

Optimizing Firewall Log File Classification with Multilayer Perceptrons

Artitayaporn Rojarath¹, Nantanee Srisaengchan², Saisunee Jabjone², Suda Tipprasert³ and Olarik Surinta^{1,*}

¹Multi-agent Intelligent Simulation Laboratory (MISL) Research Unit, Department of Information Technology, Faculty of Informatics, Mahasarakham University, Mahasarakham 44150, Thailand

²Department of Information System and Digital Innovation, Faculty of Science and Technology, Nakhon Ratchasima Rajabhat University, Nakhon Ratchasima 30000, Thailand

³Faculty of Business Administration, Rajamangala University of Technology Isan, Nakhon Ratchasima 30000, Thailand

*Corresponding Email: olarik.s@msu.ac.th

Received November 12, 2024, Revised October 18, 2025, Accepted January 8, 2026, Published July 1, 2026

Abstract. *Cyber threats remain a major concern for network security and data integrity, often causing serious consequences through unauthorized access, data breaches, and malware attacks. Intrusion detection systems are essential for countering these risks by continuously monitoring and classifying network activities. This study investigates the use of multilayer perceptrons (MLPs) to optimize firewall log file classification and improve the identification of network events. Four activation functions (Linear, Sigmoid, Tanh, and ReLU) were examined to determine the optimal configuration. The MLP model consistently achieved high performance, with accuracy exceeding 99%, and the ReLU activation function demonstrated superior effectiveness. The synthetic minority oversampling technique was applied to handle class imbalance in the firewall log dataset, improving the detection accuracy of the minority reset-both class. Moreover, feature selection using XGBoost reduced the input set from eleven to five key attributes, achieving 99.84% accuracy and improving computational efficiency. Experimental evaluations confirmed that the proposed model effectively recognizes complex nonlinear relationships in network data. These results demonstrate the potential of MLPs to enhance the accuracy, efficiency, and robustness of firewall log file classification for modern intrusion detection systems.*

Keywords: cybersecurity, feature selection, firewall log file, imbalanced data, intrusion detection systems, multilayer perceptron, neural network, multiclass classification

1 Introduction

Modern firewalls are primarily built on predefined signatures and rule-based logging mechanisms, which often fail to detect zero-day threats or advanced anomalies [1]. In recent years, machine learning techniques have been increasingly applied to classify firewall log actions with greater flexibility [2] and to integrate neural networks with oversampling strategies such as synthetic minority oversampling technique (SMOTE) for improved packet classification [3]. However, challenges persist in firewall log analysis, including excessive false positives and limited adaptability, even when ensemble learning approaches are adopted [4].

Modern firewall platforms, including next-generation firewalls (NGFWs), integrate log inspection, application awareness, and heuristic filtering capabilities. However, these systems still rely on predefined policies and signature databases, which limits adaptability to new or unseen attack behaviors [3]. Performance also tends to decline when NGFWs process large-scale or complex traffic data, and static rule-based mechanisms struggle with class imbalance, often resulting in reduced classification accuracy. In contrast, the proposed multilayer perceptron (MLP) framework uses a learning-based mechanism that captures complex feature interactions within firewall log data. The integration of oversampling and feature selection techniques enhances both robustness and efficiency, addressing adaptability and imbalance limitations typically observed in NGFW implementations.

Although substantial progress has been made, network security remains a critical concern as organizations continue to face evolving and increasingly sophisticated cyber threats. With the growing ubiquity of internet-based transactions and data exchanges, cyberattacks are becoming more frequent and diverse, employing advanced methods to evade detection. Firewalls alone cannot provide comprehensive protection against these threats [5]. Intrusion detection systems (IDS) have therefore been developed to complement firewalls by monitoring network activity and distinguishing between normal and malicious behaviors [6, 7].

Intrusion detection can be broadly categorized into misuse detection and anomaly detection. Misuse detection identifies attacks by matching events against known patterns, whereas anomaly detection focuses on deviations from established normal behavior [8]. The latter is particularly important for detecting emerging or previously unseen threats. Machine learning and deep learning models, including MLPs, have been widely applied to improve detection accuracy and reduce false positives [7, 9].

Despite the growing adoption of learning-based techniques, important research gaps remain. Most existing studies focus on traditional classifiers such as support vector machines, decision trees, and ensemble models, while MLP architectures have received limited attention in the analysis of firewall log files. Prior research has rarely examined the effects of activation function selection, feature reduction, and data imbalance handling on MLP performance. The present study addresses these gaps

by developing an optimized MLP classification framework that systematically evaluates activation functions, applies synthetic oversampling, and integrates feature selection to improve both accuracy and computational efficiency.

Recent studies have explored advanced paradigms such as unsupervised learning, reinforcement learning, and generative adversarial networks for intrusion detection. However, these models often demand extensive computational resources and complex parameter tuning, which hinder real-time deployment [10, 11, 12]. Moreover, scalability, interpretability, and resource utilization remain ongoing challenges in operational environments [1, 12, 13]. In comparison, MLP-based architectures offer a balanced trade-off between simplicity, adaptability, and predictive performance, making them a strong foundation for scalable and efficient intrusion detection systems [12, 14].

This study investigates the use of an optimized MLP network for detecting anomalous activities within firewall log data. The contributions of this research are summarized as follows:

- *Optimized neural architecture for firewall-based intrusion detection:* An optimized MLP architecture achieves higher performance than other classifiers on the firewall log dataset and demonstrates the capability of the model to learn complex patterns inherent in network behavior.
- *Comparative analysis of activation functions to address non-linearity:* A comparative study of linear, sigmoid, hyperbolic tangt (Tanh), and rectified linear unit (ReLU) functions is conducted to model non-linearity in firewall log data effectively.
- *Feature selection and class-imbalance management:* Correlation analysis and feature-importance ranking from random forest and XGBoost are used to identify pattern inputs, while the synthetic minority oversampling technique (SMOTE) ensures balanced class representation. These strategies collectively enhance generalization and interpretability.

2 Related Work

This section reviews machine learning approaches to intrusion detection, with particular emphasis on firewall log analysis and network anomaly classification.

Ertam and Kaya [15] proposed a classification method for firewall log analysis using a support vector machine (SVM) with multiple kernel functions: linear, polynomial, sigmoid, and radial basis function (RBF), for firewall log operation classification. Utilizing a dataset from Firat University containing 65,532 records with 11 features, they identified four salient attributes through systematic feature selection: port numbers, byte counts, packet volumes, and temporal information. The RBF kernel achieved an optimal precision-recall balance with an F_1 -score of 76.4%, while the sigmoid kernel, despite superior recall (98.5%), exhibited lower precision, yielding an F_1 of 74.8%. These results demonstrate the importance of kernel selection in achieving balanced classification performance.

Roempluk and Surinta [16] investigated distributed denial-of-service (DDoS) attack classification using KNN, MLP, and SVM on two benchmark datasets: KDD CUP 1999 and NSL-KDD. Duplicate instances were removed before model training to ensure data quality. After removing duplicate instances, the datasets contained approximately 529,655 and 12,354 instances, respectively. Experimental results revealed that the KNN model achieved accuracies of 99.984% and 99.087% on the KDD CUP 1999 and NSL-KDD datasets, respectively, outperforming both the SVM and MLP models.

Al-Behadili [17] conducted a comprehensive comparative evaluation of seven algorithms: decision tree, SVM, OneR, ZeroR, artificial neural network (ANN), particle swarm optimization, and multi-class classifier. Employing an 11-feature representation consistent with Ertam and Kaya [15], the study demonstrated that decision trees achieved the highest accuracy of 99.839%, where as the ZeroR model obtained the lowest accuracy of 57.437%. This notable improvement over Ertam and Kaya's F_1 suggests that tree-based methods may better capture the hierarchical decision structures inherent in firewall operations.

Aljabri *et al.* [18] conducted an experiment on a large-scale firewall log dataset containing 1,048,576 instances collected from a private organization. The dataset included 25 attributes and was divided into four classes: allow, deny, drop, and reset-both. To manage imbalance in the dataset, an undersampling technique was applied, reducing the dataset to 112,532 instances. Two feature subsets were then created, consisting of 11 and 13 attributes, respectively. The random forest algorithm achieved the highest accuracy, outperforming ANN, K-Nearest neighbor (KNN), J48, and Naïve Bayes classifiers. Specifically, the random forest achieved accuracy scores of 99.11% and 99.64% on the first and second feature subsets, respectively.

While the reviewed studies demonstrate the effectiveness of traditional machine learning for intrusion detection, several critical limitations remain, including insufficient exploration of learning architectures, inadequate handling of class imbalance and limited optimization strategies. This study addresses these issues by: (1) developing an optimized MLP architecture with systematic hyperparameter tuning specifically designed for firewall log classification; (2) conducting a comprehensive analysis of activation functions to identify effective non-linear modeling strategies; and (3) integrating feature selection with SMOTE-based class balancing to enhance model generalization.

3 Background and Theoretical Framework

This section establishes the foundational concepts and theoretical principles underlying the proposed firewall log classification methodology. We first provide essential background on firewall operations and intrusion detection integration (Section 3.1), then present the MLP architecture employed in this study (Section 3.2).

3.1 Firewall Operations and Intrusion Detection Integration

Firewalls function as the primary perimeter defense mechanism in network security architectures, establishing a controlled boundary between trusted internal networks and untrusted external environments [19]. Operating at various layers of the OSI model, firewalls examine network traffic against predefined rule sets and make filtering decisions such as *typically allow*, *deny*, or *drop*, based on packet attributes including source/destination IP addresses, port numbers, protocols, and connection states.

