

Hybrid feature selection strategies for strong network intrusion detection

Josephine Rathinam, Anandaraj Shanthi Pichandi

Presidency School of Computer Science and Engineering, Presidency University, Bengaluru, India

Article Info

Article history:

Received Sep 30, 2025

Revised Jul 25, 2026

Accepted Aug 12, 2026

Keywords:

CIC-IDS2017 dataset

Genetic algorithm

Hybrid feature selection

Multi-layer perceptron

Network intrusion detection system

ABSTRACT

High dimensional network traffic datasets pose great challenges to network intrusion detection systems (NIDS) due to redundant and irrelevant features, which decrease the detection accuracy and increase the computational cost. Traditional feature selection methods often cannot achieve a balance between classification performance and processing efficiency. This paper proposes hybrid feature selection for network intrusion detection systems (HybFS-NIDS), a hybrid feature selection framework using filter- and wrapper-based optimization methods to detect the most significant intrusion detection features to do this. The proposed model was evaluated on the CIC-IDS2017 dataset using performance metrics such as accuracy, precision, recall, F1-score, and Matthew's correlation coefficient (MCC). The experimental results selected 18 optimal features from the initial 78 features of the dataset and achieved 90.68% accuracy, 91.74% precision, 90.64% recall, 91.19% F1-score, and an MCC of 0.9148. The proposed method also achieved a 41.5% reduction in training time compared with the full-feature baseline model while improving the overall classification performance. The results show that HybFS-NIDS can successfully improve the efficiency of intrusion detection and fit for real-time cybersecurity applications with a large amount of network traffic data.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Josephine Rathinam

Presidency School of Computer Science and Engineering, Presidency University

Bengaluru, India

Email: josephine.20223cse0026@presidencyuniversity.in

1. INTRODUCTION

The growth of digital communication and the spread of internet-connected devices have led to an increase in the prevalence of cyberattacks on computing systems [1]. Network intrusion detection systems (NIDS) are used to monitor network traffic for malicious activity and to ensure data security, including availability, confidentiality, and integrity [2]. In modern high-dimensional data such as CIC-IDS2017, there exists relevant data and irrelevant information [3]. The redundant features in the dataset also contribute to the computing time, complexity, and overfitting risk of the model, which decreases detection performance [4]. Real-time intrusion detection is very difficult because of high-dimensional network traffic data [5]. Many traditional feature selection methods do not sufficiently resolve the trade-off between classification accuracy and the computing economy [6]. This makes it either have low detection accuracy or a high computing cost [7]. An efficient feature selection method that guarantees high accuracy as well as reduced complexity is therefore necessary to offer efficient real-time NIDS functionality [8].

In the case of intrusion detection, researchers have used most feature selection methods over the years [9], [10]. Because of their low computational cost, filter-based methods such as information gain (IG), chi-square test, and mutual information are often used as intrusion detection methods. They have also been

used because of the ability of wrapper techniques such as sequential forward selection (SFS) and sequential backward selection (SBS) to directly optimize feature subsets, depending on classifier performance, which has also led to their utilization. In order to strike a balance between precision and effectiveness, some hybrid systems that combine both filter and wrapper techniques have also been developed. Moreover, NIDS feature selection has also made use of meta-heuristic optimization methods such as particle swarm optimization (PSO), ant colony optimization (ACO), and genetic algorithm (GA).

These existing works have the following disadvantages: filter methods ignore feature dependencies and generally select irrelevant features along with relevant ones. Wrapper methods are computationally expensive and therefore not suitable for large-scale or real-time detection. The hybrid approaches used in previous works generally combine basic filter methods and computationally expensive wrappers, so the methods developed are not scalable. Many studies do not systematically combine the strengths of multiple advanced filter methods before applying the wrapper-based optimization method. Optimization algorithms are applied without first performing feature ranking, which needlessly searches over poor-quality features. It is suggested in this research that a novel technique called hybrid feature selection for network intrusion detection systems (HybFS-NIDS) is introduced to address the aforementioned drawbacks. The most pertinent characteristics are optimized through the integration of multiple advanced filter techniques with a meta-heuristic wrapper. By this approach, high detection accuracy is maintained, and computational complexity is reduced.

To produce feature ranking and redundancy information, three advanced filter techniques, ReliefF, fast correlation-based filter (FCBF), and minimum redundancy maximum relevance (mRMR), are first applied in the proposed method. The obtained rankings are then combined through the Borda count voting method to generate a top-ranked feature set as well. This reduced set is subjected to a wrapper based on GA. Feature subsets within this reduced collection are evaluated using a multi-layer perceptron (MLP) classifier with cross-validation. The wrapper-selected and filter-ranked features are combined by an MLP model to get a classifier-aware and statistically important feature set. This study presents HybFS-NIDS, a hybrid filter-wrapper method that uses different filter methods, the Borda count ranking method, and GA-based improvement in this paper, to select smaller feature subsets, improve the intrusion detection accuracy, reduce the processing cost and training time, and to help real-time NIDS applications. HybFS-NIDS differs from traditional methods by combining mRMR, FCBF, and ReliefF with rank aggregation and set intersection before GA optimization, thus improving the detection performance and reducing the computational complexity. Therefore, the proposed framework is applicable to enterprise and government networks, cloud security, internet of things (IoT) and industrial systems, and real-time intrusion detection.

Different filter, wrapper, ensemble, and optimization-based techniques have been used to improve feature selection in NIDS, leading to better detection accuracy and reduced dimensionality [9]–[13]. Promising results have been reported for extreme gradient boosting (XGBoost) with feature selection [11], improved grey wolf optimization (IGWO)-based feature selection [12], neural network-assisted feature selection [13], and dataset standardization methods [14]. The ensemble classification with feature selection in cloud environments [15], majority voting [16], chi-square with generative adversarial network (GAN) balancing [17], chaotic crow search [18], and bio-inspired algorithms such as PSO, grey wolf optimization (GWO), firefly algorithm (FFA), and GA [19] were used to obtain optimal feature subsets. Other studies focused on bio-inspired IoT intrusion detection [20], hybrid kernel support vector machine (SVM) with ensemble feature selection [21], IG on CIC-IDS2017 [22], weighted feature selection [23], optimal neural networks [24], and meerkat clan algorithm [25]. However, existing methods still have some limitations, such as dependency on single feature selection techniques, lack of consideration of feature dependencies, high computational cost in wrapper-based approaches, lack of systematic hybridization, and inconsistency in the reproducibility of feature subsets across different runs.

To solve these problems, the proposed HybFS-NIDS uses the optimal subsets from the filter and wrapper stages to obtain the final feature subset. Since the final feature subset is intersected with the optimal subsets of two stages, the proposed method can guarantee that the final feature subset is robust and considered as important as possible by different strategies. In comparison with existing approaches, the proposed HybFS-NIDS demonstrates improved detection accuracy, greater stability of the selected feature subset, and reduced computational complexity. The method is appropriate for high-dimensional datasets such as CIC-IDS2017 and can be adapted for deployment in real-time intrusion detection environments. The weaknesses of traditional feature selection methods in NIDS pointed out by many earlier studies. For example, few wrapper-based techniques tested on the CIC-IDS2017 dataset reported high accuracy. But they took a long time to run. Because the model had to be trained again and again while checking different feature subsets. On the other hand, filter-based methods depend only on statistical measures and often kept extra features. As a result, the improvement in accuracy was small. Also, the number of features was not eliminated effectively. Therefore, it is not always the case that only filter methods or only wrapper methods provide a good

trade-off between detection performance and computational cost. This is the reason why the HybFS-NIDS framework is required.

