizing a more comprehensive approach involving wavelet feature extraction in conjunction with an optimized Online Sequential Extreme Learning Machine (OSELM) classifier. Through this technique, multi-domain features are extracted through wavelet transforms from network traffic data of varying lengths. Wavelet transforms capture temporal and frequency-based patterns, providing both time and frequency information, unlike traditional methods that require fixed-length inputs and extensive preprocessing. The extracted features are then classified using an OSELM model, which is an online lightweight learning model and updates in real-time, making it very suitable for agile, evolving network environments. Further improvement is achieved by using the Artificial Bee Cloning Algorithm to optimize the hyperparameters of OSELM, increasing both accuracy as well as generalization. This approach provides a fast, adaptive, and resilient solution to some of the most critical problems in encrypted traffic classification like information loss due to data cutting and vulnerability to adversarial noise. The remaining sections present the steps of the proposed method which include preprocessing, wavelet-based feature extraction, OSELM classification, bee-inspied metaheuristic optimization, and traffic classification under realistic conditions.

In this article, we are concerned with the application of OSELM and the Artificial Bee Colony Algorithm (ABCA) for network traffic classification. The OSELM model's hyperparameters such as hidden layer neuron count, batch size, activation function, and the proportion of data set aside for sequential training are all tuned by the ABCA optimizer. Classification performance relies heavily on the model hyperparameters. These parameters must therefore be configured properly to improve classification accuracy, otherwise, the model's performance may be compromised.

We prefer OSELM because of its speed and ability to learn online. OSELM differentiates from other conventional neural networks which utilize backpropagation for weight adjustments because it sets hidden layer weights randomly, optimizing only the output layer. This approach improves efficiency and is more convenient for large-scale network traffic data analysis, as it cuts training time drastically.

As an optimization technique, the Artificial Bee Colony Algorithm (ABCA) is used because it is adaptive and is based on how honey bees forage. It uses a mixture of random and local search strategies which allow it to effectively navigate the search space, thus determining OSELM's optimal hyperparameters. Use of ABCA leads the model to greater classification accuracy and stability and prevents local optima convergence.

The initial step involves preprocessing the raw data. Within the context of machine learning and classification, the preprocessing phase is essential because it defines whether the raw input received is cleaned and transformed into the appropriate format to train the models. Effective preprocessing enhances model performance by eliminating noise, reducing bias, and ensuring

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

balanced data distribution.For this paper, several preprocessing steps were employed, such as shuffling, min-max normalization, 80-20 partitioning, divided labeled class oversampling, and undersampling, in order to refine the data and enhance its quality prior to classification. In order to counteract bias, shuffling has been shown to assist in randomizing the order of the dataset so that the model does not learn patterns from the dataset ordering.

Min-max normalization scales feature values to a fixed range, most commonly from 0 to 1. Such models which are sensitive to different feature magnitudes, like OSELM, require this kind of normalization. Normalizing data helps avoid scenarios in which a single feature would greatly influence the learning process, leading to more stable and effective training.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (13)$$

In this context, it is splitting the dataset into 80% for training and 20% for testing. This approach is particularly popular in the machine learning field as it provides a balanced way to train a model and still have enough data left over for testing generalization capabilities.

Undersampling tries to resolve class imbalance issues by lowering the number of instances in the majority class. In the case of network traffic classification, datasets typically have a surplus of non-VPN traffic relative to VPN traffic. We improve data balance by undersampling the dominant class, mitigating the risk of the model becoming biased toward the majority class. This allows the model to capture useful features for both VPN and non-VPN traffic. These steps taken before model training enhance the model's efficiency and accuracy, ensuring reliable and untampered classification results. After that, the model requires signal analysis and feature extraction. Wavelet Transform is one of the most insightful methods that can be used to analyze a signal and extract relevant features from complicated datasets. This method transforms a signal into its constituent frequencies, permitting a combined time and frequency domain analysis. Unlike Fourier Transform which only outputs frequency data, Wavelet Transform captures temporal and spectral information which is essential for network traffic analysis.

The original signal is decomposed into several levels in the Wavelet Transform, yielding Approximation Coefficients and Detail Coefficients. The approximation coefficients capture lower frequency signals which are less intricate and more general features, while the detail coefficients capture variations and rapid changes that happen at higher frequencies. This form of decomposition enables multi-resolution analysis of the data. In this article , we apply Daubechies-4 (db4) wavelet for feature extraction. The db4 wavelet is selected because of valuable mathematical properties which allow it to deal with non-stationary and noisy data, for example, network traffic. In addition, it is preferred in such applications because it strikes a good balance between compact support and smoothness. The network traffic data is decomposed to four levels using Wavelet Transform. From the extracted coefficients, we choose detail coefficients from levels 2 to 4 as well as the approximation coefficients from the last level as the primary features. This choice is made because mid-level detail coefficients

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

hold significant information regarding the transient changes and behavioral patterns, while the last approximation coefficients convey the signal's fundamental trend. The combination of these features improves the classification accuracy of VPN and non-VPN traffic.

Then Classification Using OSELM is performed. The OSELM network is known for its rapid training, strong generalization, and overall remarkable performance as a model. It proves especially useful in situations where data is ingested sequentially, and the system has to learn in real-time, without needing to refresh the entire model. OSELM has an architecture that includes three layers: an input layer, a hidden layer, and an output layer. Raw data is received by the input layer, which subsequently transfers it to the hidden layer. The hidden layer is made up of non-linear neurons. The output layer makes the final classification decision based on the calculated weights from the hidden to output layer. An important characteristic of OSELM is that the weights from the input layer to the hidden layer are determined at random. These fixed weights simplify the training process focusing only on the hidden to output layer weights. This optimization enhances the efficiency of OSELM for real-time applications.