Each filtering decision is recorded in firewall logs, which constitute structured event records containing critical metadata: timestamps, source and destination endpoints, protocol types, port numbers, packet/byte counts, and the action taken (allow/deny/drop/reset). These logs provide a comprehensive audit trail of network perimeter activity and serve as the primary data source for security analysis. In contrast to raw network traffic captures, firewall logs provide pre-filtered and aggregated information that balances granularity with computational tractability, making them particularly suitable for machine learning analysis.

Firewall efficacy depends critically on proper configuration; misconfigured rule sets can create security gaps or generate excessive false alarms [19, 20]. Moreover, traditional stateless firewalls and even next-generation firewalls (NGFWs) with advanced features such as deep packet inspection and application control [20] cannot detect all attack patterns, particularly those involving temporal correlations, multi-stage attacks, or subtle behavioral anomalies. This constraints motivates the application of machine learning to firewall log analysis, which enables pattern recognition beyond explicit rule-based filtering through automated learning from historical data.

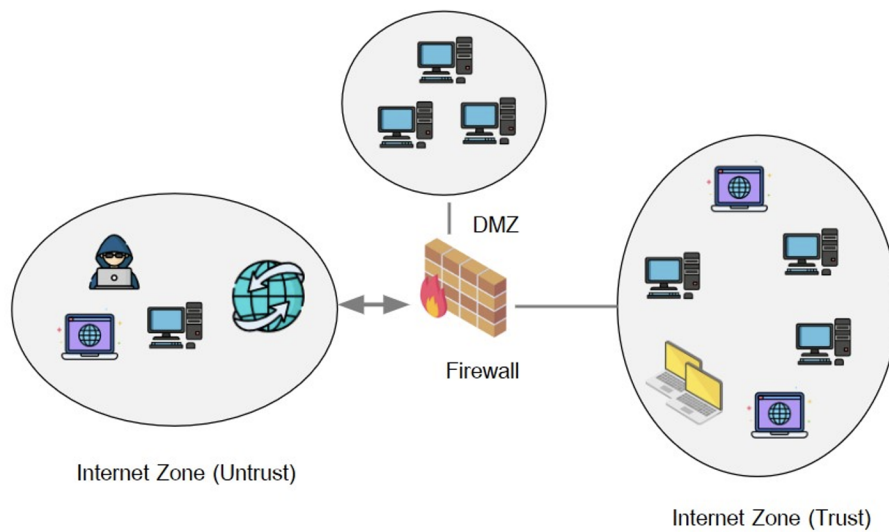


Figure 1: Schematic of network firewall connection.

Fig 1 illustrates the connection of the firewall within the network system, specifically highlighting the untrusted internet zone. The untrusted zone encompasses areas where data can freely enter and exit through global connections. In contrast, the trusted zone is confined within the organization, with access restricted to internal users. The demilitarized zone (DMZ) acts as a buffer for network servers providing public services, with certain parts accessible and others restricted, as controlled by the firewall [21].

To enhance firewall-based security, IDS provide an additional monitoring layer, available as hardware or software, designed to detect malicious activities and alert security personnel in real-time [22]. IDS can be classified into two main types: network-based IDS (NIDS), which monitors network traffic for suspicious behavior, and host-based IDS (HIDS), which tracks activities on individual devices [23]. Detection methodologies encompass signature-based approaches that match traffic against known attack patterns and anomaly-based techniques that identify deviations from baseline behavior [24]. While signature-based detection performs well at recognizing documented threats, anomaly-based detection proves more effective for novel or zero-day attacks.

Machine learning techniques have emerged as powerful tools for anomaly-based IDS, with algorithms including KNN, ANN, SVM, decision trees, random forest, and Naïve Bayes demonstrating enhanced detection accuracy by learning normal usage patterns and identifying abnormal activities [15, 16, 17, 18, 25]. This study integrates fundamental IDS concepts with firewall log analysis by employing machine learning to recognize patterns beyond explicit rule-based filtering. Our approach automatically learns from historical firewall events, combining efficient firewall log data with adaptable anomaly detection. This creates a scalable framework for identifying complex and evolving attack behaviors in network environments.

3.2 Multilayer Perceptron (MLP) Architecture

Firewall log classification can be formulated as a supervised multi-class learning problem. Let the dataset be denoted as

$$D = \{(x_i, y_i)\}_{i=1}^N \quad (1)$$

where $x_i \in \mathbb{R}^d$ represents the feature vector extracted from the i th log entry, and $y_i \in \{1, 2, \dots, K\}$ denotes the corresponding class label (e.g., allow, deny, drop, suspicious). The goal is to learn a mapping function $f: \mathbb{R}^d \rightarrow \{1, 2, \dots, K\}$ that minimizes the classification error on unseen data.

Statistical learning theory states that the ability of a model to generalize depends on model capacity, the characteristics of the training data, and the effectiveness of optimization in balancing empirical and structural risk [26, 27]. Firewall log data often contain highly non-linear attack patterns, including temporal dependencies and correlated features, which are difficult for linear classifiers to capture.

Neural networks address this limitation through the universal approximation theorem, which asserts that a feedforward network with sufficient hidden neurons and non-linear activation functions can approximate any continuous function [28, 29]. This theoretical guarantee makes neural networks particularly suitable for modeling the intricate relationships present in firewall log patterns, where the discriminative boundary between benign and malicious activities is rarely linear in the original feature space.

MLPs, a specific type of feedforward neural network, leverage one or more hidden layers to capture these complex patterns. Through backpropagation and gradient descent, MLPs iteratively adjust connection weights to minimize classification error and optimize performance [30]. This adaptability has proven effective across diverse domains, including chemical engineering [31], agriculture [32], and bioinformatics [33].

Furthermore, optimizing a neural network involves various strategies to enhance model performance, including hyperparameter tuning (such as adjusting learning rate, batch size, and selecting optimization algorithms), weight initialization, and applying regularization techniques (such as dropout, L2 regularization, and early stopping). Additionally, batch normalization and selecting suitable activation functions (such as linear, sigmoid, ReLU, and Tanh) are essential for effective training. Combining activation functions with dropout can help prevent overfitting and improve overall model performance [34, 35].

The MLP architecture consists of an input layer, hidden layer(s), and an output layer. Each neuron in layer l computes:

$$z^{(l)} = W^{(l)}x^{(l-1)} + b^{(l)} \quad (2)$$

where $z^{(l)}$ is the weighted sum in layer l , $W^{(l)}$ is the weight matrix for layer l , $x^{(l-1)}$ is the output from the previous layer $l-1$, and $b^{(l)}$ is the bias vector for layer l .

Next, the activation function is applied to the result in each layer, as described in Equation (3):

$$x^{(l)} = f(z^{(l)}) \quad (3)$$

where $x^{(l)}$ is the output (or activation) of the neuron in layer l , $f(\cdot)$ represents the activation function, and $z^{(l)}$ is the weighted sum of inputs to the neuron at layer l . In this study, the activation functions used include linear, sigmoid, Tanh, and ReLU [36, 37].

The structure of a neuron and the activation function in a perceptron is shown in Fig. 2. This figure illustrates the perceptron, a fundamental unit in neural networks, which receives multiple inputs, each assigned a weight that reflects its importance. The weighted inputs are summed, with a bias term added to adjust the activation threshold. This sum is then passed through an activation function, introducing non-linearity and enabling the perceptron to model complex patterns. Finally, the output is generated and either passed to subsequent layers or used for classification, depending on the position of the perceptron within the network.

This research primarily examines activation functions, which allow neural networks to learn complex relationships and non-linear patterns by transforming raw inputs into outputs that can be effectively processed, supporting the development of robust and efficient models. The output characteristics of the activation functions are illustrated in Fig. 3. This manuscript evaluates four activation functions, described below.

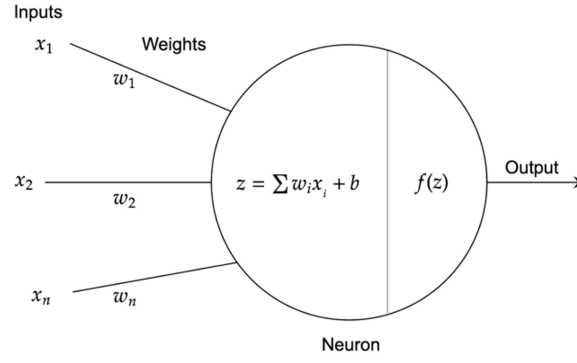


Figure 2: Structure of a neuron and activation function in a perceptron.

3.2.1 Linear Activation Function

The linear function, also known as the identity function, is the simplest activation function used in neural networks. It is a purely linear function, allowing the model to address problems where the relationship between inputs and outputs is strictly linear. The linear activation function is represented by Equation (4):

$$f(z) = z \quad (4)$$

3.2.2 Sigmoid Activation Function

The sigmoid function transforms inputs into values between 0 and 1, making it interpretable as a probability. The computation of the sigmoid function is shown in Equation (5):

$$f(z) = \frac{1}{1 + e^{-z}} \quad (5)$$

3.2.3 Hyperbolic Tangent (Tanh) Function

The Tanh function maps input values to a range between -1 and 1, offering advantages during training, particularly for handling negative input values. It is symmetric around the origin, making it effective for learning balanced data. The Tanh function is represented by Equation (6):

$$f(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (6)$$

where z is the input to the neuron and e is Euler's number, approximately equal to 2.718.

3.2.4 Rectified Linear Unit (ReLU) Function

The ReLU function outputs the input directly if it is positive; otherwise, it outputs zero. This activation selectively activates neurons with positive inputs while deactivating those with negative inputs, making the network sparse and enhancing computational efficiency [38]. The ReLU function is represented by Equation (7):

$$f(z) = \max(0, z) \quad (7)$$

where z is the input to the neuron. If $z > 0$, the output is z ; otherwise, if $z \leq 0$, the output is 0.

