TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 02/21 pp948–958
DOI: 10.26599/TST.2023.9010032
Volume 29, Number 4, August 2024

Novel Framework for an Intrusion Detection System Using Multiple Feature Selection Methods Based on Deep Learning

A. E. M. Eljialy, Mohammed Yousuf Uddin, and Sultan Ahmad*

Abstract: Intrusion detection systems (IDSs) are deployed to detect anomalies in real time. They classify a network's incoming traffic as benign or anomalous (attack). An efficient and robust IDS in software-defined networks is an inevitable component of network security. The main challenges of such an IDS are achieving zero or extremely low false positive rates and high detection rates. Internet of Things (IoT) networks run by using devices with minimal resources. This situation makes deploying traditional IDSs in IoT networks unfeasible. Machine learning (ML) techniques are extensively applied to build robust IDSs. Many researchers have utilized different ML methods and techniques to address the above challenges. The development of an efficient IDS starts with a good feature selection process to avoid overfitting the ML model. This work proposes a multiple feature selection process followed by classification. In this study, the Software-defined networking (SDN) dataset is used to train and test the proposed model. This model applies multiple feature selection techniques to select high-scoring features from a set of features. Highly relevant features for anomaly detection are selected on the basis of their scores to generate the candidate dataset. Multiple classification algorithms are applied to the candidate dataset to build models. The proposed model exhibits considerable improvement in the detection of attacks with high accuracy and low false positive rates, even with a few features selected.

Key words: feature selection; intrusion detection system; software-defined network; decision tree; random forest; logistic regression; XGB classifier; AdaBoost

1 Introduction

The exceptional growth of devices connected to the Internet has resulted in an upsurge in security threats. Intrusion detection systems (IDSs) are at the forefront of tackling security threats and detecting attacks through inbound traffic in a network[1]. Security threats breach the confidentiality, integrity, and availability of the service of a network. Security experts are rigorously engaged in developing effective methods to overcome existing and unknown threats. IDSs are a countermeasure for improving security in a network. Incoming traffic passes through IDSs, which in turn classify inbound packets as normal or an attack. This detection process tests several parameters to detect attacks. Inbound packets have numerous parameters. Including all parameters in the detection process negatively affects the efficiency and cost of IDSs. The selection of the parameters that highly contribute to anomaly detection is inevitable for building robust

• A. E. M. Eljialy and Mohammed Yousuf Uddin are with Department of Information Systems, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Alkharj, Saudi Arabia. E-mail: ae.mohammed@psau.edu.sa; m.yousuf@psau.edu.sa.
• Sultan Ahmad is with Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Alkharj, Saudi Arabia, and also with University Center for Research and Development (UCRD), Department of Computer Science and Engineering, Chandigarh University, Punjab, India. E-mail: s.alisher@psau.edu.sa.
* To whom correspondence should be addressed.
Manuscript received: 2023-03-06 ; revised: 2023-04-10; accepted: 2023-04-14

IDSs[2]. In this context, feature selection techniques come into the picture to train and build machine learning (ML) models. Feature selection requires using a suitable feature selection technique based on the data supplied and the algorithm used to build models. The performance of models is evaluated through metrics such as accuracy, detection rate, false positive rate, and false negative rate. A high detection rate and a low false alarm rate are the two main evaluation indices of effective IDSs[3]. The process of training and developing an ML model to detect anomalies is divided into four stages: feature engineering, feature selection, model creation, and model hyperparameter tuning. A dataset with benign and attack traffic is supplied to train the model. Furthermore, a new dataset with benign and attack traffic is supplied to test the model performance. The first stage of model training through feature engineering involves the cleaning and standardization of the data supplied. This stage includes null value conversion, deletion or redundant record deletion. The second stage is feature selection and is the important stage that influences overall model performance. Many feature selection techniques exist, and choosing the appropriate technique is crucial. Such a selection requires an understanding of the dataset supplied for training and the algorithm in the third stage of the model training process. The third stage is model creation using algorithms for classification and clustering and other methods. The fourth stage is hyperparameter tuning to improve the performance of the model.

The following sections of this paper are divided as follows: Related works are provided in Section 2. The proposed model is discussed in Section 3. The dataset description is given in Section 3.1; the data cleaning process is described in Section 3.2; feature selection methods are presented in Section 3.3; and classification methods are discussed in Section 3.4. The results and discussion are covered in Section 4. The conclusion and future scope are given in Section 5.

2 Related Work

Feature selection methods have been researched to reduce the dimensionality curse and improve the performance and generalization capacity of models. Dimensionality is reduced through feature selection or feature extraction. Feature selection works by removing irrelevant and redundant features, whereas feature extraction generates a new set of features that are compact and strong.

Feature selection is widely used in IDSs. It is performed with simple filter, wrapper, or embedded methods and a hybrid approach for which a method is developed by using a mixture of methods from the above categories.

Feature selection is essential for reducing complexity and improving performance. A previous work[4] proposed a hybrid feature selection technique combining the genetic algorithm with the mutual information (MI) based feature selection technique. This technique, when applied to the ADFA-LD dataset with 49 features, had better results than other approaches.

Inaccurately selected features result in a high false negative rate and low accuracy. CorrACC was proposed to select effective features to address the above problem[5]. This method is a wrapper-based feature selection method. CorrAcc and four classifiers achieved 95.0% accuracy when applied to the BoT-IoT dataset. Monika et al.[6] proposed a multiobjective feature selection method for improving cyberattack detection. This method selected fewer features than other methods. It specifically targeted DDoS attacks and achieved 99.9% accuracy on the CICIC 2017 dataset.