OSELM training follows an online sequential learning strategy where data is ingested one by one, and the output layer weights are updated incrementally. The network tries to learn and reduce the classification error, and it keeps training until it either converges or meets a predefined stopping criterion. The training process involves correcting the output layer weights with the least squares method to ensure that the network systematically computes from the given inputs to the required outputs. The choice of an activation function applied to the neurons in the hidden layer has vital influence on the performance of the model. Sigmoid, hyperbolic tangent (tanh), and ReLU are some of the commonly used activation functions. In this article, the sigmoid function is applied as an activation function because it is appropriate for binary classification problems such as distinguishing VPN from non-VPN traffic.

The performance of OSELM is affected by some parameters like the count of neurons in the hidden layer, batch size, activation function, and the percentage of data set aside for sequential training. Having a certain number of hidden neurons defines the model's ability to learn complex patterns. While having a higher number of neurons increases the cap on the amount of intricate relationships within the data that the model learns, it could also risk overfitting. The amount of samples in a batch determines the model upadtes after a set number of samples is processed. Smaller batch sizes are associated with faster convergence due to more frequent updates, but risk increased noise during learning. The choice of the activation function (e.g., sigmoid, tanh, or ReLU) is critical because it determines how well the network can model the non-linearities present in the data. For OSELM, data is usually partitioned into training and testing sets. A certain percentage is usually earmarked for training, This percentage is critical because it balances the amount of data allocated for training against the data set aside for testing.

To optimize these parameters, we use an the Artificial Bee Colony Algorithm (ABCA). ABCA is an optimization algorithm based on the foraging behavior of honey bees. In this case, ABCA is applied to fine-tune the parameters of OSELM model for better performance. ABCA mimics

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

the searching and exploitation processes of a food source by a swarm of bees. In this case, the "solutions" involve determining the optimal values for hidden neurons, batch size, activation function, and the percentage of data used for sequential training. For implementing ABCA, the algorithm is based on a population of bees, where each bee represents a candidate solution. Each bee alters the parameters of OSELM and evaluates the model's performance with a fitness function. In this case, the fitness function is the classification accuracy of the OSELM model, which directs the search process toward the optimal parameters.

In this optimization process, the goal of the objective function is to maximize the classification accuracy of the OSELM model. This now means that the ABC algorithm will update the parameters so as to maximize the classification accuracy. The objective function for this optimization is given in Equation

$$cost_{fun} = MSE = \frac{1}{n}\sum_{i-1}^{n}(Y_i - Y'_i)^2 \quad (14)$$

Ultimately, the stopping criterion for the training process is set based on the number of iterations. In other words, the algorithm stops after a set number of iterations, irrespective of whether the optimal solution has been attained. This allows control over how much computational resources are spent for attaining the optimization goal. With this, the OSELM model can be optimized accurately and efficiently which is ideal for network traffic classification, especially in the identification of VPN and non-VPN traffic.

4. **Dataset**

The ISCX 2016 Dataset

This paper utilizes the ISCX 2016 VPN and Non-VPN Traffic dataset to assess the proposed technique. The ISCX 2016 was designed to have an all-inclusive and realistic compilation of the actual network traffic [28]. The reasoning behind the dataset construction was to have a large amount as well as a wide variety of traffic data for realistic scenarios. To this end, the dataset creators established a specific set of activities capturing traffic that represented activities people do normally on the internet. As an example, a user account for Alice and Bob were created so that they would access services like Skype or Facebook so that the dataset will contain features of real world data. The dataset contains both regular sessions and sessions conducted over a Virtual Private Network (VPN). In total, 14 traffic categories were generated, some of which include VoIP, VPN-VoIP, P2P, and VPN-P2P, etc. Table 1 captures the various types of traffic and applications.

Table 1: description of the different types of traffic

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Traffic	Description
Browsing	This category includes HTTPS traffic generated by users during browsing sessions or tasks involving a web browser. For instance, while capturing voice calls using Google Hangouts, browsing traffic was also recorded, even though browsing was not the primary activity.
Email	Email traffic was captured using the Thunderbird email client and Gmail accounts for Alice and Bob. The clients were configured to send emails via SMTPS and receive emails using both POP3/SSL and IMAP/SSL protocols.
Chat	This category includes instant messaging applications such as Facebook and Hangouts via web browsers, as well as Skype, AIM, and ICQ, using the Pidgin application.
Streaming	The streaming category consists of multimedia applications requiring continuous data transfer. Traffic from YouTube (HTML5 and Flash versions) and Vimeo was captured using Chrome and Firefox browsers.
File Transfer	This category includes traffic from applications designed for sending and receiving files. The dataset includes traffic from Skype file transfers, FTP over SSH (SFTP), and FTP over SSL (FTPS) sessions.
VoIP	Voice over IP traffic was generated by applications such as Facebook, Hangouts, and Skype voice calls.
P2P	Peer-to-peer traffic was captured by downloading various .torrent files using the uTorrent and Transmission applications.

Network packet analyzers like Wireshark and tcpdump collected the traffic, which culminated in a dataset size of 28 GB. VPN traffic was created through an external VPN provider and was connected over OpenVPN (UDP mode). SFTP and FTPS traffic were created by an outside provider using the FileZilla client. To facilitate the labeling process, all non-essential services and applications were turned off during data capture to ensure only pertinent traffic, like Skype voice calls and SFTP file transfers, were captured. Only packets with source or destination IPs associated with the local client were captured using filters. The dataset contains labeled network traffic in two forms: full packet captures (pcap) and CSV files created with ISCXFlowMeter, a Java application that processes pcap files to extract certain features. The dataset, along with the tools, is available to the public for researchers.

The dataset from UNB ISCX comprises traffic data from multiple sources like YouTube, Gmail, Facebook, Skype, and Bittorrent, enabling the study of different kinds of network behavior (Table 2). It helps in building machine learning algorithms for classifying traffic, detecting anomalies, as well as in studying the pattern of VPN and non-VPN traffic in security research.