4 Experimental Results

4.1 Firewall Log File Dataset

In 2018, Ertam and Kaya [15] collected firewall log data using a Palo Alto 5020 Firewall device installed at Firat University. The dataset comprised 65,532 records, logged every 30 seconds, and categorized into four classes: allow, deny, drop, and reset-both. Each record included essential information such as port, packet, byte, and time data, totaling 11 features. Detailed feature and class descriptions are presented in Table 1 and Table 2, while Table 2 also visualizes the number of records by class.

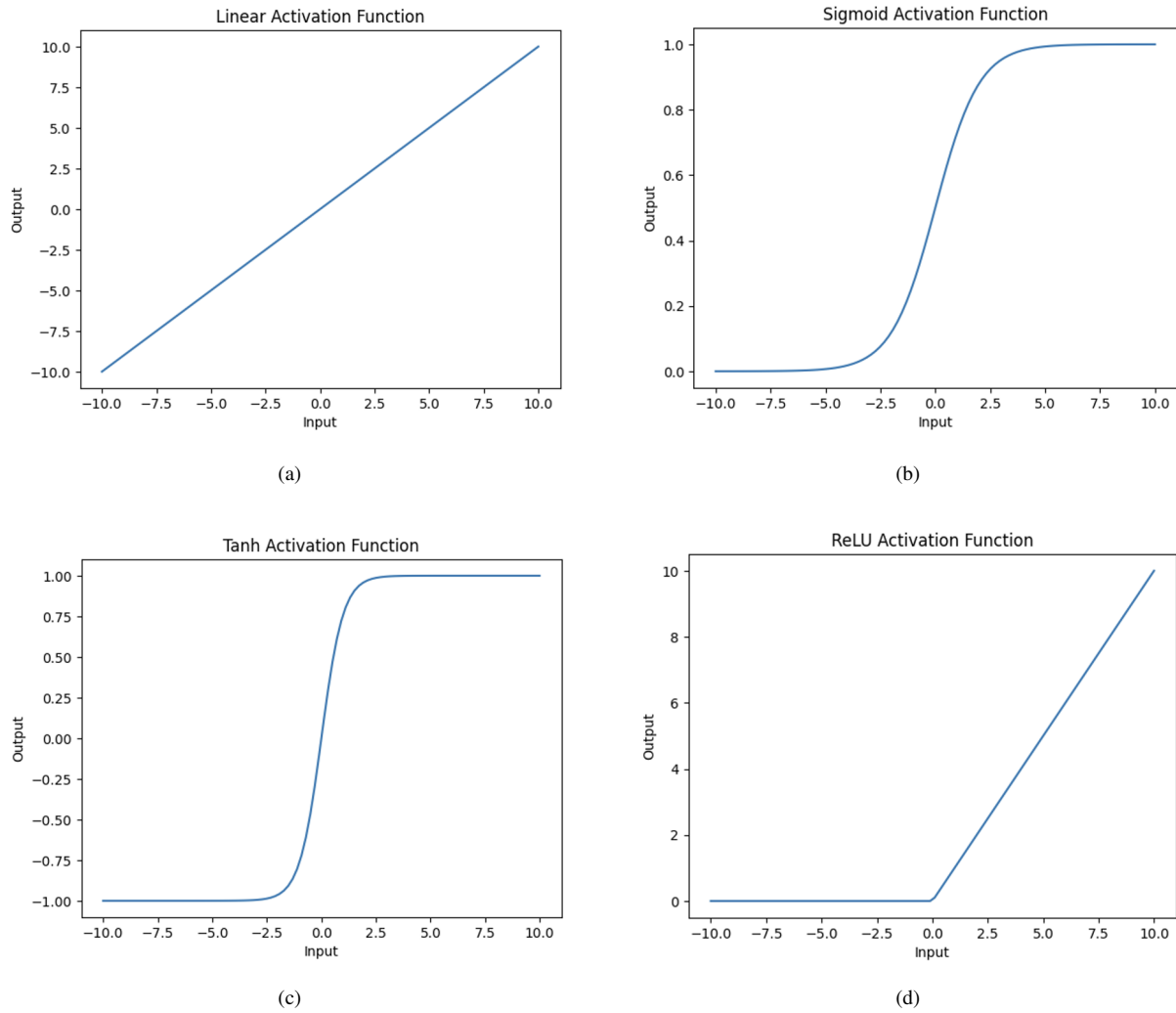


Figure 3: Comparison of activation functions: (a) Linear, (b) Sigmoid, (c) Tanh, and (d) ReLU.

Table 2 provides a summary of firewall operation data, categorizing actions taken by the firewall during network activity into four distinct classes: allow, drop, deny, and reset-both. The Allow class, representing approved data transmissions, is the most frequent, with 37,640 records. The drop class, where data transmission is blocked, accounts for 14,987 records. The deny class, involving TCP resets without a response, includes 12,851 records. Finally, the reset-both class, where TCP is reset while data is sent to both client and server, is the least frequent with only 54 records.

4.2 Experimental Settings

This study aims to identify the optimal parameters for model construction and to evaluate the performance of neural network models. The MLP framework was implemented on Google Colab and developed in Python, utilizing the scikit-learn machine learning library.

In this framework, the Firewall Log File dataset was split into training and test sets in a 7:3 ratio. The training set was used to build the model, and the test set was employed to evaluate its performance. The research framework is illustrated in Fig. 4.

To determine optimal parameters, the model was trained using the Adam optimizer, chosen for its low memory requirements. Experiments were conducted with a learning rate of 0.001, a momentum of 0.9, and a dropout rate of 0.25. Various numbers of neurons in the hidden layer were tested, including 10, 20, 50, 100, 200, 500, and 1,000.

As illustrated in Fig. 5, the proposed MLP model comprises three layers: input, hidden, and output. The input layer has 11 nodes ($\in \mathbb{R}^{11}$), the output layer has four nodes ($\in \mathbb{R}^4$), and the hidden layer contains N nodes ($\in \mathbb{R}^N$), with N varying based on the experimental setup.

Table 1: Characteristics of the Firewall Log File Dataset

Feature	Description
Source port	Client source port
Destination port	Client destination port
NAT source port	Network address translation source port
NAT destination port	Network address translation destination port
Elapsed time	Elapsed time for flow
Bytes	Total bytes
Bytes sent	Number of bytes sent
Bytes received	Number of bytes received
Packets	Total packets
Packets sent	Number of packets sent
Packets received	Number of packets received

Table 2: Firewall Operation Classes

Class	Description	Records
Allow	Allow data transmission	37,640
Drop	Block data transmission	14,987
Deny	Reset TCP with no response	12,851
Reset-both	Reset TCP with sending data to client servers	54

4.3 t-Distributed Stochastic Neighbor Embedding (t-SNE) Visualization

In this study, t-SNE was used as a dimensionality reduction technique and an effective tool for visualizing high-dimensional data into two dimensions [39]. Through t-SNE, the firewall log file dataset, containing 11 features, was represented in a two-dimensional space. As shown in Fig. 6, anomalies were detected, including clusters of deny class records appearing within the allow class. Additionally, some instances of the reset-both class were observed within the deny class, presenting challenges for accurate classification.

4.4 Imbalanced Data Preprocessing

We observed that the Firewall Log File dataset is imbalanced. As shown in Fig. 7, the distribution of firewall operations reveals that most actions allow data transmission, while fewer instances involve resetting connections (reset-both class) or blocking transmissions (deny and drop classes).

After dividing the dataset into training and test sets, a significant class imbalance was observed. The reset-both class contained only 39 records in the training set, whereas the deny and drop classes had 10,021 and 10,502 records, respectively. Additionally, the allow class had approximately 670 times more data than the reset-both class.

To address this imbalance, we applied the SMOTE [40, 41, 42], an oversampling method that generates synthetic samples to balance minority classes with majority classes. In this study, we aimed to increase the Reset-both class to 1,052 records without fully matching the levels of the major classes (around 10,000 records). Consequently, we randomly selected 1,050 records from the deny class, 902 from the drop class, and 1,052 from the allow class, totaling 3,004 records. After applying SMOTE, the reset-both class was expanded to 1,052 records, which were then added to the training set.

The t-SNE visualization in Fig. 8(b) illustrates the increased data distribution in the reset-both class after applying the SMOTE technique.

4.5 Evaluation Metrics

We evaluated the performance of the MLP models for firewall log file classification using several metrics, including accuracy, precision, recall, and F_1 -score [16, 35, 37, 43, 44]. A brief description of these evaluation metrics is provided below:

Accuracy refers to the ratio of correctly classified instances to the total number of instances in the dataset. It provides an overall measure of model performance.

Precision focuses on the proportion of positive classifications that are actually correct, reflecting the quality of positive predictions.

Recall measures how well the model identifies positive instances, emphasizing the model's ability to detect true positives.

F_1 is used to evaluate the performance of a classification model, particularly in cases of imbalanced datasets. It is especially useful when both precision and recall are critical to the model's effectiveness.