The rest of this paper is organized as follows: section 2 explains the dataset and the proposed HybFS-NIDS framework. Section 3 presents the research method in detail. Section 4 presents the experimental setup and discusses the results and comparisons with existing methods. Section 6 is the summary of the conclusions and proposals for future work.

2. PROPOSED METHOD

This section outlines the methodology, namely HybFS-NIDS. This method is used to combine different filter-based ranking methods and a wrapper-based meta-heuristic optimization to find out the most relevant and non-redundant features to be used in intrusion detection. The last optimized subset is tested by a MLP classifier. The performance is then assessed based on various measures to ensure its robustness and utility.

2.1. Dataset description

The CIC-IDS2017 dataset is used as the experimental framework, consisting of a vast intrusion detection dataset that covers a large variety of normal traffic and numerous types of attacks, including denial of service (DoS), distributed denial of service (DDoS), intrusion, brute force, online, and botnets. The dataset consists of actual traffic conditions and contains over 80 network parameters like flow time, packet size metrics, header contents, and inter-arrival time (IAT). Before feature selection was done, data preprocessing was done in order to have high data quality:

- Missing value imputation—missing (records) of some flow-based features were imputed based on the mean (numerical) and mode (categorical) attributes.
- Duplicate removal—duplicates were removed so as to prevent bias in the feature selection process.
- Outlier handling—the interquartile range (IQR) was used to detect extreme values, which were capped to decrease skewness.
- Normalization—all of the numeric features were normalized with min-max normalization to bring all values within the range [0, 1], so that each feature contributes the same distance-based computation.

In order to measure the effects of preprocessing, Table 1 provides percentages of records affected at every step. This guarantees transparency in terms of data quality improvement before feature selection. After the dataset was preprocessed, the resulting dataset was prepared to run the hybrid feature selection pipeline to produce a subset of features optimized for intrusion detection.

Table 1. Impact of data preprocessing on the CIC-IDS2017 dataset

Preprocessing step	Affected records	Percentage (%)	Description of action
Missing value imputation	12,845	1.73%	Numerical features imputed using mean; categorical using mode
Duplicate removal	8,214	1.10%	Exact duplicate flow records removed
Outlier handling (IQR capping)	24,506	3.29%	Extreme values capped to IQR bounds
Normalization	100% of numerical features	—	Min-max scaling to range [0,1]

3. HYBFS-NIDS ALGORITHM AND IMPLEMENTATION

The HybFS-NIDS algorithm integrates both filter-based and wrapper-based selection strategies, thereby combining the advantages of the two approaches. Filter methods are fast and effective since they rapidly screen the features in accordance with their statistical significance. In contrast, wrapper methods consider subsets of characteristics in the environment of a particular classifier in order to enhance the predicted accuracy. The HybFS-NIDS algorithm is illustrated in Algorithm 1.

Algorithm 1. HybFS-NIDS

Input: preprocessed CIC-IDS2017 dataset (cleaned, normalized, and outlier-handled).

Output: optimized feature subset for improved NIDS accuracy and efficiency.

Steps:

- 1: Dataset loading
Load the preprocessed dataset into the feature selection environment.
- 2: Filter-based selection
ReliefF: weigh features by their ability to separate close but differently labeled samples.
FCBF: use symmetrical uncertainty (SU) to retain relevant, non-redundant features.
mRMR: select features maximizing relevance to the target while minimizing redundancy.
- 3: Filter results aggregation
Apply Borda count to merge rankings from all filter methods, selecting the top-N features with highest aggregate scores.

- 4: Wrapper-based selection (meta-heuristic)
 Use a GA:
 Encode feature subsets as chromosomes. Evolve via selection, crossover, and mutation.
 Fitness: MLP classifier performance via cross-validation.
- 5: Hybrid selection
 Intersect top filter-selected features with GA-optimized wrapper results to retain only consistently.

3.1. Step 1: filter-based selection

The process of filter selection that involves the use of filters begins by applying three different algorithms, which are ReliefF, FCBF, and mRMR, to assess and rank the significance of individual features. These different rankings are then combined by the use of the Borda count technique. This strategy sums up the scores to determine the attributes that have a high score across all approaches. The N features with the highest cumulative scores are chosen, thus ending the filter-based selection stage.

Between HybFS-NIDS, ReliefF, FCBF, and mRMR, the latter is selected because of the necessity to process high-dimensional cybersecurity data more efficiently. Most standard filter techniques, such as IG and chi-square, typically examine the relationship of each feature to the target class. However, they do not address the interrelationships of features in the right way. Comparatively, mRMR picks features by retaining those having strong relatedness to the target as well as eliminating those that have similar information. It prevents repetition. Moreover, FCBF assists in this process by removing irrelevant features with help of SU.

This SU works well for large datasets such as CIC-IDS2017. ReliefF adds one more benefit by evaluating features using individual data instances and capturing local patterns in the data. This makes it useful for identifying small differences between normal and attack traffic. By combining these three methods, the HybFS-NIDS framework creates a more balanced and reliable feature ranking process than only relying on basic statistical filters.

3.1.1. ReliefF algorithm

ReliefF assesses the significance of a feature by repeatedly selecting samples and comparing the differences between the closest neighbors belonging to the same class, called the nearest hit, and those belonging to other classes, called the nearest miss. Greater weight is assigned to features that more effectively distinguish between classes. The corresponding weight update equation is given as (1), where $W[A]$ represents the weight assigned to feature A , m denotes the total number of samples, and $diff(.)$ indicates the normalized difference calculated between the corresponding feature values.

$$W[A] = W[A] - \frac{diff(A_{nearest\ hit})}{m} + \frac{diff(A_{nearest\ miss})}{m} \quad (1)$$

3.1.2. Fast correlation-based filter

In FCBF, SU is employed to identify features that are strongly related to the target variable while remaining non-redundant with respect to each other. The SU metric is defined as (2). Where $IG(X|Y)$ denotes the IG and $H(.)$ represents the entropy function.

$$SU(X, Y) = \frac{2IG(X|Y)}{H(X) + H(Y)} \quad (2)$$

3.1.3. Minimum redundancy maximum relevance

The mRMR approach chooses features by maximizing their relevance to the target variable while simultaneously minimizing redundancy among the selected features. The scoring function is defined as (3), where $I(.,.)$ denotes the mutual information, S indicates the feature subset, and c refers to the class label. The filter-based feature selection process is illustrated in Figure 1.

$$\max \left[\frac{1}{|S|} \sum_{f \in S} I(f; c) - \frac{1}{|S|^2} \sum_{f_i, f_j \in S} I(f_i; f_j) \right] \quad (3)$$

3.2. Step 2: rank aggregation via Borda count

The rankings produced by the three filter techniques are combined by using the Borda count method as shown in (4). Here N denotes the total number of features, M represents the number of ranking approaches, and $Rank_k(f)$ is the ranking position assigned to feature f by method k . The features that have obtained the highest combined scores are selected for the wrapper stage.