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Table 2: The mapping relationship between application types and traffic types

Traffic Type	Application Type
Web Browsing	Firefox and Chrome
Email	SMTPS, POP3S, IMAPS
Chat	ICQ, AIM, Skype, Facebook, Hangouts
Streaming	Vimeo, YouTube
File Transfer	Skype, FTPS, SFTP (using FileZilla and external services)
VoIP	Facebook, Skype, Hangouts voice calls (1-hour duration)
P2P	uTorrent, Transmission (Bittorrent)

5. Evaluation Metrics

In this artcle, the performance of the proposed model is analyzed based on its accuracy, precision, recall, and F1 score. These metrics are well-known in classification problems because they holistically evaluate a model's ability to not only classify network traffic but also deal with skewed data distributions..

Accuracy: Accuracy is essential metric for measuring the correctness of the classifier based on the evaluating the proportion of correctly classified instances against the total number of instances. In terms of network traffic classification, accuracy can be formally defined as:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (15)$$

where:

- TP (True Positives) represents the number of correctly classified network flows that belong to a specific application.
- TN (True Negatives) denotes the number of correctly classified flows that do not belong to a specific application.
- FP (False Positives) refers to the number of flows incorrectly classified as belonging to an application.
- FN (False Negatives) refers to the number of flows belonging to an application but classified otherwise.

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

While accuracy is helpful in balanced class situations, it may fail to explain a classifier's effectiveness in a network traffic classification context due to class imbalance.

Precision: Precision assesses the proportion of accurately identified instances relative to the total identified for a particular class. In the context of network traffic classification, understanding the classifier's capability to reduce false positive errors is very important, especially when some traffic types are infrequent or of high importance. The formula for precision is given as

$$Precision = \frac{TP}{TP+FP} \qquad (16)$$

A highly precise score reflects the model's ability to accurately detect genuine traffic without much misclassification, which is crucial when differentiating among various network applications.

Recall: Recall, or sensitivity, also known as true positive rate, measures the share of instances for a class that are correctly classified. It evaluates the model's capabilities in identifying pertinent network flows and emphasizes minimizing false negatives. Recall is defined as

$$Recall = \frac{TP}{TP+FN} \qquad (17)$$

In network traffic classification, recall is very important in cases where the goal is to ensure that critical or security-sensitive data traffic types are captured by the model and not ignored.

F1 Score: F1 score is a metric that captures both precision and recall as the harmonic mean of the two, thereby incorporating both false positives and false negatives. In the case of network traffic classification with an imbalanced dataset, it is necessary to maximize both precision and recall. Therefore, the F1 score becomes critical in such situations. The F1 score is computed as**:**

$$F_1 Score = \frac{2\times(Precision\times Recall)}{Precision+Recall} \qquad (18)$$

Evaluating a model's performance across various types of network traffic is easier using the classification model's F1 score since it encompasses both precision and recall.

6. Simulation results

Identifying and distinguishing VPN from non-VPN traffic constitutes a notable gap in contemporary network management and security. Because of the growing popularity of VPNs and their use in concealing traffic patterns, there is now a demand for classification methods which are both efficient and accurate. The high-dimensional and dynamic character of the network data adds to this challenge. Moreover, many traditional methods lack real-time processing capabilities or are not robust to adversarial perturbations which limits their practical usefulness. This article addresses some of these gaps by presenting a novel approach called OSELM-Wavelet, designed to speed up processing using online sequential learning and wavelet feature extraction.

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

OSELM-Wavelet method integrates the artificial bee colony optimization technique with wavelet analysis for fast adaptive classification which allowing for real-time usage. Simulations were carried out in MATLAB 2024a on a high-performance machine whose specifications are presented in Table 3 Therefore this configuration sufficed for the efficient processing of the OSELM-Wavelet model.

Table 3: System specifications for simulations

Component	Specification
Software	MATLAB 2024a
CPU	Intel Core i7-13650HX
RAM	16 GB
GPU	NVIDIA RTX 4060, 8 GB GDDR6

The OSELM-Wavelet method provides a quick and flexible solution for distinguishing between VPN and non-VPN traffic. It integrates wavelet transform for multi-domain feature extraction with classification via Online Sequential Extreme Learning Machine (OSELM). Artificial Bee Colony (ABC) algorithm is utilized to optimize OSELM parameter settings with the goal of enhancing classification accuracy in both standard and hostile environments.

To begin with, feature extraction is done using Daubechies 4 (db4) wavelet transform on raw traffic packets, where each packet is decomposed to four levels. As is customary, the first detail level (d1) which contains high-frequency noise is discarded. The remaining components (d2, d3, d4, and a4) are used as frequency-domain features. At the same time, the original packet data (which is a time-domain signal) is kept to retain temporal information. Hence, the resultant feature vector is a combination of time-domain (x) signal and frequency-domain (d2–a4) signals, thus enriching the representation of traffic features.

In order to conform with OSELM's input size restriction, both the time domain and frequency domain contribute 400 elements each, leading to an overall feature vector of 800 elements. In the time domain, packets are either padded or truncated to a size of 400 bytes. In the frequency domain, to prevent size imbalance that may lead to one of the four components dominating and thus equal contribution from all frequency bands, instead of using standard wavelet downsampling, each of the four components is resized to exactly 100 bytes.

The classification accuracy and robustness, especially concerning resilience to noise and adversarial manipulation, have been improved with this dual-domain and size-normalized feature construction. With respect to real-time applications, the approach is efficient and capable of identifying patterns in VPN versus non-VPN traffic. In the figure 1, we show processed input signals for one random sample of the dataset. As shown, the time domain signal

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

is zero-padded to a length of 400, while the other frequency domain signals are of length 100 packet bytes.