The equations for the metrics are as follows:

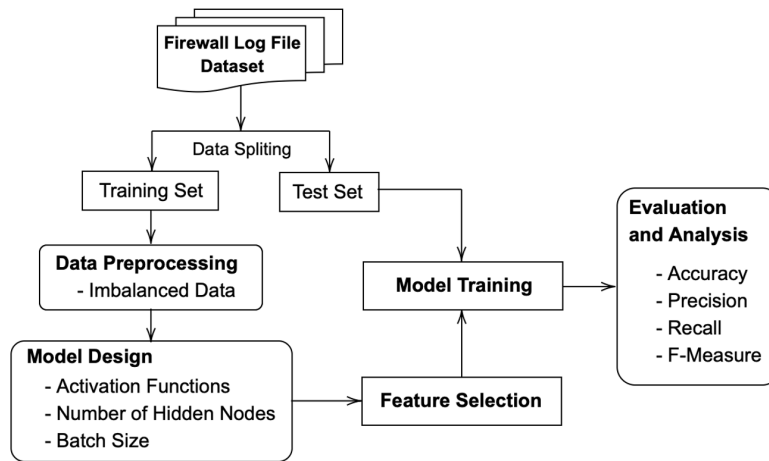


Figure 4: Framework of the MLP for firewall log file classification.

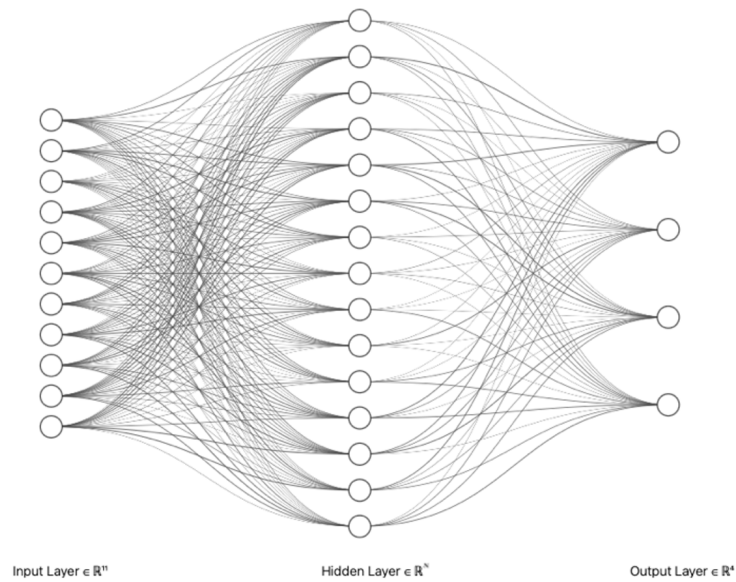


Figure 5: The proposed MLP model for firewall log file classification.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (8)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (9)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (10)$$

$$F_1 = \frac{2 \cdot (\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}} \quad (11)$$

where TP , TN , FP , and FN represent true positives, true negatives, false positives, and false negatives, respectively.

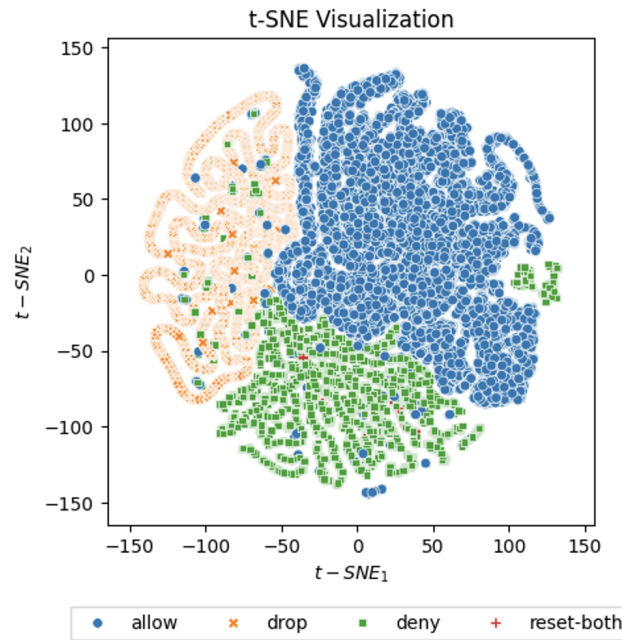


Figure 6: Visualization of the Firewall Log File dataset using t-SNE for dimensionality reduction.

Table 3: Classification performance of MLP models on the Firewall Log File dataset, showing average accuracy (%) on the validation set, $\pm$ standard deviation, and test accuracy (%). Bold numbers indicate the best model performance

Hidden Nodes	Activation Functions							
	Linear		Sigmoid		ReLU		Tanh	
	Validation	Test	Validation	Test	Validation	Test	Validation	Test
10	92.43 $\pm$ 0.01212	94.95	95.85 $\pm$ 0.00042	97.82	96.21 $\pm$ 0.00529	98.01	88.47 $\pm$ 0.10321	97.36
20	95.42 $\pm$ 0.01061	97.93	95.73 $\pm$ 0.00554	97.10	96.50 $\pm$ 0.01278	99.44	92.22 $\pm$ 0.04729	98.02
50	94.06 $\pm$ 0.01534	95.76	95.97 $\pm$ 0.00307	98.33	97.68 $\pm$ 0.00501	99.75	95.59 $\pm$ 0.00541	96.98
100	96.06 $\pm$ 0.00622	98.34	95.97 $\pm$ 0.00202	97.82	97.56 $\pm$ 0.01037	99.79	96.03 $\pm$ 0.00266	97.76
200	96.29 $\pm$ 0.00123	98.28	96.91 $\pm$ 0.00335	98.36	97.78 $\pm$ 0.00407	99.81	95.77 $\pm$ 0.00468	97.68
500	95.66 $\pm$ 0.01489	95.55	95.81 $\pm$ 0.00766	98.16	97.98 $\pm$ 0.00112	99.49	95.15 $\pm$ 0.01149	97.98
1000	95.77 $\pm$ 0.00831	98.52	96.12 $\pm$ 0.00246	98.12	97.76 $\pm$ 0.00276	99.66	95.43 $\pm$ 0.00985	98.25
Average	95.09	97.08	96.85	97.96	97.35	99.42	94.09	97.72

4.6 Experimental Results

4.6.1 Experiments on Optimizing Multilayer Perceptron Models

This research focused on optimizing of MLP models through the implementation of four activation functions: linear, sigmoid, Tanh, and ReLU. The evaluation process utilized the Firewall Log File dataset. The dataset exhibited class imbalance across four output classes: allow (37,640 instances), deny (14,987 instances), drop (12,851 instances), and reset-both (54 instances). The experimental evaluation analyzed multiple parameters of the MLP model, emphasizing hidden node configurations and activation functions.

Table 3 presents a comparative analysis of accuracy (%) results for the MLP model. The evaluation utilized the hold-out method with three experimental iterations on the training (validation $\pm$ standard deviation) and test sets across seven hidden node configurations (10, 20, 50, 100, 200, 500, and 1,000) and four activation functions: linear, sigmoid, ReLU, and Tanh.

The experimental results indicate that the MLP model implementing the ReLU activation function demonstrated superior performance compared to alternative activation functions (linear, sigmoid, and Tanh) across seven hidden node configurations for both validation and test sets.

In configurations with ten hidden nodes, ReLU achieved the highest test accuracy (98.01%), while the Linear function exhibited the lowest performance (94.95%). As the number of hidden nodes increased, ReLU maintained consistent superior performance, achieving maximum accuracy of 99.81% with 200 hidden nodes. In contrast, linear function demonstrated relatively lower performance, achieving 96.29% accuracy with the same node configuration. The sigmoid, Tanh, and linear functions exhibited comparable performance patterns, maintain average test accuracies consistently approximately 97%.

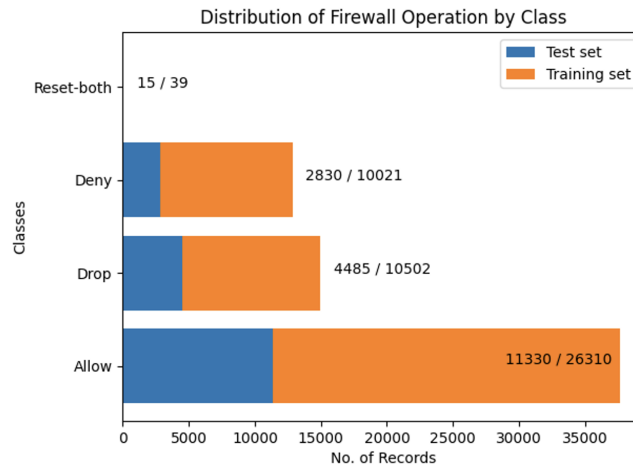


Figure 7: Visualization of firewall operation distribution by class.

Table 4: Classification performance (precision, recall, F_1) of MLP with different activation functions using 200 hidden nodes

Activation Function	Class	Precision (%)	Recall (%)	F_1 (%)
Linear	Allow	99.70	99.67	99.68
	Deny	99.69	80.40	89.01
	Drop	94.56	100.00	97.20
	Reset-both	0.30	13.33	0.58
Sigmoid	Allow	99.95	98.98	99.46
	Deny	99.03	94.15	96.54
	Drop	91.80	100.00	95.72
	Reset-both	0.00	0.00	0.00
ReLU	Allow	99.93	99.95	99.94
	Deny	99.68	97.01	98.33
	Drop	98.79	100.00	99.39
	Reset-both	5.81	33.33	9.90
Tanh	Allow	99.91	98.78	99.34
	Deny	98.74	93.17	95.88
	Drop	90.60	100.00	95.07
	Reset-both	0.00	0.00	0.00

The comparative analysis established ReLU as the most effective activation function for this classification task, particularly in configurations with 200 hidden nodes. The performance ranking of activation functions, based on accuracy metrics, positioned sigmoid, Tanh, and linear in descending order of effectiveness.

The confusion matrices shown in Fig. 9 provide an in-depth analysis of firewall operation classification for each MLP model with different activation functions. The confusion matrix for the ReLU activation function (see Fig. 9(d)) demonstrates accurate classification across all classes. The model correctly classified 11,325 instances of allow, 4,464 of deny, and 3,830 of drop.

