# Design and Implementation of Network Intrusion Detection System based on Machine Learning

Rongguo Fu
*Information Center*
*Jiangsu Earthquake Agency*
Nanjing, Jiangsu, China

2025 International Conference on Intelligent Systems and Computational Networks (ICISCN) | 979-8-3315-2924-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICISCN64258.2025.10934502

***Abstract*—Traditional rule or signature-based Intrusion Detection Systems (IDS) often struggle to effectively identify and defend against new, complex, and evolving cyber attacks. Therefore, this study explores machine learning (ML)-based network intrusion detection technologies to enhance the accuracy and flexibility of detection. Methodologically, this paper adopts Deep Neural Networks (DNN) as the core algorithm and combines it with the Random Forest algorithm from ensemble learning to construct a hybrid model that improves the performance and accuracy of network intrusion detection. The system design employs a layered architecture, encompassing five levels: data collection, data preprocessing, feature extraction, intrusion detection, and alerting and response, ensuring the system's scalability, maintainability, and efficiency. The research results indicate that the hybrid model of DNN and Random Forest achieved an accuracy rate of 98.5% on the test set, significantly higher than models using DNN or Random Forest alone. The system can complete real-time classification of network traffic within milliseconds, meeting the real-time requirements of IDS. Moreover, the model demonstrates strong robustness in the presence of noisy data and unknown attack types, maintaining a high detection rate while reducing the false positive rate. ML-based network IDS showcases its potential in addressing complex network threats, promising to drive technological innovation and development in the field of cybersecurity. This study successfully designs and implements an ML-based network IDS that employs an ensemble learning approach combining DNN and Random Forest, significantly improving the accuracy and robustness of network attack detection. The test set accuracy reached 98.5%, and the system meets real-time requirements, with an average processing time of 50 milliseconds.**

***Keywords—machine learning, intrusion detection system, deep neural network, random forest.***

## I. Introduction

With the continuous progress of network technology, the security threat of cyberspace is becoming more and more serious, and network intrusions occur frequently, which brings great economic losses and severe information leakage risks to society. Traditional rule-based or signature-based network intrusion detection systems (IDS) are often unable to effectively identify and defend against new, complex and changeable network attacks, which urges us to explore more intelligent and adaptive network intrusion detection technologies.

In the past literature, IDS based on machine learning (ML) has been extensively studied, which reveals the great potential of ML in the field of network security [1-3]. ML can learn rules and patterns from a large number of data, and then predict and classify unknown data, which provides a new idea for solving the shortcomings of traditional IDS [4]. Common ML models, such as support vector machine (SVM), decision tree, random forest and neural network, have been applied to the field of intrusion detection and achieved remarkable results [5-6]. However, these studies also face many challenges, such as feature selection of high-dimensional data, improvement of model generalization ability, optimization of real-time detection efficiency and reduction of false positives and false negatives [7]. In addition, how to design an efficient and practical IDS with practical application scenarios is still an important research direction.

In view of the above challenges and shortcomings, this paper designs and implements a network IDS based on ML to deal with the increasingly complex network threats. Through in-depth analysis of network traffic data, effective features are extracted, and an efficient and accurate intrusion detection model is constructed. This paper hopes to improve the detection performance and robustness of IDS. In this paper, Deep Neural Network (DNN) is adopted as the core algorithm, and its powerful nonlinear mapping ability and highly automated feature learning ability are fully utilized to process complex and high-dimensional network traffic data. At the same time, in order to further improve the generalization ability and stability of the model, this paper also introduces the random forest algorithm in ensemble learning, which is fused with DNN to give play to their complementary advantages.

The contribution of this paper is mainly reflected in the following aspects: First, a network IDS based on ML is designed and implemented, which adopts a layered architecture and has good scalability, maintainability and efficiency; Secondly, by combining DNN and random forest algorithm, the accuracy and robustness of intrusion detection are significantly improved; Finally, the real-time and stability of the system are verified by testing in the actual network environment, and its potential in dealing with complex network threats is demonstrated.

The organizational structure of this paper is as follows:

(1) Introduce the background and challenges of network intrusion detection and the application status of ML in IDS in detail.

(2) Elaborate the design idea, system architecture and algorithm implementation of network IDS based on ML.

(3) Evaluate the performance of the system through experimental tests, and analyze its advantages and disadvantages.

(4) Summarize the whole paper and look forward to the future research direction.