$$Score(f) = \sum_{k=1}^M (N - Rank_k(f)) \quad (4)$$

3.3. Step 3: wrapper-based selection with genetic algorithm

The wrapper-based feature selection process is initiated by encoding prospective feature subsets as chromosomes, which are then evolved by the GA operations, which include selection, crossover, and mutation. The efficacy of each subset is evaluated by the performance of an MLP classifier. The procedure is repeated until convergence is achieved or a stopping criterion is met. This results in the identification of the optimal feature subset that maximally enhances classification performance. Within the GA framework, a chromosome is encoded as a binary string in which a value of 1 denotes that a feature is selected, whereas 0 indicates that it is not included. The fitness function is defined as (5). Where α and β represent weights (here, each weight is 0.5 to be equally important). The GA is used to evolve the population by making use of selection, crossover, and mutation to maximize classification. This wrapper-based selection of features is illustrated in Figure 2.

$$Fitness = \alpha \cdot Accuracy + \beta \cdot F1 - score \quad (5)$$

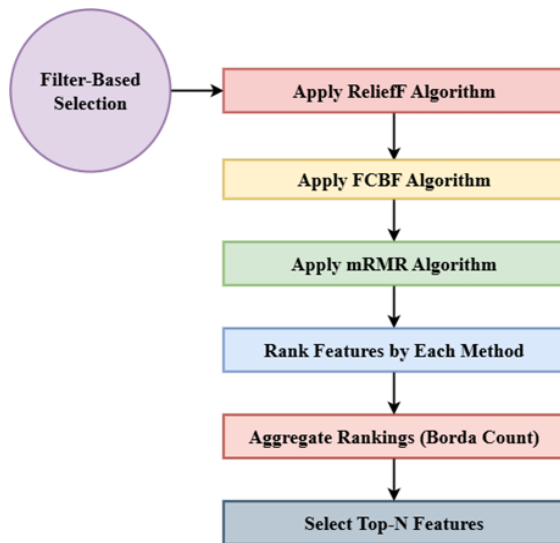


Figure 1. Filter-based feature selection

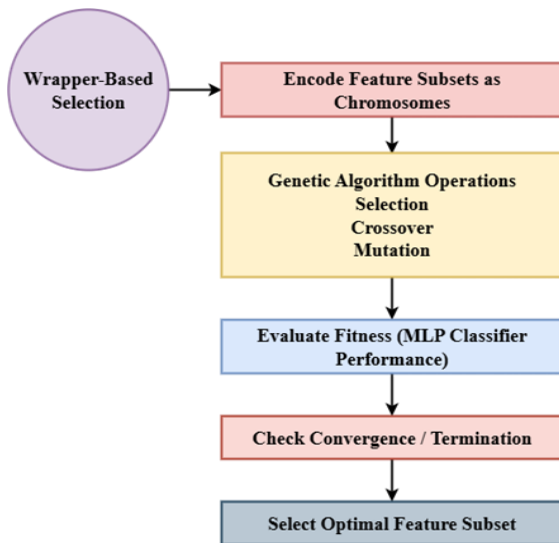


Figure 2. Wrapper-based feature selection

3.4. Step 4: hybrid selection (intersection)

The hybrid intersection process begins with the feature subsets that were selected in the filter-based and the GA-based processes. Then, it runs a set intersection to retain only features that appear important to both methods. This results in the optimized feature set, which terminates the hybrid selection process. This guarantees that only features that are important in all stages are retained, and a balance between statistical significance and prediction is achieved.

The definitions of the feature sets are provided to explain the hybrid intersection step clearly. First, the top-N features that had been chosen by the three filter methods with the application of the Borda rank aggregation are FRelief, FFCBF, FFCBF, and FmRMR. As a whole, individual filters provide their lists of rankings, and these rankings are merged to create a common ranking.

Next, fuzzy genetic algorithm (FGA) is the optimized feature subset used by the GA during the wrapper step. The phase is used to determine a good combination of features through testing different subsets. Now comes the main idea. In order to be included in the final subset, a feature must meet two conditions. It should be included in the composite filter set, and it should also be chosen by the GA. So, both stages must agree. This means that the feature is statistically significant and also useful to the classifier.

The aggregated filter set is defined as $FFilter = \{f | f \text{ appears in at least two of the three filter rankings}\}$. In practice, this means a feature should be supported by at least two filters before moving forward. One reason for this rule may be to avoid depending too much on a single method. Sometimes one filter may favor a feature that others do not. The process seems more stable by having to agree. Meanwhile, the final decision is determined by the GA, which verifies the result of the features operating together in the model. The final optimized feature subset is thus computed as in (6).

$$FFinal = FFilter \cap FGA \quad (6)$$

This majority-based consensus (at least two filters) ensures robustness by reducing the influence of any single ranking method, while the subsequent intersection with the GA-selected subset guarantees predictive effectiveness through classifier-based validation. The hybrid intersection is indicated in Figure 3.

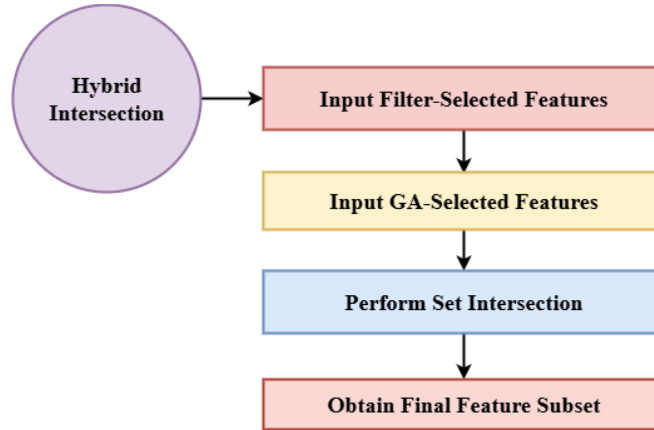


Figure 3. Hybrid intersection

3.5. Step 5: model evaluation metrics

The final subset is validated with the help of an MLP classifier and assessed with the help of various metrics:

- i) Accuracy: indicates the overall classification correctness by calculating the proportion of correctly predicted instances among all samples (7). Where true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). This metric was used to measure the percentage of accurate predictions of all the classes.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (7)$$

- ii) Precision: evaluates the accuracy of positive predictions by determining the proportion of correctly predicted positive instances among all predicted positives in (8).

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

Precision is considered important for evaluating how effectively false alarms are minimized by the NIDS. Recall measures the proportion of actual positive instances that are correctly identified by the model in (9).

$$Recall = \frac{TP}{TP + FN} \quad (9)$$

High recall is ensured so that most attack instances are correctly detected.

- iii) F1-score: represents the harmonic average of precision and recall, offering a combined evaluation of both measures as defined in (10).

$$F1-score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \quad (10)$$

By this balanced assessment, the detection performance and the false alarms are maintained.

- iv) Matthew's correlation coefficient (MCC): serves as a balanced performance metric for binary classification tasks by taking into account all prediction outcomes as described in (11).

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (11)$$

MCC is selected as it accounts for all four confusion matrix categories and remains dependable even when the dataset is imbalanced. Table 2 presents the parameter configuration for HybFS-NIDS.

