Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.040567
Article

Fusion of Feature Ranking Methods for an Effective Intrusion Detection System

Seshu Bhavani Mallampati[1] and Seetha Hari[2,*]

[1]School of Computer Science and Engineering, VIT-AP University, Andhra Pradesh, India
[2]Center of Excellence, AI, and Robotics, VIT-AP University, Andhra Pradesh, India
*Corresponding Author: Seetha Hari. Email: seetha.hari@vitap.ac.in
Received: 23 March 2023; Accepted: 19 May 2023; Published: 30 August 2023

Abstract: Expanding internet-connected services has increased cyberattacks, many of which have grave and disastrous repercussions. An Intrusion Detection System (IDS) plays an essential role in network security since it helps to protect the network from vulnerabilities and attacks. Although extensive research was reported in IDS, detecting novel intrusions with optimal features and reducing false alarm rates are still challenging. Therefore, we developed a novel fusion-based feature importance method to reduce the high dimensional feature space, which helps to identify attacks accurately with less false alarm rate. Initially, to improve training data quality, various preprocessing techniques are utilized. The Adaptive Synthetic oversampling technique generates synthetic samples for minority classes. In the proposed fusion-based feature importance, we use different approaches from the filter, wrapper, and embedded methods like mutual information, random forest importance, permutation importance, Shapley Additive exPlanations (SHAP)-based feature importance, and statistical feature importance methods like the difference of mean and median and standard deviation to rank each feature according to its rank. Then by simple plurality voting, the most optimal features are retrieved. Then the optimal features are fed to various models like Extra Tree (ET), Logistic Regression (LR), Support vector Machine (SVM), Decision Tree (DT), and Extreme Gradient Boosting Machine (XGBM). Then the hyperparameters of classification models are tuned with Halving Random Search cross-validation to enhance the performance. The experiments were carried out on the original imbalanced data and balanced data. The outcomes demonstrate that the balanced data scenario knocked out the imbalanced data. Finally, the experimental analysis proved that our proposed fusion-based feature importance performed well with XGBM giving an accuracy of 99.86%, 99.68%, and 92.4%, with 9, 7 and 8 features by training time of 1.5, 4.5 and 5.5 s on Network Security Laboratory-Knowledge Discovery in Databases (NSL-KDD), Canadian Institute for Cybersecurity (CIC-IDS 2017), and UNSW-NB15, datasets respectively. In addition, the suggested technique has been examined and contrasted with the state of art methods on three datasets.

Keywords: Cyber security; feature ranking; imbalance; preprocessing; IDS; SHAP

1 Introduction

The rapid advancement of network-based technologies and their applications has resulted in many risks and illegal activities. Cyber scamming, crypto trojans, and phishing are examples of frequent yet dangerous cyber assaults that deliberately seek out and exploit the user's sensitive data [1]. Innovation in security mechanisms is required to address these issues. According to an IBM security report, the overall average cost of a data breach worldwide climbed to $4.35 million in 2022, as shown in Fig. 1 [2]. By observing Fig. 1, it is evident that the average cost of the data breach increased by 12.7% from 2020 to 2022, from $3.86 million to $4.35 million.

Figure 1: The total average cost of a data breach globally [2]

Furthermore, according to checkpoint research [3], education and research are still the most targeted sectors. These organizations experience an average of 2,297 attacks per week, a rise of 44% from 2021. Furthermore, healthcare remains one of the most targeted sectors globally, with a 69% rise from 2021.

Even though security procedures are now more widely recognized, no network can be completely secure with current technologies. Various security mechanisms, including firewalls, data encryption, and user authentication, are employed to stop cyberattacks, yet the frequency of attacks is rising daily rather than decreasing [4]. In this regard, intrusion detection systems are one of the solutions regularly used to monitor the network, identify potential threats, and find security flaws [5]. The two primary classes of attack identification are signature-based and anomaly-based.

Signature-Based: The attack signatures that distinguish legitimate traffic from malicious traffic are identified using signature-based detection techniques, which depend on known attack patterns for identification. Popular signature-based detection methods are Spectral analysis, SNORT, and Bro network analysis framework [6].

Anomaly-Based: This approach is based on characterizing network behaviour. If network behaviour follows the established behaviour, it is either acknowledged or triggered by an anomaly detection event. The anomaly-based IDSs thought to be adaptable, although they have a significant risk of generating false positives [7].

Network traffic has increased significantly in quantity, features, and frequency. Thus, it is challenging to classify network traffic. Various IDS datasets are generated by collecting unprocessed network traffic to analyze the network traffic [8]. Several networking tools, including Wireshark and Nmap, are utilized [9] and stored as Tcpdump or PCAP files to record raw network data. As a result, the IDS datasets used to evaluate performance include high-dimensional network feature space [10].

Bellman observed that the "curse of dimensionality" is a slew of issues caused when processing high-dimensional data [11].

In recent years, the dimensionality of datasets utilized in machine learning (ML) applications has grown significantly [12]. Because of the enormous search space, it is challenging to retrieve pertinent information about a particular area of interest [13]. Therefore, feature selection is essential in dealing with massive datasets by discarding irrelevant and duplicated data. The added benefits of feature selection are conserving storage space, improving computation time, boosting the classification models' predictive accuracy, and making them easier to understand [14].

Feature selection (FeS) approaches shrink the initial feature space without transforming it, preserving the original attributes and allowing for coherent interpretation. Other advantages of FeS include producing models with fewer attributes that are simpler to interpret, easier to visualize, and require less memory [15]. Feature selection methods are of three categories such as filter, wrapper, and embedded forms. Filter techniques utilize statistical data-dependent methods to choose the best feature subset for classification. These approaches are computationally quick and independent of the classifier type, but they overlook the significance of various dimensions when selecting the best feature set [16].

On the other hand, wrapper-based feature selection techniques use the classification model to identify optimal feature subsets. But these wrapper methods have limitations, including a high computational cost and the possibility of overfitting. Finally, embedded-based approaches handle feature selection and classification simultaneously, and they do so as a part of the training process. Based on the significance of the extracted characteristics, it chooses the best features [17].

An IDS can detect anomalies depending on how many features it has. Data mining and ML approaches aim to improve detection accuracy and decrease the false positive rate for IDS. The current algorithms failed to identify the network breach despite employing all the attributes. Therefore, we proposed a fusion of feature selection methods for determining the most significant features contributing to a model's predictive accuracy. This strategy combines the results of multiple feature selection methods to generate more robust and precise features. The scientific basis of the proposed method is that a fundamental hypothesis says that merging the outcomes of various feature selection methods can produce more trustworthy features. The fusion of feature selection methods mitigates the risk of overfitting, removes irrelevant features and enhances the model performance.

Therefore, in this work, we propose a fusion of feature ranking methods based on the feature importance, such as mutual information importance (MI), permutation importance (PI), random forest importance (RFI), SHAP feature importance (SFI), and statistical methods like the difference between mean and median (DMM) and standard deviation (SD).

The critical contributions of this proposed work are as follows:

- This research aims to provide a fast and efficient Intrusion detection mechanism.
- Handled imbalanced data by generating synthetic samples for better classification performance.
- Proposed a fusion of feature ranking techniques to select the optimal subset of features.
- The detection performance of the suggested technique was compared with the existing state-of-the-art methods.

The structure of the article is as follows. Section 2 reflects the literature survey. Section 3 states the proposed method. Section 4 shows the experimental results and analysis. Section 5 shows the summary of the proposed work.

2 Related Works

Several IDS and classification strategies have been used in recent decades to produce quicker and more accurate results.

Osanaiye et al. [18] proposed a filter-based ensemble feature selection method for detecting cloud Distributed Denial of Service attacks. They used filter-based methods like chi-squared (Chi^2), gain ratio (GR), information gain (IG), and Relief techniques to identify essential features. Then 13 attributes are selected from four feature selection methods. Finally, the optimal features are trained using a decision tree (DT) classifier and detected attacks accurately. But they have not addressed the class imbalance in the NSL-KDD dataset. Bansal et al. [19] proposed an IDS based on XGBM for detecting Denial-of-Service attacks in the network. They further tweaked the XGBM parameters to optimize performance by employing a sparse matrix and flags on every potential value. They have conducted multiple experiments on the CIC-IDS 2017 dataset.

