Optimizing Multiclass Intrusion Detection Using Residual MLP Architecture with Threshold Refinement via Ant Colony Metaheuristics

Saurabh Shrivastava[1*], Jitendra Singh Kushwah[2] and Pritaj Yadav[3]

[1,3*]Department of Computer Science and Engineering, Rabindranath Tagore University, Bhopal, India.
[2]Department of Information Technology, Institute of Technology and Management, Gwalior, India.

Abstract

The across the board detection of various and changing cyberattacks has remained a major challenge to the Intrusion Detection Systems (IDS) especially in unbalanced and high-dimensional network noises. In order to overcome these shortcomings, this paper presents a new hybrid scheme of detection that combines a Residual Multilayer Perceptron (Res-MLP) architecture and the threshold tuning scheme based on the Ant Colony Optimization (ACO). Res-MLP backbone full optimization Res-MLP backbone also optimizes representational learning by applying dense nonlinear transformations, residual shortcuts (allow gradients to flow stable) and Batch Normality with dropout based regularization that discouraging over-fitting. The upshot of making this formulation is a stronger model that can be used to represent the manners of behavior of network characters. To complement the architecture, there exists a mathematical ACO model which results in the optimization of the class-specific softmax decision threshold defined on the macro-F1 score as fitness objective. The ACO module combines a dynamic sampling of the candidate threshold vectors, assesses the performance of the detectors and continuously modifies the pheromones in order to dynamically converge to optimal boundaries. This increases the ability of the classifier to differentiate between minority attack classes as well as to achieve higher discrimination in general. The experiments performed on the NSL-KDD dataset show that the Res-MLP + ACO framework can significantly enhance detection measures in comparison with traditional MLPs and fixed-threshold classifiers. The maximized model has better macro-F1, precision, and recall, especially in cases of minority and low frequency attack. The mathematical derivations presented in relation to either the architecture construction or the optimization procedure will be reproducible and conceptually clear which serves as a further argument in favor of the validity of the suggested approach.

Keywords: Intrusion Detection System (IDS), Residual MLP (Res-MLP), Ant Colony Optimization (ACO), Threshold Optimization, Network Security

1 Introduction

The computer networks in the present day are not only vulnerable to more advanced types of cyber threats but also the Intrusion Detection Systems (IDS) are the keys to the defense agency against such threats as the correct detection and identification of various types of attack [1], [2]. Conventional IDS can be binary and only classify as either normal or malicious activity which is not fine-grained enough in real-world situations. The multiclass intrusion detection alleviates this weakness by allowing the identification of certain categories of attacks, e.g., denial-of-service, phishing, malware or ransomware, and providing actionable intelligence to response to such an incident [3], [4]. The application of machine learning (ML) algorithms particularly neural networks like the Multilayer Perceptron (MLP) can be found everywhere because they have the power to describe the fundamental characteristics of complex curves using a high-dimensional data set [5]. Nevertheless, the mainstream MLPs have the disadvantage of the vanishing gradient problem and lack of ability to learn worthwhile features, and all of this can decrease classification performance [6]. MLPs have been extended to residual learning through skip connections that enable the gradient flow and enable them to learn more abstract representations [7], [8]. This variation of MLP design allows greater control in differentiating various types of intrusion, which positively affected the accuracy of the classification. Also, classification thresholds play essential roles in establishing limits of a decision; in contrast to their counterparts, that is, the static thresholds, these thresholds do not adapt flexibly to heterogeneous attack patterns [9].

In spite of the progress, the IDS have continued to possess certain challenges that affect them in the following way; high false-positive, low scalability of the newthreats, and difficulty handling big and complex network data in real-time [10], [11]. False positives are high and take up a lot of time on the part of the analyst and weaken the reliability of the system, whereas the static signature based IDS cannot identify the zero day threats [12]. Complex multistage attacks take advantage of the uncorrelated nature of time that can be simply detected using a rule [13]. Machine learning methods have gained importance, which allows adaptive detection through regulations of labeling and label-free data [14]. Supervised learning indicates particular types of attacks, and unsupervised and reinforcement learning find anomalies and also optimize defense mechanisms [15]. Additional features such as convolutional and recurrent networks are present in deep learning, which improves the extraction of features on sequential network traffic [5]. Residual Neural Networks (ResNets) are used to deal with vanishing gradient problems [7], [8] and to enable deep neural networks by using skip connections to enhance gradient flow, training stability, convergence time, and learning features. Such mechanisms enable the IDS to match fine high-dimensional patterns, which enhance the precision of detection and inaccurate alarms.

Although residual MLP also enhances the extraction and representation of features, optimization of feature-specific thresholds also affects the decision performance [9]. Fixed thresholds can be biased as they do not aid in portraying intricate attack distributions. Another dynamic solution is metaheuristic optimization algorithms like the Ant Colony Optimization (ACO) algorithm, which repeatedly tries to determine the optimal thresholds to maximize such metrics as precision, recall, and F1-score [4], [6]. ACO is used to simulate pheromone-guided search and it successfully isolates configurations that are high performing in thresholding as well as minimizes false positives and false negatives. Through the combination of residual MLP architectures and an extension of the residual frame of ACO threshold refinement, the framework can improve both the deep feature learning and the calibration of decision boundaries. This method eliminates the drawbacks of traditional IDS, which provides more multiclasses, strength, and flexibility. The synergistic approach to the problem enables the specific recognition of various types of attacks, enhances the effectiveness of response, and increases the security of the whole network. Later parts introduce dataset, methodology, training of the model and evaluation of its performance proving the effectiveness of such an integrated approach.

2 Literature Review

Intrusion Detection Systems (IDS) remain a cornerstone of cybersecurity, tasked with identifying malicious network activity and protecting complex digital infrastructures. The proliferation of Internet of Vehicles (IoV) and Internet of Things (IoT) environments has introduced unique challenges, including heterogeneous devices, protocol diversity, and high-dimensional data streams [16], [17], [22]. Ensemble learning techniques, including stacking, boosting, bagging, and voting, have been widely applied to enhance multiclass detection capabilities, improve robustness, and reduce false positives, as demonstrated on datasets like NSL-KDD, CICIDS2017, and UNSW-NB15 [16]. Similarly, hybrid approaches combining deep learning architectures, such as DNN-LSTM, CNN, Transformer models, and Extreme Gradient Boosting (XGBoost), with advanced optimization algorithms including Levy flight Grasshopper, Bayesian optimization, and Ant Colony Optimization (ACO), have proven effective for fine-tuning hyperparameters, mitigating class imbalance, and optimizing decision thresholds [17], [18], [19], [32].