Siddiqi and Pak implemented a process flow for filter-based feature selection with five different transformation techniques to improve the efficiency and effectiveness of the feature selection method implemented in a previous work[7]. Compared with the existing process flow and feature selection methods, their process flow method selected a more relevant set of features with higher efficiency and accuracy. Information gain and gain ratio were applied to select subsets. Union operations on subsets were obtained, and the top 50.0% of IG and GR features were selected. The selected features were evaluated and validated on BoT-IoT and KDDCup99 datasets, respectively, and the JRip classifier proposed by Pushparaj and Deepak exhibited 99.9% accuracy on both datasets[8].

False alarm rate is also an important issue in IDSs. Moustafa and Slay proposed a feature selection method based on the central points of attribute values[9]. UNSW-NB15 and NSL-KDD datasets were used in this experiment. The mode and mean of the data in a short time dataset were divided into small parts to identify statistical characteristics. The results showed that the processing time was reduced when the data

were divided into small parts. Evaluation results after using expectation maximization clustering, logistic regression, and Naïve Bayes yielded accuracies of 77.2%, 83.0%, and 79.5%, respectively, and false alarm rates of 14.2%, 17.5%, and 61.4%, respectively. Lonely logistic regression also provided good results. Rene and Shalini proposed another approach similar to our work. Specifically, they proposed a hybrid method for feature selection that combined the best features of different feature selection methods[10]. They used the feature selection methods CfsSubsetEval, GainRatioAttributeEval, OneRAttributeEval, and SymmetricalUncert AttributeEval after they combined features that were selected from the NSL-KDD dataset. They utilized Bayesian network-, regression-, nearest neighbor-, tree-, and SVM-based classifiers in their experiment. Their results showed that their hybrid approach significantly improved detection rates and accuracy.

Feature selection can be accomplished with the simple filter methods proposed by Hong and Haibo[11]. These methods are based on χ^2 and an enhanced C4.5 algorithm for building a lightweight network intrusion detection system. An evaluation was performed on KDDCup99 datasets. Training time significantly reduced from 0.13 s to 0.02 s, and testing time decreased from 0.22 s to 0.03 s. Reducing the number of features did not compromise the effectiveness of the model. Emmanuel et al.[12] proposed a feature selection method with a rule-based hybrid feature selection method. The results of this method were proportional to the performance of the base classifier. Kumar et al. used a metaheuristic search algorithm to select relevant features. Their methodology utilized a hybrid of the gray wolf optimization and particle swarm optimization algorithms to select relevant features from the KDDCup99, NSL-KDD, and CICIDS-2017 datasets. An evaluation was performed with the random forest (RF) algorithm classifier. The results of their approach were compared with those of other approaches[13].

Bostani and Sheikhan proposed a hybrid feature selection method. They combined the binary gravitational search algorithm (BGSA) and MI to improve the efficiency of the standard BGSA algorithm. BGSA acted as a global search algorithm, and MI functioned as a filter-based method. The results of the above experiment were compared with those of ReliefF, mRMR, MIFS-U, and χ^2[14]. Naung et al. proposed a framework based on sequential detection architecture. In this framework, data were collected and categorized on the basis of attack type. Eight types of attacks were included. The feature selection module selected the features relevant to the attack class, and correlation-based feature selection was employed to build feature sets. Each feature set was trained with the following ML algorithms: artificial neural network, J48 decision tree, and Naïve Bayes. A model with high accuracy was selected as a subengine for each attack. Incoming network traffic passed through the attack detector, and the extracted features sequentially passed through each subengine to detect attacks. The above scheme showed a higher detection rate and lower processing time than formal detection schemes[15]. Alper and Pelin proposed an intelligent automated real-time intrusion detection and mitigation solution to detect attacks on SDN-based IoT networks. This approach involved automated flow feature extraction and flow classification with RF classifiers at the SDN application layer. Related works are summarized in Table 1[16].

3 Proposed Model

As shown in Figure 1, the stages of the proposed model are data preprocessing, feature selection, and classification with multiple classification algorithms. The data preprocessing stage involves cleaning the dataset to remove attributes with null values and high variance to generate a clean dataset. In the feature selection stage, Step 1 involves the application of multiple feature selection methods, including filter and wrapper methods, on the clean dataset. Step 2 combines all the selected features from the output of the previous step. Step 3 includes counting the number of times a feature or attribute is selected by different feature selection methods and sorting features. A feature with a high count is assigned a high score. In our proposed model, features with scores greater than 4 are selected, and the candidate dataset is generated. Classification starts with splitting the candidate dataset into training and testing datasets, applying multiple classification methods on the training dataset to generate models, and supplying the test dataset to analyze model performance.

3.1 Dataset description

The SDN-based IoT network dataset is specific to software-defined networks. All the features in this

Table 1 Related work.