Kannari et al. [20] proposed an IDS to reduce the detection model computation time and resource usage. Initially, they used recursive feature elimination to remove the irrelevant features, and they selected 21 most essential attributes out of 42 of the NSL-KDD Dataset. The optimal features are passed to RF to detect attacks in the network effectively. Najar [21] proposed an IDS to detect attacks on the NSL-KDD Dataset. They used Random Forest (RF) and multilayer perceptron (MLP) to classify the attack. Initially, they used principal component analysis and extracted ten optimal features. Further, the optimal features are passed to RF to identify binary attacks. Kasongo et al. [22] proposed a wireless IDS for providing security to various communication infrastructures by applying a wrapper-based feature extraction unit with a base classifier as an ET. Then the optimal feature vector is trained using the feed-forward deep neural network. They tested the model on UNSW-NB15 and the AWID datasets. Their experimental results proved that their model performs better.

Saha et al. [23] proposed an ensemble feature selection technique to train various machine learning, deep learning, and unsupervised learning methods by using them. They have conducted multiple experiments on UNSW-NB 15, and their results proved that neural networks (NN), long-short-term memory networks (LSTM), and Gated Recurrent Units (GRU) outperformed other methods. Mhawi et al. [24] proposed a hybrid feature selection method by combining correlation feature selection with RF. The optimal features fed to K-Nearest Neighbor (KNN), SVM, RF, and Naive Bayes (NB). These four improved classifiers have been used as AdaBoosting and bagging by employing the average voting method. They tested the model with two feature subsets containing 13 and 30 features on CIC-IDS 2017 data set. Their results showed that with 30 optimal features, the model detects attacks accurately.

Ali et al. [25] proposed a soft voting mechanism using an AutoML concept to identify network intrusions. Initially, they used various sampling methods to handle class imbalance. Gradient Boost, RF, Extra Tree, and MLP were employed to create a soft-voting model for classification. They tested the model on UNSW-NB 15 and CIC-IDS 2017 datasets. Henry et al. [26] suggested an IDS based on Convolution neural networks and Gated Recurrent Unit (CNN-GRU). Pearson's Correlation is applied to remove the correlated features. Then the optimal features are trained by using (CNN-GRU). They have conducted multiple experiments on CIC-IDS 2017 dataset. Table 1 shows the summary of the literature survey.

Table 1: Summary of related works on three datasets

Model	Dataset	Acy	Pe	RC	F1-Mes	Tr. Time (s)	Te. Time (s)	Limitations
[18]	NSL-KDD	99.6	NA	NA	NA	0.78	NA	They have not addressed the class imbalance in the dataset.
[19]	NSL-KDD	99.5	NA	NA	NA	NA	NA	They have not mentioned training time.
[20]	NSL-KDD	99.8	99.9	99.69	99.78	20.948	NA	RF with more trees may cause the algorithm to run slowly. Therefore, using various encoding techniques, such as entity embedding and the one-hot hash trick, without reducing the number of estimators can enhance the processing speed.
[21]	NSL-KDD	99.1	99.9	98.2	98.78	NA	NA	They have not addressed parameter tuning.
[22]	UNSW-NB15	90.8	80.3	98.3	88.45	NA	NA	They have not addressed the class imbalance in the dataset.
[23]	UNSW-NB15	87.2	87.2	87.1	87.9	78.32	NA	The performance of their model can be improved by using dynamic tuning instead of grid search cross-validation.
[23]	UNSW-NB15	86.8	82.4	93.7	87.7	474.75	NA	
[23]	UNSW-NB15	86.5	81.7	94.1	87.4	427.58	NA	
[24]	CIC-IDS2017	99	NA	NA	NA	NA	NA	They have selected 30 features. They can remove more attributes to improve performance.
[25]	CIC-IDS2017	98.4	97.5	99.4	98.4	NA	NA	Identifying the optimal set of classifiers to make an ensemble model is computationally costlier.
[26]	CIC-IDS2017	98.7	NA	NA	NA	1128.5	NA	1) They have not addressed the class imbalance. 2) Model performance can be improved by using optimization.

Note: *NA-not available.

Selecting relevant features is a challenging task in IDS since no single feature selection algorithm gives optimal features that would show predictive performance and be robust changes to the input data. Various ensemble feature approaches have been explored in the literature, but the novel fusion of feature selection methods could still enhance the model's performance. Further, most studies did not consider the classification's training (Tr.) and testing (Te.) time. Therefore, to overcome these issues in the proposed work, we used various feature ranking methods and determined the most frequently contributed features, which enhanced the prediction accuracy.

3 Proposed Method

This section includes a detailed discussion of the suggested methodology. Fig. 2 provides the framework of the proposed model.

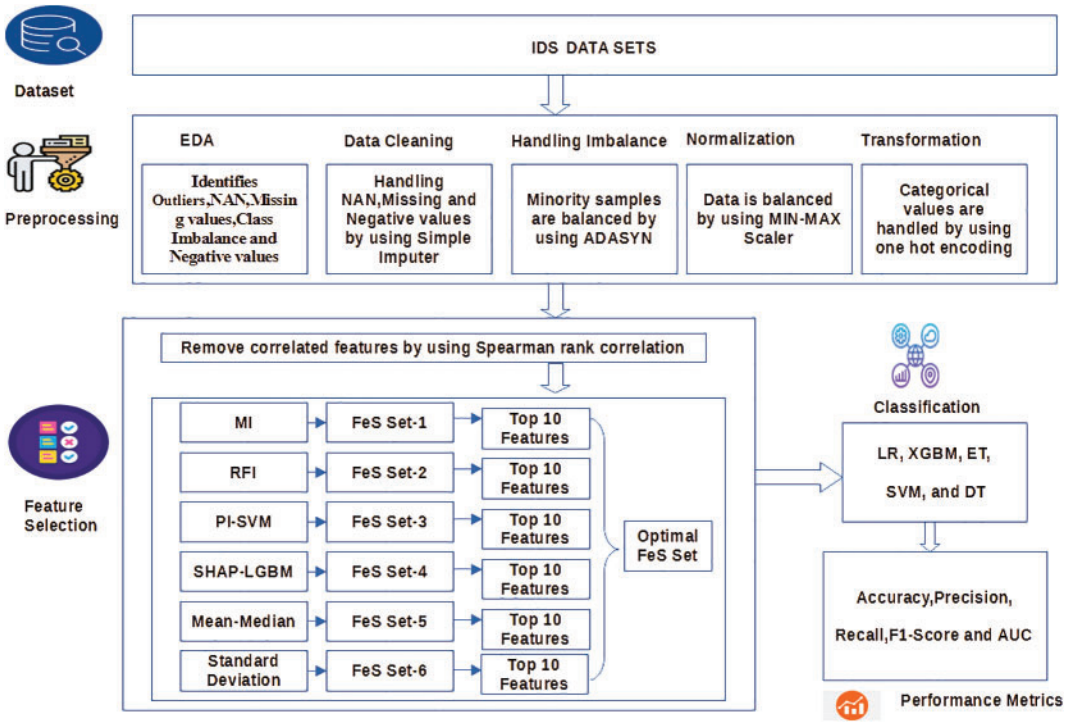


Figure 2: The proposed IDS framework

3.1 Dataset Acquisition

NSL-KDD, UNSW NB-15, and CICIDS-2017 are three intrusion detection datasets used to evaluate the effectiveness of the proposed method. These datasets comprised various network characteristics generated by multiple network configurations. Additionally, these datasets include both synthetic and authentic network traffic. As a result, the effectiveness of the suggested approach can be unambiguously supported by employing diverse network traffic from three independent datasets. The following is a quick explanation of each dataset.

3.1.1 NSL-KDD

Tavallaee et al. [27] suggested NSLKDD as a replacement for KDD Cup 99. NSL-KDD has developed from the KDD CUP 99 dataset by eliminating missing and redundant samples to reduce classifier bias. It contains 42 features, including class labels, divided into four main categories: time-based network traffic statistical features, Transmission Control Protocol Connection features, host-based operating features, and host-based network traffic statistical features. The NSL-KDD Dataset includes separate training and test datasets with 125,973 and 22,544 data samples, respectively. Further, it contains four different types of attacks Denial of Service (DoS), Probe, Remote to Local (R2L), and User to Root (U2R) attacks. This work considers a training data set for experimental analysis.

3.1.2 UNSW-NB15

Moustafa [28] generated this dataset by setting up the synthetic infrastructure at the Australian Centre for Cyber Security using the IXIA tool. 'Tcpdump' was used to record 100 GB of unprocessed network traffic. Twelve models are used to extract the features using the Argus and Bro-IDS tools. The dataset comprises 2.5 million records, covering nine attack classes and one normal class: backdoor, analysis, DoS, Fuzzers, generic, reconnaissance, exploits, worms, and shellcode. It has 49 features divided into six categories: time, flow, content, basic, labelled, and additional generated features. For experimental analysis, we have considered UNSW-NB 15 training and testing datasets which contain 175341 and 82332 records.