## II. Analysis and Design of Intrusion Detection Model

### A. Intrusion Detection Algorithm

In this article, deep neural networks (DNN) are used as the core algorithm to construct network IDS [8-9]. DNN performs

well in processing complex and high-dimensional network traffic data due to its powerful nonlinear mapping ability and highly automated feature learning ability, making it particularly suitable for identifying subtle patterns and abnormal features in network intrusion behavior.

DNN is developed from Multi Layer Perceptron (MLP), which forms deep structures by stacking multiple hidden layers, each layer containing multiple neurons connected by weights.

The input layer receives raw feature data, the hidden layer is responsible for feature extraction and transformation, and the output layer provides the final classification result (such as normal or intrusion). The MLP structure is shown in Figure 1.

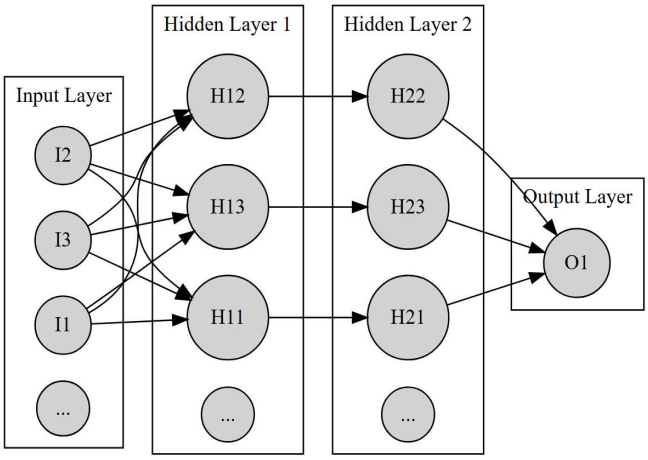


Fig. 1. MLP structure

In the training process, the weight parameters in the network are constantly adjusted by optimization techniques such as back propagation algorithm and gradient descent to minimize the prediction error, so as to learn the inherent laws and feature representation of data [10].

For DNN, its forward propagation process can be expressed as in eq. (1) and (2)

$$z^l = W^l \cdot a^{l-1} + b^l \tag{1}$$

$$a^l = \sigma(z^l) \tag{2}$$

Where $z^l$ is the input of the $l$ layer, $W^l$ is the weight matrix of the $l$ layer, $b^l$ is the bias vector of the $l$ layer, $a^l$is the activation output of the $l$layer, and $\sigma(\cdot)$ is the activation function.

Commonly used activation functions include Sigmoid, ReLU, etc. For example, Sigmoid function is defined as in eq. (3)

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{3}$$

The loss function is used to evaluate the difference between the prediction result of the model and the real label. The commonly used loss function is cross entropy loss. For the classification problem, its expression is shown in eq. (4)

$$L(y, \hat{y}) = -\sum_i y_i \, log(\hat{y}_i) \tag{4}$$

Where $y$ is the unique thermal coding representation of the real tag, and $\hat{y}_i$ is the probability distribution predicted by the model.

According to the loss function, the gradient of each layer's weight is calculated for updating the weight. For the weight of the $l$ layer, the gradient calculation is as follows in eq. (5)

$$\frac{\partial L}{\partial W^{(l)}} = \delta^{(l)} \left(a^{(l-1)}\right)^T \tag{5}$$

Where $a^{(l-1)}$ is the activation output of the $l-1$ layer. $\delta^{(l)}$ is the error of the first $l$ layer, which is calculated as follows in eq. (6)

$$\delta^{(l)} = \left(\frac{\partial L}{\partial z^{(l)}}\right) \otimes \sigma'\left(z^{(l)}\right) \tag{6}$$

$\sigma'\left(z^{(l)}\right)$ is the derivative of the activation function, and $\otimes$ represents the product of the element level.

Use gradient descent to update weights, for example shown in eq. (7)

$$W^{(l)} = W^{(l)} - \alpha \frac{\partial L}{\partial W^{(l)}} \tag{7}$$

Here, $\alpha$ is the learning rate.

Similar to weight updating, the updating formula of offset is shown in eq. (8)

$$b^{(l)} = b^{(l)} - \alpha \frac{\partial L}{\partial b^{(l)}} \tag{8}$$

Where $\frac{\partial L}{\partial b^{(l)}}$ is usually directly equal to $\delta^{(l)}$.