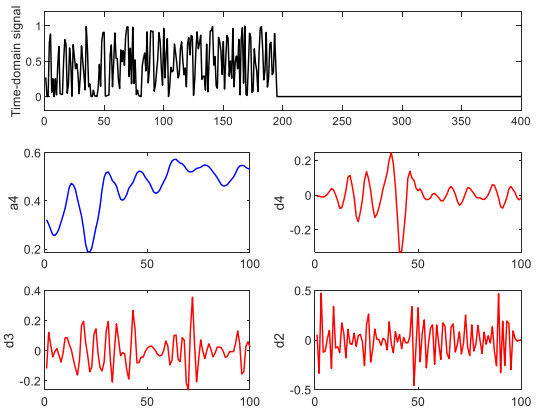

Figure 1: The processed time-domain and frequency-domain input signals for a random sample of the dataset

In this study, having built the feature vector with 800 features (400 in the time domain and 400 in the frequency domain), the OSELM network was trained and optimized with the Artificial Bee Colony (ABC) algorithm. The ABC algorithm was applied to adjust the four main parameters: the hidden neuron count, activation function, batch size, and the percentage of training samples designated for the sequential training portion.

The hidden neurons were limited in number between 20 and 200 to ensure model complexity did not hinder computational efficiency or vice versa. Among the three possible activation functions: linear, tanh, and sigmoid, one was chosen using a roulette wheel method which provides randomized yet diverse exploration to prevent early convergence to suboptimal functions.

The batch size was tuned within the interval of 256 to 512 to ensure stable training and effective GPU memory consumption on an NVIDIA RTX 4060. The sequential data training ratio was set between 30% and 70% to make sure both the initial training and sequential training phases had enough data while retaining the flexibility of the OSELM model.

In terms of optimization, data was randomly split into two segments where 20% was set aside for validation and the other 80% was separated into initial and sequential training groups according to some optimal ratio. With respect to the optimization done with the ABC Algorithm, its cost function was set to the model's prediction error on the validation set measured by mean squared error (MSE), which indicates the generalization ability of the model.

Taking into consideration the system specifications, which include an Intel Core i7-13650HX processor, the ABC algorithm was performed for 60 iterations with a colony size of 8 which ensured a good balance between search diversity and computational efficiency. The

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

acceleration coefficient was set to 1 which provided a neutral balance between exploration and exploitation, ensuring stable convergence during the optimization process.

Figure 2 illustrates the convergence curve of the ABC algorithm for 1000 function evaluations. While the validation mean squared error (MSE) dropped from 0.03722 to 0.02487, it reached a plateau pretty quickly. The fast convergence of the ABC algorithm demonstrates its capability to effectively traverse the parameter space. Most of the progress was made in the first 400 evaluations, indicating that the selected colony size and iteration number were adequate to discover a solution that was nearly optimal. The parameters after optimization are given in Table 4: the hidden neurons were 199, the batch size was 501, the activation function used was sigmoid, and the proportion of the sequential training data was 0.3697 (36.97%). The values near the maxima for hidden neurons (199) and batch size (501) suggest that the model is aided by high capacity and large batch updates, which is likely due to the complexity stemming from the 800 features in the input space. The sigmoid activation function will help capture the subtle variations of VPN and non-VPN traffic because it is a smooth non-linear function.

The 36.97% share regarding sequential training indicates a real drift towards initial training (63.03%) which ensures a good foundational model is built before incrementally updating. This aligns with the method's emphasis on real-time adaptability.

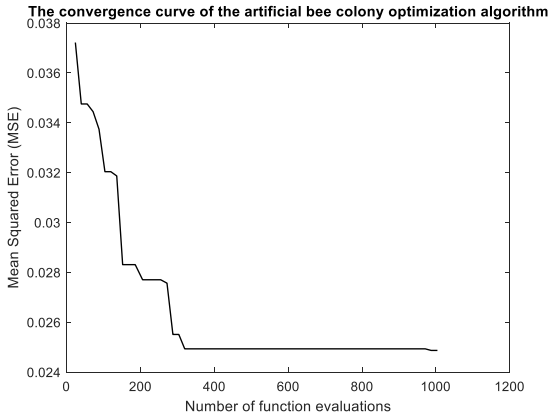

Figure 2: Convergence curve of the artificial bee colony optimization algorithm

Table 4: Optimized parameters for the OSELM-Wavelet method obtained from ABC algorithm.

Parameter	Value
The number of hidden neurons	199
Batch size	501

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Activation function	Sigmoid
Probability of sequential training data	0.3697

The training and evaluation of the OSELM-Wavelet model were conducted with optimized parameters. Results for both normal and adversarial traffic classification will be discussed in the following subsections.

Normal traffic classification results

Evaluating a classification model's effectiveness can be done using a confusion matrix as it compares predicted labels against true labels. A confusion matrix breaks and summarizes the model's performance against its predictions including true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) which informs about the precision and recall of the model in question. The accuracy of the model can also be determined from these values. The confusion matrix's diagonal values are correct predictions while off-diagonal values are errors. Here, TP means non-VPNs accurately recognized and TN means VPNs correctly identified. VPNs mislabeled as non-VPNs are FP, and non-VPNs incorrectly labeled as VPNs are FN.

The confusion matrix illustrated in Figure 3 gives a first glance towards evaluating the model performance of the training data. As seen from the matrix, for 498 true VPN samples, 480 were correctly classified which makes it a striking 96.4% classification with 18 misclassified as non-VPN (3.6%). For the case of 582 true non-VPN samples, 571 were classified correctly (98.1%) while 11 were classified as VPN (1.9%). The overall accuracy on the training set can be computed as

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Samples}} = \frac{480+571}{498+582} = \frac{1051}{1080} \approx 97.3\% \quad (19)$$

Precision, recall, and F1-score for each class further quantify the model's performance. For the VPN class:

$$\text{Precision}_{\text{VPN}} = \frac{\text{True VPN}}{\text{Predicted VPN}} = \frac{480}{480+11} = \frac{480}{491} \approx 97.8\% \quad (20)$$

$$\text{Recall}_{\text{VPN}} = \frac{\text{True VPN}}{\text{Actual VPN}} = \frac{480}{498} \approx 96.4\% \quad (21)$$

$$\text{F1-Score}_{\text{VPN}} = 2 \times \frac{\text{Precision}_{\text{VPN}} \times \text{Recall}_{\text{VPN}}}{\text{Precision}_{\text{VPN}} + \text{Recall}_{\text{VPN}}} = 2 \times \frac{0.978 \times 0.964}{0.978 + 0.964} \approx 0.971 \quad (22)$$