3.1.3 CIC-IDS 2017

Sharafaldin et al. [8] generated CIC-IDS 2017 IDS dataset by producing and collecting network traffic. The dataset includes regular traffic and traffic generated by fourteen attacks collected in five days. They used the B-profile technique to produce benign human web activity and generate standard Hypertext Transfer Protocol Secure (HTTPS), Hypertext Transfer Protocol (HTTP), Secure Shell (SSH), and File Transfer Protocol (FTP) traffic. The entire CICIDS-2017 dataset comprises eight CSV files comprising 22,73,097 normal and 5,57,646 attack samples. It contains 80 features collected with the CICFlowMeter. Further, it has seven attack categories: DoS, Distributed Denial-of-Service (DDoS), Patator, Web attacks, Infiltration, Bot, and Portscan attacks. As the original dataset was more, this work considers a subset for experimental evaluation.

3.2 Preprocessing

It is a crucial phase in any ML model that aids in enhancing data quality and extracts insightful knowledge from the data. Preprocessing entails cleaning and organizing raw data to make it suitable for building and training ML models. It consists of the following stages, which are detailed more below.

3.2.1 Exploratory-Data-Analysis (EDA)

It interprets datasets by summarizing their essential properties and frequently visualizing them. This process is crucial, especially when modelling the data using machine learning. We identified the data type by performing EDA on NSL-KDD, UNSW-NB15, and CIC-IDS2017. They contain duplicate records and class imbalance. Further, the CIC-IDS 2017 dataset has Not a Number (NaN), missing and negative values.

3.2.2 Data Cleaning/Handling Noisy Data

Most machine learning algorithms demand samples with no missing data. Because model accuracy is affected when data contains missing values or noise. The suggested work eliminates duplicates to reduce computation overheads. Then missing or NaN, inf, and negative values are handled using mean imputation, where mean imputation minimizes the variance of the imputed values.

3.2.3 Transformation

The datasets include categorical features. Non-numerical data is converted into numerical ones using the Label Encoding approach because machine learning algorithms can interpret numerical values. It will change every distinct non-numerical value of an attribute to an integer, starting from 0 to n−1.

3.2.4 Normalization

Data normalization entails scaling the value of each feature into a well-proportioned range to remove the bias in favour of characteristics with higher values from the dataset. In this study, we employed a Min-Max scaler, which shortens the attributes to a range while preserving the actual distribution. The values are tweaked to have the highest value be one and the lowest value is zero. The mathematical notation form is

$$S_{normalized} = \frac{S - Minimum(S)}{Maximum(S) - Minimum(S)} \tag{1}$$

where S is the sample.

3.2.5 Handling Imbalance

When performing EDA, we observed that the three datasets contain a class imbalance (**CIb**). The classifier's performance drops when there are proportionally more negative samples than positive ones. Various methods are used in literature for balancing minority classes, which improves the misclassification penalty. In this work, we used Adaptive synthetic sampling (ADASYN) [29], an oversampling method, to balance the minority class samples.

Algorithm 1: Handling Imbalance

Input: Training dataset $S = \{s_i, y_i\}$ with n instances, and d dimension data; where $i = \{1, 2, \ldots n\}$; s_i is the sample in the subspace X and target variable y; where $y \varepsilon \{0, 1\}$

- α specifies the required balance level; $\alpha \varepsilon \{0, 1\}$ if $\alpha = 1$ the dataset is entirely balanced.
- n_{maj} is the number of majority instances and $n_{\min}$ the number of minority samples, respectively.
- $n_{\min} \leq n_{maj}$ and $n_{maj} + n_{\min} = n$.

Methodology:

1. Let the number of synthetic samples to be created for the minority class is 'C.'
   $C = (n_{maj} - n_{\min}) * \alpha$
2. For each minority class sample s_i, identify its K nearest neighbors.
3. 1) Compute τ_i where τ_i is the number of majority instances from the K neighbors of s_i
   2) Calculate the ratio of the majority instance by $\gamma_i = \frac{\tau_i}{K}; i = \{1, 2 \ldots .n_l\}, \gamma_i \in \{0, 1\}$

(Continued)

Algorithm 1 (continued)

3) Further, normalize γ_i by using $\hat{\gamma} = \frac{\gamma_i}{\sum_{i=1}^{m_l} \gamma_i}$

4) Determine the number of synthetic data instances to be generated for each minority sample

$$c_i = \hat{\gamma}_i * C$$

4. For each minority sample s_i generate c_i synthetic samples as follows
   Do steps 1 to 3
   a) Select a random minority instance s_{km} from K neighbors of s_i
   b) Synthetic sample $s_{new} = s_i + (s_{km} - s_i) * \eta$ where η is a random number between zero and one.

End

Output: Balanced dataset.

3.3 Proposed Feature Selection

Previous studies showed that single feature selection techniques could have distinct biases, whereas fusion of feature ranking has the advantage of mitigating and compensating for these biases. We used Spearman rank correlation to remove the correlated features in the proposed work. Further, we employed feature ranking methods like mutual information, permutation, random forest, SHAP feature importance, and statistical techniques like the difference between mean, median, and standard deviation. The hyper paraments of feature ranking methods are tuned using Halving Random Search CV. Table 2 shows the optimal parameters. The top contributed features of each feature importance method are depicted in Tables 3–5 for the three datasets. The attributes common in the four ranking techniques are selected as optimal features and shown in Table 6. The proposed feature selection process is in Algorithm 2.

Table 2: Parameters of feature ranking methods

Dataset	Model	Parameters
UNSW-NB15	SHAP-LGBM	learning_rate = 1, max_depth = 30, min_child samples = 14
	PI-SVM	gamma = 1, C = 1.0, kernel = 'rbf'
	RF	n_estimators = 100, criterion = 'gini', max_depth = 9
NSL-KDD	SHAP-LGBM	learning_rate = 1.2, max_depth = 47, min_child samples = 16
	PI-SVM	C = 0.1, gamma = 1.0, kernel = 'rbf'
	RF	n_estimators = 100, criterion = 'gini', max_depth = 9
CIC-IDS 2017	SHAP-LGBM	learning_rate = 1, max_depth = 45, min_child samples = 21
	PI-SVM	C = 1.0, kernel = 'rbf', gamma = 'auto'
	RF	n_estimators = 100, criterion = 'gini', max_depth = 9

Table 3: Most contributed features of various FeS methods on the NSL-KDD dataset

Feature name	SD	DMM	SFI	RFI	MI	PI (SVM)	Frequency
dst_host_srv_count	✓	✓	✓	✓	✓	✓	6
dst_host_same_src_port_rate	✓	✓	✓	✓	✓	✓	6
src_bytes	✓	✓	×	✓	✓	✓	5
same_srv_rate	✓	×	✓	✓	✓	✓	5
dst_host_count	✓	×	✓	✓	✓	✓	5
Flag	✓	×	✓	✓	✓	✓	5
Service	✓	×	✓	✓	✓	✓	5
Count	✓	✓	✓	✓	✓	×	5
Hot	×	×	✓	✓	✓	✓	4
rerror_rate	✓	✓	×	×	×	✓	3

Table 4: Most contributed features of various FeS methods on the UNSW-NB15 dataset

Feature name	DMM	SD	RF	SFI	MI	PI (SVM)	Frequency
ct_srv_src	✓	✓	✓	✓	✓	✓	6
Smean	✓	✓	✓	✓	✓	✓	6
Proto	✓	✓	✓	✓	✓	✓	6
Sttl	×	✓	✓	✓	✓	✓	5
Service	✓	✓	✓	✓	×	✓	5
ct_src_ltm	✓	✓	×	✓	×	✓	4
ct_dst_src_ltm	✓	✓	✓	×	✓	×	4
ct_src_dport_ltm	✓	×	✓	✓	✓	×	4
Dur	✓	×	✓	×	✓	×	3
State	×	×	✓	×	✓	✓	3

Table 5: Most contributed features of various FeS methods on the CIC-IDS2107 dataset