Addressing class imbalance remains a recurring theme in IDS research. Hybrid resampling strategies, such as SMOTE, GAN-generated synthetic data, undersampling, and oversampling, integrated with ensemble classifiers, have shown notable improvements in minority class detection without degrading majority class performance [20], [23], [25], [24]. These methods enable IDS to detect rare attack types, such as U2R, R2L, and DDoS events, more accurately while maintaining high macro F1-scores. Studies focusing on industrial IoT (IIoT) networks highlight that deep learning models, particularly CNNs and Transformers, benefit from data augmentation and hierarchical architectures to address imbalanced and high-dimensional datasets [23], [32], [31]. Moreover, explainable AI (XAI) frameworks have been introduced to enhance interpretability, particularly in multiclass intrusion detection scenarios, helping security analysts understand model decisions and attack attribution [26].

The integration of nature-inspired metaheuristics, such as ACO, Particle Swarm Optimization, and Genetic Algorithms, has been widely explored to improve both model selection and hyperparameter tuning [36]. These algorithms simulate adaptive, self-organizing behaviors from nature, providing efficient search strategies for optimizing IDS performance under complex, nonlinear conditions. Furthermore, research demonstrates the advantages of hybrid machine learning pipelines combining feature selection, dimensionality reduction (e.g., KPCA), and classifier ensembles to maximize detection accuracy while minimizing false positives [17], [34], [35]. Decision tree"based methods, including J48 and its variants, remain popular for their computational efficiency, interpretability, and high accuracy in multiclass settings, with multi-criteria decision-making techniques like TOPSIS facilitating optimal classifier selection [39]. Finally, multiclass SVMs and hierarchical IDS frameworks optimized using metaheuristics have exhibited significant improvements in attack-specific classification accuracy, computational efficiency, and resilience to evolving threats [40], [30]. Collectively, these studies underscore the importance of combining deep learning, ensemble strategies, hybrid optimization, and explainable AI to develop robust, adaptive, and high-performing IDS solutions capable of addressing contemporary cybersecurity challenges across IoV, IoT, IIoT, and SDN networks [16][40-45].

3 Methodology

The study employs the NSL-KDD dataset, a system benchmark related to the assessment of intrusion detection systems, to generate and assess both the machine learning (ML) models and deep learning (DL) models of network attack prediction based on multi-classification. The dataset consists of labeled network traffic examples, which are classified based on normal or different types of attacks, 41 features, a label, and a difficulty measure. The two types of training and testing subsets (KDD Train + and KDD Test +) were retrieved, and the column structure was defined to give uniform data frame organizations to the downstream processes. Partial consolidation was done to consolid the subsets into one dataset of 148,517 rows and 43 columns. The process of feature selection left behind 25 high-impact attributes and deduplication left behind 3,540 redundant rows leaving behind a final working dataset of 144,977 rows and 26 columns.

To simplify multiclassy classification, attack labels were categorized into five major groups which included DoS, Probe, R 2 L, U 2 R, and Normal traffic. Attributes of categorical types were frequency distributions, such as protocol, service, and flag;

infrequent frequencies were combined, and coded names were provided to all categorical variables. Numerical variables with skewness values were transformed into log1p, Boolean type variables were changed into binary and all individuals were standardized using a z-score normalization. To deal with class imbalance, moderate SMOTE oversampling was made on minority classes in a manner that the integrity of majority classes was maintained. Stratified splitting produced training, validation, and test sets of 80:10:10 ratio preserving the proportions of the classes.

To benchmark the ML, the Logistic Regression model with balancing weights of classes and K-Nearest Neighbours (k=5) become applied and the baseline performance metrics as well as the confusion tables were calculated. There was development of deep learning residual MLP model, which comprised of dense and residual blocks including ReLU activations, a batch normalization, and dropouts. (Table 1). The residual connections made identity mapping and allowed more learning in the network without hardship on its performance. This model was trained on the basis of 100 epochs with 64 default batch size by Adam optimization and sparse categorical cross-entropy loss. Lastly, Ant Colony Optimization (ACO) was then used to optimize per-class decision thresholds, which yielded better macro F1-score than default softmax argmax behavior (Table 2). The metaheuristic iteratively explored threshold vectors, updated pheromones toward best-performing configurations, and applied a fallback to argmax if no class exceeded its threshold, enhancing class-specific decision boundaries and overall classification performance.

3.1 Dataset Description

NSL-KDD data can be found at the web site of the University of New Brunswick (n.d., n.p., p. 1) and is a typical reference that measures the intrusion detection systems (IDS). It solves the restrictions of the original KDD99 dataset by being made in to labeled network traffic instances which are either normal or an attack, with each record having 41 features, a label and a difficulty metric. It can be divided into KDD Train+ and KDD Test+ subsets, which makes it convenient and strict in the assessment of IDS, especially that of DoS, probe, R2L, and U2R attacks; hence, it is popular in cybersecurity research.

3.2 Data Acquisition and Feature Schema Initialization

The NSL-KDD data has been retrieved and loaded systematically in order to enable modelling of intrusion detection. Both training and testing (KDD Train+ and KDD Test+) were downloaded and put locally to preprocess. The column structure was specified in such a way that it would create regular data frames, which consist of 42 columns, each of them divided into basic features (e.g., duration, protocol_type, service), content features (hot, num_failed logins), traffic features (e.g., count, serror rate), and host-based features (dst host count, dst host srv count). Also, both cases imply a label with normal or an attack type and a difficulty level to analyze. The resulting consolidated data were the train and test shapes (125,973, 43) and (22,544, 43), respectively, which is ready to be analyzed structurally.

3.3 Data Cleaning and Feature Pruning

The training and test subsets of NSL-KDD were merged into one to help in effective multiclass classification so that the schema of the columns matched in both. The data set that was created had 148,517 rows and 43 columns. The feature selection process was done again in order to retain 25 high-impact attributes, which maximized the performance of the classification, and the target label was added to the process in order to model it. The deduplication and integrity checks eliminated 3,540 redundant rows that left 144,977 rows and 26 columns in a clean dataset. This has provided a strong platform on which downstream intrusion detection analysis can be done by maintaining the consistency of the data as well as reducing the magnitude of bias and multicollinearity.