For the non-VPN class:

$$\text{Precision}_{\text{non-VPN}} = \frac{\text{True non-VPN}}{\text{Predicted non-VPN}} = \frac{571}{571+18} = \frac{571}{589} \approx 96.9\% \quad (23)$$

$$\text{Recall}_{\text{non-VPN}} = \frac{\text{True non-VPN}}{\text{Actual non-VPN}} = \frac{571}{582} \approx 98.1\% \quad (24)$$

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

$$\text{F1-Score}_{\text{non-VPN}} = 2 \times \frac{\text{Precision}_{\text{non-VPN}} \times \text{Recall}_{\text{non-VPN}}}{\text{Precision}_{\text{non-VPN}} + \text{Recall}_{\text{non-VPN}}} = 2 \times \frac{0.969 \times 0.981}{0.969 + 0.981} \approx 0.975 \quad (25)$$

The training accuracy OSELM-Wavelet achieved (97.3%) along with the F1-scores of 0.971 for VPN traffic and 0.975 for non-VPN confirmed its learning capabilities on the dataset. The slightly better recall for non-VPN traffic (98.1%) over VPN traffic (96.4%) indicates that the model performs better in classifying non-VPN traffic, likely because non-VPN packets have distinct, less complex temporal and spectral signatures compared to the more complex and masked VPN traffic. Nonetheless, the low misclassification rates (3.6% for VPN and 1.9% for non-VPN) show that the model's performance was well-balanced for both classes, which confirms the adequacy of dual domain (time and frequency) feature extraction as well as the nearly balanced dataset (1.2:1 ratio post-undersampling).

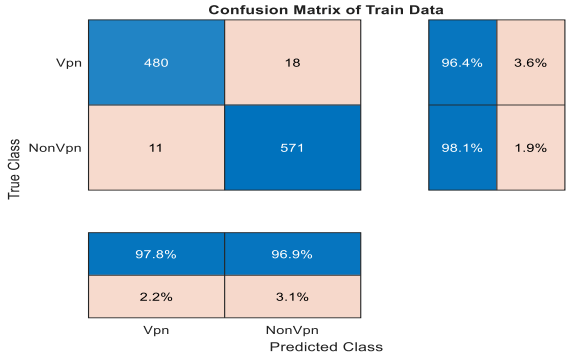

Figure 3: Confusion matrix of training data for the OSELM-Wavelet method

Figure 4 shows the ROC curve which assesses the model's test set performance discriminative ability. Receiver Operating Characteristic (ROC) curves are particularly important for assessing classification performance. The ROC curve graphs the true positive rate (sensitivity) against the false positive rate (1-specificity) for a number of different thresholds. The AUC (Area Under the Curve) measures the overall performance of the model concerning class separation. AUC values near 1 demonstrate effective class separation, while values approaching 0.5 indicate ineffectiveness. ROC is most advantageous where there is class imbalance.

The AUC metric for VPN classification is as high as 0.9956, demonstrating remarkable distinction between VPN and non-VPN samples. The operating point for the VPN model that is marked on the curve has a TPR of about 0.98 and an FPR of 0.02 which is consistent with the high recall reported in the confusion matrix. Such a high value of AUC is an indication that the OSELM-Wavelet method is capable of accurately identifying VPN traffic despite the complications of encapsulation, protocol diversity, and the rest of the VPN traffic within the ISCX dataset. The close proximity of the operating point to the top-left corner reinforces the conclusion of the model achieving high TPR with low FPR which is essential for practical implementation in network monitoring systems.

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

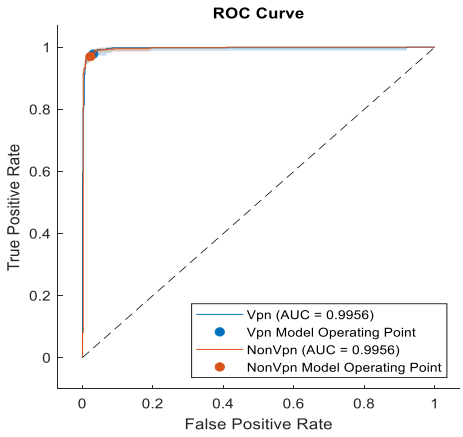

Figure 4 ROC curve for the OSELM-Wavelet method on test data

The findings reveal that the OSELM-Wavelet method performs exceptionally well on normal traffic, exhibiting high accuracy, balanced F1-scores, and an excellent AUC. The slight bias towards non-VPN recall might be due to the method's responsiveness to the unique frequency-domain characteristics of non-VPN traffic as captured by the wavelet transform's d2, d3, d4, and a4 components. Nonetheless, the low misclassification rates and near-perfect AUC suggest that the method generalizes very well to normal traffic and establishes a strong baseline for evaluation under adversarial testing in the following subsection.

Adversarial traffic classification results

This checks the robustness of the OSELM-Wavelet method against adversarial traffic, which is important in practical situations where an attacker may try to evade classification using intentional modifications. To gain deeper insights into the method's performance, three scenarios are studied: (1) training and testing with normal data, (2) training with normal data and testing with adversarial data, (3) training and testing with adversarial data. Examples of attacks were created using the Fast Gradient Sign Method (FGSM), a well-known method for generating adversarial modifications. FGSM was used with a perturbation limit of

controlling the maximum permissible distortion, and a step size of , determining the extent of every gradient change. Such settings were taken to ensure the introduced changes were realistic and the adversarial influences on network traffic were targeted and controlled, maintaining the corrupted packets within a realistic range (normalized data in [0, 1]). The adversarial examples were crafted by applying FGSM was computed as

$$x' = x + \grave{o} \cdot \text{sign}(\nabla_x \text{L}(f_\theta(x), y)) \quad (26)$$

where L is the loss function, f_θ is the trained OSELM-Wavelet model, y is the true label, and $\nabla_x \text{L}$ is the gradient of the loss with respect to the input. The step size $\alpha = 0.0187$ was used

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

to approximate the perturbation within the $\epsilon = 0.015$ bound, ensuring the adversarial examples remain close to the original data while maximizing misclassification likelihood.