Feature name	DMM	SD	SFI	RF	PI (SVM)	MI	Frequency
Flow duration	✓	✓	✓	✓	✓	✓	6
Destination port	✓	✓	✓	✓	✓	✓	6
Init_Win_bytes_backward	✓	✓	×	✓	✓	✓	5
Bwd packet length min	✓	×	✓	×	✓	✓	4
Total Fwd packets	×	×	✓	✓	✓	✓	4
Total backward packets	×	×	✓	✓	✓	✓	4
Total length of Fwd packets	×	×	✓	✓	✓	✓	4
Fwd Packet Length Min	×	×	×	✓	✓	✓	3
Idle Std	✓	✓	×	×	×	×	2
FIN Flag Count	✓	✓	×	×	×	×	2

Table 6: Optimal features from the three datasets

S. No. ↓/Dataset →	UNSW-NB15	NSL-KDD	CIC-IDS2017
1	ct_srv_src	dst_host_srv_count	Flow duration
2	Smean	dst_host_same_src_port_rate	Destination port
3	Proto	Count	Init_Win_bytes_backward
4	Sttl	src_bytes	Bwd packet length min
5	ct_src_ltm	same_srv_rate	Total Fwd packets
6	Service	dst_host_count	Total backward packets
7	ct_dst_src_ltm	Flag	Total length of Fwd packets
8	ct_src_dport_ltm	Service	–
9	–	Hot	–

Algorithm 2: Proposed feature selection process

Input: Training data set S with an original feature set $S^1 = \{s_1, s_2, s_3, .., s_p\}$ and class label $y \in \{0, 1\}$.

Process:

1. Removal of constant features
   If (values of features (s_m) in all rows==constant;
   Eliminate s_m, then the new feature set
   $S^2 = S^1 - s_m$; *where* $S^2 = \{s_1, s_2, ...s_n\}$; $(n \leq p)$
   Repeat the process until all features with constant values removed
2. Eliminate all duplicate attributes
   if $(s_c == s_d)$; *where* $(c, d) \in \{1, 2, 3, \ldots n\}$; $(c! = d)$
   eliminate s_c from S^2; $S^3 = S^2 - s_c$; *where* $S^3 = \{s_1, s_2, s_3, ..., s_d, ...s_k\}$; $(k \leq n)$
   Repeat the process until all duplicates are removed.
3. Removal of correlated features
   if $(\zeta(s_d, s_f) \geq 0.8) d, f \in \{1, 2, 3 \ldots k\}$; $d \neq f$
   Then remove s_f the new feature set $S^4 = S^3 - s_f$; *where* $S^4 = \{s_1, s_2...s_i\}$; $i \leq k$
   here ζ is the Spearman's rank correlation coefficient
   $$\zeta = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$ *where* d_i is the difference of two feature ranks ,
   andn = number of obervations
   Repeat the process until all the correlated features are removed.
4. The attributes obtained from step 3 are fed to each feature ranking technique
5. For each feature ranking technique, calculate feature rank on S^4
6. Sort the features based on feature rank
7. *if* (*feature rank*$(s_a) \leq 0$) discard s_a *update feature set* $S^5 = S^4 - s_a$; *where* $S^5 = \{s_1, s_2...s_g\}$; $g \leq i$
8. Determine the occurrence frequency (ψ) of each feature among all the feature ranking methods

(Continued)

Algorithm 2 (continued)

9. *if* $(\psi(s_d) \geq \eta)$, then add it to the final optimal feature set $S^6 = \{S^5 \cup s_d\}$, where η is the threshold of occurrence frequency.

Output: The optimal feature set is $S^6 = \{s_1, s_2 ... s_e\}$ *where*$(e \leq g)$

3.3.1 Feature Selection Methods Used for Fusion

a) Mutual Information (MI)

MI can be employed in information theory to evaluate any arbitrary dependency between random variables [30]. It specifically evaluates the average amount of information transmitted between two random variables. If two random variables, S and Y, are independent, if S does not contain any information about Y and vice versa, then MI is zero. The mathematical form of MI is

$$MI(S;T) = H(S) - H\left(\frac{S}{T}\right) = H(T) - H\left(\frac{T}{S}\right) = H(S) + H(T) - H(S;T) \tag{2}$$

where $H(S)$, $H(T)$ are the entropy of the random variables. $H\left(\frac{T}{S}\right), H\left(\frac{S}{T}\right)$ Conditional entropy and H(S; T) are the joint entropy of the random variable.

Entropy is the degree of uncertainty in its information, and its mathematical form is

$$H(S) = -\sum_{s \in S} P(s) \log P(s) \tag{3}$$

where $P(s)$ is the probability distribution.

The entropy of a combined probability distribution or a multi-valued random variable is known as joint entropy, and its mathematical form is

$$H(S;T) = -\sum_{s \in S} \sum_{t \in T} P(s,t) \log P(s,t) \tag{4}$$

where $P(s, t)$ denotes the joint probability distribution

Conditional entropy $H\left(\frac{S}{T}\right)$ is the average degree of uncertainty regarding a variable S following the observation of a second random variable T. The mathematical form is

$$H\left(\frac{S}{T}\right) = \sum_{t \in T} P(t) \left[-\sum_{s \in S} P\left(\frac{s}{t}\right) \log P\left(\frac{s}{t}\right) \right] \tag{5}$$

where $P\left(\frac{s}{t}\right) = \frac{P(s,t)}{p(t)}$ is the conditional probability of given 't'. We can determine from the preceding equation that mutual information is the interconnectedness of the two variables' uncertainty levels, expressed in Eq. (3). MI values are always non-negative, that is $MI \geq 0$.

b) Standard Deviation (SD)

It is a statistical technique that measures how far attributes deviate from the mean. The examination of SD reveals that a high value for the SD says that the feature values are dispersed across an extensive range of values. A low value for the SD suggests that the feature values are nearer to the mean [31]. As a result, feature selection using SD selects features with a high SD value because

successful prediction outcomes can be determined when the values are over a broad spectrum. The SD of a feature s_m is determined by

$$SD\,(s_m) = \sqrt{\frac{1}{n-1}\sum_{k=1}^{n}\left(s_{ki}\sum_{k=1}^{n}s_{ki}\right)} \tag{6}$$

where $k = 1, 2, 3,\ldots, d;\ i = 1, 2, 3,\ldots, n;\ \ s_{ki}$ is the value of sample k on its attribute m.

c) Difference Between Mean and Median (DMM)

They are descriptive statistical indicators used to characterize data distribution. Furthermore, these statistical metrics represent the relative magnitude of variation in a data distribution [32]. De Nijs et al. [33] stated that the difference between the mean and median can use as feature selection. The mathematical form is

$$score(s_i) = |mean - median| \tag{7}$$

Examining the mean-median difference reveals that a high difference value implies variance over an extensive range of values. Hence attributes with a high difference value can be regarded as a significant feature for successful prediction and classification [34].

d) Random Forest Feature Importance (RFI):

An ensemble classifier known as Random Forest supports a variety of feature relevance metrics and is constructed using several decision trees [35]. Feature selection is made directly by Random Forest while a classification rule is applied. Gini significance index (GI) and permutation importance index are the two typically employed feature importance measures in RFI. We used GI-based RFI to extract the feature importance in the proposed work. Gini impurity illustrates how effectively a split divides the total samples of binary classes in a given node. The mathematical form of GI is

$$GI(\tau) = 1 - P_1^2 - P_0^2 \tag{8}$$

The drop in Gini impurity caused by the best split $\triangle GI_f(\tau, \mathrm{T})$ is noted and calculated for each node τ in each tree T in the forest, separately for each attribute, and it is mathematical form is

$$GI\,(f) = \sum_{T}\sum_{\tau}\triangle GI_f\,(\tau, \mathrm{T}) \tag{9}$$

where $GI(f)$ is the gini importance of a feature f.

e) Permutation Importance (PI)

Breiman [35] suggested PI for RFI. Using this methodology, Fisher et al. [36] presented a model-agnostic feature importance. According to that approach, a model M(SVM) is trained on data DD with the feature set $S = \{s_1, s_2, s_3, \ldots . s_n\}$ and target variable Y where $Y\varepsilon\ \{0, 1\}$

1) Calculate model error on the feature set S;

$$error_{ori} = L(y, M) \tag{10}$$

2) For every feature, s_i; where $i = \{1, 2 \ldots n\}$ do;
3) Permute feature s_i in the data DD to create a feature matrix DD_{perm}.

It dissociates the feature from the actual outcome y.