DNN has a strong feature learning ability and a good ability to deal with nonlinear problems. It can capture high-level data representation by increasing the depth of hidden layer, which is especially suitable for network intrusion detection. It can learn and identify abnormal network traffic patterns and effectively detect new and unknown attacks. However, DNN also has obvious shortcomings, including long training time, large demand for computing resources, easy over-fitting and poor model interpretability.

### B. *Ensemble learning*

In order to improve the performance and accuracy of intrusion detection model based on DNN, a number of optimization strategies are implemented, including determining the best learning rate through grid search and cross-validation, to ensure that the weight update is stable and efficient; Choose the most suitable number of hidden layers and neurons through experimental comparison to balance the complexity of the model and the risk of over-fitting; L2 regularization technique is adopted, and the sum of squares of weights is added to the loss function as a penalty term, which effectively controls the weight growth and further prevents over-fitting.

In order to further improve the generalization ability and stability of the model, the random forest algorithm in ensemble learning is used to fuse with DNN, and the model structure after fusion is shown in Figure 2. Random forest can reduce the deviation and variance of a single model and improve the overall performance by constructing multiple decision trees and taking their average prediction results [11].

In this paper, the output of DNN is regarded as one of the input characteristics of random forest, and the integration characteristics of random forest are used to further improve the accuracy of intrusion detection.

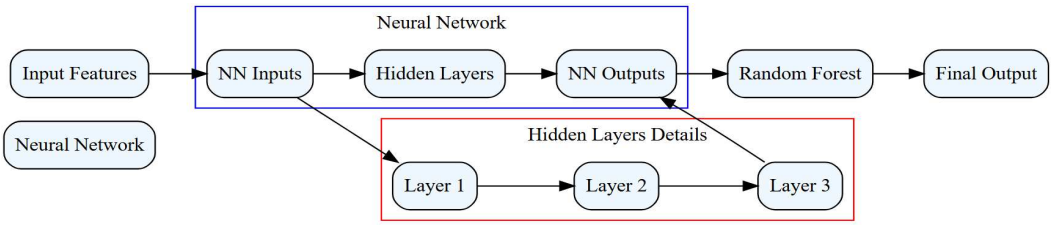


Fig. 2. Model structure combining DNN and random forest

For random forests, the prediction results can be expressed as shown in eq. (9)

$$\hat{y} = \frac{1}{N}\sum_{i=1}^{N} T_i\,(x) \tag{9}$$

Where $\hat{y}$ is the final prediction result, $N$ is the number of decision trees, and $T_i(x)$ is the prediction result of the $i$ decision tree on input $x$.

## III. System Design And Implementation

### A. *System architecture*

In this study, a network IDS based on ML is designed, and the hierarchical architecture is adopted to ensure its scalability, maintainability and efficiency. The system consists of five layers: data acquisition, data preprocessing, feature extraction, intrusion detection and alarm and response. Each layer communicates with each other through standardized interfaces. The system architecture is shown in Figure 3. The data acquisition layer collects network traffic data in real time; Data preprocessing layer cleans, transforms and normalizes data; The feature extraction layer uses DNN and other technologies to mine valuable features; Intrusion detection layer combines DNN and random forest model to classify in real time; The alarm and response layer generates an alarm and takes corresponding defensive measures after detecting the intrusion [12].

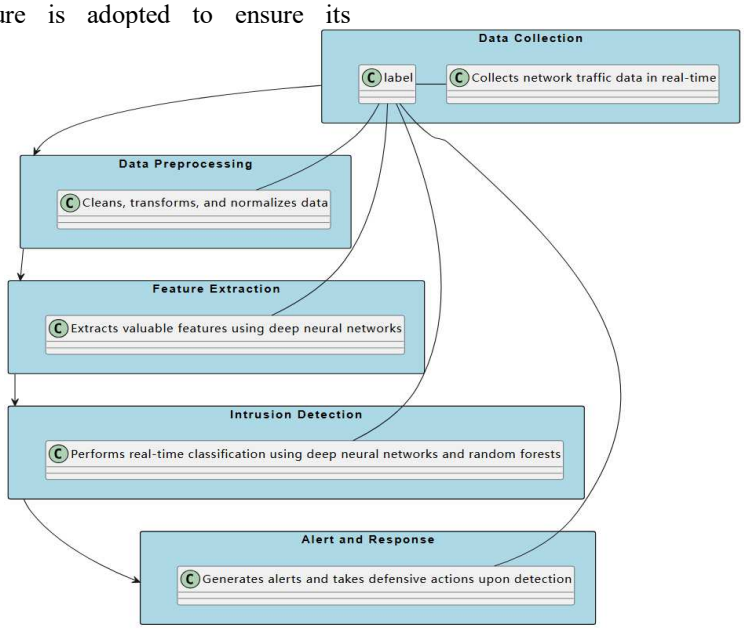


Fig. 3. Network IDS architecture based on ML

### B. *System function module*

The system function module is shown in Figure 4. It mainly includes data acquisition, data preprocessing, feature extraction, intrusion detection, alarm and response. The data acquisition module supports a variety of network protocols and data formats, and has data filtering and caching mechanisms; The data preprocessing module realizes data cleaning, format conversion and normalization; The feature