Table 5 summarizes the accuracy of the OSELM-Wavelet method across the three scenarios, evaluated on the test set.

Table 5: Accuracy of OSELM-Wavelet under different training and inference scenarios

Training Data	**Inference Data**	**Accuracy (%)**
Normal Data	Normal Data	97.3148
Normal Data	Adversarial Data	85.2778
Adversarial Data	Adversarial Data	92.0370

The first scenario, training and inference on normal data,serves as the baseline The accuracy of 97.3148% VPN and non-VPN traffic behavior under normal conditions are differentiated effectively through dual-domain feature extraction (time and frequency) and ABC-optimized parameters, which explains the high baseline AUC of 0.9956 alongside the F1-scores. As noted prior, these values demonstrate the exceptional results derived from clean traffic.

For the second case, the model underwent evaluation on adversarial data after training on normal data. The accuracy dropped significantly to 85.2778%, which is a reduction of 12.037% from the baseline. This decline demonstrates how untrained models are delicate to adversarial tweaks and modifications. Though 85.2778% accuracy is still considerably lower than the baseline, the value suggests that the model has some defenses against adversarial attacks. This is particularly true when considering the random guessing benchmark of 50%. This suggests that at least some aspects of the feature extraction strategy, especially the discarding of the noisy d1 component, help reduce the impact of adversarial influences. Nonetheless, the large reduction in accuracy highlights the importance of devising better strategies tailored for adversarial situations.

In the third scenario, training and inference performed on adversarial data achieves an accuracy of 92.0370%. Although this is still 5.2778% below the baseline set by normal data, it is an improvement of 6.7592% over the second scenario. Training on adversarial data enables the OSELM-Wavelet model to adjust decision boundaries based on learned distortions, thus figuring out FGSM's perturbations. OSELM’s ability to learn sequentially is quite useful in this case since the model can adjust its weights to better counteract adversarial samples. The impact of ABC-optimized sigmoid activation function on model smoothing also FGSM’s extreme gradient change effects lessened could explain the improved accuracy. Still, the existing

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

accuracy gap indicates that the model's performance is not fully recovered despite the effectiveness of adversarial training.

In the OSELM-Wavelet method, the clear benefits of adversarial training are shown in the comparison of the three scenarios, where robustness improves by 6.7592% with both training and inference on adversarial data. Still, the consistent gap of 5.2778% between normal and adversarial performance suggests that the method's feature-wavelet dependence, while useful for normal traffic, remains a weakness under attack. The frequency-domain signal downsampling and upsampling to 100 bytes each preserves feature representation consistency; however, these processes may also uniformly transmit adversarial perturbations across all levels and so amplify their impact. Although the OSELM-Wavelet method is fast and its performance on normal traffic is good, the need for an alternative approach tailored for greater accuracy and robustness is underscored by its performance in adversarial situations.

7. Comparison

A thorough evaluation of the method OSELM-Wavelet is done concerning several advanced techniques for classifying VPN and non-VPN traffic in this section. The central findings as well as the key defining features and performance metrics of prior research and this method are presented in Table 5, marking the notable strengths, weaknesses, and trade-offs in accuracy, robustness, and computational efficiency.

Works like [29] and [70] use network packet statistical features to identify VPN and non-VPN traffic. The methodology in [29] achieves a detection accuracy of 95.02% using a Random Forest classifier, with precision and recall scores of 95.4% and 97%, and an F1-score of 96.2%. It would be ideal for VPN traffic detection due to the high recall, but the lower precision and F1-score indicate some level of imbalanced classification with more frequent false positive detections. [30] uses AdaBoost and LightGBM and obtained 93.1% accuracy with precision of 91.3%, recall of 94%, and an F1-score of 92.6%. For fairness, only the binary classification case of [30] is used here. This method shows flexibility to different dataset sizes through the use of temporal features, but the lower accuracy and precision suggest struggles with complex traffic pattern datasets. Both [29] and [30] have the ability to handle variable-length packet flows, a trait shared with the methods proposed here, but the reliance on averaged statistical features rather than full packet flow summaries means crucial details, including raw packet flow data, may be ignored. More importantly, both methods lack framework for adversarial robustness making them less useful in applications where security is a concern and adversarial threat is present.

In [31], the authors used Artificial Neural Networks (ANN) to classify VPN traffic with 98.79% accuracy, 97.94% precision, 98.54% recall, and F1 score of 96.84%. The ANN's performance recall was good because the advanced feature extraction techniques improved its ability to tell the difference between VPN and non-VPN traffic. However, like [29] and [30], this approach suffered from the use of feature summarization which ignores deep packet inspection and thus does not provide a full view of the traffic, missing its complex or

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

adversarial intricacies. Furthermore, the lack of adversarial training in [31] makes it susceptible to performance degrading perturbations in practical environments.

Deep learning approaches like [32] and [33] process trimmed packet flows of uniform size, sacrificing the ability to handle variable-length flows. This is done in the name of standardization during preprocessing. In [32] BERT-based ET-CNN (BCFNet) achieves high benchmark accuracy, precision, recall, and F1 score of 100, 100, 100, and 100 respectively for VPN/non-VPN binary classification. While this method performs well on normal traffic classification, its dependence on fixed-size flows renders it unable to deal with variable-length packets. In addition, the lack of robustness against adversarial attacks is a critical limitation for security-sensitive applications. A similar approach in [33] which combines CNN and Swin Transformer achieves 96.7%, 95.7%, 97.3%, and 96.2% for accuracy, precision, recall, and F1 score respectively. While this method outperforms several leading models on normal traffic classification, it suffers from the same shortcomings as [35): inability to process variable-length flows and the absence of considerations for adversarial conditions.