4) Using the permuted data's predictions as a basis, estimate error

$$error_{perm} = L(y, M(DD_{perm})) \tag{11}$$

5) Evaluate the permutation importance (j_i) of a feature as s_j

$$j_i = \frac{1}{n}\sum_{n=1}^{n} error_{ori} - error_{perm} \tag{12}$$

f) Shapley Additive exPlanations (SHAP)

An idea from game theory [37] was adapted to create the unified model SHAP framework, which was put forth by Lundberg et al. [38] for interpreting predictions. Regarding computing, SHAP results in Shapley values, linear combinations of binary variables representing model predictions. Consider a scenario S is a subset of n features. $X = \{x_i/l\varepsilon[1, \ldots n]\}$ reflects the dataset's feature values vector. The payout of the S features values is denoted by $val(S)$ which is the final prediction. The mathematical form of the shapely value ϕ_l of the feature l is

$$\phi_l\,(val) = \sum_{S\subseteq\{x_1,x_2,sx\,\ldots\,x_n\}\setminus\{x_l\}} \frac{|S|\,!\,(n-|S|-1)!}{n!}\,(val\,(S\cup\{s_l\}-val\,(S) \tag{13}$$

The Shapely value measures the average strength with which an attribute influences predictions. SHAP aims to calculate each feature's contribution to the forecast, and the mathematical form is

$$M\,(y') = \phi_0 + \sum_{l=1}^{M}\phi_l y_l^{'} \tag{14}$$

where M is the ML model. In this work, we used a Light gradient boosting machine (LGBM) as a base classifier, $y^{'}\,\varepsilon\,\{0, 1\}^M$ is the coalition vector of the attributes utilized, M represents the largest coalition size, and $\phi_l \in \mathrm{R}$ is the Shapley value of the feature l. Therefore, features with higher Shapley values are essential when using SHAP to gauge the significance of various features.

4 Results and Analysis

The proposed model was executed on Windows 10 Pro, Intel Core i7–10750H processor running at 2.60 GHz with 64 GB RAM and contains 2 GB of GeForce GTX 1080 Ti graphics. The experimental environment uses Python 3.8 programming language and libraries. The proposed model was analyzed using three publicly available datasets mentioned in Section 3.1. To analyze the performance of the proposed model, we carried out experiments in two cases as

1) Performance evaluation on imbalanced data with optimal features.
2) Performance evaluation on balanced data with optimal features

4.1 Performance Evaluation of Imbalanced Data with Optimal Features

The fusion of feature ranking method is used to select the essential features, where the top contributed attributes are chosen based on the occurrence frequency specified by the threshold (η). We examined the feature frequency and set the η as 4. Even though each feature selection uses a separate ranking algorithm, Tables 3–5 show that some features are identical among all the methods; features that won a simple plurality vote were determined using η are shown in Table 6. Then they are used to train the ML models like DT, LR, XGBM, SVM, and ET for attack classification. Tables 7–9 show

the experimental results obtained without balancing the data of NSL-KDD, UNSW-NB 15, and CIC-IDS2107 datasets, respectively.

Table 7: Results of imbalance data with optimal features on NSL-KDD

Model	*Acy*	*Pe*	*Rc*	*F*1−*Mes*	AUC	Tr. time (s)	Te. time (s)
XGBM	99.02	99.16	98.75	98.95	99.01	3.13	0.06
LR	87.33	87.02	95.65	86.33	87.22	2.52	0.004
DT	99.13	99.35	98.78	99.07	99.11	1.17	0
SVM	97.59	96.66	92.67	94.6	95.85	50.9	17.64
ET	98.27	99.64	92.74	96.07	96.32	3.4	0.1

Table 8: Results of imbalance data with optimal features on UNSW-NB15

Model	*Acy*	*Pe*	*Rc*	*F*1−*Mes*	AUC	Tr. time (s)	Te. time (s)
XGBM	83.23	76.9	99.4	86.71	81.41	30.6	0.17
LR	67.78	71.7	68.51	70.07	67.69	2.5	0.01
DT	82.14	76.49	97.53	85.74	80.41	1.8	0.01
SVM	80.57	74.25	99.05	84.88	78.49	426.54	277.56
ET	77.41	77.07	83.95	80.36	76.67	15.8	1.1

Table 9: Results of imbalance data with optimal features on CIC-IDS2107

Model	*Acy*	*Pe*	*Rc*	*F*1−*Mes*	AUC	Tr. time (s)	Te. time (s)
XGBM	99.93	99.11	96.34	97.7	98.16	59	0.15
LR	98.23	42.7	44.75	73.7	71.9	31.7	0.07
DT	99.95	98.67	98.71	98.69	99.34	163.9	0.13
SVM	99.09	80.21	54.1	64.61	79.94	79335.1	745.6
ET	99.92	98.41	96.54	97.47	98.25	266.83	7.3

It is observed from Table 7–9 most of the model's accuracy was reasonable and, in some cases, even better than balanced data. However, due to an imbalance in the data, the models in this instance are skewed towards the majority class samples. Data is to be unreliable, noisy, and unpredictable, with variations in format.

However, dealing with unbalanced data *F1-Mes* is also a crucial measure to consider. Among the models, XGBM performs better with an accuracy of 99.02%, 83.23%, and 99.93% with NSL-KDD, UNSW-NB15, and CIC-IDS 2017 datasets, respectively. But by observing *F1-Mes*, and *Rc,* they are not up to the mark; this may happen due to fewer minority samples. On the other hand, LR performance was not good compared to other models due to fewer minority instances. LR claims 87.33%, 67.78%, and 98.23% accuracy for the three datasets, NSL-KDD, UNSW-NB15, and CIC-IDS 2017.

4.2 Performance Evaluation on Balanced Data with Optimal Features

To improve the performance of *Acy, Pe, Rc,* and *F1-Mes,* we balanced the minority samples by using an over-sampling ADASYN technique, which increases the minority class samples, to address the imbalance issues. Tables 10–12 show the results obtained on balanced data with optimal features. The ADASYN approach not only minimizes the learning bias brought on by the initial imbalanced data distribution, but it may also adaptively adjust the decision boundary. Further, the ML models' hyperparameters are tuned using a halving random search CV to enhance the performance, as shown in Table 13. Halving Random Search CV employs sequential halving (SH) to search parameter space. SH is similar to a game among possible parameter combinations. SH is an iterative selection procedure considering all parameter combinations with limited resources during the first iteration. Then a subset of these parameters is used for the following iteration, which will receive more resources. The number of training samples is often the resource for parameter tweaking.

Table 10: Results on balance data with optimal features on NSL-KDD

Model	*Acy*	*Pe*	*Rc*	$F1-Mes$	AUC	Tr. time (s)	Te. time (s)
XGBM	99.86	99.83	99.88	99.86	99.86	1.7	0.01
LR	91.53	91.33	90.44	90.88	91.45	0.8	0
DT	99.4	99.41	99.3	99.35	99.39	0.2	0
SVM	97.26	98.31	95.79	97.04	97.17	90.39	27.4
ET	99.73	99.64	99.17	99.4	99.53	2.6	0.4

Table 11: Results on balance data with optimal features on UNSW-NB15

Model	*Acy*	*Pe*	*Rc*	$F1-Mes$	AUC	Tr. time (s)	Te. time (s)
XGBM	92.42	90.48	96.37	93.33	91.97	1.5	0.06
LR	78.94	94.59	65.5	77.4	80.45	1.17	0.003
DT	88.07	86.28	93.13	89.58	87.5	0.5	0.01
SVM	86.73	82.63	96.1	88.86	85.68	1158	654
ET	88.69	84.42	97.42	90.46	87.7	18.13	2.2

Table 12: Results on balanced data with optimal features on CIC-IDS2107

Model	*Acy*	*Pe*	*Rc*	$F1-Mes$	AUC	Tr. time (s)	Te. time (s)
XGBM	99.68	99.48	99.88	99.68	99.68	1.2	0.09
LR	83.02	76.24	95.96	84.97	83.02	0.5	0.002
DT	99.36	99.36	99.39	99.36	99.36	0.2	0
SVM	83.33	75.85	97.81	85.44	83.33	1413	134
ET	99.57	99.39	99.75	99.57	99.57	4.2	0.45