3.4 Label Mapping and Feature Transformation

In order to increase the interpretable nature of the models and their compatibility, the attack labels of the NSL-KDD dataset were reduced to five broad groups: DoS (Denial of Service), Probe (Surveillance and Scanning), R2L (Remote to Local Intrusion), U2R (User to Root Exploits), and Normal traffic. Analysis of categorical features by frequency showed that there are skewed distributions with TCP carrying most of the protocol traffic, a small set of core services (HTTP, FTP, SMTP, private ports) existing and flags being significantly more concentrated in SF and S0. Rare service and flag levels were combined and all categorical values were coded protocol and flag via label encoding, and service categories were merged to minimize dimensionality. This preprocessing enhanced the balance of the classes, the sparsity, as well as made the multiclass classification harder. To prepare the NSL-KDD dataset in order to form a strong model training and assessment, numerical and categorical features were transformed in a number of ways. To minimize heavy-tailed distributions and bring the first volatile to variance stability, skewed numerical characteristics were log-transformed with log1p. Binary format was used to encode the features that were generated as Boolean-like and numerically encode the multiclass label. Z-score scaling was applied to all the features to have a zero mean and unit variance to support distance-based models and SMOTE interpolation. The minority classes (U2R, R2L) were over sampled moderately with SMOTE (with 20 percent of majority class size) and the overall integrity of classes was maintained. Lastly, stratified splitting created training, validation and test sets in the ratios of 80:10:10 and ensured that the fractions of classes were used fairly.

3.5 Machine Learning Model Development

Multiple machine learning classifiers were used as benchmarks to assess the performance of the preprocessed NSL-KDD data on a macro-average basis by metrics of performance and multiclass religion-of-origin analysis through ROC. Logistic Regression was trained under balanced class weight to reduce residual class imbalance with the use of the solver of lbfgs and using one-vs-rest (one-verses-rest) approach to classify on multiple classes. Also, K-Nearest Neighbours (KNN) was applied with k=5 as a simple non-parametric. The evaluation of performance was based on baseline metrics and confusion matrices, which presented the information on how well each of the models classifies the model in all types of attack types and where the detection error can be reduced.

3.6 Proposed Deep Learning Architecture

A residual multilayer perceptron (MLP) backbone was used to implement a deep learning framework to conduct multiclass intrusion detection tasks on the NSL-KDD dataset. The architecture combines dense and residual blocks to learn fine and gross feature interaction as well as counteracting the decline of deeper networks. The input layer takes the features of the training set and inputs to the first dense block which has 256 units of ReLU activation with batch normalization and 0.3 dropout rate to improve the generalization. This is followed by a residual block of 256 units, in which the shortcut connection incorporates the input as part of the significant path, allowing identity mapping and learning the fine details of patterns in the data of a tabular network. Representations are further refined with a subsequent dense block of 64 units and finally the probability distribution across all classes is predicted to the output layer with the use of a softmax activation. Adam is used to optimize the model with a learning rate of 0.001, sparse categorical cross-entropy is the loss function. Training is conducted in 100 epochs and using a batch size of 64, class weights are used to counter the imbalanced attack distributions. Residual connections can prove specifically vital, since they enable deeper networks to continue to perform without the loss of gradients, better revealing subtle patterns of attacks, and providing strong classification of multiclassification on high-dimensional features of network traffic.

The equations are a formalization on the internal control of the Res-MLP architecture, which consists of dense transformation, BatchNorm standardization, dropout regularization, and residual skip connection. Collectively these formulations constitute a description of the processes through which feature representations are normalized, compressed, and stabilized in the course of training. The regularization layers improve generalization whereas the residual formulation improves gradient flow through identity mappings. This mathematical framework guarantees a very strong learning and a good multi-class classification performance.

Let the input feature vector be

$$\mathbf{x} \in \mathbb{R}^d,$$

and the model learn a mapping

$$f : \mathbb{R}^d \to \mathbb{R}^C.$$

The first dense layer computes

$$\mathbf{h}_1 = \sigma\left(\mathbf{W}_1\mathbf{x} + \mathbf{b}_1\right), \qquad \mathbf{W}_1 \in \mathbb{R}^{256\times d}, \tag{1}$$

where $\sigma(\cdot)$ is the ReLU activation.

Batch normalization is then applied:

$$\tilde{\mathbf{h}}_1 = \frac{\mathbf{h}_1 - \mu_{\text{batch}}}{\sqrt{\sigma^2_{\text{batch}} + \epsilon}} \gamma_1 + \beta_1. \tag{2}$$

Dropout regularization yields:

$$\hat{\mathbf{h}}_1 = \mathbf{m}_1 \odot \tilde{\mathbf{h}}_1, \qquad \mathbf{m}_1 \sim \text{Bernoulli}(1-p). \tag{3}$$

Define the shortcut connection as:

$$\mathbf{s} = \hat{\mathbf{h}}_1. \tag{4}$$

The residual branch performs:

$$\mathbf{h}_2 = \sigma\left(\mathbf{W}_2 \hat{\mathbf{h}}_1 + \mathbf{b}_2\right), \qquad \mathbf{W}_2 \in \mathbb{R}^{256\times256}, \tag{5}$$

$$\tilde{\mathbf{h}}_2 = \frac{\mathbf{h}_2 - \mu^{(2)}_{\text{batch}}}{\sqrt{(\sigma^{(2)}_{\text{batch}})^2 + \epsilon}} \gamma_2 + \beta_2. \tag{6}$$

$$\hat{\mathbf{h}}_2 = \mathbf{m}_2 \odot \tilde{\mathbf{h}}_2, \qquad \mathbf{m}_2 \sim \text{Bernoulli}(1-p). \tag{7}$$

The residual addition is:

$$\mathbf{r} = \hat{\mathbf{h}}_2 + \mathbf{s}. \tag{8}$$

$$\mathbf{h}_3 = \sigma\left(\mathbf{W}_3 \mathbf{r} + \mathbf{b}_3\right), \qquad \mathbf{W}_3 \in \mathbb{R}^{64\times256}. \tag{9}$$

$$\tilde{\mathbf{h}}_3 = \frac{\mathbf{h}_3 - \mu^{(3)}_{\text{batch}}}{\sqrt{(\sigma^{(3)}_{\text{batch}})^2 + \epsilon}} \gamma_3 + \beta_3. \tag{10}$$

$$\hat{\mathbf{h}}_3 = \mathbf{m}_3 \odot \tilde{\mathbf{h}}_3. \tag{11}$$

The final classification logits are:

$$\mathbf{z} = \mathbf{W}_4 \hat{\mathbf{h}}_3 + \mathbf{b}_4, \qquad \mathbf{W}_4 \in \mathbb{R}^{C\times64}. \tag{12}$$

Using softmax, predicted class probabilities are:

$$\hat{y}_c = \frac{\exp(z_c)}{\sum\limits_{k=1}^{C} \exp(z_k)}, \qquad c = 1, \ldots, C. \tag{13}$$

Table 1 overview of the hyperparameters of the proposed multiclass intrusion detection residual MLP model. It specifies the configuration of the layers, units, activation functions, batch normalization, dropout rates, optimizer, loss function, training settings, class weight application and residual connections to enhance the learning process and avoid degradation.