The effectiveness of OSELM-Wavelet is indicative with metrics like an accuracy of 97.31%, precision of 97.25%, recall of 97.35%, and F1-score of 97.30% on normal traffic. It differs from works [29], [30], and [31] as it implements adversarial training (FGSM, $\epsilon = 0.015$), which boosts its robustness to 92.0370% accuracy under adversarial . This is a notable improvement over other approaches which do not consider these factors. OSELM-Wavelet's primary advantage is its online training and inference capabilities through the online sequential extreme learning machine framework. This allows the model to adaptively adjust its weights with incoming data, proving useful in dynamic network settings where traffic patterns change over time (e.g., new protocols or shifts in user behavior). Through online retraining, these modifications are executable in real-time monitoring sans full retraining cycles. With a training time of 0.054 seconds and an inference time of 0.004 seconds per packet, both times enhanced by the OSELM framework, the system is agile for developmental and deployable purposes.

Nevertheless, OSELM-Wavelet is dependent on wavelet feature extraction which necessitates the input data to be of a uniform size. This means that it cannot accommodate variable-length packet flows in the data streams, or capture all of the information in the packet flow. This represents a limitation in comparison with the other proposed method.

Table 6: Comparison of proposed methods with state-of-the-art techniques for VPN traffic classification

Received: July 21, 2025

International Journal of Applied Mathematics
Volume 38 No. 3s, 2025
ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

Ref.	Method	Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
[29]	Statistical features and random forest	ISCX VPN-nonVPN	95.02	95.4	97	96.2
[30]	Adaboost and light GBM	ISCX VPN-nonVPN	93.1	91.3	94	92.6
[31]	ET-BERT and 1D-CNN (BCFNet)	ISCX VPN-nonVPN	100	100	100	100
[32]	Artificial Neural Networks (ANN)	ISCX VPN-nonVPN	98.79	97.94	98.54	96.84
[33]	CNN and Swin Transformer	ISCX VPN-nonVPN	96.7	95.7	97.3	96.2
Ours	PGD trained BiLSTM-EfficientNetB0	ISCX VPN-nonVPN	99.81	99.82	99.81	99.81
Ours	OSELM-Wavelet	ISCX VPN-nonVPN	97.31	97.25	97.35	97.30

8. Conclusion

Classifying network traffic, specifically in differentiating between VPN and non-VPN traffic, is still a major problem for network administrators because the use of VPNs to mask traffic, network data patterns, and adversarial attacks are becoming more common. Many existing approaches suffer from limitations such as lacking real-time processing capability, being fragile to perturbations, or poor management of variable-length packet streams, which makes them unreliable for security-sensitive applications. To address these issues, this thesis puts forward a new and effective approach—OSELM-Wavelet—that integrates wavelet-based frequency analysis with time-domain signal processing, online sequential learning, and optimization using the Artificial Bee Colony (ABC) algorithm to achieve high classification accuracy, fast execution, and robustness against adversarial manipulation.

In the OSELM-Wavelet method that has been proposed, a flow of network traffic is processed using discrete wavelet transform with the Daubechies 4 (db4) wavelet which decomposes the signal up to four levels. In this case, we will obtain detail coefficients d2, d3, d4 along with approximation coefficient a4 to obtain important information in the frequency domain while d1 is discarded due to being the lowest level signal that is very noisy. Simultaneously, the unprocessed time signal is kept to augment the frequency domain representation. In order to achieve a uniform size for training, both domains are set to the same fixed lengths of 400 resulting in 800 dimensions after concatenation. The d1 component, which is the noisiest part is discarded. An Online Sequential Extreme Learning Machine (OSELM) suitable for real-time incremental training is then applied. OSELM's are very useful in constantly changing environments like networks as they do not require full retraining for new data.

Received: July 21, 2025

International Journal of Applied Mathematics

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

For best results, four critical hyperparameters of the OSELM model, hidden neurons, activation function, batch size, and the proportion of sequential training data are optimized using ABC metaheuristic. Using a bee colony of size 8 and a fixed acceleration coefficient of 1, the ABC algorithm balances exploration and exploitation while minimizing the mean squared error (MSE) on a validation set over 60 iterations. The optimal configuration was found to consist of 199 hidden neurons, a sigmoid activation function selected via roulette wheel mechanism, batch size of 501, and a sequential training data ratio of 36.97%. With these parameters, the OSELM was capable of effectively learning and generalizing to new patterns in the traffic data.

With the OSELM-Wavelet method, we were able to achieve 97.31% accuracy, 97.25% precision, 97.35% recall, 97.30% F1 score, and an AUC of 0.9956 under normal testing conditions. In addition, the model's robustness was tested using FGSM adversarial training with $\epsilon = 0.015$. When trained with adversarial samples, the model maintained an accuracy of 92.0370% which is only a 5.28% drop in performance. The model performed exceptionally well with a training time of 0.054 seconds and 0.004 seconds per packet during inference for real-time scenarios such as live network streaming and adaptive intrusion detection.

References

[1] Dias, K. L., Pongelupe, M. A., Caminhas, W. M., & De Errico, L. (2019). An innovative approach for real-time network traffic classification. *Computer networks*, *158*, 143-157.

[2] A. Azab, M. Khasawneh, S. Alrabaee, K. K. R. Choo, and M. Sarsour, "Network traffic classification: Techniques, datasets, and challenges," *Digital Communications and Networks*, Vol. 10, No. 3, pp. 676-692, 2024.

[3] R. H. Hwang, M. C. Peng, C. W. Huang, P. C. Lin, and V. L. Nguyen, "An unsupervised deep learning model for early network traffic anomaly detection," *IEEE Access*, Vol. 8, pp. 30387-30399, 2020.

[4]G. D'Angelo and F. Palmieri, "Network traffic classification using deep convolutional recurrent autoencoder neural networks for spatial–temporal features extraction," *Journal of Network and Computer Applications*, Vol. 173, 102890, 2021.