Although the training set included only a few records of the reset-both class, the MLP model with the ReLU function was still able to learn the pattern and correctly classify three records. Misclassifications were minimal, with only a few errors in the deny and reset-both classes, indicating a high level of classification accuracy.

Table 4 presents a comparative analysis of the classification performance, measured by precision, recall, and F_1 , for a MLP model employing four distinct activation functions. The results indicate that the ReLU function achieved the highest performance for the allow, deny, and drop classes, with precision and recall exceeding 98% and an F_1 reaching 99.94% for allow class. However, the reset-both class exhibited significantly lower performance across all activation functions, with ReLU achieving a slightly higher F_1 of 9.90% compared to near-zero values for the other functions. In conclusion, the ReLU function provided the best classification performance, particularly for the major classes.

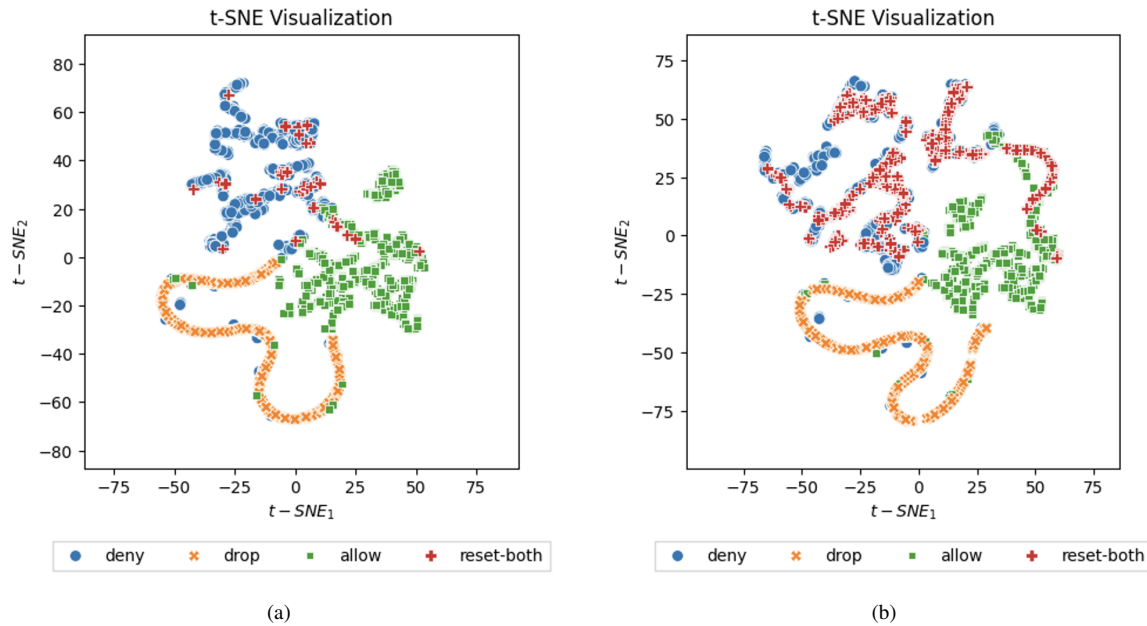


Figure 8: Visualization of the reset-both class data (a) before and (b) after applying the SMOTE technique using t-SNE.

4.6.2 Experiments on Feature Selection Techniques

We conducted experiments evaluating three feature selection methods: correlation analysis and feature importance (via random forest and XGBoost).

For the correlation analysis, we generated a triangle correlation heatmap to visualize the pairwise correlation coefficients between variables in the Firewall Log File dataset. As illustrated in Fig. 10, the heatmap revealed strong positive correlations (indicated by lighter colors) between several features, including bytes, bytes sent, bytes received, packets, packets sent, and packets received.

Additionally, we employed the random forest and XGBoost feature importance methods to compute relevance scores for each feature. The resulting importance scores are presented in Fig. 11(a) and Fig. 11(b), respectively. Based on these scores, the random forest approach selected eight features, while the XGBoost method identified five relevant features.

Table 5: Comparison of features selected by different feature selection methods

Feature	Correlation	Random Forest	XGBoost
Source port		✓	
Destination port		✓	✓
NAT source port		✓	✓
NAT destination port		✓	
Elapsed time		✓	✓
Bytes	✓	✓	✓
Bytes sent	✓	✓	
Bytes received	✓	✓	✓
Packets	✓		
Packets sent	✓		
Packets received	✓		

Table 5 summarizes the features identified through correlation analysis and feature importance evaluation using random forest and XGBoost methods. The MLP models were subsequently trained with these selected features. Each model employed 200 hidden nodes and the ReLU activation function, following the configuration determined in earlier experiments. Utilizing XGBoost-based feature selection reduced the feature set from eleven to five attributes while maintaining comparable accuracy, confirming that the reduced subset preserved sufficient discriminative information for effective firewall log classification.

The experimental results, as shown in Table 6, indicate that the feature selection method utilizing the XGBoost algorithm achieved an accuracy of 99.84% on the Firewall Log File dataset. In contrast, the correlation-based feature selection approach yielded a comparatively lower accuracy of 75%.

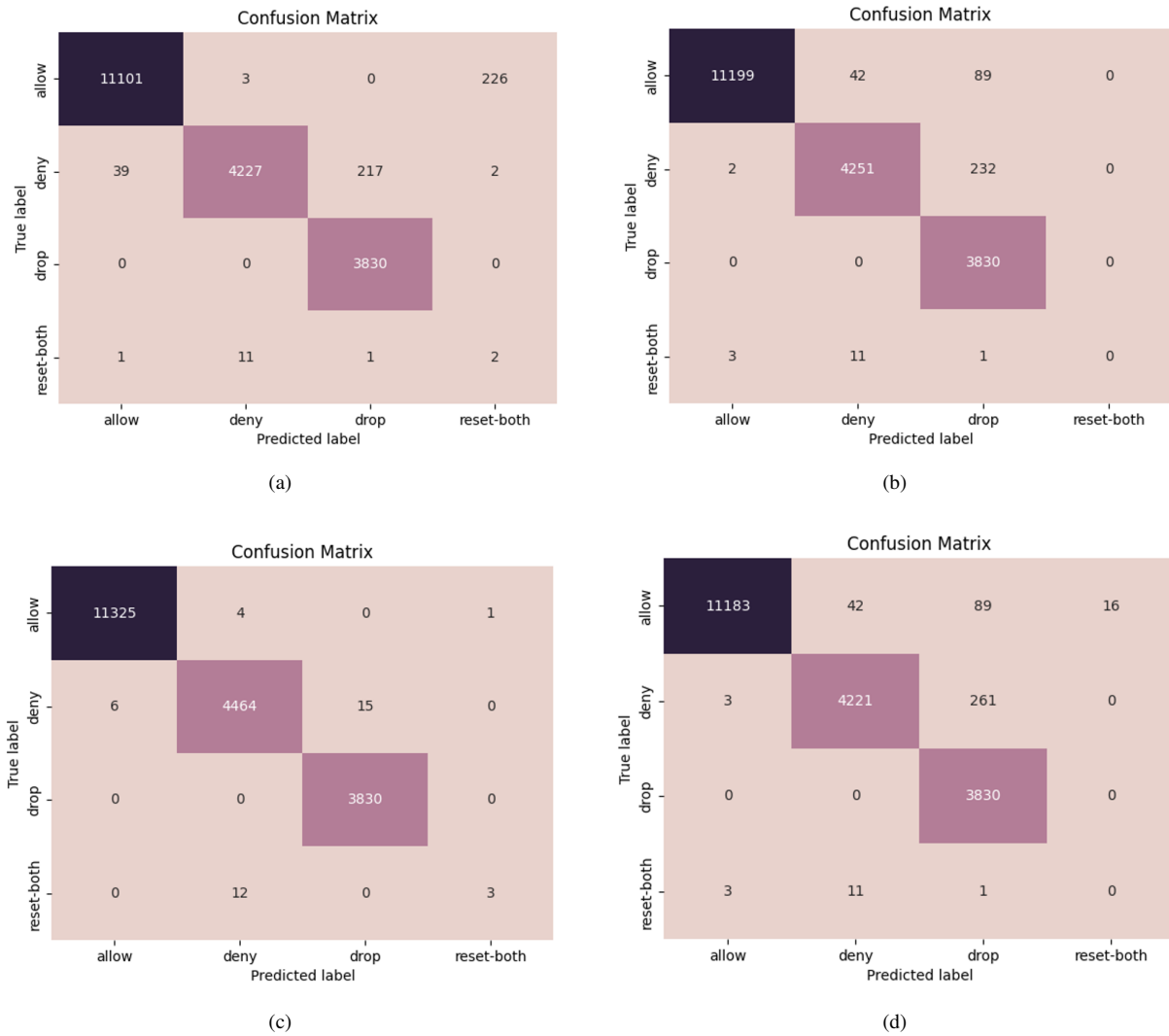


Figure 9: Confusion matrices for four activation functions: (a) Linear, (b) Sigmoid, (c) ReLU, and (d) Tanh.

4.6.3 Comparison with Existing Methods on the Firewall Log File Dataset

In this section, a comparative analysis is presented between the optimized MLP model configuration (200 hidden nodes and ReLU activation) and three existing machine learning methods: decision tree, random forest, and SVM.