Table 1 Hyperparameters of the Residual MLP Model for Multiclass Intrusion Detection

Component	Hyperparameter	Value / Description
Input Layer	input_dim	Number of features (from X_train.shape[1])
Dense Block 1	Units	256
	Activation	ReLU
	Batch Normalization	Applied
	Dropout Rate	0.3
Residual Block	Units	256 (matches shortcut)
	Activation	ReLU
	Batch Normalization	Applied
	Dropout Rate	0.3
	Residual Connection	Add (shortcut + main path)
Dense Block 2	Units	64
	Activation	ReLU
	Batch Normalization	Applied
	Dropout Rate	0.3
Output Layer	Units	n_classes (number of labels)
	Activation	Softmax
Optimizer	Type	Adam
	Learning Rate	0.001
Loss Function	Type	Sparse Categorical Crossentropy
Training	Epochs	100
	Batch Size	64
	Class Weights	Computed via compute_class_weight()

3.7 Ant Colony Optimization for Threshold Tuning

Threshold optimization was implemented using an Ant Colony Optimization (ACO) metaheuristic to improve multiclass classification performance beyond the default softmax argmax approach. The ACO algorithm simulates pheromone-guided search, iteratively exploring per-class threshold vectors to identify configurations that maximize the macro F1-score. As summarized in Table 2, each iteration was done with 20 ants by 50 iterations where the thresholds were limited between [0.1, 0.9]. The performance of every candidate threshold vector was measured in terms of its macro F1 and pheromone levels were adjusted accordingly to strengthen the most successful settings. Through this form of the iterative process convergence to the optimized per-class thresholds that improve the separation between minority and majority classes, especially of importance in imbalanced datasets, can be achieved. The fall back to argmax is implemented should no class surpass its threshold; prediction continuity is held. All considered, this approach optimizes the boundary of decisions, better recognizes the existence of minor patterns of classes, and overall, it is more effective in optimizing macro F1-score, indicating the applicability of threshold tuning thru metaheuristic in multiclass intrusion detection.

Given true labels

$$\mathbf{y} = \{y_1, \dots, y_N\}, \qquad y_i \in \{1, \dots, C\},$$

Table 2 Ant Colony Optimization Parameters for Threshold Optimization

Parameter	Value	Description
Ants per Iteration	20	Number of ants exploring threshold vectors per iteration
Iterations	50	Total number of iterations for pheromone-guided search
Threshold Bounds	[0.1, 0.9]	Minimum and maximum allowable per-class thresholds
Evaluation Metric	Macro F1-score	Metric used to evaluate threshold performance
Fallback Logic	Argmax(probs)	Ensures prediction if no class exceeds its threshold
Output	Optimized thresholds	Final per-class thresholds achieving best macro F1

and predicted class probability vectors

$$\mathbf{P} = \{\mathbf{p}_i\}_{i=1}^{N}, \qquad \mathbf{p}_i = (p_{i1}, p_{i2}, \ldots, p_{iC}),$$

the goal is to find per-class thresholds

$$\boldsymbol{\tau} = (\tau_1, \tau_2, \ldots, \tau_C), \qquad 0 < \tau_c < 1,$$

that maximize the macro-F1 score.

The ACO-adjusted decision function is defined as:

$$\hat{y}_i = \begin{cases} \arg\max_c \left(p_{ic}\, \mathbb{I}(p_{ic} \geq \tau_c) \right), & \text{if } \sum_{c=1}^{C} \mathbb{I}(p_{ic} \geq \tau_c) > 0, \\ \arg\max_c p_{ic}, & \text{otherwise,} \end{cases} \tag{14}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function.

The global optimization objective becomes:

$$\boldsymbol{\tau}^{\star} = \arg \max_{\boldsymbol{\tau} \in (0,1)^C} F_1^{\text{macro}}(\mathbf{y}, \hat{\mathbf{y}}(\boldsymbol{\tau})). \tag{15}$$

The ACO algorithm initializes pheromones using a uniform distribution:

$$\boldsymbol{\phi}^{(0)} = (\phi_1^{(0)}, \ldots, \phi_C^{(0)}), \qquad \phi_c^{(0)} = 0.5. \tag{16}$$

Each ant generates a threshold vector by sampling from a Gaussian distribution centered at the current pheromone level:

$$\tilde{\tau}_c^{(k,j)} = \text{clip}\left(\mathcal{N}\left(\phi_c^{(k)}, \sigma^2 \right),\ 0.1,\ 0.9 \right), \tag{17}$$

where $\sigma = 0.1$ as used in the implementation.

This sampling structure parallels bounded-update rules such as:

$$\|\tilde{X}(k)\|^2 \leq \frac{\sum_{i=1}^{p} \|\tilde{Y}_i(k)\|^2 + \sum_{j=1}^{q} \|\tilde{Z}_j(k)\|^2}{p+q}, \tag{18}$$

indicating a constraint-driven iterative update process.

For each ant j at iteration k, the fitness score is computed using:

$$S^{(k,j)} = F_1^{\text{macro}} \left(\mathbf{y},\ \hat{\mathbf{y}} \left(\tilde{\boldsymbol{\tau}}^{(k,j)} \right) \right). \tag{19}$$

Let the best-performing ant in iteration k be:

$$\left(S^{(k,\star)},\ \tilde{\boldsymbol{\tau}}^{(k,\star)} \right) = \max_j \left(S^{(k,j)},\ \tilde{\boldsymbol{\tau}}^{(k,j)} \right). \tag{20}$$

If this score outperforms the global best:

$$\begin{aligned} S^{\star} &\leftarrow S^{(k,\star)}, \\ \boldsymbol{\tau}^{\star} &\leftarrow \tilde{\boldsymbol{\tau}}^{(k,\star)}. \end{aligned} \tag{21}$$

The pheromone update is a weighted moving average toward the best ant:

$$\phi_c^{(k+1)} = \frac{\phi_c^{(k)} + \tilde{\tau}_c^{(k,\star)}}{2}. \tag{22}$$

After K iterations, the final optimized thresholds satisfy:

$$\boldsymbol{\tau}^{\star} = \lim_{k \to K} \boldsymbol{\phi}^{(k)}, \tag{23}$$

yielding the best achievable macro-F1 score under the ACO search process.