SNo	Reference	Feature Selection	Classification	Dataset	Advantage	Limitation
1	Vijayanand et al.[4]	Genetic algorithm with MI	SVM, artificial neural network	ADFA-LD, KDD	High accuracy	Partial dataset and small training dataset
2	Mohammed Shafiq et al.[5]	CorrACC	C4.5, Naïve Bayes, RF, SVM	BoT-IoT	High accuracy	Sensitivity and specificity
3	Monika et al.[6]	Multiobjective feature selection	Extreme learning machine	CICIDS2017	Achieved 99.0% accuracy with six features	Limited to DDoS attacks
4	Murtaza and Wooguil[7]	Normalization filter-based feature selection	Yeo Johnson and Pearson	Unknown	Unknown	Undefined
5	Pushparaj and Deepak[8]	Information gain and gain ratio	JRip classifier	BoT-IoT and KDDCup99 datasets	Accuracies of 99.9% and 99.5% for BoT-IoT and KDD Cup	Limited to DoS and DDoS attacks
6	Moustafa and Slay[9]	Central points of attributes and association rule mining	Expectation maximization clustering, logistic regression	Dataset	Unknown	Central points of attributes and association rule mining
7	Rene and D. Shalini[10]	CFS, GR, OneR, SU	Bayes net, logistic, IB1 NBTree, and SGD with SVM	NSL-KDD	High performance	F-measure
8	Hong and Haibo[11]	χ^2 and enhanced C4.5	Decision tree	KDD Cup	Reduced training time	Old dataset
9	Emmanuel et al.[12]	Rule-based hybrid method	Deep learning	NSL-KDD	Deep learning	Single dataset
10	Kumar et al.[13]	Metaheuristic	GWO, CSA, DSAE	NSL-KDD	Performance	Computational complexity
11	Bostani and Sheikhan[14]	BGSA	Hybrid algorithm	NSL-KDD	Accurate and low cost	Single dataset
12	Naung et al.[15]	Correlation	ANN, J48	N-BaIoT	Accuracy of 99.0%	Limited to Botnet
13	Sarica and Angin[16]	Feature importance	RF	SDN dataset	High detection accuracy	Single classifier

dataset can be accessed in real time by using any SDN application. The dataset used in this study was generated by using five IoT devices in the year 2020. It consists of 210 000 records with 33 attributes. The target variable contains 350 00 samples of each attack category and 35 000 records of normal traffic. Attack categories in this dataset include DoS, DDoS, port scanning, OS fingerprinting, and fuzzing. This dataset was selected because it represents the most common attack categories in IoT networks[17]. Table 2 shows the dataset features and their descriptions.

3.2 Data cleaning

Data cleaning is an essential part of ML. Clean datasets result in reliable results and accurate model generation. Most recorded data, except for the most organized and synthetic datasets, are likely to contain some level of noise. Imperfect data might originate from a variety of sources. The accuracy of a classifier may be highly dependent on the quality of the training data. Therefore, a classifier constructed from the noise-free version of the same dataset using the same technique[18] may be less accurate and less compact than the one constructed from the SDN dataset used in this study. In our dataset, the two attributes srcPort and dstPort with null values were removed. The category attribute was not considered in this study because we are interested in attack packets instead of attack types.

3.3 Feature selection

Given that meaningful information must be extracted from the multidimensional feature space, high-