As shown in Table 7, the comparative evaluation was conducted on the Firewall Log File dataset. The MLP model, with its optimal configuration identified in previous experiments, achieved an accuracy of 99.84%. In contrast, the decision tree and random forest algorithms demonstrated accuracies of 99.839% and 99.7%, respectively, while the SVM obtained 98.5% accuracy. These results indicate that the optimized MLP model outperformed the other existing techniques in classifying the firewall log data.

The optimized MLP architecture achieved accuracy levels comparable to intrusion detection models [2, 15, 17] while demonstrating measurable operational improvements. Consistent performance under varied hyperparameter configurations, in conjunction with effective mitigation of class imbalance through SMOTE-based oversampling, demonstrates strong generalization capability. Dimensionality reduction using XGBoost feature importance analysis decreased the feature space from eleven to five attributes with less than 0.5% difference in accuracy.

5 Discussion

Our study demonstrated that MLP models consistently achieved accuracy above 99% across four activation functions. Additionally, we monitored for potential overfitting and found that the models successfully addressed these challenges, even with imbalanced data. The detailed findings are presented below.

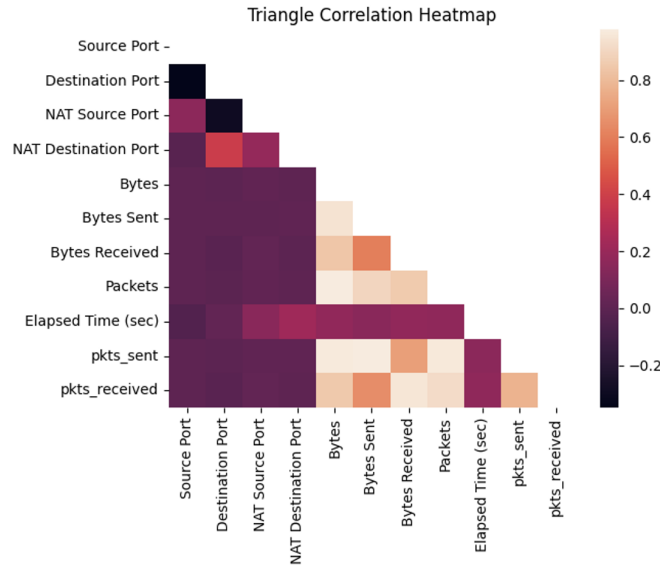


Figure 10: Feature selection using triangle correlation. Lighter colors indicate features with higher correlation.

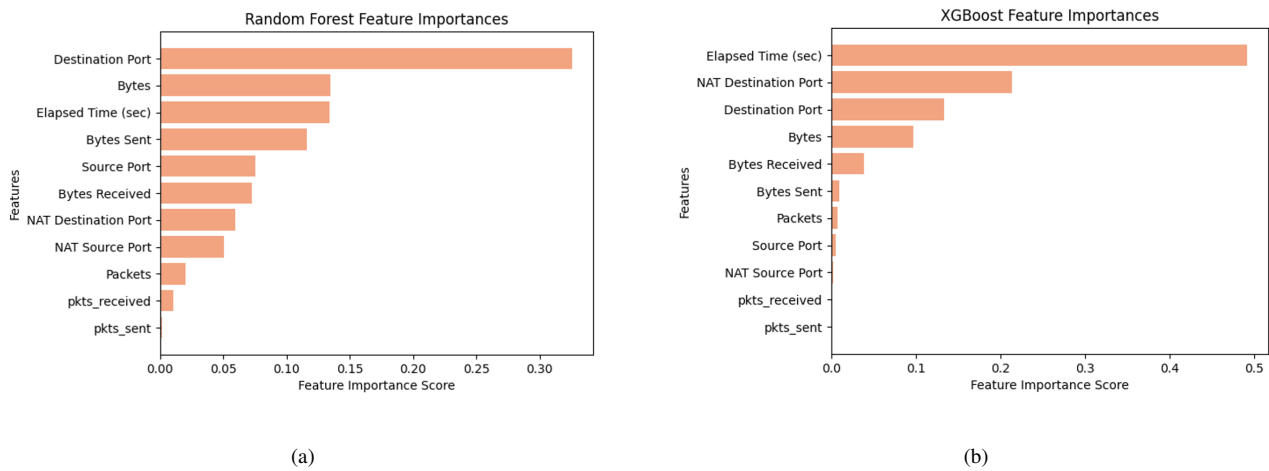


Figure 11: Feature selection with (a) random forest and (b) XGBoost methods.

5.1 Comparison with Conventional Firewall Capabilities

Existing firewalls, including NGFWs, provide essential log inspection and traffic classification functions based on predefined rules and signature-based detection. Although effective for identifying known attack patterns, these systems demonstrate reduced accuracy when analyzing complex or high-dimensional network data and lack adaptive mechanisms for addressing imbalanced log distributions. The proposed MLP framework addresses these limitations by modeling nonlinear feature interactions within firewall log data to support adaptive classification of majority and minority traffic classes. The incorporation of SMOTE balances class distributions, thereby improving detection accuracy for less frequent firewall log classes.

5.2 Imbalanced Data Problem

In this case, the imbalance in the Firewall Log File dataset is evident, with the reset-both class containing only 54 instances (see Fig. 6), approximately 267 and 697 times fewer than the drop and allow classes, respectively. In the test set (see Fig. 9(c)), only 3 out of 15 reset-both instances were correctly classified, with 12 misclassified as deny class.

The confusion matrix demonstrates that the proposed method addresses class imbalance effectively, achieving high true positive rates across all classes, including minority classes such as deny and reset-both. Despite the imbalance, the model maintains a low misclassification rate, as indicated by minimal off-diagonal values. The results indicate that the model accurately

Table 6: Model performance with different feature selection methods using the ReLU activation function

Feature Selection Method	Accuracy (%)
Correlation	75.11
Random Forest	99.57
XGBoost	99.84

Table 7: Classification performance on the Firewall Log File dataset compared with existing methods

Reference	Method	Accuracy (%)
[17]	Decision Tree	99.839
[2]	Random Forest	99.70
[15]	Support Vector Machine	98.50
Our Proposed	MLP (200 hidden nodes and ReLU)	99.84

identifies instances of minority classes (deny and drop) without defaulting heavily to the majority class (allow) when adequate instances are available. To address this, we used SMOTE to increase the reset-both class from 29 to 1,052 records. However, the model struggled with imbalance when the minority class contained very few records.

The balanced distribution of correct predictions across classes suggests that the model achieves strong precision and recall for each class. Techniques such as class weighting, resampling, or specialized algorithms for imbalanced data appear to have contributed to this robust performance, reducing bias toward the more frequent classes.

5.3 Convergence and Overfitting Problem

The accuracy and loss graphs in Fig. 12 illustrate model performance over 200 epochs for both training and test sets. The accuracy graph (Fig. 12(a)) indicates rapid improvement during the initial epochs, surpassing 98% around epoch 25 and approaching 99% thereafter, signaling stable performance with minimal overfitting. Similarly, the loss graph (Fig. 12(b)) shows a sharp decline in the early epochs, stabilizing near zero for both sets, which reflects effective error minimization. The close alignment and convergence of training and test curves in both graphs suggests strong generalization, indicating that the model effectively balances learning from training data while maintaining performance on unseen data.

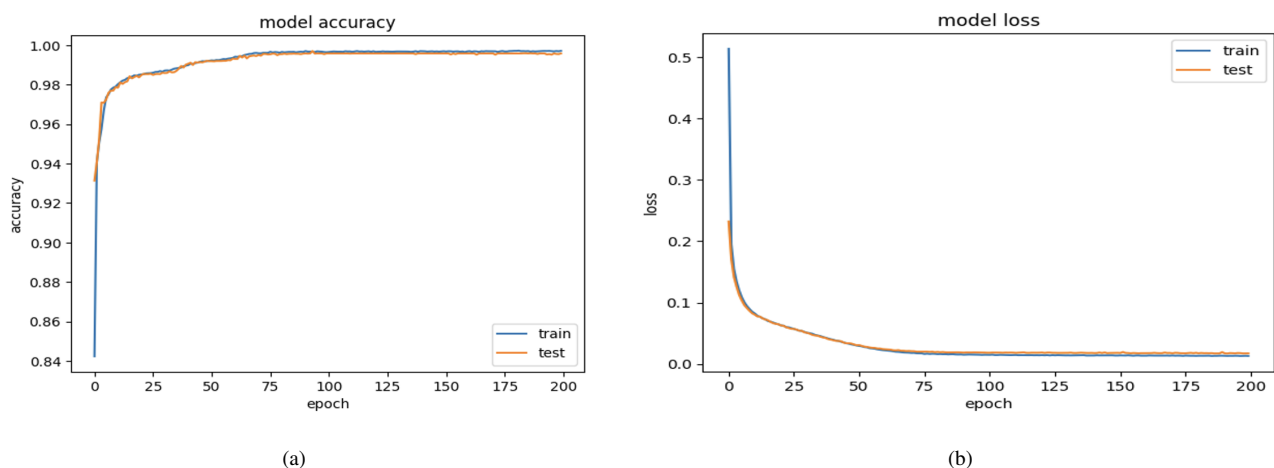


Figure 12: Illustration of (a) accuracy and (b) loss graphs over epochs during training with the ReLU activation function.

5.4 Optimize the Activation Function for Multilayer Perceptron Model

In our study, we optimized the proposed MLP model by experimenting with four activation functions. The ReLU activation function, as illustrated in Fig. 13, consistently achieved the highest accuracy across various hidden node configurations, frequently surpassing 99%. This performance emphasizes the effectiveness of ReLU in handling complex, non-linear patterns within the data. Additionally, ReLU enables the model to converge more rapidly and effectively during training. In contrast, the linear activation function exhibits significant fluctuations, particularly a marked drop in accuracy at 500 hidden nodes. This instability suggests that the Linear function struggles to capture non-linear relationships, limiting its reliability in this context.