Table 2. Experimental parameter settings for HybFS-NIDS

Component	Parameter	Value
Filter stage	Initial number of features	78
	Top-N features selected after Borda aggregation	30
GA (wrapper stage)	Population size	50
	Number of generations	100
	Selection method	Tournament selection ($k = 3$)
	Crossover rate	0.8
	Mutation rate	0.02
	Chromosome representation	Binary encoding
	Fitness function weights (α, β)	0.5, 0.5
MLP classifier	Input layer size	Depends on selected features
	Hidden layers	2
	Neurons (hidden layer 1)	64
	Neurons (hidden layer 2)	32
	Activation function	ReLU (hidden), softmax (output)
	Optimizer	Adam
	Learning rate	0.001
	Batch size	128
	Epochs	50
	Loss function	Categorical cross-entropy
Evaluation	Cross-validation	10-fold

The HybFS-NIDS method consists of statistical feature ranking and evolutionary search to determine the most appropriate set of features of the CIC-IDS2017 dataset. The method ensures that computational and prediction power are high through ReliefF, FCBF, and mRMR, as well as Borda count aggregation and GA-based wrapper selection. Not only will the final model be capable of determining the intrusions with great accuracy, but it will also be capable of addressing real-world intrusion detection situations quite well, as the various evaluation measures are used, including accuracy, precision, recall, F1-score, and MCC. The optimized feature subset developed using this process is anticipated to decrease computational overhead while improving the detection capabilities of NIDS systems.

- Computational complexity analysis: let, n is number of samples, m is number of features, G is number of GA generations, P is GA population size, and T_{MLP} is time complexity of MLP training.
- Filter stage: ReliefF requires nearest neighbor search for each sample $\rightarrow$ complexity: $O(n^2m)$. FCBF computes symmetrical uncertainty between features $\rightarrow$ complexity: $O(m^2n)$. mRMR is mutual information pairwise computation $\rightarrow$ complexity: $O(m^2n)$. Borda aggregation is sorting and summation $\rightarrow$ complexity: $O(m \log m)$. Overall filter stage complexity: $O(n^2m + m^2n)$.
- Wrapper stage (GA): for each generation is fitness evaluation: P chromosomes $\times$ MLP training $\rightarrow O(P \times T_{MLP})$. For G generations: $O(G \times P \times T_{MLP})$.
- Hybrid intersection: set intersection operation is $O(m)$. Overall complexity, the total computational complexity of HybFS-NIDS is $O(n^2m + m^2n + G \times P \times T_{MLP})$.

The computational complexity analysis shows that HybFS-NIDS is computationally costly during the training and feature-selection stage, especially when the number of samples and features is high. The major computational burden is from the quadratic operations in the filter step, especially the nearest-neighbor computation of ReliefF, and the GA-based wrapper step that requires training the MLP classifier for a number of chromosomes over many generations. This leads to relatively high computational costs in the development of the offline model for the proposed method. However, the computational cost is mainly in the feature selection and training stage, but not in the final intrusion detection process. After optimization, HybFS-NIDS reduces the original feature space to a compact subset of 18 features. It reduces the amount of data processed in subsequent intrusion detection. Thus, the proposed method offers a trade-off between higher offline computational cost and reduced feature processing requirements during deployment.

Since wrapper evaluation dominates runtime, the GA stage is the primary computational bottleneck, and the filter stage greatly reduces dimensionality prior to wrapper optimization, thus decreasing the overall cost of processing. The complete experimental flow of the HybFS-NIDS framework is illustrated in Figure 4. The procedures were performed in an identifiable sequence to ensure that other individuals can perform the same procedure without confusion.

First, raw data of the traffic flow provided in the CIC-IDS2017 dataset were collected and pre-processed. Mean imputation was used to remove duplicate records and missing values. Practically, this stage of cleaning the data is aimed at preventing unexpected errors in the future.

Next, the encoding of categorical features into numerical ones was done. Min-max scaling was used to normalize the numerical attributes. This step maintains all values at a similar range, which typically facilitates easier learning by the classifier.

The features were ranked by three distinct filter methods after preprocessing: ReliefF, FCBF, and mRMR. Both approaches resulted in their ranking lists. The results of these rankings were then combined with the Borda count to select the N candidate features. This suggests that features supported by multiple filters are given more importance.

The next step was the GA wrapper. It was run under fixed parameters, such as population size, mutation rate, crossover rate, and number of generations. An MLP classifier was used to evaluate each feature combination during the search process. In simple terms, the GA experimented with various subsets and kept the better ones based on classifier performance.

Finally, 10-fold cross-validation was used to validate the selected feature subset. This repetition of testing helps to enhance the statistical stability and reduces the effect of random data splits. All the experiments were performed in a controlled computational environment. So, the results should be consistent and reasonably easy to replicate under similar conditions.

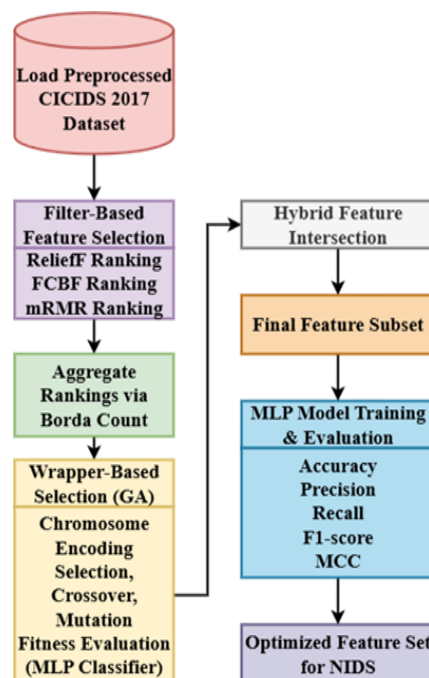


Figure 4. Flow diagram of HybFS-NIDS methodology

4. RESULTS AND DISCUSSION

4.1. Experimental setup

This study was carried out in the Java programming language and in the Apache NetBeans IDE 15, operating on Windows 11. Java (JDK 1.8) and NetBeans were advantageous selections due to their large libraries and debugging tools that helped to create and test the offered hybrid feature selection algorithm. Weka is an open-source software that was used to perform machine learning classification and evaluation. Weka's comprehensive selection of classifiers, which includes the MLP, allowed Weka to effectively train models, validate them, and evaluate their performance. Here, Weka 3.9.5 is used. This configuration helped to make sure that the HybFS-NIDS algorithm was rigorously evaluated in a stable and reproducible setting.

4.2. Comparison results

Table 3 compares the proposed algorithm of HybFS-NIDS with other sophisticated methods that were presented in the literature. They are evaluated on accuracy, precision, recall, F1-score, and MCC.

These metrics together provide a broad picture of the classification efficiency of the system and point out its key strengths.

Table 3. Performance metrics comparison of NIDS models

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	MCC (%)
Bulavas <i>et al.</i> [26]	81.00	77.00	83.00	76.00	61.06
Ramzan <i>et al.</i> [27]	73.00	72.00	79.00	68.00	49.22
Oliveira <i>et al.</i> [28]	76.22	78.68	73.75	75.79	54.13
Al Farsi <i>et al.</i> [29]	82.00	84.00	80.00	80.00	64.87
HybFS-NIDS	90.68	91.74	90.64	91.19	91.48

From Table 3, the HybFS-NIDS algorithm performs better than all the other existing models in terms of all performance results. The model is performing better in the overall prediction with a high accuracy of 90.68%. The accuracy (91.74%) and recall (90.64%) scores show that it is useful to correctly identify attacks with low false positives. It has a high F1-score (91.19%), which gives a good balance of precision and recall, which are essential to intrusion detection as both false positives and false negatives can have serious repercussions. The MCC of 91.48% suggests that the classifier is reliable and works well in dealing with intrusion datasets containing uneven numbers of attack types. The advantages arise due to a hybrid feature selection strategy, which eliminates redundant features and optimizes the subset to enhance the classifier. This minimizes noise and increases the generalizability of the model. The MLP classifier was selected because several recent intrusion detection studies, including [26]–[29], used MLP-based models for evaluation. Using the same classifier ensures a fair comparison under similar conditions. Moreover, MLP is effective in handling complex nonlinear relationships in high-dimensional cybersecurity data. The main objective of this study is to evaluate the effectiveness of the proposed hybrid feature selection method rather than compare multiple classifiers. Therefore, maintaining a consistent MLP architecture helps ensure that performance improvements are mainly due to the proposed HybFS-NIDS framework. The achieved results, including 90.68% accuracy, 91.19% F1-score, and 91.48% MCC, confirm the effectiveness of the proposed feature selection approach.