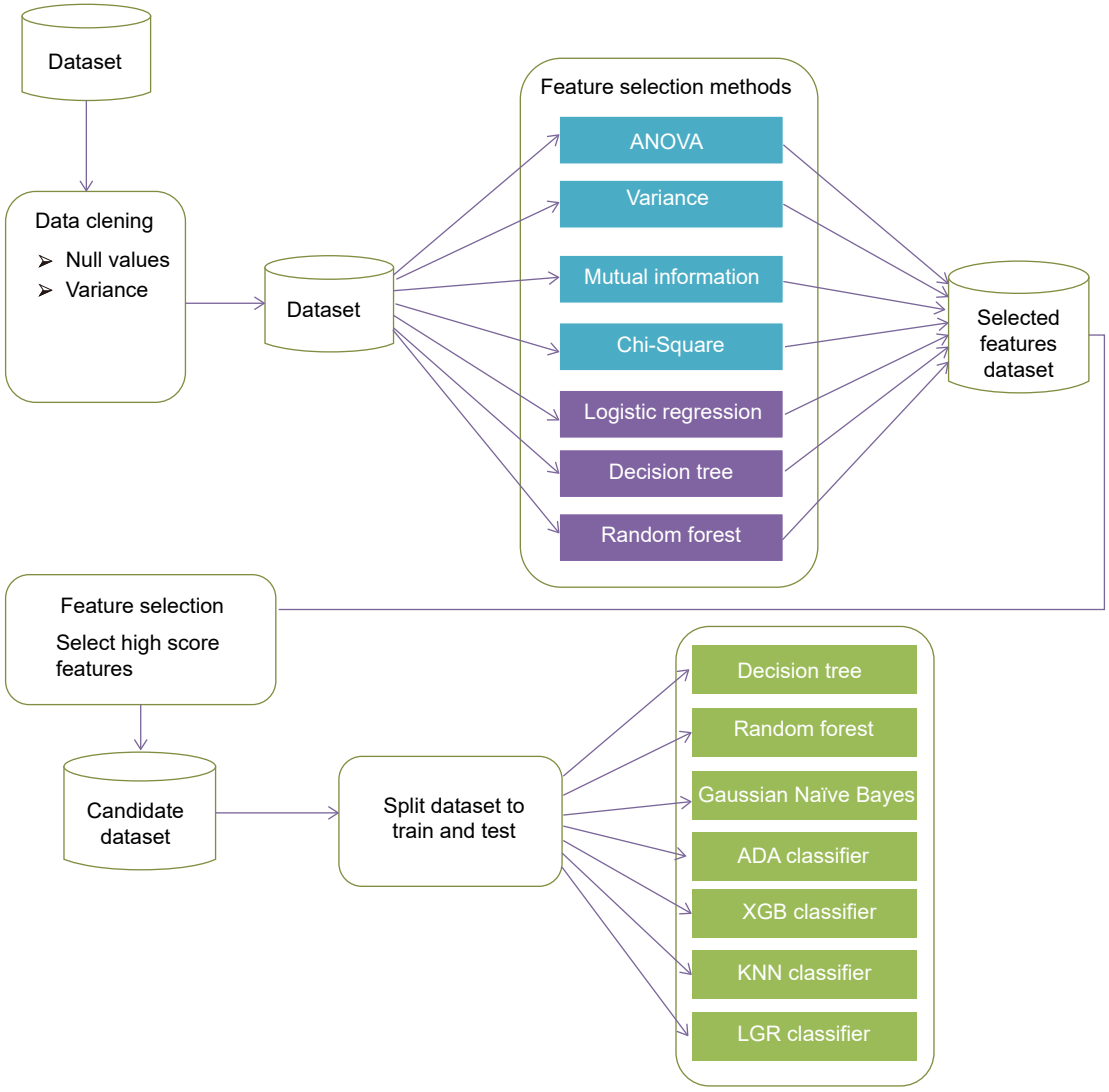

Fig. 1 Proposed model of feature selection.

dimensional attribute sets can have irrelevant features that introduce additional noise and difficulty to the learning algorithm. High-dimensional attribute sets can also contain a large number of dimensions. A decrease in the dimensionality of features can reduce the amount of time needed to complete the learning process, and careful feature selection can lead to an improvement in overall model performance. In addition, the utilization of feature selection strategies can help decrease instances of overfitting; this effect, in turn, contributes to the improved generalization of models[19]. We used filter techniques and wrapper methods, which are subcategories of feature selection methods, to choose the aspects that contribute to the quality of the generated model.

Table 3 contains a rundown of the several approaches that were used in this study. The score of each feature was generated by identifying the number of times a feature was selected by a feature selection technique. Eight methods were used on datasets. A minimum score of four was required to meet the 50.0% target. A selected feature from a particular method is denoted as "1" in Table 4. The "Score" column in the same table shows the final score. The top 17 features out of the 33 features contained by the SDN dataset were selected to build the candidate dataset. This approach resulted in a considerable decrease in dimensionality, which shortened execution time and increased the accuracy of detecting malicious traffic in SDN networks. Filter methods measure the importance of features on the basis of their correlation with the dependent variable. Filter techniques are considerably faster than wrapper methods because they do not require model training. The filter methods used in this

Table 2 Dataset description.

SNo	Feature	Description
1	srcMAC	Source MAC address
2	dstMAC	Destination MAC address
3	srcIP	Source IP address
4	dstIP	Destination IP address
5	srcPort	Source port number
6	dstPort	Destination port number
7	last_seen	Record last time
8	Protocol	Textual representation of the network protocol
9	proto_number	Numerical representation of the network protocol
10	Dur	Record total duration
11	Mean	Average duration of aggregated records
12	Stddev	Standard deviation of aggregated records
13	Min	Minimum duration of aggregated records
14	Max	Maximum duration of aggregated records
15	Pkts	Total count of packets in the transaction
16	Bytes	Total number of bytes in the transaction
17	Spkts	Source-to-destination packet count
18	Dpkts	Destination-to-source packet count
19	Sbytes	Source-to-destination byte count
20	Dbytes	Destination-to-source byte count
21	Srate	Source-to-destination packets per second
22	Drate	Destination-to-source packets per second
23	Sum	Total duration of aggregated records
24	TnBPSrcIP	Total number of bytes per source IP
25	TnBPDstIP	Total number of bytes per destination IP
26	TnP_PSrcIP	Total number of packets per source IP
27	TnP_PDstIP	Total number of packets per destination IP
28	TnP_PerProto	Total number of packets per protocol
29	TnP_Per_Dport	Total number of packets per destination port
30	N_IN_Conn_P_DstIP	Number of inbound connections per source IP
31	N_IN_Conn_P_SrcIP	Number of inbound connections per destination IP
32	Attack	Attack or not
33	Category	Traffic category

study are ANOVA, variance, correlation, MI, and χ^2 test. In wrapper methods, the feature selection procedure is determined by a particular ML algorithm that is applied to the given dataset. It employs a greedy search strategy by comparing all potential feature combinations for evaluation[20]. The wrapper methods used in this study are logistic regression, decision tree, and RF.

Table 3 Feature selection methods.

Filter method	Wrapper method
Variance	Logistic regression
ANOVA classifier	Decision tree
MI	RF
χ^2 test	

3.3.1 Variance

The variance threshold is a straightforward feature selection technique that functions as a baseline. It removes any and all characteristics with a variance that falls short of a certain threshold. Given that it discards all features that have zero variance by default, it removes characteristics that always have the same value across all samples[21]. We assumed that features with high variance may contain useful information; however, we must note that we ignored the relationship between feature variables or between feature and target variables. This situation is one of the disadvantages of using filter methods.

$$\sigma^2 = \frac{\sum_{i=n}^{n}(x_i-\mu)^2}{n} \tag{1}$$

where x_i represents individual observations, n is the number of observations, and μ is the mean of the dataset. The variance measures the spread or dispersion of a set of data around its mean.