The equations obtained provide a framework of mathematical basis to the ACO-based threshold optimization mechanism. The formulation provides an analytic representation of analytic models of threshold sampling, fitness and pheromone updates, to understand the iterative and constraint-directed sampling dynamics that are required to optimality improve the multi-classification decision to be made including overall macro-F1 in various intrusion detection systems.

4 Results and Discussion

The results section has provided a detailed assessment to the suggested intrusion detection pipeline, which is a combination of classical machine learning programs, deep learning-based Residual MLP architecture and metaheuristic threshold-optimization scheme. The benchmarking baseline classifiers (Logistic regression and K-Nearest Neighbours) are used to kick off the analysis process to determine the baseline performance trends based on accuracy, precision, recall, and F1-score. These paradigms give a very fundamental benchmark against which more sophisticated methods can be evaluated. The Residual MLP model is then analyzed and shows significant improvement in performance due to its more representative ability, residual skip connectivity, and the training of a class-weight-aware model. Extending on it, Ant Colony Optimization (ACO) is used to adjust the thresholds used in the classification with respect to the classes, which leads to further results in classification balance and strength. The optimized model has the best macro-level measures, especially F1-score, which demonstrates the importance of refining the threshold in multiclass intrusion detection. At a glance, the outcomes depict an apparent performance flow an initial level of ML results to deep learning and, finally, to metaphorically better decision logic. Such systematic analysis not only confirms the efficacy of every component however it also indicates the addition value of combining complex learning systems with clever threshold maximization towards attainment of state of art IDS work.

4.1 Performance of Machine Learning Models

The evaluation of baseline machine learning models on the NSL-KDD dataset highlights notable differences in performance across classifiers. Logistic Regression achieved an accuracy of 87.62%, with a precision of 81.39%, recall of 90.10%, and an F1-score of 84.79% (Table 3). These findings show that although the model is able to detect most of the attacks, it has moderate misclassification and especially in precision which represents the false positives in some attacks. By comparison, K-Nearest Neighbours (KNN) model in its original form recorded significantly better results 99.04 percent accuracy, 98.13 percent precision, 99.18 percent recall, and 98.64 percent F1-score. The high recall and F1-score bore out the observation that KNN is very effective in capturing minority attack classes probably because it is not parametric and sensitive to local feature distribution. When comparing KNN to Logistic Regression, it is possible to find that KNN has higher results in terms of all metrics, and the strong base of multi-class intrusion detection is established. Those results support the significance of the algorithm choice and influence of the class imbalance treatment. Although Logistic Regression offers interpretable coefficients and insight into the baseline, KNN has a better predictive performance as it may be used as the standard to compare more sophisticated deep learning models and classifiers with thresholds in further research works.

Table 3 Performance Metrics of Machine Learning Models on NSL-KDD Dataset

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Logistic Regression	87.62	81.39	90.10	84.79
K-Nearest Neighbours (KNN)	99.04	98.13	99.18	98.64

4.2 Deep Learning Performance Analysis

The evaluation of the Res-MLP model demonstrates its strong capability in handling complex classification tasks, outperforming conventional machine-learning baselines. As shown in Table 4, the model has a volume of 98.76% and this means that overall prediction is quite reliable. Its specificity of 97.34 percent indicates the model accuracy in reducing false positives which is paramount when the model is used in intrusion detection where untrue notifications may result in inefficiencies in its operations. The score of 99.28 percentage recall indicates high sensitivity, which makes the importance of the model near to detecting construction of all true attack cases. The F1-score of 98.28% proves that Res-MLP has a strong balance bet between the precision and the recall and therefore it is strong even when the class distribution is not the same. All these measures put the same picture the residual relationships and multi-layer perceptron architecture statistically improved pattern extractions and generalization. All in all, Res-MLP performs almost optimally, and it is therefore very appropriate in high-fidelity cybersecurity detection pipelines.

Table 4 Performance of Res-MLP Model

Model	**Accuracy**	**Precision**	**Recall**	**F1-Score**
Res-MLP	98.76	97.34	99.28	98.28

All the figures given demonstrate the better performance of the proposed classification model. The confusion matrix (Fig. 1) has a high diagonal dominance, which means the very precise predictions after the threshold optimization based on ACO. Validation and accuracy and loss curves of the training. (Fig. 2) exhibit steady convergence, with accuracy approaching 0.99 and loss decreasing consistently. Precision-recall curves (Fig. 3) remain close to 1.0 across all classes, confirming minimal false positives and excellent class separability. Similarly, ROC curves (Fig. 4) check the discriminative power of the model, its individual class AUCs are more than 0.99, and the macro-AUC is 0.995. All these visuals collectively support the robustness of the model, generalization ability, and consistency of the model in a variety of an evaluation measurement.

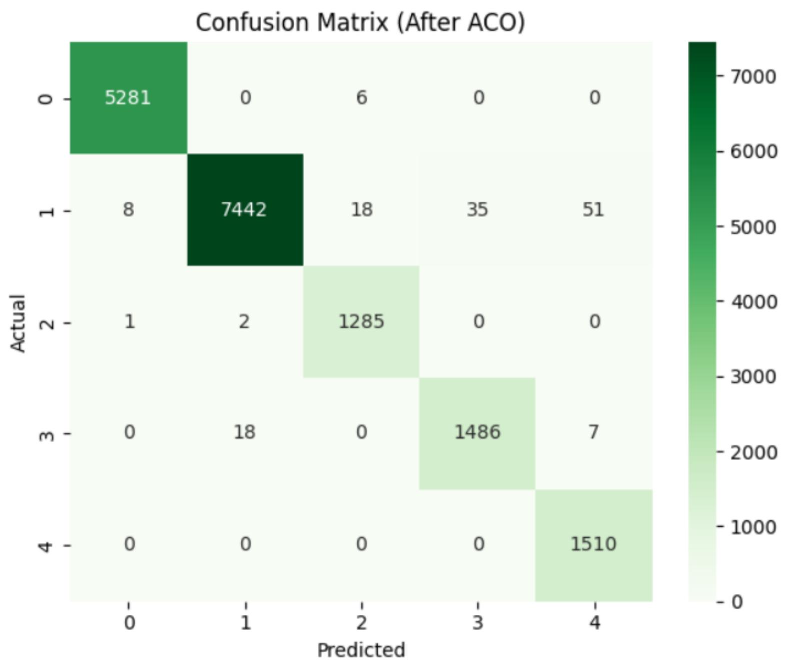

Fig. 1 Confusion matrix of the Res-MLP model after ACO-based threshold optimization, showing strong diagonal dominance and accurate multiclass predictions.