4.3. Discussions

The experimental results in Figures 5 to 9 show the effectiveness of HybFS-NIDS over the current approaches. As shown in Figure 5, the proposed model yields better accuracy. This is attributed to the integration of statistical and wrapper-based feature selection methods, which select the most significant features and improve the classification performance. Figure 6 indicates the accuracy, and HybFS-NIDS has shown a considerable improvement. This is directly due to the removal of extraneous and unnecessary characteristics, and thus minimizes the amount of false positives, and reported incursions are highly reliable. Figure 7 represents the recall comparison. The high recall of HybFS-NIDS depicts that it has a high ability to identify real intrusions with a low false negative rate. This has a significant role in network security, in which false detections can cause serious vulnerabilities.

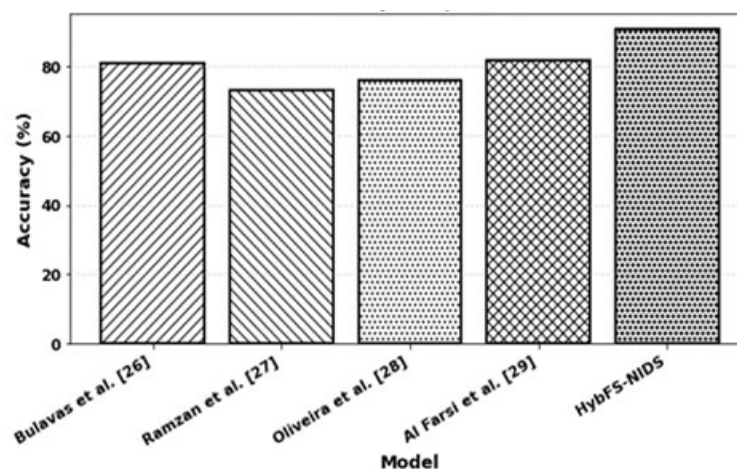


Figure 5. Accuracy comparison

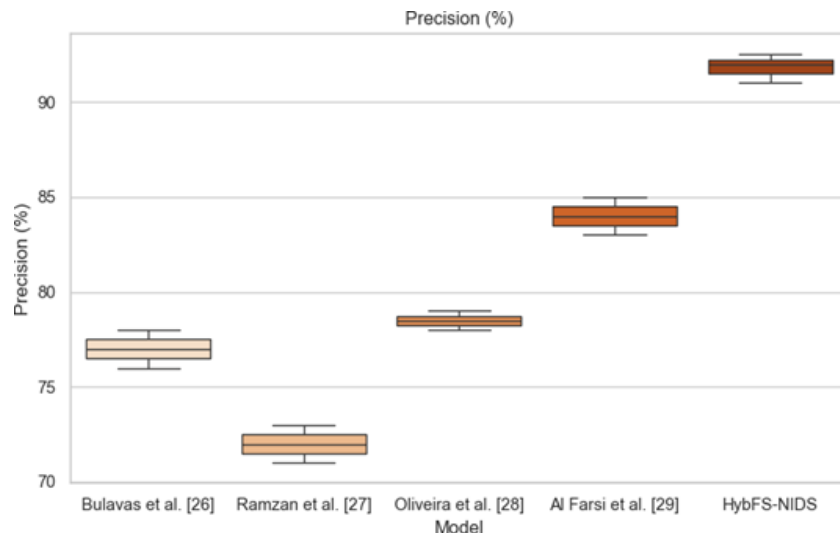


Figure 6. Precision comparison

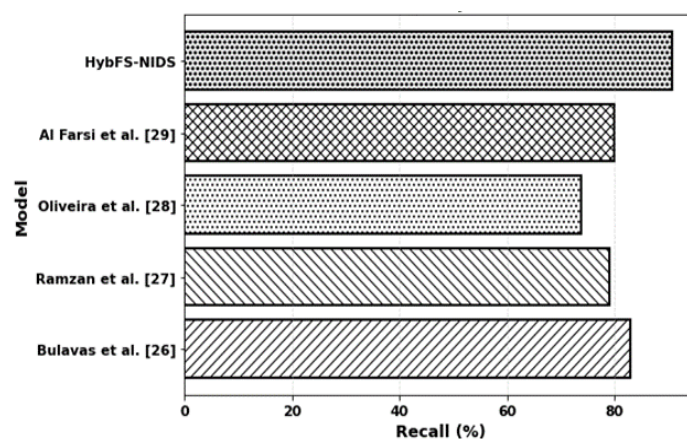


Figure 7. Recall comparison

The F1-scores are shown in Figure 8, and this result indicates that HybFS-NIDS is the leading performer, which affirms that the model maintains a fair and strong trade-off between precision and recall and enhances the overall effectiveness of the detection system. The comparison of MCC is given in Figure 9. The HybFS-NIDS received the best score. The high MCC value indicates that the classification performance is effective even when the dataset is class-imbalanced. This reflects the robustness and dependability of the algorithm.

The experimental results and analysis show that the HybFS-NIDS algorithm can be used for improving the performance of intrusion detection. It combines several filtering methods with a wrapper approach based on GA. It successfully identifies the most relevant information, improves classification accuracy, and obtains better precision, recall, F1-score, and MCC metrics than previous models. The results indicate that HybFS-NIDS can be utilized to offer a feasible solution for effective and sound network intrusion detection for real-world applications. In order to further test the effectiveness of HybFS-NIDS, it was compared with other popular techniques of feature selection like IG, SFS, and PSO. All the methods were tested in the same setup. They were trained on the same MLP model and the same CIC-IDS2017 dataset; therefore, the comparison would be fair.

In practice, IG reduces features quickly. That is helpful. But it does not always drop features that store similar information. This can lead to minor variations in performance when the model is trained again. One potential cause is that IG will examine individual features separately and not comprehensively examine the interaction of features. Instead, SFS constructs the feature set in stages. This tends to increase the quality of the chosen subset. However, the process is more time-consuming. It makes numerous combinations; hence the training is carried out repeatedly. In a real-life case, if a security team needs to update a model quickly after a new type of attack appears, this extra time may become a problem. PSO also

gave good accuracy. It is a smarter way of searching as compared to basic greedy methods. However, it seems to require careful parameter tuning. In some cases, the results vary with the way in which it is initialized. So, the processing load may increase.

HybFS-NIDS is more balanced. It minimizes features, and performance is still maintained. In most cases, the training time was also less. This implies that an integration of the techniques could be useful in the management of accuracy and cost. It might not work flawlessly in all cases; however, in general, it appears to be more stable and practical compared to using only one traditional or evolutionary method.