3.3.2 Correlation

The metric used in feature selection considers groups of features that are highly associated with the target variable but are uncorrelated with each other[22].

$$r = \frac{\sum(x-\overline{x})(y-\overline{y})}{\sqrt{\sum(x-\overline{x})\sum(y-\overline{y})^2}} \tag{2}$$

3.3.3 ANOVA test

Analysis of variance, also known as the ANOVA test, is a statistical test that is used to determine whether or not a statistically significant difference exists between two or more categorical groups by testing for differences between means on the basis of variance[23].

$$f = \frac{Between-groupvariance}{Within-groupvariance} = \frac{MSB}{MSW} \tag{3}$$

Table 4 Ranking of selected features.

Feature	ANOVA	Variance	Corr	M-Info	χ^2	LR	DT	RF	Score
N_IN_Conn_P_SrcIP	1	1	1	1	1	0	1	1	7
TnBPDstIP	1	1	1	1	1	1	0	0	6
Dur	1	1	0	1	1	0	1	1	6
N_IN_Conn_P_DstIP	1	1	0	1	1	0	1	1	6
Dbytes	0	1	0	1	1	1	1	1	6
Sbytes	0	1	0	1	1	1	1	1	6
TnP_Per_Dport	1	1	1	1	1	0	0	0	5
last_seen	1	1	0	1	1	1	0	0	5
Protocol	1	1	1	1	0	0	1	0	5
TnP_PDstIP	1	1	1	1	0	1	0	0	5
TnBPSrcIP	0	1	0	1	1	1	1	0	5
Spkts	0	1	0	1	0	1	1	1	5
Stddev	1	1	1	1	0	0	0	0	4
Bytes	0	1	0	1	1	1	0	0	4
Mean	1	1	0	1	0	0	1	0	4
Srate	1	1	0	1	0	0	0	1	4
TnP_PerProto	0	1	1	1	0	1	0	0	4

3.3.4 MI

MI is a metric that measures the connection between two concurrently sampled random variables. Specifically, it measures the average amount of information conveyed from one random variable to another[24]. The MI between two random variables, X and Y, can be represented as

$$I(X:Y) = H(X) - H(X|Y) \tag{4}$$

where $I(X{:}Y)$ is the MI, $H(X)$ is the entropy of X, and $H(X|Y)$ is the conditional entropy of X given Y.

3.3.5 χ^2 test

χ^2 feature selection is a statistical technique for choosing features on the basis of their association with the target variable. This technique determines the χ^2 statistic between each feature and the target variable, then selects the features with the highest statistical value. Features with high statistics are seen to have a strong relationship with the target variable, whereas those with low statistics are deemed to have a weak relationship. Typically, the characteristics with the highest statistics are selected for predictive analytics[25].

$$x^2 = \sum \frac{(o_i - E_i)^2}{E_i} \tag{5}$$

3.3.6 Wrapper methods

Wrapper methods are a type of feature selection technique used in ML. They use the performance of a learning algorithm to evaluate the relevance of each feature in datasets. The basic idea is to fit a learning algorithm to the data, then use the performance of the algorithm as a measure of the importance of each feature[26]. The most common wrapper methods are

- Forward selection: This method starts with an empty feature set and iteratively adds features to the set, selecting the feature that results in the best performance in accordance with some criteria.
- Backward elimination: This method starts with the full feature set and iteratively removes features from the set, selecting the feature to remove on the basis of its effect on the performance of the algorithm.
- Recursive feature elimination: This method starts with the full feature set and iteratively removes features on the basis of their weight coefficients in the learning algorithm. Features are removed until a desired number of features is reached, or the performance of the algorithm reaches a certain threshold. Wrapper methods can be computationally expensive because they require training and evaluating a learning algorithm for each iteration. They are also sensitive to the choice and parameters of the learning algorithm as well as the performance criteria used to evaluate the feature set. Despite these limitations, wrapper methods are widely used because they can provide insight into the relationship between features and target variables and can be effective when the relationship between features and target variables is complex and cannot be easily captured by simple

statistical methods.

3.4 Classification methods

We used seven different ML classifiers, including decision tree (C4.5), RF, Gaussian Naïve Bayes ML, ADA, XGB, KNN, and LGR. The algorithms employed are listed in Table 4. In terms of accuracy, precision, sensitivity, and specificity, all of the utilized ML identifiers generated extremely promising results for effective feature selection for anomaly and intrusion detection in IoT traffic by utilizing the feature set determined by our suggested method. The decision tree uses nonparametric supervised learning techniques. Its objective is to develop a model that predicts the value of a target variable by learning basic decision rules inferred from data attributes. RF is a popular supervised ML algorithm for classification and regression issues[27]. It generates decision trees on the basis of distinct samples and takes their majority vote for classification and average in case of regression. The Naïve Bayes approach is based on the application of the Bayes theorem with the "naïve" assumption of conditional independence between each pair of features given the value of the class variable. The core notion of Adaboost is setting the weights of classifiers and training the data sample in each iteration to ensure the correct prediction of uncommon observations. Any ML method that takes weights on the training set can be used as a basic classifier. XGBoost is a gradient-boosted decision tree solution optimized for speed and performance in competitive ML[28]. The KNN classification of an item is determined by the majority vote of its neighbors, with the object being allocated to the class that is most prevalent among its *k*-closest neighbors. Logistic regression calculates the likelihood of an event, such as malicious or benign, given a dataset of independent factors[29].