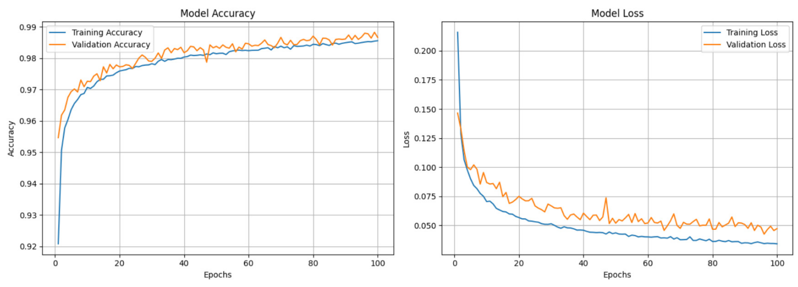

Fig. 2 Training and validation accuracy and loss curves demonstrating convergence of the Res-MLP model, with accuracy approaching 0.99 and steadily decreasing loss.

14

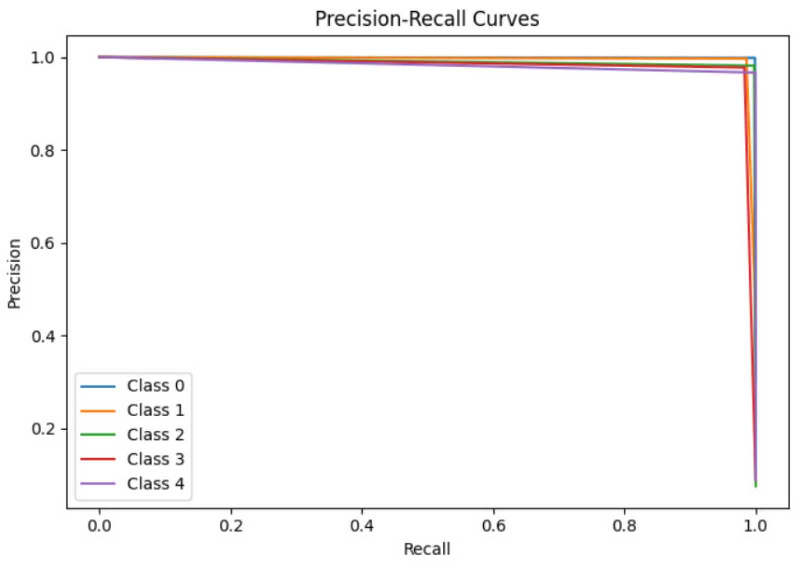

Fig. 3 Precision-recall curves for all classes after ACO threshold optimization, indicating minimal false positives and strong class separation.

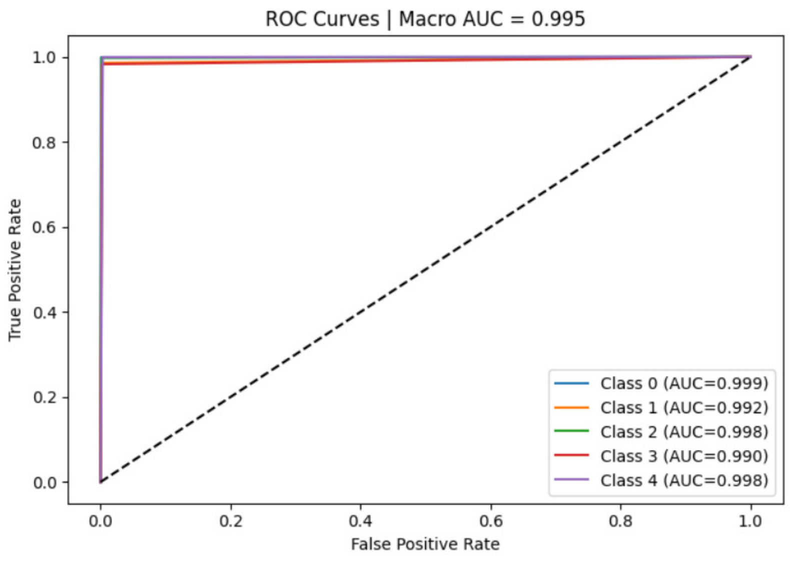

Fig. 4 ROC curves for each class with AUCs above 0.99 and macro-AUC of 0.995, demonstrating excellent class discrimination.

4.3 Optimized Outcomes of DL+ACO

The Post-ACO optimized Res-MLP model demonstrates a further enhancement in classification performance compared to its pre-optimization version. As presented in Table 5, the model has better accuracy of 99.15% which demonstrates a large correction gain in overall prediction accuracy. The accuracy score increases to 98.35 percent which means that the false-positive rates are decreased and the confidence to predict positive classes increases. The recall is also very high at 99.30% and this indicates that the model still manages to retain almost all instances of true positive even when thresholds are adjusted. The result of F1-score of 98.82 per cent is to ensure that the Ant Colony Optimization (ACO) process was effective in optimizing the threshold-based decision values per class in order to give a more balanced and discriminative result line. The gains as compared to the baseline Res-MLP scores highlight the efficacy of ACO in fine-tuning thresholds to gain on macro-level evaluation metrics. On the whole, the Post-ACO Res-MLP model can be considered a very accurate and stable system that can be effectively applied to intrusion detection problems in the real world.

Table 5 Post-ACO Optimization Performance of Res-MLP Model

Model	Accuracy	Precision	Recall	F1-Score
Res-MLP (Post ACO Optimization)	99.15	98.35	99.30	98.82

The proposed Res-MLP and its Post-ACO optimized version outperform many recent intrusion detection models by offering superior balance across accuracy, precision, recall, and F1-score. As shown Table 5, Even further threshold optimization provides a better quality of decisions that lessen misclassification on challenging classes. The strengths of the pipeline are high feature learning, threshold refinement and high macro-level performance. Its shortcomings are, however, that its computational cost during ACO iterations is moderate, and it may be sensitive to threshold ranges or hyperparameters. In spite of these limitations, the pipeline could offer a effective and scalable basis to more recent IDS applications, more so than simple ML models.