To make the evaluation more reliable, the HybFS-NIDS framework was tested in a 10-fold cross-validation with the CIC-IDS2017 dataset. Simply, the data was split into ten parts. Testing was done on each part, and training was done on the rest of the parts. This was done until all the parts were tested. Final results are not obtained as the outcome of a single run. Rather, the average performance of the ten consecutive runs was reported. The standard deviation was also calculated to show the level of stability of the results. A small variation indicates that the model performs consistently across different runs. Practically, this repeated testing is performed to reduce the effects of random data partitioning. A single split could possibly provide a little more or a little less result by chance. The repetition of the process makes the performance estimate look more dependable and closer to how the model might behave with unseen data.

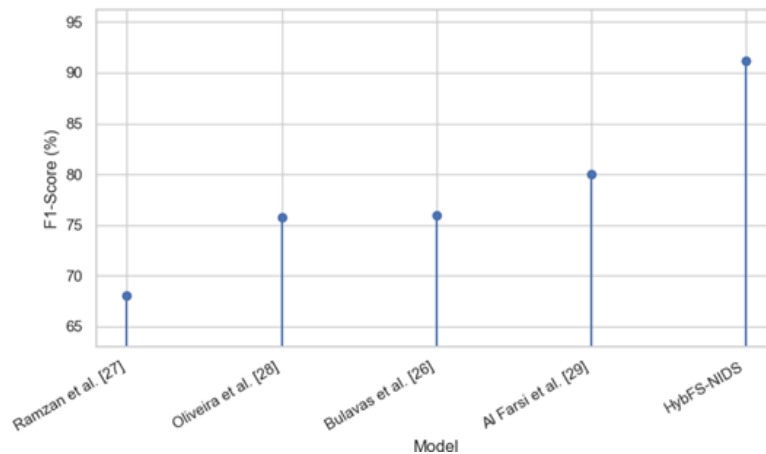


Figure 8. F1-score comparison

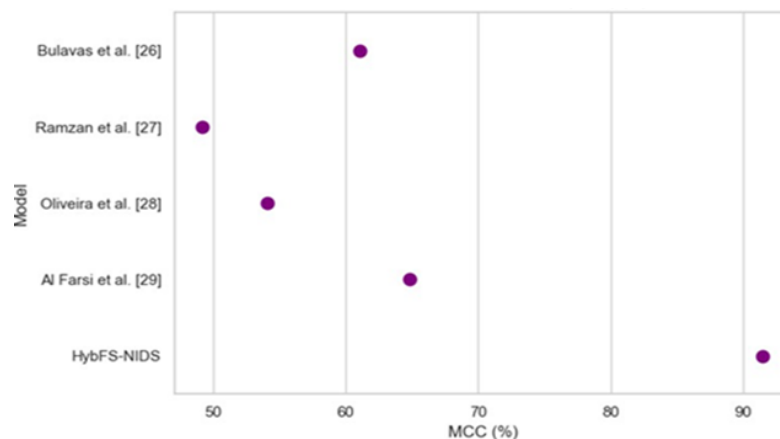


Figure 9. MCC comparison

Table 4 presents the performance of the proposed HybFS-NIDS framework using 10-fold cross-validation. This cross-validation approach is utilized to evaluate both the stability and generalization capability of the model. The obtained mean accuracy of 90.68% and MCC of 0.9148 indicate that the proposed framework maintains consistent intrusion detection performance across different data partitions with low variation.

Table 4. Performance with 10-fold cross-validation

Metric	Value
Accuracy (%)	90.68
Precision (%)	91.74
Recall (%)	90.64
F1-score (%)	91.19
MCC	0.9148

The ablation study helps to understand the role of each part of the framework. Since the elements were tested separately, it was clarified that every stage introduces some improvement. The difference between these changes will not seem to be very dramatic, yet when combined, they create a noticeable difference.

As an example, to illustrate, the stability of feature ranking can be improved by the Borda aggregation step. It is not based on a single filter but a combination of their opinions. This seems to reduce the chance of random differences in ranking. The refined list is then used by the GA wrapper to find a combination of features that gives a better view of the classifier's true performance. In other words, the model dictates the final choice.

The overall results of the full hybrid setup are the best in the case that all the parts are applied. It is more accurate and MCC, as well as uses fewer features. This acts as a pointer that the multi-stage design is effective. This may be because every step addresses a different problem; ranking stability is to be resolved first, then subset optimization, which leads to a more balanced and efficient final model.

Table 5 provides an ablation study to analyze the contribution of each component in the proposed HybFS-NIDS framework. The results show that using only filter-based feature selection achieved 86.21% accuracy with 35 selected features. Incorporating the Borda ranking strategy improved the performance to 88.73% while reducing the feature count to 30. Further integration of the GA increased the accuracy to 89.42% with 22 features. The complete HybFS-NIDS framework achieved the best performance with 90.68% accuracy, 91.19% F1-score, and MCC of 0.9148 using only 18 features. The results demonstrate that each component is helpful for the improvement of both feature optimization and classification performance.

Table 5. Ablation study of HybFS-NIDS components

Configuration	Accuracy (%)	F1-score (%)	MCC	#Features
Filters only	86.21	86.88	0.842	35
Filters + Borda	88.73	89.15	0.879	30
Filters + GA	89.42	89.90	0.895	22
Full HybFS-NIDS	90.68	91.19	0.9148	18

This study analyzes the run-time to demonstrate the efficiency of the proposed framework. The reduction of the feature set from 78 features to 18 features reduced the training time from 142.5 s to 83.4 s (41.5%). This reduction reduces computational cost, while still maintaining high intrusion detection performance, thus making HybFS-NIDS suitable for real-time and resource-constrained security applications. Table 6 shows the comparison of the training time of the full-feature MLP model and the proposed HybFS-NIDS framework.

Table 6. Training time comparison

Model	#Features	Training time (sec)	Reduction (%)
Full-feature MLP (78)	78	142.5	—
HybFS-NIDS (18)	18	83.4	41.5%

The real-world performance of HybFS-NIDS was measured in terms of inference latency, throughput, and memory usage besides the training time (Table 7). The framework only used 18 selected features with an inference latency of 3.8 ms, a throughput of 2,630 flows/s, and reduced the memory utilization from 512 MB to 298 MB (41.8% reduction). The results show that HybFS-NIDS can be efficiently deployed with strong intrusion detection performance, and therefore, it is suitable for real-time and resource-constrained cybersecurity environments.

To assess the robustness and generalization performance of HybFS-NIDS, this study performed further experiments on UNSW-NB15 and NSL-KDD datasets using the same preprocessing, feature selection, GA optimization, and MLP settings. The framework achieved an accuracy of 89.12% and 92.03% with 20 and 16 selected features, respectively. The consistent results show that HybFS-NIDS can

significantly reduce redundant features while maintaining the robust performance of intrusion detection on different cybersecurity datasets.

To improve the interpretability of the proposed HybFS-NIDS framework, explainability analysis was conducted to examine the semantic importance of the selected optimal features. Since the final hybrid selection process retained only 18 highly discriminative features, additional analysis was performed using Shapley additive explanations (SHAP) to evaluate the contribution of each feature toward intrusion classification decisions. Table 8 shows explainability analysis of important features using SHAP.