extraction module uses DNN to learn features automatically, and optimizes the feature set through feature selection and updating mechanism. Intrusion detection module combines DNN and random forest model for real-time classification to support model training and performance evaluation; The alarm and response module designs alarm information generation algorithm, provides multi-channel alarm notification, and allows customization of response strategy.

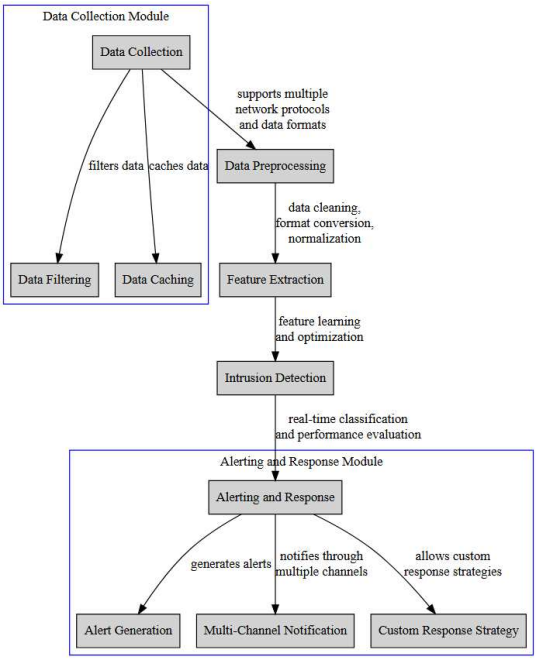


Fig. 4. System function module

## IV. Experiment And Result Analysis

### A. Data set construction

The dataset used in this study originates from the publicly available KDD CUP 1999 cybersecurity dataset and traffic data captured from the actual network environment through collaboration with partner institutions. The KDD CUP 1999 dataset is a widely used benchmark for network intrusion detection, containing a large amount of normal traffic and various types of attack traffic, such as DoS (Denial of Service), Probe (scanning), U2R (User to Root privilege escalation), and R2L (Remote to Local privilege escalation). Specific categories are listed in Table 1. The data captured from the real network environment provides more realistic network traffic characteristics, enhancing the generalization capability of the model.

TABLE I. KDD CUP 1999 Dataset Attack Category

| Attack category | Name | Describe |
|---|---|---|
| Denial of service attack | DoS | An attack that prevents legitimate users from accessing services by consuming system resources. |
| Scanning detection | Probe | The act of collecting information about a target system by sending a specific packet, usually in preparation for a subsequent attack. |
| User to root privilege elevation | U2R | It refers to an attack in which the attacker upgrades from ordinary user's authority to super user's (root) authority, which is common in exploiting system vulnerabilities. |
| Remote to local privilege elevation | R2L | Refers to the attacker executing code or command remotely through the network and gaining local rights of the target system. |

Data set is cleaned, labeled and divided. Data labeling divides data into normal and attack categories according to known attack types and normal behavior patterns. The data set is divided into training set, verification set and test set according to the proportion of 70%, 15% and 15%, which are used for model training, parameter tuning and performance evaluation respectively. This data set is diverse, authentic and large-scale, covering a variety of network attack types and normal traffic, which can fully and truly reflect network behavior and provide sufficient support for DNN training.

### B. Construction of experimental environment

The construction of experimental environment includes hardware and software. In terms of hardware, Intel Xeon E5-2690 CPU, 64GB RAM, NVIDIA Tesla K80 GPU and 1TB SSD server configuration are selected. In terms of software, Ubuntu 18.04 LTS operating system is run, Python 3.12 is used for programming, and TensorFlow 2.3.0 deep learning framework and Scikit-learn 0.23.2ML library are relied on. TensorFlow was chosen as the experimental tool to construct and train DNN, Scikit-learn provided random forest algorithm, Jupyter Notebook was used to record the experiment and show the results, and GridSearchCV conducted grid search to determine the best parameters. In configuration, DNN uses Adam optimizer, and the learning rate and parameters of random forest are optimized by experiments. At the same time, cross-validation is used to evaluate the model performance and reduce the risk of over-fitting.