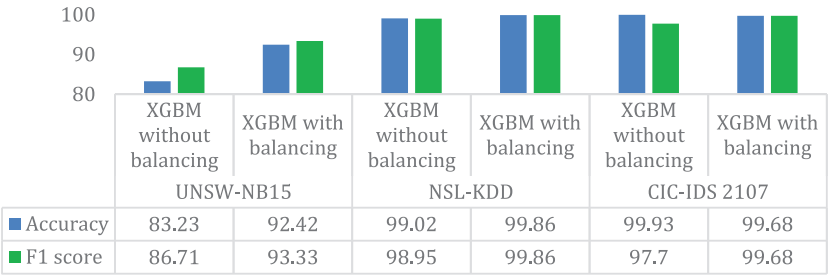
Table 13: Hyperparameters

UNSW-NB15	XGBM	learning_rate = 0.1, subsample = 1.0, max_depth = 9, n_estimators = 27
	LR	penalty = none, C = 54
	DT	max_depth = 32, criterion = gini
	SVM	gamma = 0.02, C = 1.0, kernel = rbf
	ET	criterion = entropy, max_depth = 45, n_estimators = 75
NSL-KDD	XGBM	learning_rate = 0.39, max_depth = 14, n_estimators = 84
	LR	Penalty = l2, C = 100
	DT	max_depth = 150, criterion = entropy
	SVM	C = 0.1, gamma = 0.03 kernel = rbf
	ET	criterion = entropy, max_depth = 77, n_estimators = 92
CIC-IDS 2017	XGBM	learning_rate = 0.7, max_depth = 9, n_estimators = 77
	LR	penalty = none, C = 100
	DT	max_depth = 23, criterion = gini
	SVM	C = 1.0, kernel = rbf, gamma = auto
	ET	criterion = gini, max_depth = 56, n_estimators = 100

Once the data is balanced and hyperparameters are tuned, we observed the detection rate improved in three data sets. By analyzing experimental results, SVM takes more training time when compared with imbalanced data. Because when there is an imbalance in the data, the SVM can get biased towards the majority class samples, resulting in poor classification results. To solve this problem, we applied an oversampling technique ADASYNC to balance the dataset. This method raises the number of minority samples in the dataset, which can lengthen the time required for SVM training. It may happen because the SVM needs to consider all the instances in the dataset to locate the decision border between the classes during training. When there are more samples, the SVM's computational complexity increases, making the training process take significantly longer. At the same time, DT and LR take less training time. But when compared with other performance metrics, XGBM performs better. XGBM performs better with all three datasets by comparing imbalance and balanced data cases because the XGBM tree employs a series of decision trees, each learning from the tree before it and influencing the currently processed tree.

Consequently, they make the model more robust and produce an effective learner. By observing the experimental results of the CIC-IDS 2017 data set from Tables 9 and 12, In the case of imbalanced data, the accuracy of XGBM, SVM, ET, and DT was higher when compared to the balanced data. It may happen due to more majority samples, and the models are biased towards majority samples, but when observing precision, recall, $F1-Mes$ and AUC of imbalanced data was not good. Therefore, when the data is balanced, the bias is reduced, and the performance is improved.

Fig. 3 shows accuracy and *F1-Mes* comparison on the three datasets with the XGBM classifier. When ADASYN increased the minority samples, accuracy improved by 9.19% on UNSW-NB15% and 0.84% on NSL-KDD datasets. *F1-Mes* raises by 6.62% on UNSW-NB15%, 0.95% on NSL-KDD, and 1.98% on CIC-IDs 2017 data sets, respectively.



	UNSW-NB15		NSL-KDD		CIC-IDS 2107	
	XGBM without balancing	XGBM with balancing	XGBM without balancing	XGBM with balancing	XGBM without balancing	XGBM with balancing
Accuracy	83.23	92.42	99.02	99.86	99.93	99.68
F1 score	86.71	93.33	98.95	99.86	97.7	99.68

Figure 3: Analysis of the proposed model on imbalance and balanced data

To examine the proposed model detection, we plot AUC (Area Under the Curve) and ROC (Receiver Operating Characteristic) curves as depicted in Figs. 4–6 for the three balanced datasets. The AUC-ROC curve is a statistic for evaluating binary classification tasks. Where ROC is a probability curve, AUC represents the degree of separability. It illustrates the True Positive Rate (TPR) against False Positive Rate (FPR) on the y and x-axis at various threshold levels. Hence, a higher AUC suggests that a model has an excellent detecting capabilities rate.

By observing Figs. 4–6, ROC curves of the three datasets are in the upper left corner of the graph, indicating the detection rate is higher with less false alarm rate. We observed that AUC values are more than 0.9 for three data sets implying that our proposed feature selection model functions most effectively using XGBM with optimal features. Based on the findings, it is possible to conclude that the performance of the proposed model is superior in terms of accuracy, precision, and recall.

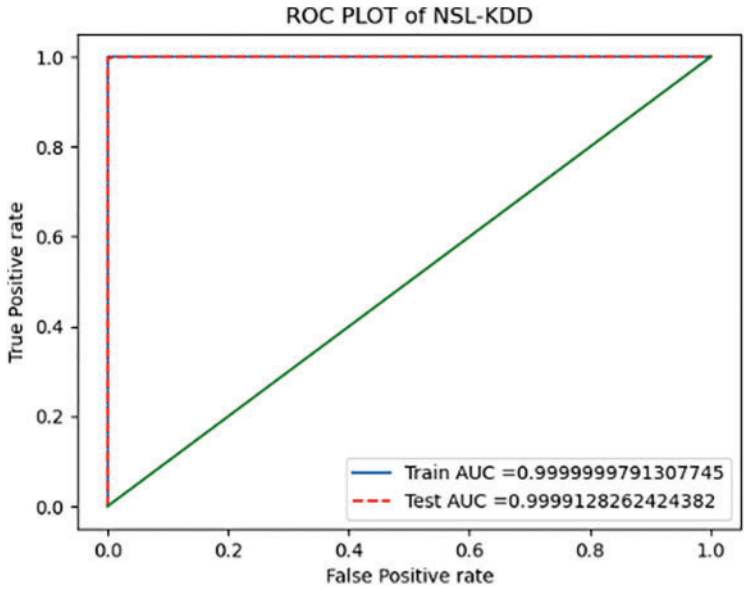


Figure 4: ROC curve proposed feature selection with XGBM on NSL-KDD

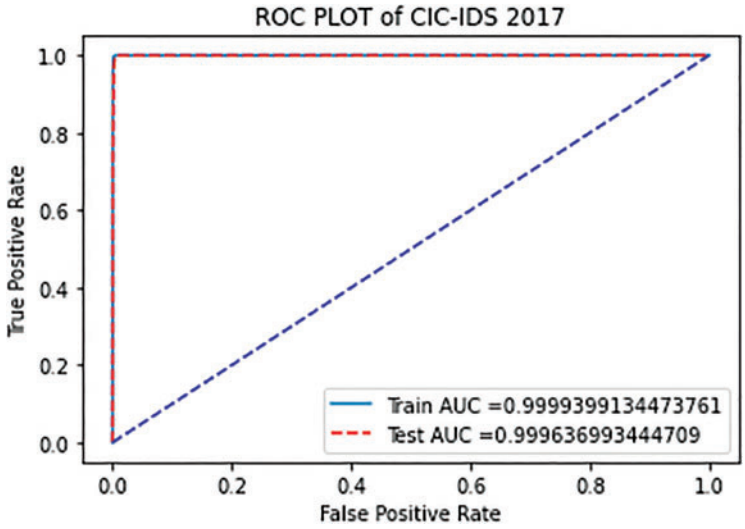


Figure 5: ROC curve proposed feature selection with XGBM on CIC-IDS 2017

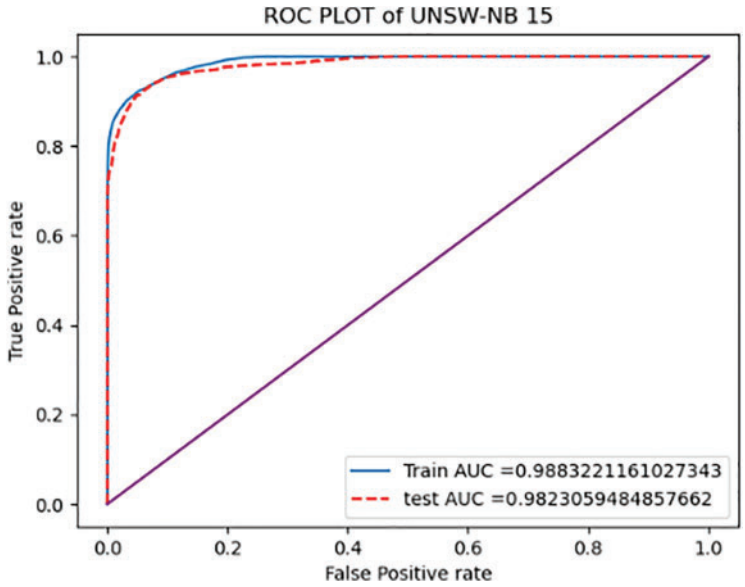


Figure 6: ROC curve proposed feature selection with XGBM on UNSW-NB 15

Further, we compare the proposed feature selection method with existing feature extraction and selection methods [39], like Pearson correlation coefficient (PCC), Information Gain (IG), and Principal component analysis (PCA). Table 14 shows the comparison results.