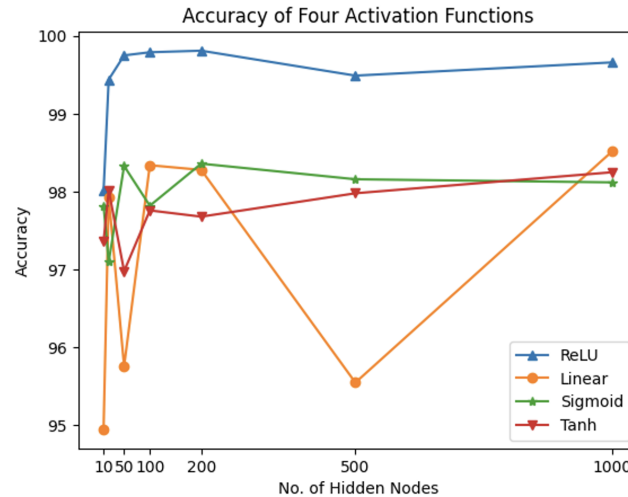


Figure 13: Accuracy comparison of four activation functions (Linear, Sigmoid, ReLU, and Tanh) for different hidden layer sizes.

6 Conclusion

This study examined the effectiveness of multilayer perceptron (MLP) neural networks for classifying firewall log data, emphasizing activation function selection, feature optimization, and data imbalance reduction. The optimized MLP model consistently achieved high performance, with accuracy exceeding 99% across configurations, while the ReLU activation function produced the most reliable results for major traffic classes. Applying the synthetic minority oversampling technique enhanced the classification accuracy of the reset-both class and improved overall model balance, a challenge commonly observed in traditional firewall systems. Moreover, employing XGBoost-based feature selection reduced the feature set from eleven to five attributes, achieving 99.84% accuracy while improving computational efficiency. Evaluation metrics, including precision, recall, and F_1 , confirmed the robustness of the proposed approach in capturing complex, nonlinear traffic behaviors. The MLP-based framework demonstrates significant advantages over conventional rule-based firewalls by providing higher classification accuracy, better handling of imbalanced data.

Future work may focus on advanced optimization techniques, alternative oversampling methods to achieve better class balance, and validation on diverse datasets to enhance model generalizability. Integrating the MLP with ensemble learning could further improve robustness, while adaptive modeling approaches may increase responsiveness to evolving cyber threats in real-time network environments. Although this study primarily employed the Firewall Log File dataset, the proposed MLP architecture has strong potential for generalization to other intrusion detection contexts. Benchmark datasets such as NSL-KDD [45] and CIC-IDS2017 [46] contain similar log structures and feature representations, making them suitable for future testing to assess robustness under varied network conditions. From an implementation perspective, the framework could also be incorporated into next-generation firewalls [47] or security information and event management (SIEM) [48] systems to enable real-time anomaly detection and classification in operational settings.

CRedit Authorship Contribution Statement

Artitayaporn Rojarath: Conceptualization, Methodology, Resources, Validation, Formal analysis, Investigation, Writing - Original draft, Funding acquisition **Nantane Srisaengchan:** Methodology, Software, Investigation, Data curation, Formal analysis, Writing - original draft **Saisunee Jabjone:** Validation, Formal analysis **Suda Tipprasert:** Validation, Formal analysis **Olarik Surinta:** Conceptualization, Methodology, Validation, Formal analysis, Writing - review & editing, Supervision.

Data Availability Statement

The data that support the findings of this study are openly available in Kaggle at <https://www.kaggle.com/datasets/tunguz/internet-firewall-data-set>, reference number [15].

Acknowledgements

This research project was financially supported by Faculty of Informatics, Mahasarakham University, Thailand.

Declaration of Generative AI

During the preparation of this manuscript, the authors used Grammarly, ChatGPT, and Claude to improve readability and language. The authors reviewed and edited all content and take full responsibility for the publication.

Conflicts on Interest

The authors declare no conflicts of interest.

References

- [1] A. Pinto, L.-C. Herrera, Y. Donoso, and J. A. Gutierrez, "Survey on intrusion detection systems based on machine learning techniques for the protection of critical infrastructure," *Sensors*, vol. 23, no. 5, p. 2415, Feb. 2023. [Online]. Available: <https://doi.org/10.3390/s23052415>
- [2] B. A. Al-Tarawneh and H. Bani-Salameh, "Classification of firewall logs actions using machine learning techniques and deep neural network," in *the 4th International Computer Sciences and Informatics Conference (ICSIC)*. AIP Publishing, 2023, p. 050003. [Online]. Available: <https://doi.org/10.1063/5.0174750>
- [3] A. Korkmaz, S. Bulut, T. Talan, S. Kosunalp, and T. Iliev, "Enhancing firewall packet classification through artificial neural networks and synthetic minority over-sampling technique: An innovative approach with evaluative comparison," *Applied Sciences*, vol. 14, no. 16, p. 7426, Aug. 2024. [Online]. Available: <https://doi.org/10.3390/app14167426>
- [4] S. J. Moore, F. Cruciani, C. D. Nugent, S. Zhang, I. Cleland, and S. Sani, "Deep learning for network intrusion: A hierarchical approach to reduce false alarms," *Intelligent Systems with Applications*, vol. 18, p. 200215, May 2023. [Online]. Available: <https://doi.org/10.1016/j.iswa.2023.200215>
- [5] M. Patel, P. Amritha, V. B. Sudheer, and M. Sethumadhavan, "DDoS attack detection model using machine learning algorithm in next generation firewall," *Procedia Computer Science*, vol. 233, p. 175–183, 2024. [Online]. Available: <https://doi.org/10.1016/j.procs.2024.03.207>
- [6] S. S. Bamber, A. V. R. Katkuri, S. Sharma, and M. Angurala, "A hybrid CNN-LSTM approach for intelligent cyber intrusion detection system," *Computers & Security*, vol. 148, p. 104146, Jan. 2025. [Online]. Available: <https://doi.org/10.1016/j.cose.2024.104146>
- [7] R. A. Abed, E. K. Hamza, and A. J. Humaidi, "A modified CNN-IDS model for enhancing the efficacy of intrusion detection system," *Measurement: Sensors*, vol. 35, p. 101299, Oct. 2024. [Online]. Available: <https://doi.org/10.1016/j.measen.2024.101299>
- [8] K. Suresh, K. Jayasakthi Velmurugan, R. Vidhya, S. Rahini sudha, and Kavitha, "Deep anomaly detection: A linear one-class SVM approach for high-dimensional and large-scale data," *Applied Soft Computing*, vol. 167, p. 112369, Dec. 2024. [Online]. Available: <https://doi.org/10.1016/j.asoc.2024.112369>
- [9] S. Muruganandam, R. Joshi, P. Suresh, N. Balakrishna, K. H. Kishore, and S. Manikanthan, "A deep learning based feed forward artificial neural network to predict the K-barriers for intrusion detection using a wireless sensor network," *Measurement: Sensors*, vol. 25, p. 100613, Feb. 2023. [Online]. Available: <https://doi.org/10.1016/j.measen.2022.100613>
- [10] J. F. Cevallos M., A. Rizzardi, S. Sicari, and A. Coen Porisini, "Deep reinforcement learning for intrusion detection in internet of things: Best practices, lessons learnt, and open challenges," *Computer Networks*, vol. 236, p. 110016, Nov. 2023. [Online]. Available: <https://doi.org/10.1016/j.comnet.2023.110016>
- [11] W. Lim, K. S. C. Yong, B. T. Lau, and C. C. L. Tan, "Future of generative adversarial networks (GAN) for anomaly detection in network security: A review," *Computers & Security*, vol. 139, p. 103733, Apr. 2024. [Online]. Available: <https://doi.org/10.1016/j.cose.2024.103733>