### C. System performance evaluation

Through the test in the actual network environment, the system can complete the real-time classification of network traffic in milliseconds, and meet the real-time requirements of

IDS. Figure 5 shows the real-time performance of the system when processing network traffic data, and the processing time fluctuates to some extent, reflecting the influence of packet content complexity, system load and resource contention. The average processing time is about 50 milliseconds, which meets the high real-time requirements. However, the "long tail" phenomenon that the processing time is significantly prolonged occasionally may be caused by large data packets, high system load or resource bottleneck. Overall, the system shows good real-time and stability, but it still needs to be optimized for the fluctuation of processing time and extreme delay to improve processing efficiency and user experience.

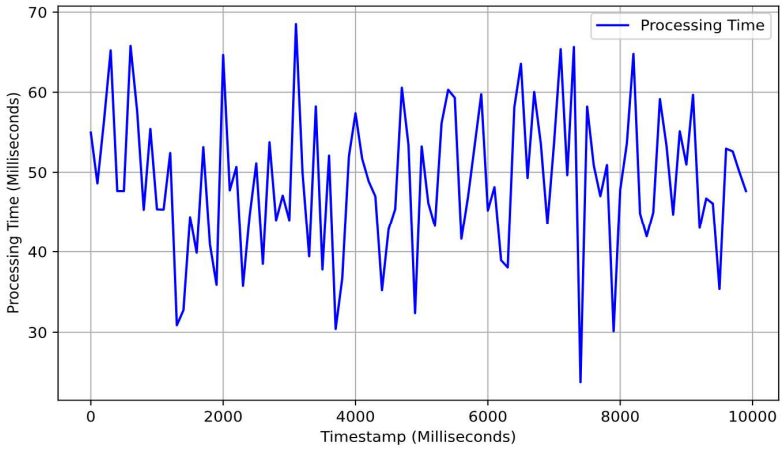


Fig. 5. Real-time processing of network traffic data by the system

On the test set, the accuracy of the model combining DNN and random forest reaches 98.5%, which is higher than that of the model using DNN or random forest alone, and effectively improves the accuracy of intrusion detection. Table 2 shows that the model combining DNN and random forest is superior to the single model in accuracy, recall and F1-score. Specifically, the accuracy of the fusion model reaches 98.5%, which is 1.3 and 1.7 percentage points higher than that of DNN and random forest respectively. The recall rate is 0.98, which is slightly higher than that of DNN 0.96 and random forest 0.95. F1-score is 0.985, which comprehensively reflects the accuracy and robustness of the model. This shows that the fusion model significantly improves the performance of network intrusion detection by combining the advantages of the two methods, and becomes a more effective detection method.

TABLE II. COMPARISON OF DIFFERENT MODELS IN ACCURACY, RECALL AND F1-SCORE

| Model | Accuracy | Recall | F1-score |
|---|---|---|---|
| DNN | 97.2% | 0.96 | 0.965 |
| Random Forest | 96.8% | 0.95 | 0.960 |
| DNN+Random Forest | 98.5% | 0.98 | 0.985 |

By introducing the integration characteristics of L2 regularization and random forest, the model shows strong robustness in the face of noisy data and unknown attack types, and can maintain a high detection rate and reduce the false alarm rate. Under different network environments and traffic characteristics, the system can maintain stable performance, which proves its good generalization ability. Figure 6 shows the robustness of the model under different conditions. It can be seen that under "normal" conditions, the detection rate of the model is high and the fluctuation is small. However, under the condition of "noise", the detection rate decreases slightly and the fluctuation increases. For two "unknown attack types", the detection rate of the model is further reduced and fluctuates more, especially under the condition of "unknown attack type 2". This shows that the model still maintains a certain detection ability in the face of noise data and unknown attack types, but its performance is really affected.