4 Results and Discussion

The SDN dataset is a large collection of records containing 30.2 million records. In this study, a portion of the dataset with 210 000 records, which included 35 000 records from each category of attacks, was selected. In the beginning, the dataset had 33 features. It was preprocessed and cleaned to produce a clean dataset that contained 27 features. Preprocessing step source, destination mac address, and IP address attributes were removed, and several other attributes with low correlation were dropped from the dataset. The protocol attribute contained the three unique values tcp, udp, and icmp, for which dummy numerical columns were created. Seven of the 33 columns were eliminated from the dataset. SrcPort and dstPort contained null values, and the category column was also deleted because we are working on binary classification in the proposed work. A clean dataset with 27 columns and 210 000 instances in total was obtained. In this research context, a "clean dataset" has all relevant attributes. Additional datasets were separated into training and test versions. The training and testing sets comprised 70.0% and 30.0% of the dataset, respectively. The split was performed by using a random state value drawn from 10 different instances. The classifiers included in Table 4 were used in the training and testing phases of model development. Additional models were constructed by using the classifiers in Table 4 for the two candidate datasets. The first potential dataset had 17 features and a feature score either larger than or equal to 4. The second potential dataset consisted of columns with a score that was more than or equal to 5 and 12 characteristics. Accuracy and the F1 score were the metrics used to evaluate the models' performances. The accuracies of the models using the dataset that contained all features were compared with those of the models using the two candidate datasets. The comparison of accuracy is presented in Table 5, and the comparison of the F1 score is shown in Table 6. The results showed that the accuracy of the decision tree was 0.987 for all feature and candidate sets with scores greater than or equal to 4 and 0.982 for the candidate dataset with scores greater than or equal to 5. KNN showed an accuracy of 0.90 for all features with scores greater than or equal to 4 and 0.854 for the candidate dataset with scores greater than or equal to 5. The F1 scores of KNN also showed some variation. The results demonstrated that the candidate dataset with scores greater than or equal to 4 performed equally as the dataset with all features. The comparison of accuracy is shown visually by using a histogram in Figure 2, and the F1 score is presented in Figure 3. Our findings illustrated the relevance of our method as shown in Table 7. Our IDS model identified malicious traffic with a small number of features without losing its accuracy, provided that these attributes were chosen by using the appropriate feature selection method. In this study, the XGB, decision tree, RF, and ADA classifiers achieved higher accuracy than the other classifiers. Shafiq et al[5]. used a novel

CorrACC feature selection metric approach with a decision classifier, RF, and SVM and achieved 95.0% accuracy. Our proposed method achieved the highest accuracy of 99.0%.

5 Conclusion

In the proposed work, feature selection was based on a score and each selected feature obtained a score. The feature selection methods utilized in this work can be classified into two categories: filter- and wrapper-based methods. Five filter-based approaches and three classifier-based methods were implemented. The dataset with 33 attributes consisted of 27 attributes after it was preprocessed. The candidate datasets with scores greater than or equal to 4 had 17 characteristics,

Table 5 Classification algorithms.

S.No	Algorithm
1	Decision tree
2	RF
3	Gaussian Naïve Bayes
4	ADA classifier
5	XGB classifier
6	KNN classifier
7	LGR classifier

Table 6 Comparison of the accuracies of the models.

Classifier	All selected feature	Feature score 4	Feature score 5
Decision tree	0.987	0.987	0.982
RF	0.988	0.988	0.984
Gaussian Naïve Bayes	0.720	0.720	0.762
ADA classifier	0.986	0.986	0.983
XGB classifier	0.990	0.990	0.984
KNN classifier	0.900	0.900	0.854
LGR classifier	0.830	0.830	0.836

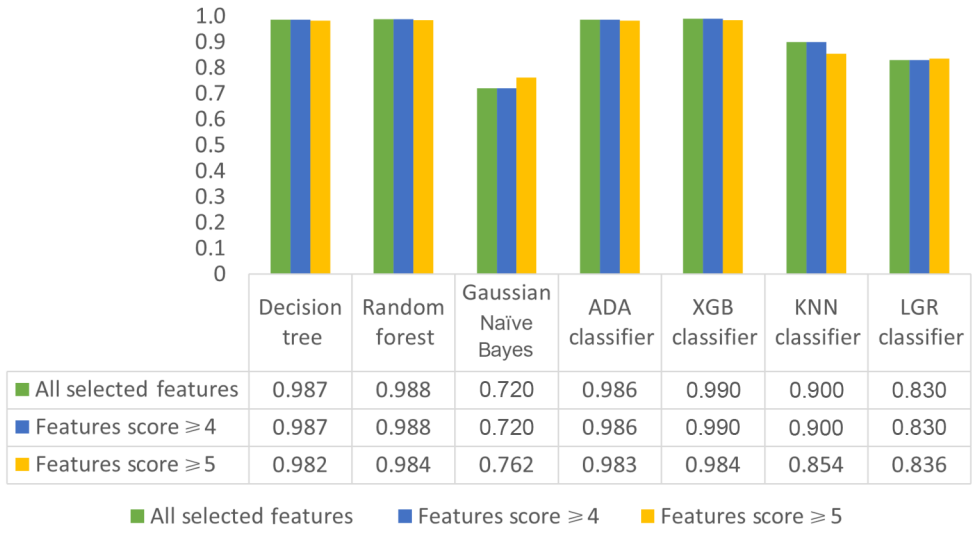

Fig. 2 Comparison of the accuracies of the models.