- [12] M. L. Ali, K. Thakur, S. Schmeelk, J. DeBello, and D. Dragos, "Deep learning vs. machine learning for intrusion detection in computer networks: A comparative study," *Applied Sciences*, vol. 15, no. 4, p. 1903, Feb. 2025. [Online]. Available: <https://doi.org/10.3390/app15041903>
- [13] K. S. Adewole, A. Jacobsson, and P. Davidsson, "Intrusion detection framework for internet of things with rule induction for model explanation," *Sensors*, vol. 25, no. 6, p. 1845, Mar. 2025. [Online]. Available: <https://doi.org/10.3390/s25061845>
- [14] J. Khan, R. Elfakharany, H. Saleem, M. Pathan, E. Shahzad, S. Dhou, and F. Aloul, "Can machine learning enhance intrusion detection to safeguard smart city networks from multi-step cyberattacks?" *Smart Cities*, vol. 8, no. 1, p. 13, Jan. 2025. [Online]. Available: <https://doi.org/10.3390/smartcities8010013>
- [15] F. Ertam and M. Kaya, "Classification of firewall log files with multiclass support vector machine," in *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*. IEEE, Mar. 2018, p. 1–4. [Online]. Available: <https://doi.org/10.1109/isdfs.2018.8355382>
- [16] T. Roempluk and O. Surinta, "A machine learning approach for detecting distributed denial of service attacks," in *2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT-NCON)*. IEEE, Jan. 2019, p. 146–149. [Online]. Available: <https://doi.org/10.1109/ecti-ncon.2019.8692243>
- [17] H. Al-Behadili, "Decision tree for multiclass classification of firewall access," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 3, p. 294–302, Jun. 2021. [Online]. Available: <https://doi.org/10.22266/ijies2021.0630.25>
- [18] M. Aljabri, A. A. Alahmadi, R. M. A. Mohammad, M. Aboulmour, D. M. Alomari, and S. H. Almotiri, "Classification of firewall log data using multiclass machine learning models," *Electronics*, vol. 11, no. 12, p. 1851, Jun. 2022. [Online]. Available: <https://doi.org/10.3390/electronics11121851>
- [19] J. M. Kizza, *Firewalls*. Springer International Publishing, 2024, p. 265–294. [Online]. Available: https://doi.org/10.1007/978-3-031-47549-8_12
- [20] R. S. Yadav and P. Likhar, *Firewall: A Vital Constituent of Network Security*. Springer Nature Singapore, 2024, p. 47–67. [Online]. Available: https://doi.org/10.1007/978-981-97-0407-1_3
- [21] R. J. Shimonski, W. Schmied, T. W. Shinder, V. Chang, D. Simonis, and D. Imperatore, *Firewall and DMZ Design*. Elsevier, 2003, p. 321–368. [Online]. Available: <https://doi.org/10.1016/b978-193183688-3/50011-0>
- [22] A. Khraisat, I. Gondal, P. Vamplew, and J. Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 2, no. 1, p. 20, Jul. 2019. [Online]. Available: <https://doi.org/10.1186/s42400-019-0038-7>
- [23] F. A. P. Kuswara, H. H. Nuha, and V. Suryani, "Intrusion detection system using incremental learning method," in *2023 11th International Conference on Information and Communication Technology (ICoICT)*. IEEE, Aug. 2023, p. 588–593. [Online]. Available: <https://doi.org/10.1109/icoict58202.2023.10262799>
- [24] G. Long and Z. Zhang, "PUNet: A semi-supervised anomaly detection model for network anomaly detection based on positive unlabeled data," *Computers, Materials & Continua*, vol. 81, no. 1, p. 327–343, Oct. 2024. [Online]. Available: <https://doi.org/10.32604/cmc.2024.054558>
- [25] S. Ransewa, N. Elz, N. Thanon, and S. Intajag, "Anomaly detection using source port data with Shannon entropy and EWMA control chart," in *2018 18th International Conference on Control, Automation and Systems (ICCAS)*, Oct. 2018, pp. 596–601. [Online]. Available: <https://ieeexplore.ieee.org/document/8571559>
- [26] V. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks*, vol. 10, no. 5, pp. 988–999, 1999. [Online]. Available: <https://doi.org/10.1109/72.788640>
- [27] L. Oneto, S. Ridella, and D. Anguita, "Informed machine learning: Excess risk and generalization," *Neurocomputing*, vol. 646, p. 130521, Sep. 2025. [Online]. Available: <https://doi.org/10.1016/j.neucom.2025.130521>
- [28] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks," *Neural Networks*, vol. 2, no. 3, p. 183–192, Jan. 1989. [Online]. Available: [https://doi.org/10.1016/0893-6080\(89\)90003-8](https://doi.org/10.1016/0893-6080(89)90003-8)
- [29] W. Kumagai, A. Sannai, and M. Kawano, "Universal approximation with neural networks on function spaces," *Journal of Experimental & Theoretical Artificial Intelligence*, vol. 36, no. 7, p. 1089–1100, Sep. 2022. [Online]. Available: <https://doi.org/10.1080/0952813x.2022.2123563>

- [30] A. Waheed, F. Subhan, M. Mohd Su'ud, and M. Mansoor Alam, "Molding robust S-box design based on linear fractional transformation and multilayer perceptron: Applications to multimedia security," *Egyptian Informatics Journal*, vol. 26, p. 100480, Jun. 2024. [Online]. Available: <https://doi.org/10.1016/j.eij.2024.100480>
- [31] I. Shomope, M. Tawalbeh, A. Al-Othman, and F. Almomani, "Predicting biohydrogen production from dark fermentation of organic waste biomass using multilayer perceptron artificial neural network (MLP-ANN)," *Computers & Chemical Engineering*, vol. 192, p. 108900, Jan. 2025. [Online]. Available: <https://doi.org/10.1016/j.compchemeng.2024.108900>
- [32] A. Singh, M. Imtiyaz, R. Isaac, and D. Denis, "Comparison of soil and water assessment tool (SWAT) and multilayer perceptron (MLP) artificial neural network for predicting sediment yield in the Nagwa agricultural watershed in Jharkhand, India," *Agricultural Water Management*, vol. 104, p. 113–120, Feb. 2012. [Online]. Available: <https://doi.org/10.1016/j.agwat.2011.12.005>
- [33] M. S. Tan, P.-L. Cheah, A.-V. Chin, L.-M. Looi, and S.-W. Chang, "A review on omics-based biomarkers discovery for Alzheimer's disease from the bioinformatics perspectives: Statistical approach vs machine learning approach," *Computers in Biology and Medicine*, vol. 139, p. 104947, Dec. 2021. [Online]. Available: <https://doi.org/10.1016/j.compbiomed.2021.104947>
- [34] R. C. Gerum, A. Erpenbeck, P. Krauss, and A. Schilling, "Sparsity through evolutionary pruning prevents neuronal networks from overfitting," *Neural Networks*, vol. 128, p. 305–312, Aug. 2020. [Online]. Available: <https://doi.org/10.1016/j.neunet.2020.05.007>
- [35] O. Eric, R.-M. O. M. Gyening, O. Appiah, K. Takyi, and P. Appiahene, "Cocoa beans classification using enhanced image feature extraction techniques and a regularized artificial neural network model," *Engineering Applications of Artificial Intelligence*, vol. 125, p. 106736, Oct. 2023. [Online]. Available: <https://doi.org/10.1016/j.engappai.2023.106736>
- [36] C. Wang, J. Liang, and Q. Deng, "Dynamics of heterogeneous Hopfield neural network with adaptive activation function based on memristor," *Neural Networks*, vol. 178, p. 106408, Oct. 2024. [Online]. Available: <https://doi.org/10.1016/j.neunet.2024.106408>
- [37] M. Hasan, M. Rahman, M. S. Islam, W. H. Chan, Y. M. Alginahi, M. N. Kabir, S. A. Bakar, and D. Ramasamy, "Artificial neural network modeling for predicting thermal conductivity of EG/water-based CNC nanofluid for engine cooling using different activation functions," *Frontiers in Heat and Mass Transfer*, vol. 22, no. 2, p. 537–556, May 2024. [Online]. Available: <https://doi.org/10.32604/fhmt.2024.047428>
- [38] S. A. A. Fazel, "Prediction of bubble departing diameter in pool boiling of mixtures by ANN using modified ReLU," *Heliyon*, vol. 10, no. 11, p. e31261, Jun. 2024. [Online]. Available: <https://doi.org/10.1016/j.heliyon.2024.e31261>
- [39] M. C. Cieslak, A. M. Castelfranco, V. Roncalli, P. H. Lenz, and D. K. Hartline, "t-distributed stochastic neighbor embedding (t-SNE): A tool for eco-physiological transcriptomic analysis," *Marine Genomics*, vol. 51, p. 100723, Jun. 2020. [Online]. Available: <https://doi.org/10.1016/j.margen.2019.100723>
- [40] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, p. 321–357, Jun. 2002. [Online]. Available: <https://doi.org/10.1613/jair.953>
- [41] A. Rojarath, W. Songpan, E. Okafor, and O. Surinta, "Enhancing personality characteristics analysis with smote and association rule mining: A case study on introverts and extroverts," *ICIC Express Letters*, vol. 19, no. 6, pp. 597–606, Jun. 2025. [Online]. Available: <https://doi.org/10.24507/icicel.19.06.597>
- [42] A. Rojarath, W. Songpan, and O. Surinta, "Depression classification with imbalanced data problems: Literature survey," *Engineering Access*, vol. 11, no. 2, pp. 185–199, Jun. 2025. [Online]. Available: <https://doi.org/10.14456/mijet.2025.18>
- [43] S. Phiphitphatphaisit and O. Surinta, "Multi-layer adaptive spatial-temporal feature fusion network for efficient food image recognition," *Expert Systems with Applications*, vol. 255, p. 124834, Dec. 2024. [Online]. Available: <https://doi.org/10.1016/j.eswa.2024.124834>
- [44] A. Rabruen, P. Pokkate, O. Surinta, and S. Khruahong, "Sentiment analysis of Thai laborers' perceptions of working abroad: A machine learning approach using Youtube comments," *ICIC Express Letters, Part B: Applications*, vol. 16, no. 3, pp. 333–341, Mar. 2025. [Online]. Available: <https://doi.org/10.24507/icicelb.16.03.333>
- [45] M. Tavallaei, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA)*. IEEE, Jul. 2009, p. 1–6. [Online]. Available: <https://doi.org/10.1109/cisda.2009.5356528>

- [46] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *the 4th International Conference on Information Systems Security and Privacy (ICISSP)*. SCITEPRESS - Science and Technology Publications, Jan. 2018, pp. 108–116. [Online]. Available: <https://doi.org/10.5220/0006639801080116>
- [47] N. Pandey, N. S. M, M. Sumathi, I. Alam, S. Das, and R. Mondal, "Next-generation firewalls: Enhancing network security with application-awareness," in *2025 International Conference on Automation and Computation (AUTOCOM)*, mar 2025, pp. 595–600. [Online]. Available: <https://doi.org/10.1109/AUTOCOM64127.2025.10956381>
- [48] A. M. Winkler and P. Sharma, "Proactive threat detection in enterprise systems using Wazuh: A MITRE ATT&CK evaluation," *Computers & Security*, vol. 159, p. 104702, Dec. 2025. [Online]. Available: <https://doi.org/10.1016/j.cose.2025.104702>