[5]S. Dong, "Multi class SVM algorithm with active learning for network traffic classification," *Expert Systems with Applications*, Vol. 176, 114885, 2021.

[6]Y. Wang, X. Yun, Y. Zhang, C. Zhao, and X. Liu, "A multi-scale feature attention approach to network traffic classification and its model explanation," *IEEE Transactions on Network and Service Management*, Vol. 19, No. 2, pp. 875-889, 2022.

[7]R. Bozkır, M. Cicioglu, A. Calhan, and C. Togay, "A new platform for machine-learning-based network traffic classification," *Computer Communications*, Vol. 208, pp. 1-14, 2023.

Received: July 21, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[8] J. Zhao, X. Jing, Z. Yan, and W. Pedrycz, "Network traffic classification for data fusion: A survey," *Information Fusion*, Vol. 72, pp. 22-47, 2021.

[9]S. M. Rachmawati, D. S. Kim, and J. M. Lee, "Machine learning algorithm in network traffic classification," In: *Proc. of the 2021 International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 1010-1013, October 2021.

[10]M. Shafiq, Z. Tian, A. K. Bashir, A. Jolfaei, and X. Yu, "Data mining and machine learning methods for sustainable smart cities traffic classification: A survey," *Sustainable Cities and Society*, Vol. 60, 102177, 2020.

[11]Y. Yang, Y. Yan, Z. Gao, L. Rui, R. Lyu, B. Gao, and P. Yu, "A network traffic classification method based on dual-mode feature extraction and hybrid neural networks," *IEEE Transactions on Network and Service Management*, Vol. 20, No. 4, pp. 4073-4084, 2023.

[12]M. Al-Fayoumi, M. Al-Fawa'reh, and S. Nashwan, "VPN and Non-VPN Network Traffic Classification Using Time-Related Features," *Computers, Materials & Continua*, Vol. 72, No. 2, 2022.

[13]W. Zheng, J. Zhong, Q. Zhang, and G. Zhao, "MTT: an efficient model for encrypted network traffic classification using multi-task transformer," *Applied Intelligence*, Vol. 52, No. 9, pp. 10741-10756, 2022.

[14]M. Lotfollahi, M. Jafari Siavoshani, R. Shirali Hossein Zade, and M. Saberian, "Deep packet: A novel approach for encrypted traffic classification using deep learning," *Soft Computing*, Vol. 24, No. 3, pp. 1999-2012, 2020.

[15]M. M. Raikar, S. M. Meena, M. M. Mulla, N. S. Shetti, and M. Karanandi, "Data traffic classification in software defined networks (SDN) using supervised-learning," *Procedia Computer Science*, Vol. 171, pp. 2750-2759, 2020.

[16] Zhao, J., Wang, Z., & Park, D. S. (2012). Online sequential extreme learning machine with forgetting mechanism. *Neurocomputing*, *87*, 79-89.

[17] Wang, X., & Han, M. (2014). Online sequential extreme learning machine with kernels for nonstationary time series prediction. *Neurocomputing*, *145*, 90-97.

[18] Zou, H., Lu, X., Jiang, H., & Xie, L. (2015). A fast and precise indoor localization algorithm based on an online sequential extreme learning machine. *Sensors*, *15*(1), 1804-1824.

[19] Chan, Y. T. (2012). *Wavelet basics*. Springer Science & Business Media.

[20] Bhatnagar, N. (2020). *Introduction to wavelet transforms*. Chapman and Hall/CRC.

[21] Walnut, D. F. (2013). *An introduction to wavelet analysis*. Springer Science & Business Media.

Received: July 21, 2025

Volume 38 No. 3s, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

[22] Akujuobi, C. M. (2022). *Wavelets and wavelet transform systems and their applications*. Berlin/Heidelberg, Germany: Springer International Publishing.

[23] Karaboga, D. (2010). Artificial bee colony algorithm. *scholarpedia*, *5*(3), 6915.

[24] Karaboga, D., & Akay, B. (2009). A comparative study of artificial bee colony algorithm. *Applied mathematics and computation*, *214*(1), 108-132.

[25] Bansal, J. C., Sharma, H., & Jadon, S. S. (2013). Artificial bee colony algorithm: a survey. *International Journal of Advanced Intelligence Paradigms*, *5*(1-2), 123-159.

[26] Gao, W. F., & Liu, S. Y. (2012). A modified artificial bee colony algorithm. *Computers & Operations Research*, *39*(3), 687-697.

[27] Sonmez, M. (2011). Artificial Bee Colony algorithm for optimization of truss structures. *Applied Soft Computing*, *11*(2), 2406-2418.

[28] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and VPN traffic using time-related features," In *Proceedings of the 2nd International Conference on Information Systems Security and Privacy (ICISSP)*, pp. 407-414, February 2016.

[29] M. Al-Fayoumi, M. Al-Fawa'reh, and S. Nashwan, "VPN and Non-VPN Network Traffic Classification Using Time-Related Features," *Computers, Materials & Continua*, Vol. 72, No. 2, 2022.

[30] G. Abbas, U. Farooq, P. Singh, S. S. Khurana, and P. Singh, "Feature engineering and ensemble learning-based classification of VPN and non-VPN-based network traffic over temporal features," *SN Computer Science*, Vol. 4, No. 5, 546, 2023.

[31] P. Zhu, G. Wang, J. He, Y. Dong, and Y. Chang, "An encrypted traffic identification method based on multi-scale feature fusion," *Array*, Vol. 21, 100338, 2024.

[32] S. A. A. Mohamed and S. Kurnaz, "Classified VPN network traffic flow using time related to artificial neural network," *Computers, Materials and Continua*, Vol. 80, No. 1, 819-841, 2024.

[33] Y. Wang, Y. Gao, X. Li, and J. Yuan, "Encrypted traffic classification model based on SwinT-CNN," In *2023 4th International Conference on Computer Engineering and Application (ICCEA)*, pp. 138-142, IEEE, April 2023

Received: July 21, 2025