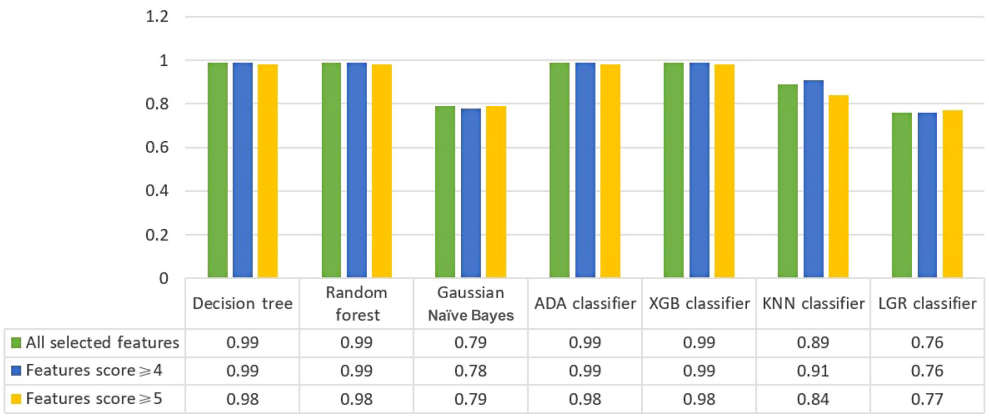

Fig. 3 Comparison of the F1 scores of the models.

Table 7 F1 scores of the models.

Classifier	All selected feature	Feature score	Feature score
Decision tree	0.99	0.99	0.98
RF	0.99	0.99	0.98
Gaussian Naïve Bayes	0.79	0.78	0.79
ADA classifier	0.99	0.99	0.98
XGB classifier	0.99	0.99	0.98
KNN classifier	0.89	0.91	0.84
LGR classifier	0.76	0.76	0.77

whereas those with scores greater than or equal to 5 had 12 attributes. Seven of the most prominent classification algorithms for anomaly detection were employed to examine their performance, and the results demonstrated that the careful selection of features not only minimized dimensionality but also preserved the accuracy of detection systems. Our work on functionality entailed the multiclass identification of attack type and data from non-SDN-based network configurations. This study utilized a sample of the available dataset to accommodate our system's limitations. A high-performance computing environment wherein we can simulate the suggested system with a complete dataset is planned for the future. This approach will not only help us understand our suggested system further but it will also reveal the hardware requirements for establishing an anomaly detection system.

Acknowledgment

The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number (IF2/PSAU/2022/01/23126)

References

[1] R. Gopi, M. Mathapati, B. Prasad, S. Ahmad, F. N. Al-Wesabi, M. Abdullah Alohali, and A. Mustafa Hilal, Intelligent DoS attack detection with congestion control technique for VANETs, *Comput. Mater. Continua*, vol. 72, no. 1, pp. 141–156, 2022.

[2] H. Hindy, D. Brosset, E. Bayne, A. K. Seeam, C. Tachtatzis, R. Atkinson, and X. Bellekens, A taxonomy of network threats and the effect of current datasets on intrusion detection systems, *IEEE Access*, vol. 8, pp. 104650–104675, 2020.

[3] S. Ahmad, S. Jha, A. Alam, M. Alharbi, and J. Nazeer, Analysis of intrusion detection approaches for network traffic anomalies with comparative analysis on botnets (2008–2020), *Secur. Commun. Netw.*, vol. 2022, pp. 1–11, 2022.

[4] R. Vijayanand, D. Devaraj, and B. Kannapiran, A novel intrusion detection system for wireless mesh network with hybrid feature selection technique based on GA and MI, *J. Intell. Fuzzy Syst.*, vol. 34, no. 3, pp. 1243–1250, 2018.

[5] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, IoT malicious traffic identification using wrapper-based feature selection mechanisms, *Comput. Secur.*, vol. 94, pp. 101863, 2020.

[6] M. Roopak, G. Y. Tian, and J. Chambers, Multi-objective-based feature selection for DDoS attack detection in IoT networks, *IET Netw.*, vol. 9, no. 3, pp. 120–127, 2020.

[7] M. A. Siddiqi and W. Pak, Optimizing filter-based feature selection method flow for intrusion detection system, *Electronics*, vol. 9, no. 12, pp. 2114, 2020.

[8] P. Nimbalkar and D. Kshirsagar, Feature selection for intrusion detection system in Internet-of-Things (IoT), *ICT Express*, vol. 7, no. 2, pp. 177–181, 2021.

[9] N. Moustafa and J. Slay, A hybrid feature selection for network intrusion detection systems: Central points, arXiv preprint arXiv: 1707.05505, 2017.

[10] J. Rene Beulah and D. Shalini Punithavathani, A hybrid feature selection method for improved detection of wired/wireless network intrusions, *Wirel. Pers. Commun.*, vol. 98, no. 2, pp. 1853–1869, 2018.

[11] H. Dai and H. Li, A lightweight network intrusion detection model based on feature selection, in Proc. 2009 15th IEEE Pacific Rim Int. Symp. on Dependable Computing, Shanghai, China, 2009, pp. 165–168.

[12] F. E. Ayo, S. O. Folorunso, A. A. Abayomi-Alli, A. O. Adekunle, and J. B. Awotunde, Network intrusion detection based on deep learning model optimized with rule-based hybrid feature selection, *Inf. Secur. J. A Glob. Perspect.*, vol. 29, no. 6, pp. 267–283, 2020.

[13] P. K. Keserwani, M. C. Govil, and E. S. Pilli, An optimal intrusion detection system using GWO-CSA-DSAE model, *Cyber Phys. Syst.*, vol. 7, no. 4, pp. 197–220, 2021.