Table 7. Real-time deployment performance analysis

Model	#Features	Inference latency (ms)	Throughput (flows/sec)	Memory usage (MB)
Full-feature MLP (78)	78	6.7	1,540	512
HybFS-NIDS (18)	18	3.8	2,630	298

Table 8. Explainability analysis of important features using SHAP

Feature	Network significance	SHAP importance
Flow duration	Captures abnormal session duration patterns	High
Total forward packets	Indicates unusual packet transmission behavior	High
Flow bytes/s	Detects high-volume traffic anomalies	High
Packet length mean	Represents malicious payload variations	High
Destination port	Identifies suspicious communication targets	Medium-high
Flow IAT mean	Measures irregular packet arrival timing	Medium-high
Fwd packet length Std	Detects packet size inconsistency	Medium
Active mean	Captures abnormal active traffic periods	Medium

The explainability results indicated that traffic flow statistics, packet length characteristics, IAT attributes, and destination port-related features contributed significantly to distinguishing malicious traffic from normal network behavior. Features such as flow duration, total forward packets, flow bytes/s, packet length mean, destination port, and flow IAT mean consistently produced higher SHAP importance scores, indicating strong influence on attack prediction outcomes. In particular, abnormal packet transmission rates and irregular timing behavior were observed to contribute strongly to the detection of DDoS, brute-force, and botnet-related attacks.

Similarly, destination port details were useful for finding unusual traffic patterns that could indicate unauthorized login attempts. The explainability analysis confirms the statistical relevance and semantic meaningfulness of the picked features, boosting the trustworthiness and ease of understanding of HybFS-NIDS. For the adversarial evasion and poisoning attacks, HybFS-NIDS still achieved a high robustness with the accuracy of 87.42% and 85.97%, respectively. However, more research is needed on issues like class imbalance, scalability, and unseen attacks. These recent works have obtained promising results using hybrid learning and feature selection techniques [30]–[34]; however, most of the works mainly focus on improving the classification performance. On the other hand, the proposed HybFS-NIDS framework combines several advanced filter techniques with a wrapper based on a GA and Borda rank aggregation to produce a concise and informative feature subset, resulting in better detection performance and lower computational complexity.

5. CONCLUSION

This paper proposes the HybFS-NIDS framework to improve feature selection and intrusion detection on the CIC-IDS2017 dataset. The framework combines filter and wrapper-based feature selection methods to select 18 key features from an initial set of 78. Consequently, it achieved 90.68% accuracy, 91.74% precision, 90.64% recall, 91.19% F1-score, and an MCC of 0.9148 while reducing training time by 41.5%. The results indicate that HybFS-NIDS can be applied in real-time cybersecurity applications, even in computing resource-constrained environments. However, the framework was evaluated only on the dataset of CIC-IDS2017. In future work, this study plan to validate the framework on more recent datasets, test the framework in real-time environments, integrate explainable artificial intelligence, and devise more efficient feature reduction techniques to combat evolving cyber threats.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Josephine Rathinam	✓	✓	✓	✓	✓	✓		✓	✓		✓		✓	
Anandaraj Shanthi Pichandi		✓		✓	✓	✓				✓		✓		

C : Conceptualization

M : Methodology

So : Software

Va : Validation

Fo : Formal analysis

I : Investigation

R : Resources

D : Data Curation

O : Writing - Original Draft

E : Writing - Review & Editing

Vi : Visualization

Su : Supervision

P : Project administration

Fu : Funding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The datasets used in this study, including CIC-IDS2017, UNSW-NB15, and NSL-KDD, are publicly available from their respective dataset providers. The data supporting the findings of this study are available from the corresponding author, [JR], upon reasonable request.

REFERENCES

- [1] K. Swarnalatha, N. Narisetty, G. R. Kancherla, and B. Bobba, "Analyzing resampling techniques for addressing the class imbalance in NIDS using SVM with random forest feature selection," *International Journal of Experimental Research and Review*, vol. 43, pp. 42–55, 2024, doi: 10.52756/ijerr.2024.v43spl.004.
- [2] M. S. Habeeb and T. R. Babu, "Coarse and fine feature selection for network intrusion detection systems (IDS) in IoT networks," *Transactions on Emerging Telecommunications Technologies*, vol. 35, no. 4, 2024, doi: 10.1002/ett.4961.
- [3] A. R. Muhsen, G. G. Jumaa, N. F. A. Bakri, and A. T. Sadiq, "Feature selection strategy for network intrusion detection system (NIDS) using Meerkat Clan algorithm," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 16, pp. 158–171, 2021, doi: 10.3991/ijim.v15i16.24173.
- [4] A. Ahmed, M. Asim, I. Ullah, Zainulabidin, and A. A. Ateya, "An optimized ensemble model with advanced feature selection for network intrusion detection," *PeerJ Computer Science*, vol. 10, pp. 1–32, 2024, doi: 10.7717/peerj-cs.2472.
- [5] M. M.-eddine, A. Guezzaz, S. Benkirane, and M. Azrou, "An efficient network intrusion detection model for IoT security using K-NN classifier and feature selection," *Multimedia Tools and Applications*, vol. 82, no. 15, pp. 23615–23633, 2023, doi: 10.1007/s11042-023-14795-2.
- [6] M. Awad and S. Fraihat, "Recursive feature elimination with cross-validation with decision tree: feature selection method for machine learning-based intrusion detection systems," *Journal of Sensor and Actuator Networks*, vol. 12, no. 5, Sep. 2023, doi: 10.3390/jsan12050067.
- [7] J. Li, M. S. Othman, H. Chen, and L. M. Yusuf, "Optimizing IoT intrusion detection system: feature selection versus feature extraction in machine learning," *Journal of Big Data*, vol. 11, no. 1, 2024, doi: 10.1186/s40537-024-00892-y.
- [8] M. T. Abdelaziz *et al.*, "Enhancing network threat detection with random forest-based NIDS and permutation feature importance," *Journal of Network and Systems Management*, vol. 33, no. 1, 2025, doi: 10.1007/s10922-024-09874-0.
- [9] F. G. F. Gharehchopogh and S. B. S. Bagirzada, "Enhancing network intrusion detection with mountain gazelle optimizer-based feature selection," in *PAHTEL-Proceedings of Azerbaijan High Technical Educational Institutions*, 2024, vol. 43, no. 8–1, pp. 520–529. doi: 10.36962/pahtei4308012024-58.
- [10] A. K. Mananayaka and S. S. Chung, "Network intrusion detection with two-phased hybrid ensemble learning and automatic feature selection," *IEEE Access*, vol. 11, pp. 45154–45167, May 2023, doi: 10.1109/ACCESS.2023.3274474.
- [11] G. M. Hassan, A. Gumaei, A. Alanazi, and S. M. Alzanin, "A network intrusion detection approach using extreme gradient boosting with max-depth optimization and feature selection," *International Journal of Interactive Mobile Technologies*, vol. 17, no. 15, pp. 120–134, 2023, doi: 10.3991/ijim.v17i15.37969.
- [12] M. Yuvaraja, S. Arunkumar, P. V. Kumar, and L. M. I. Sheela, "Improved grey wolf optimization- (IGWO-) based feature selection on multiview features and enhanced multimodal-sequential network intrusion detection approach," *Wireless Communications and Mobile Computing*, vol. 2023, 2023, doi: 10.1155/2023/8478457.
- [13] Y. Wang, Z. Liu, W. Zheng, J. Wang, H. Shi, and M. Gu, "A combined multi-classification network intrusion detection system based on feature selection and neural network improvement," *Applied Sciences*, vol. 13, no. 14, 2023, doi: 10.3390/app13148307.
- [14] R. F. Moyano *et al.*, "A feature selection approach towards the standardization of network security datasets," in *2023 IEEE 9th International Conference on Network Softwarization: Boosting Future Networks through Advanced Softwarization, NetSoft 2023*, 2023, pp. 257–261, doi: 10.1109/NetSoft57336.2023.10175497.
- [15] S. Krishnaveni, S. Sivamohan, S. Sridhar, and S. Prabhakaran, "Network intrusion detection based on ensemble classification and feature selection method for cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 34, no. 11, 2022, doi: 10.1002/cpe.6838.
- [16] D. R. Patil and T. M. Pattewar, "Majority voting and feature selection based network intrusion detection system," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 22, no. 6, 2022, doi: 10.4108/eai.4-4-2022.173780.