We have obtained 9, 8, 7 optimal features for NSL-KDD, UNSW-NB 15 and CIC-IDS 2017 datasets in our proposed method. So, we compared the existing feature selection methods such as PCC, IG and PCA, respectively, with 9, 8, 7 on NSL-KDD, UNSW-NB 15 and CIC-IDS 2017 datasets. By observing Table 14, we can infer that the proposed feature selection method outperformed the existing techniques with XGBM classifier for three datasets.

Table 14: Comparison of proposed FeS with other other FeS/extraction methods

Dataset	Model	FeS	No. of features	*Acy*	*Pe*	*Rc*	*F*1−*Mes*	AUC	Tr. time (s)
NSL-KDD	XGBM	PCC	9	99.10	98.85	99.37	99.11	99.10	17.7
	XGBM	IG	9	99.52	99.63	99.41	99.52	99.52	6.4
	XGBM	PCA	9	91.5	86.24	99.03	92.19	91.52	3.9
	XGBM	Proposed	9	99.86	99.83	99.88	99.86	99.86	1.5
UNSW-NB15	XGBM	PCC	9	50.86	57.94	39.21	46.77	52.17	14.8
	XGBM	IG	8	86.6	81.55	97.78	88.93	85.34	8.3
	XGBM	PCA	8	69.95	67.85	86.33	75.98	68.10	18.7
	XGBM	Proposed	8	92.42	90.48	96.37	93.33	91.97	5.5
CIC-IDS 2017	XGBM	PCC	7	97.35	96.19	98.60	97.38	98.35	5.5
	XGBM	IG	7	90.49	85.49	97.52	91.11	90.49	2.45
	XGBM	PCA	7	98.49	97.91	99.10	98.50	98.49	7.6
	XGBM	Proposed	7	99.68	99.48	99.88	99.68	99.68	4.5

4.3 Comparative Analysis

This section compares the proposed model metrics with various existing techniques for attack detection using the three datasets. Table 15 shows the comparative analysis of the proposed model with existing models.

Table 15: Comparative analysis of the proposed model with existing models

Dataset	Model	No. of FeS	*Acy*	*Pe*	*Rc*	*F*1−*Mes*	AUC	Tr. Time	FeS	CIb
NSL-KDD	RF [20]	21	99.83	99.9	99.62	99.78	NA	20.94	Yes	No
	ResNet 50 [40]	12	97.25	92	91	94	NA	NA	Yes	No
	ELM [41]	NA	96.53	NA	NA	NA	NA	4.64	Yes	Yes
	DM [42]	NA	99.80	99.83	99.84	99.83	NA	NA	Yes	No
	Proposed model	9	99.86	99.83	99.88	99.86	99.86	1.5	Yes	Yes
UNSW-NB 15	FFDNN [22]	22	85.48	NA	NA	NA	NA	NA	Yes	No
	ResNet 50 [40]	12	92.18	93	91	89	NA	NA	Yes	No
	DM [42]	NA	90.98	87.37	99.72	93.34	NA	NA	Yes	No
	XGBM [43]	19	90.85	83.33	98.38	88.4584	NA	NA	Yes	No
	Proposed model	8	92.42	90.48	96.37	93.33	91.97	5.5	Yes	Yes
CIC-IDS 2017	EL [24]	30	99	NA	NA	NA	NA	NA	Yes	No
	EL [24]	78	92	NA	NA	NA	NA	NA	No	No
	ResNet 50 [40]	12	95.23	95.63	95.25	94.92	NA	NA	Yes	No
	DM [42]	NA	98.97	99.9	94.43	97.08	NA	NA	Yes	No
	EL [44]	15	88.92	NA	NA	NA	NA	NA	Yes	No
	Proposed model	7	99.68	99.48	99.88	99.68	99.68	4.5	Yes	Yes

Note: *NA-not available.

To reduce resource utilization and computational time, Kannari et al. [20] suggest an IDS model using Recursive feature elimination with RF classifier to identify attacks. They tested their model on the NSL-KDD data set. Shaikh et al. [40] suggested an IDS detect attacks in the network by using CNN and resnet50. They evaluated their model on NSL-KDD, UNSW-NB15, and CIC-IDS 2017. To prevent intrusion in a cloud-based IoT environment, Lin et al. [41] developed an IDS using multi-feature extraction Extreme Learning Machine (MELM) to detect attacks on the NSL-KDD dataset. Yousefnezhad et al. [42] increased the detection rate and reduced the false alarm rate by proposing an ensemble classification model using Dempster–Shafer technique (DM) to detect assaults in the network traffic. They trained their model on datasets like NSL-KDD, UNSW-NB 15, and CIC-IDS2017.

Kasongo et al. [22] recommended wrapper-based feature extraction by using an Extra tree (ET) to select optimal features. Then they used a feed-forward deep neural network to detect attacks in the wireless networks and trained their model on the UNSW-NB15 dataset. Mhawi et al. [24] suggested a hybrid feature selection using correlation feature selection and Forest Panelized Attributes (CFS–FPA). Further, the optimal features are given to an ensemble classifier to identify attacks in the network. They tested their model on CIC-IDS 2017 dataset. Kasongo et al. [43] suggested a feature selection model using XGBoost to select the optimal features. Then they are fed to the DT classifier to classify the attacks. Finally, they evaluated their model on UNSW-NB 15 dataset. Abbas et al. [44] proposed ensemble-based (EL) IDs to detect threats in IoT networks using CIC-IDS 2017.

Even Mhawi et al. [24] have obtained an accuracy of nearly 99% on the CIC-IDS 2017 dataset, but their model uses 30 attributes which is more when compared to our proposed model. Moreover, Kannari et al. [20] attained an accuracy of nearly 99% on the NSL-KDD dataset, but compared with our proposed model, the training time and the number of features are more for their models. Compared to earlier techniques, our suggested method outperforms the others since most solutions did not address the class imbalance. In the proposed work, we addressed class imbalance by using ADASYN. It is evident from the experimental results that when the data is balanced, our model outperforms other models with less training time and fewer features.

Finally, with less number of optimal features, our model performed well with an accuracy of 99.85%, 92.4%, and 99.68%, the precision of 99.85%, 90.48%, 99.48%, recall of 99.83%, 96.37%, 99.88%, $F1-Mes$ of 99.84%, 93.33%, 99.68%, and AUC of 99.86%, 91.975%, 99.68% on NSL-KDD, UNSW-NB15, and CIC-IDS 2017 datasets, respectively.

5 Conclusion and Future Work

Intrusion detection systems with redundant and irrelevant features significantly impact the results. To counteract the considerable influence, we proposed a fusion of feature ranking to select the most contributed features. Initially, the network traffic is preprocessed by removing duplicate records and handling missing NaN and negative values using mean imputation. The uneven distribution of data is balanced. Further, the fusion of feature importance is applied to retrieve the top ten features from each feature selection method. Then plurality voting is used to select the optimal features. Then the optimal features are fed to various ML models. Among them, XGBM outperforms other ML models. Hyperparameters are tuned to enhance the model performance by halving the random search CV. The proposed model was evaluated using publicly accessible IDS datasets such as NSL-KDD, UNSW-NB15, and CIC-IDS 2107. Finally, our proposed IDS produced superior outcomes with fewer features than existing approaches. The limitation of the proposed work is it can be computationally expensive, especially for big datasets, to run numerous feature selection methods for fusion which