[14] H. Bostani and M. Sheikhan, Hybrid of binary gravitational search algorithm and mutual information for feature selection in intrusion detection systems, *Soft Comput.*, vol. 21, no. 9, pp. 2307–2324, 2017.

[15] Y. N. Soe, Y. Feng, P. I. Santosa, R. Hartanto, and K. Sakurai, Machine learning-based IoT-botnet attack detection with sequential architecture, *Sensors*, vol. 20, no. 16, pp. 4372, 2020.

[16] A. K. Sarica and P. Angin, Explainable security in SDN-based IoT networks, *Sensors*, vol. 20, no. 24, pp. 7326, 2020.

[17] A. Kaan Sarica and P. Angin, A novel SDN dataset for intrusion detection in IoT networks, in Proc. 2020 16th Int. Conf. Network and Service Management (CNSM), Izmir, Turkey, 2020, pp. 1–5.

[18] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, Foundations of feature selection. Feature Selection for High-Dimensional Data. Cham: Springer, 2015: 13-28.

[19] K. Filus, P. Boryszko, J. Domańska, M. Siavvas, and E. Gelenbe, Efficient feature selection for static analysis

vulnerability prediction, *Sensors*, vol. 21, no. 4, pp. 1133, 2021.
[20] N. Kunhare, R. Tiwari, and J. Dhar, Particle swarm optimization and feature selection for intrusion detection system, *Sādhanā*, vol. 45, no. 1, pp. 1–14, 2020.
[21] A. Bommert, T. Welchowski, M. Schmid, and J. Rahnenführer, Benchmark of filter methods for feature selection in high-dimensional gene expression survival data, *Brief Bioinform*, vol. 23, no. 1, pp. bbab354, 2022.
[22] H. Zhou, X. Wang, and R. Zhu, Feature selection based on mutual information with correlation coefficient, *Appl. Intell.*, vol. 52, no. 5, pp. 5457–5474, 2022.
[23] M. Alassaf and A. M. Qamar, Improving sentiment analysis of Arabic tweets by one-way ANOVA, *J. King Saud Univ. Comput. Inf. Sci.*, vol. 34, no. 6, pp. 2849–2859, 2022.
[24] R. Lamba, T. Gulati, and A. Jain, A hybrid feature selection approach for Parkinson's detection based on mutual information gain and recursive feature elimination, *Arab. J. Sci. Eng.*, vol. 47, no. 8, pp. 10263–10276, 2022.
[25] S. K. Dey, K. M. M. Uddin, H. M. H. Babu, M. M. Rahman, A. Howlader, and K. M. Aslam Uddin, Chi2-MI: A hybrid feature selection based machine learning approach in diagnosis of chronic kidney disease, *Intell. Syst. Appl.*, vol. 16, pp. 200144, 2022.
[26] Jack Tan, How to improve data quality for machine learning? https://towardsdatascience.com/how-to-improve-data-preparation-for-machine-learning-dde107b60091 Accessed on January 1, 2023.
[27] A. Rajagopal, S. Ahmad, S. Jha, R. Alagarsamy, A. Alharbi, and B. Alouffi, A robust automated framework for classification of CT covid-19 images using MSI-ResNet, *Comput. Syst. Sci. Eng.*, vol. 45, no. 3, pp. 3215–3229, 2023.
[28] M. A. Talukder, K. F. Hasan, M. M. Islam, M. A. Uddin, A. Akhter, M. Abu Yousuf, F. Alharbi, and M. Ali Moni, A dependable hybrid machine learning model for network intrusion detection, *J. Inf. Secur. Appl.*, vol. 72, pp. 103405, 2023.
[29] S. Ahmad and M. Yousuf Uddin, An intelligent irrigation system and prediction of environmental weather based on nano electronics and Internet of Things devices, *J. Nanoelectron. Optoelectron.*, vol. 18, no. 2, pp. 227–236, 2023.

A.E.M Eljialy is an assistant professor in the Department of Information Systems at Prince Sattam Bin Abdulaziz University Alkharj, Saudi Arabia. His work focuses on delivering lectures such as Database security, Information system analysis & design and Database auditing. He has published many articles about database security, data science and deep learning. Right now, he is working on a project for a database security model to curb security vulnerabilities.

Mohammed Yousuf Uddin is a lecturer in the Department of Information Systems at Prince Sattam Bin Abdulaziz University, Alkharj, Saudi Arabia. He has more than 15 years of teaching and research experience. He received the master degree from Osmania University, Hyderabad. He was working as a lecturer in Yamen earlier. He published many research papers in reputed journals and conferences. His research area includes digital forensics, big data, data science and the Internet of Things. He has presented his research papers at many national and international conferences.

Sultan Ahmad is an IEEE Member. He received the PhD (CSE) degree from Glocal University and master of Computer Science and Applications degree from the prestigious Aligarh Muslim University, India, with distinction marks in 2006. Presently he is working as a senior lecturer in the Department of Computer Science, College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Alkharj, Saudi Arabia. He is also Adjunct Professor at Chandigarh University, Gharuan, Punjab, India. He has a unique blend of education and experience. He has more than 15 years of teaching and research experience. He has around 75 accepted and published research papers and book chapters in reputed SCI, SCIE, ESCI and SCOPUS-indexed journals and conferences. He has an Australian Patent in his name also. He has authored 4 books that are available on Amazon. His research area includes intelligence computing, Data Science, machine learning, and the Internet of Things. He has presented his research papers at many national and international conferences. He is a member of IEEE, IACSIT and the Computer Society of India