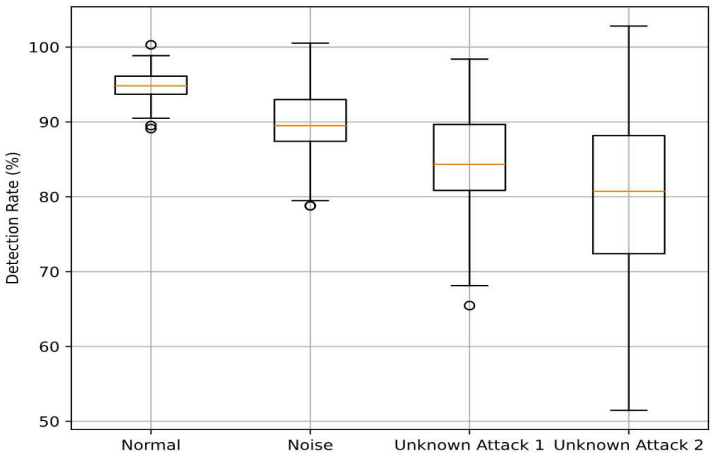


Fig. 6. Robustness under different conditions

## V. Conclusion

In this study, a network IDS based on ML is designed and implemented, and the integrated learning method of DNN and random forest is adopted, which significantly improves the accuracy and robustness of network attack detection. The experiment uses KDD CUP 1999 data set and actual network traffic data. The results show that the average processing time of the IDS is 50 milliseconds, and the accuracy of the test set is 98.5%, which is better than that of a single model. Nevertheless, the system still faces challenges such as feature selection of high-dimensional data and improvement of model generalization ability. Future research will focus on optimizing algorithm parameters, enhancing generalization ability and improving data processing efficiency, so as to further enhance the practicability and effectiveness of the system. Network IDS based on ML shows its potential in dealing with complex network threats and is expected to promote technological innovation and development in the field of network security.

## References


[1] Onyema, E. M. , Dalal, S. , Romero, Carlos Andrés Tavera, Seth, B. , Young, P. , & Wajid, M. A. (2022). Design of intrusion detection system based on cyborg intelligence for security of cloud network traffic of smart cities. Journal of Cloud Computing, 11(1), 1-20.

[2] Abdulganiyu, O. H. , Tchakoucht, T. A. , & Saheed, Y. K. (2024). Towards an efficient model for network intrusion detection system (ids): systematic literature review. Wireless networks, 30(1), 453-482.

[3] Zhu, Y. , Gaba, G. S. , Almansour, F. M. , Alroobaea, R. , & Masud, M. (2021). Application of data mining technology in detecting network intrusion and security maintenance. Journal of Intelligent Systems, 30(1), 664-676.

[4] Yogesh, & Goyal, L. M. (2024). Deep learning based network intrusion detection system: a systematic literature review and future scopes. International Journal of Information Security, 23(6), 3433-3463.

[5] Bhutta, A. A. , Nisa, M. U. , & Mian, A. N. (2024). Lightweight real-time wifi-based intrusion detection system using lightgbm. Wireless networks, 30(2), 749-761.

[6] Cui, M. , Chen, J. , Qiu, X. , Lv, W. , Qin, H. , & Zhang, X. (2024). Multi-class intrusion detection system in sdn based on hybrid bilstm model. Cluster Computing, 27(7), 9937-9956.

[7] Varzaneh, Z. A. , & Rafsanjani, M. K. (2021). Intrusion detection system using a new fuzzy rule-based classification system based on genetic algorithm. Intelligent Decision Technologies, 15(2), 231-237.

[8] Li, J. , Zhang, H. , Liu, Z. , & Liu, Y. (2023). Network intrusion detection via tri-broad learning system based on spatial-temporal granularity. The Journal of Supercomputing, 79(8), 9180-9205.

[9] Rajabi, S. , Asgari, S. , Jamali, S. , & Fotohi, R. (2024). An intrusion detection system using the artificial neural network-based approach and firefly algorithm. Wireless Personal Communications, 137(4), 2409-2440.

[10] Xiao, J. , Yang, L. , Zhong, F. , Chen, H. , & Li, X. (2023). Robust anomaly-based intrusion detection system for in-vehicle network by graph neural network framework. Applied Intelligence, 53(3), 3183-3206.

[11] Lin, L. , Zhong, Q. , Qiu, J. , & Liang, Z. (2025). E-gracl: an iot intrusion detection system based on graph neural networks. The Journal of Supercomputing, 81(1), 1-31.

[12] Aswani, I., Kar, N.K., Ganguly, T., Ramesh, G.P. and Tejaswini, N.P., 2023, February. A Fault Diagnosis of Sound and Vibration Signals Using Statistical Features and Machine Learning Algorithm. In 2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS) (pp. 1-7). IEEE.