could be challenging. In the future, we will expand our methodology to distinguish multiple attacks by considering metaheuristic algorithms on IoT datasets.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] V. S. D. Priya and S. S. Chakkaravarthy, "Containerized cloud-based honeypot deception for tracking attackers," *Scientific Reports*, vol. 13, no. 1, pp. 1–14, 2023.
[2] IBM, "IBM security's cost of a data breach report 2022," 2022.
[3] CYBER ATTACK check point' s 2022 Mid-year report, 2022.
[4] S. B. Mallampati and H. Seetha, "A review on recent approaches of machine learning, deep learning, and explainable artificial intelligence in intrusion detection systems," *Majlesi Journal of Electrical Engineering*, vol. 17, no. 1, pp. 29–54, 2023.
[5] K. S. Babu and Y. N. Rao, "Improved monarchy butterfly optimization algorithm (IMBO): Intrusion detection using MapReduce framework based optimized ANU-net," *Computers, Materials & Continua*, vol. 75, no. 3, pp. 5887–5909, 2023.
[6] T. Mahjabin, Y. Xiao, G. Sun and W. Jiang, "A survey of distributed denial-of-service attack, prevention, and mitigation techniques," *International Journal of Distributed Sensor Networks*, vol. 13, no. 12, pp. 1–33, 2017.
[7] V. Jyothsna, V. V. Rama Prasad and K. Munivara Prasad, "A review of anomaly-based intrusion detection systems," *International Journal of Computer Applications*, vol. 28, no. 7, pp. 26–35, 2011.
[8] I. Sharafaldin, A. H. Lashkari and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSP 2018*, Funchal, Madeira, Portugal, pp. 108–116, 2018.
[9] G. Bagyalakshmi, G. Rajkumar, N. Arunkumar, M. Easwaran, K. Narasimhan *et al.,* "Network vulnerability analysis on brain signal/image databases using Nmap and wireshark tools," *IEEE Access*, vol. 6, pp. 57144–57151, 2018.
[10] M. Ring, S. Wunderlich, D. Scheuring, D. Landes and A. Hotho, "A survey of network-based intrusion detection data sets," *Computers and Security*, vol. 86, pp. 147–167, 2019.
[11] R. E. Bellman, "Preface," in *Dynamic Programming*. Princeton, New Jersey: Princeton University Press, 1957.
[12] P. Hui Li, J. Xu, Z. Yi Xu, S. Chen, B. Wei Niu *et al.,* "Automatic botnet attack identification based on machine learning," *Computers, Materials & Continua*, vol. 73, no. 2, pp. 3847–3860, 2022.
[13] B. Pes, "Ensemble feature selection for high-dimensional data: A stability analysis across multiple domains," *Neural Computing and Applications*, vol. 32, no. 10, pp. 5951–5973, 2020.
[14] B. Seijo-Pardo, I. Porto-Díaz, V. Bolón-Canedo and A. Alonso-Betanzos, "Ensemble feature selection: Homogeneous and heterogeneous approaches," *Knowledge-Based Systems*, vol. 118, pp. 124–139, 2017.
[15] P. Drotár, M. Gazda and L. Vokorokos, "Ensemble feature selection using election methods and ranker clustering," *Information Sciences*, vol. 480, pp. 365–380, 2019.
[16] A. Tiwari and A. Chaturvedi, "A hybrid feature selection approach based on information theory and dynamic butterfly optimization algorithm for data classification," *Expert Systems and Applications*, vol. 196, no. February, pp. 116621, 2022.
[17] R. K. Batchu and H. Seetha, "A generalized machine learning model for DDoS attacks detection using hybrid feature selection and hyperparameter tuning," *Computer Networks*, vol. 200, pp. 108498, 2021.

[18] O. Osanaiye, H. Cai, K. K. R. Choo, A. Dehghantanha, Z. Xu *et al.,* "Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing," *Eurasip Journal on Wireless Communications and Networking*, vol. 2016, no. 130, pp. 1–10, 2016.
[19] A. Bansal and S. Kaur, "Extreme gradient boosting based tuning for classification in intrusion detection systems," in *Proc. ICACDS 2018*, Dehradun, India, vol. 905, pp. 372–380, 2018.
[20] P. R. Kannari, N. S. Chowdary and R. Laxmikanth Biradar, "An anomaly-based intrusion detection system using recursive feature elimination technique for improved attack detection," *Theoretical Computer Science*, vol. 931, pp. 56–64, 2022.
[21] A. A. Najar, "DDoS attack detection using MLP and random forest algorithms," *International Journal of Information Technology*, vol. 14, no. 5, pp. 2317–2327, 2022.
[22] S. M. Kasongo and Y. Sun, "A deep learning method with wrapper-based feature extraction for wireless intrusion detection system," *Computers and Security*, vol. 92, pp. 101752, 2020.
[23] S. Saha, A. T. Priyoti and A. Sharma, "Towards an optimized ensemble feature selection for DDoS detection using both supervised and unsupervised method," *Sensors*, vol. 22, no. 23, pp. 1–17, 2022.
[24] D. N. Mhawi, A. Aldallal and S. Hassan, "Advanced feature-selection-based hybrid ensemble learning algorithms for network intrusion detection systems," *Symmetry*, vol. 14, no. 7, pp. 1–17, 2022.
[25] M. Ali, N. Iqbal, H. Jamil and D. Kim, "An optimized ensemble prediction model using AutoML based on soft voting classifier for network intrusion detection," *Journal of Network and Computer Applications*, vol. 212, pp. 103560, 2023.
[26] A. Henry, S. Gautam, S. Khanna, K. Rabie, T. Shongwe *et al.,* "Composition of hybrid deep learning model and feature optimization for an intrusion detection system," *Sensors*, vol. 23, no. 2, pp. 890, 2023.
[27] M. Tavallaee, E. Bagheri, W. Lu and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *IEEE Symp. on CISDA 2009*, Ottawa, Canada, pp. 1–6, 2009.
[28] J. S. Moustafa Nour, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data Set)," in *Proc. MilCIS*, Canberra, ACT, Australia, pp. 1–6, 2015.
[29] H. He, Y. Bai, E. A. Garcia and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. of the Int. Joint Conf. on Neural Networks*, Hong Kong, China, March, pp. 1322–1328, 2008.
[30] M. V. Brahmam, S. Gopikrishnan, K. R. Sravan and M. S. Bhavani, "Pearson correlation based outlier detection in spatial-temporal data of IoT networks," in *Proc. ICIDCA 2021*, Coimbatore, India, vol. 96, pp. 1019–1028, 2022.
[31] J. Xie, M. Wang, S. Xu, Z. Huang and P. W. Grant, "The unsupervised feature selection algorithms based on standard deviation and cosine similarity for genomic data analysis," *Frontiers in Genetics*, vol. 12, no. May, pp. 1–17, 2021.
[32] U. De Moncton, N. Brunswick, T. L. Hung and V. Nam, "The mean and median absolute deviations," *Mathematical and Computer Modelling*, vol. 34, no. 7–8, pp. 921–936, 2001.
[33] R. De Nijs and T. L. Klausen, "On the expected difference between mean and median introduction," *Electronic Journal of Applied Statistical Analysis*, vol. 6, no. 1, pp. 110–117, 2014.
[34] A. Thakkar and R. Lohiya, "Fusion of statistical importance for feature selection in deep neural network-based intrusion detection system," *Information Fusion*, vol. 90, pp. 353–363, 2023.
[35] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.
[36] A. Fisher, C. Rudin and F. Dominici, "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously," *Journal of Machine Learning Research*, vol. 20, pp. 1–81, 2019.
[37] L. S. Shapley, "A value for N-person games," in *Classics in Game Theory, 1*. New Jersey, United States: Princeton University Press, pp. 69, 1997.
[38] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions," in *Proc. NIPS*, Long Beach, CA, USA, pp. 4766–4775, 2017.
[39] C. Do Xuan, H. Thanh and N. T. Lam, "Optimization of network traffic anomaly detection using machine learning," *International Journal of Electrical and Computer Engineering*, vol. 11, no. 3, pp. 2360–2370, 2021.

[40] A. Shaikh and P. Gupta, "Real-time intrusion detection based on residual learning through ResNet algorithm," *International Journal of Systems Assurance Engineering and Management*, 2022. https://doi.org/10.1007/s13198-021-01558-1

[41] H. Lin, Q. Xue and D. Bai, "Internet of things intrusion detection model and algorithm based on cloud computing and multi-feature extraction extreme learning machine," *Digital Communication Networks*, vol. 9, no. 1, pp. 111–124, 2022.

[42] M. Yousefnezhad, J. Hamidzadeh and M. Aliannejadi, "Ensemble classification for intrusion detection via feature extraction based on deep learning," *Soft Computing*, vol. 25, no. 20, pp. 12667–12683, 2021.

[43] S. M. Kasongo and Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset," *Journal of Big Data*, vol. 7, no. 105, pp. 1–20, 2020.

[44] A. Abbas, M. A. Khan, S. Latif, M. Ajaz, A. A. Shah *et al.,* "A new ensemble-based intrusion detection system for internet of things," *Arabian Journal for Science and Engineering*, vol. 47, no. 2, pp. 1805–1819, 2022.