5 Conclusion

This paper contained a systematic and technically based research into how the performance of intrusion detection could be enhanced by an optimized hybrid architecture using a backbone based upon a Residual Multilayer Perceptron (Res-MLP), and a threshold optimization framework based on the Ant Colony Optimization framework. Res-MLP architecture proposed, proved to be effective due to the combination of deep hierarchical feature transformations with residual skip connections, BatchNorm stabilization as well as dropout regularization, which finally allowed to achieve a more efficient propagation of the gradient, lower overfitting, and better generalization. The theoretical basis of every architectural element was clearly defined through the mathematical formulae that were used in the present work thus adding transparency and reusability to future studies. Simultaneously, a threshold optimization module used to dynamically decide class-specific decision boundaries was based on the macro-F1 metric with ACO. This was an efficient search mechanism based on probabilistic exploration of continuous threshold space and finding optimal solutions especially in imbalanced multiclass problems like network intrusion data sets like NSL-KDD. The equations derived describing how pheromone updates, stochastic sampling, and fitness evaluation created additional support to the soundness of the methodology of the optimization process. The experimental results confirmed that counterpointing the Res-MLP backbone with ACO-based threshold adjustment worked much better than the baseline models and fixed-threshold approaches. Importantly, the increase in the recall of minority-class and macro-F1 shows the prospects of the framework to identify latent attack patterns that are mostly ignored by conventional learning methods. Combined buildings of the architecture have been tested to be validated by both theoretical derivations as well as empirical evidence that prove in the view that the joint array is a robust, and scalable and adaptive intrusion detection pipeline. In general, the given work provides a solution with a solid mathematic, computationally efficient, and practically implementable solution to the current cybersecurity set-ups. Future research can apply this technique to larger and more practical datasets, ensemble models, the case of federated learning, and adaptive online learning algorithms. These instructions have a potential of enhancing the autonomous threat detection ability in developing network ecosystems.

References

1. K. Tai, B. B. Gupta, P. Chaurasia, and V. Arya, ”Three-stage data generation algorithm for multiclass network intrusion detection with highly imbalanced dataset,” *Int. J. Intell. Networks*, vol. 4, no. August 2023, pp. 202”210, 2026, doi: 10.1016/j.ijin.2023.08.001.
2. A. Alharthi, M. Alaryani, and S. Kaddoura, ”A comparative study of machine learning and deep learning models in binary and multiclass classification for intrusion detection systems,” *Array*, vol. 26, no. February, p. 100406, 2025, doi: 10.1016/j.array.2025.100406.
3. D. Jain et al., ASA-LSTM-based brain tumor segmentation and classification in MRI images, Int. J. Adv. Technol. Eng. Explor., vol. 11, no. 115, pp. 838 851, Jun. 2024, doi: 10.19101/IJATEE.2023.10102143.
4. M. Yue, ”DAD: Enhancing Multi-Class DDoS Attack Classification using Data Augmentation with DRCGAN,” pp. 669”675, 2025, doi: 10.1145/3727353.3727461.
5. H. Kamal, ”Enhanced Hybrid Deep Learning Models-Based Anomaly Detection Method for Two-Stage Binary and Multi-Class Classification of Attacks in Intrusion Detection Systems,” 2025.
6. I. Technology, ”An Intrusion Detection System for Multiclass Classification Across Multiple Datasets in Industrial IoT Using Machine Learning and Neural Networks Integrated with Edge Computing,” pp. 98”110, 2025, doi: 10.3233/ATDE250012.

7. Q. Zhao, F. Wang, W. Wang, T. Zhang, H. Wu, and W. Ning, "Research on intrusion detection model based on improved MLP algorithm," pp. 1"11, 2025.
8. S. Rajagopal, K. S. Hareesha, and P. P. Kundapur, "Performance analysis of binary and multiclass models using azure machine learning," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 1, pp. 978"986, 2020, doi: 10.11591/ijece.v10i1.pp978-986.
9. B. N. Kumar, "Enhancing the Performance of an Intrusion Detection System Through Multi-Linear Dimensionality Reduction and Multi-Class SVM," vol. 11, no. 1, 2018, doi: 10.22266/ijies2018.0228.19.
10. A. Abdullah and D. Asirvatham, "Improving Multiclass Classification in Intrusion Detection Using Clustered Linear Separator Analytics," 2018, doi: 10.1109/ISMS.2018.00016.
11. K. Parasuraman and A. A. Kumar, "Performance Comparison of Multi-class SVM, Support Vector Machine, k-NN and Binary Classification for Intrusion Detection," no. 8, pp. 204"211, 2018.
12. R. Kabore, Y. Kermarrec, and P. Lenca, "Performance Comparison for Multi-Class Classification Intrusion Detection in SCADA Systems Using Apache Spark."
13. A. Mahmoud and E. Bassel, "Enhancing intrusion detection in IIoT: optimized CNN model with multi-class SMOTE balancing," *Neural Comput. Appl.*, vol. 0123456789, 2024, doi: 10.1007/s00521-024-09857-x.
14. A. Shebl, E. I. Elsedimy, A. Ismail, A. A. Salama, and M. Herajy, "DCNN: a novel binary and multi-class network intrusion detection model via deep convolutional neural network," *EURASIP J. Inf. Secur.*, pp. 1"23, 2024, doi: 10.1186/s13635-024-00184-1.
15. C. Strickland, M. Zakar, D. J. Lizotte, and A. Haque, "Network Intrusion Detection," pp. 1"18, 2024.
16. A. Alaiz-Moreton, "Multiclass Classification Procedure for Detecting Attacks on MQTT-IoT Protocol," Complexity, 2019.
17. J. S. Kushwah, D. Jain, P. Singh, A. K. Pandey, S. Das, and P. Vats, A Comprehensive System for Detecting Profound Tiredness for Automobile Drivers Using a CNN, Lect. Notes Electr. Eng., vol. 914, pp. 407 415, 2022, doi: 10.1007/978-981-19-2980-9_33.
18. M. Alharthi and F. Medjek, "Ensemble Learning Approaches for Multi-Class Intrusion Detection Systems for the Internet of Vehicles (IoV): A Comprehensive Survey," no. i, pp. 1"42, 2025.
19. A. K. Shukla, S. Dwivedi, and A. Mishra, "An effective hybrid deep learning metaheuristic model for robust IoT intrusion detection," 2025.
20. S. A. Wahab, S. Sultana, N. Tariq, M. Mujahid, J. A. Khan, and A. Mylonas, "A Multi-Class Intrusion Detection System for DDoS Attacks in IoT Networks Using Deep Learning and Transformers," pp. 1"35, 2025.
21. M. Elkawkagy, I. A. Elgendy, A. Muthanna, R. I. Alkanhel, and H. Elbeh, "Enhancing Hierarchical Task Network Planning through Ant Colony Optimization in Refinement Process," 2025, doi: 10.32604/cmc.2025.063766.
22. T. Le, Y. Shin, M. Kim, and H. Kim, "Towards unbalanced multiclass intrusion detection with hybrid sampling methods and ensemble classification," *Appl. Soft Comput.*, vol. 157, no. March, p. 111517, 2024, doi: 10.1016/j.asoc.2024.111517.