- [17] A. Alabrah, "A novel study: GAN-based minority class balancing and machine-learning-based network intruder detection using chi-square feature selection," *Applied Sciences*, vol. 12, no. 22, 2022, doi: 10.3390/app122211662.
- [18] H. Al-Zoubi and S. Altaamneh, "A feature selection technique for network intrusion detection based on the chaotic crow search algorithm," in *2022 International Conference on Intelligent Data Science Technologies and Applications, IDSTA 2022*, 2022, pp. 54–60, doi: 10.1109/IDSTA55301.2022.9923108.
- [19] O. Almomani, "A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms," *Symmetry*, vol. 12, no. 6, pp. 1–20, 2020, doi: 10.3390/sym12061046.
- [20] P. J. Prakash and B. Lalitha, "A novel intrusion detection system for RPL based iot networks with bio-inspired feature selection and ensemble classifier," *i-manager's Journal on Software Engineering*, vol. 13, no. 2, pp. 1–19, 2021, doi: 10.21203/rs.3.rs-442429/v1.
- [21] A. Abbas, M. A. Khan, S. Latif, M. Ajaz, A. A. Shah, and J. Ahmad, "A new ensemble-based intrusion detection system for internet of things," *Arabian Journal for Science and Engineering*, vol. 47, pp. 1805–1819, Aug. 2021, doi: 10.1007/s13369-021-06086-5.
- [22] Kurniabudi, D. Stiawan, Darmawijoyo, M. Y. B. Idris, A. M. Bamhdi, and R. Budiarto, "CICIDS-2017 dataset feature analysis with information gain for anomaly detection," *IEEE Access*, vol. 8, pp. 132911–132921, 2020, doi: 10.1109/ACCESS.2020.3009843.
- [23] G. Tripathi, V. K. Singh, V. Sharma, and M. V. Vinodbhair, "Weighted feature selection for machine learning based accurate intrusion detection in communication networks," *IEEE Access*, vol. 12, pp. 20973–20982, 2024, doi: 10.1109/ACCESS.2024.3362794.
- [24] S. G. Pran and S. Raja, "An efficient feature selection and classification approach for an intrusion detection system using optimal neural network," *Journal of Intelligent and Fuzzy Systems*, vol. 44, no. 5, pp. 8561–8571, 2023, doi: 10.3233/JIFS-222120.
- [25] K. Kurniabudi, D. Stiawan, Darmawijoyo, M. Y. B. Idris, B. Kerim, and R. Budiarto, "Important features of CICIDS-2017 dataset for anomaly detection in high dimension and imbalanced class dataset," *Indonesian Journal of Electrical Engineering and Informatics*, vol. 9, no. 2, pp. 498–511, 2021, doi: 10.52549/ijeei.v9i2.3028.
- [26] V. Bulavas, V. Marcinkevičius, and J. Rumiński, "Study of multi-class classification algorithms' performance on highly imbalanced network intrusion datasets," *Informatica*, vol. 32, no. 3, pp. 441–475, 2021, doi: 10.15388/21-INFOR457.
- [27] M. Ramzan *et al.*, "Distributed denial of service attack detection in network traffic using deep learning algorithm," *Sensors*, vol. 23, no. 20, Oct. 2023, doi: 10.3390/s23208642.
- [28] N. Oliveira, I. Praça, E. Maia, and O. Sousa, "Intelligent cyber attack detection and classification for network-based intrusion detection systems," *Applied Sciences*, vol. 11, no. 4, pp. 1–21, 2021, doi: 10.3390/app11041674.
- [29] A. Al Farsi, A. Khan, M. M. B.-Suwailam, and M. R. Mughal, "Comparative performance evaluation of machine learning algorithms for cyber intrusion detection," *Preprints*, 2024, doi: 10.20944/preprints202412.0497.v1.
- [30] M. G. Yaseen, A. H. Ali, and M. Alajani, "An enhanced hybrid genetic-JAYA algorithm for feature selection and SVM parameter optimization in intrusion detection systems: evaluation on the CICIDS dataset," *Shifra*, vol. 2025, pp. 98–109, 2025, doi: 10.70470/shifra/2025/006.
- [31] P. Mamatha, S. Balaji, and S. S. Anuraghav, "Development of hybrid intrusion detection system leveraging ensemble stacked feature selectors and learning classifiers to mitigate the DoS attacks," *International Journal of Computational Intelligence Systems*, vol. 18, no. 1, 2025, doi: 10.1007/s44196-025-00750-6.
- [32] D. R. Patil, T. M. Pattewar, T. S. Shinde, K. S. Kumavat, and S. N. Deshpande, "Optimizing network intrusion detection systems through ensemble learning and feature selection using the CIC-IDS2017 dataset," *Informatica*, vol. 49, no. 4, 2025, doi: 10.31449/inf.v49i4.7678.
- [33] K. Okokpuije, K. D. O. Maha, M. O. Ogundele, O. Oguntuyo, A. P. Ijeh, and I. P. Okokpuije, "Development of a network intrusion detection system using hybridised recurrent neural networks and random forest models," *NIPES - Journal of Science and Technology Research*, vol. 7, no. 1, pp. 1669–1675, 2025, doi: 10.37933/nipes/7.4.2025.SI195.
- [34] R. Magar, P. Dhole, and B. Gawali, "A hybrid random forest and LSTM-based intrusion detection system using CICIDS2017 network traffic data," *International Journal of Computer Science Engineering and Technology*, vol. 10, no. 2, pp. 304–312.

BIOGRAPHIES OF AUTHORS



Josephine Rathinam    is an assistant professor at Presidency University and a Ph.D. scholar specializing in cyber security. She is an academician, researcher, author, mentor, and HARVEST Club Coordinator, passionate about teaching, emerging technologies, and student innovation. Her research and publications include an IEEE-Scopus paper on deep learning and book chapters on IoT-based plant disease identification and machine learning-based voice assistants. Through mentoring and leadership, she inspires students and contributes to advancements in AI, robotics, and cybersecurity. She can be contacted at email: josephine.r@presidencyuniversity.in.



Dr. Anandaraj Shanthi Pichandi    is an accomplished academician with 19 years of teaching and research experience. As professor and head of the department in Presidency University, Bangalore, he has played a key role in academic leadership, research promotion, and student development. His expertise spans across network security, big data, and blockchain technology, where he actively contributes through teaching, research, and mentorship. He is dedicated to creating a research-driven learning environment and has guided numerous projects that address contemporary challenges in computer science and engineering. His commitment to innovation and knowledge sharing continues to inspire both colleagues and students. He can be contacted at email: anandaraj@presidencyuniversity.in.