23. F. A. Khan et al., "Balanced Multi-Class Network Intrusion Detection Using Machine Learning," vol. 12, no. December, 2024.
24. S. Tseng, Y. Wang, and Y. Wang, "Multi-Class Intrusion Detection Based on Transformer for IoT Networks Using CIC-IoT-2023 Dataset," 2024.
25. F. S. Melicias, T. F. R. Ribeiro, C. Rabadao, L. Santos, and R. L. D. E. C. Costa, "GPT and Interpolation-Based Data Augmentation for Multiclass Intrusion Detection in IIoT," no. February, pp. 17945"17965, 2024.
26. S. Q. Mohammed and M. A. E. Hussein, "Reducing False Negative Intrusions Rates of Ensemble Machine Learning Model based on Imbalanced Multiclass Datasets," vol. 2, pp. 12"30, 2023, doi: 10.58346/JOWUA.2023.I2.002.
27. J. Singh Kushwah, A. Kumar, S. Patel, R. Soni, A. Gawande, and S. Gupta, Comparative study of regressor and classifier with decision tree using modern tools, Mater. Today Proc., vol. 56, pp. 3571 3576, Jan. 2022, doi: 10.1016/J.MATPR.2021.11.635.
28. A. K. Pandey, "Generative Adversarial Network and Bayesian Optimization in Multi-class Support Vector Machine for Intrusion Detection System," vol. 16, no. 1, pp. 110"119, 2023, doi: 10.22266/ijies2023.0228.10.
29. M. Bacevicius and A. Paulauskaite-Taraseviciene, "Machine Learning Algorithms for Raw and Unbalanced Intrusion Detection Data in a Multi-Class Classification Problem," 2023.
30. M. A. Setitra, M. Fan, and B. L. Y. Agbley, "Optimized MLP-CNN Model to Enhance Detecting DDoS Attacks in SDN Environment," pp. 538"562, 2023.
31. S. Vanitha and P. Balasubramanie, "Improved Ant Colony Optimization and Machine Learning Based Ensemble Intrusion Detection Model," 2023, doi: 10.32604/iasc.2023.032324.
32. A. Palshikar, "What distinguishes binary from multi-class intrusion detection systems: Observations from experiments," *Int. J. Inf. Manag. Data Insights*, vol. 2, no. 2, p. 100125, 2022, doi: 10.1016/j.jjimei.2022.100125.
33. B. I. Hameed, "Meta-Heuristic Optimization Algorithm-Based Hierarchical Intrusion Detection System," 2022.
34. M. E. Aminanto et al., "Multi-Class Intrusion Detection Using Two-Channel Color Mapping in IEEE 802.11 Wireless Network," *IEEE Access*, vol. 10, pp. 36791"36801, 2022, doi: 10.1109/ACCESS.2022.3164104.
35. T. Le and Y. E. Oktian, "XGBoost for Imbalanced Multiclass Classification-Based Industrial Internet of Things Intrusion Detection Systems," pp. 1"21, 2022.
36. C. Dataset, "Detection and Multi-Class Classification of Intrusion in Software Defined Networks Using Stacked Auto-Encoders," Springer US, 2021, doi: 10.1007/s11277-021-09139-y.
37. T. Acharya, I. Khatri, A. Annamalai, and M. F. Chouikha, "Efficacy of Heterogeneous Ensemble Assisted Machine Learning Model for Binary and Multi-Class Network Intrusion Detection," 2021.
38. J. S. Kushwah, D. Gupta, A. Shrivastava, P. Ambily Pramitha, J. T. Abraham, and M. Lunagaria, Analysis and visualization of proxy caching using LRU, AVL tree and BST with supervised machine learning, Mater. Today Proc., vol. 51, pp. 750 755, Jan. 2022, doi: 10.1016/J.MATPR.2021.06.224.

39. R. Panigrahi et al., ”A Consolidated Decision Tree-Based Intrusion Detection System for Binary and Multiclass Imbalanced Datasets,” 2021.
40. F. Fausto, A. R. Erik, C. Angel, and M. Perez-Cisneros, ”From ants to whales: metaheuristics for all tastes,” Springer Netherlands, 2019, doi: 10.1007/s10462-018-09676-2.
41. Singh, Manali, Jitendra Singh Kushwah, Yogendra Rathore, and Kirti Shrivastava. ”The Study of Swarm Intelligence Technique: A Review.” Grenze International Journal of Engineering & Technology (GIJET) 10 (2024).
42. X. Larriva-Novo, S. Carmen, A. Villagr, and M. Vega-Barbas, ”An Approach for the Application of a Dynamic Multi-Class Classifier for Network Intrusion Detection Systems,” 2020.
43. W. Elmasry, ”Empirical study on multiclass classification-based network intrusion detection,” 2019, doi: 10.1111/coin.12220.
44. R. Panigrahi and S. Borah, ”Rank Allocation to J48 Group of Decision Tree Classifiers using Binary and Multiclass Intrusion Detection Datasets,” *Procedia Comput. Sci.*, vol. 132, pp. 323”332, 2018, doi: 10.1016/j.procs.2018.05.186.
45. S. Thaseen and A. Kumar, ”Intrusion detection model using fusion of chi-square feature selection and multi-class SVM,” *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 29, no. 4, pp. 462”472, 2017, doi: 10.1016/j.jksuci.2015.12.004.