Arabian Journal for Science and Engineering (2022) 47:1353–1374
https://doi.org/10.1007/s13369-021-05947-3

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Analysis of Machine Learning Classifiers for Early Detection of DDoS Attacks on IoT Devices

Vimal Gaur[1,2] · **Rajneesh Kumar**[3]

Received: 26 December 2020 / Accepted: 27 June 2021 / Published online: 8 July 2021

Abstract
Distributed denial-of-service attacks are still difficult to handle as per current scenarios. The attack aim is a menace to network security and exhausting the target networks with malicious traffic from multiple sites. Although a plethora of conventional methods have been proposed to detect DDoS attacks, so far the rapid diagnosis of these attacks using feature selection algorithms is a daunting challenge. The proposed system uses a hybrid methodology for selecting features by applying feature selection methods on machine learning classifiers. Feature selections methods, namely chi-square, Extra Tree and ANOVA have been applied on four classifiers Random Forest, Decision Tree, *k*-Nearest Neighbors and XGBoost for early detection of DDoS attacks on IoT devices. We use the CICDDoS2019 dataset containing comprehensive DDoS attacks to train and assess the proposed methodology in a cloud-based environment (Google Colab). Based on the experimental results, the proposed hybrid methodology provides superior performance with a feature reduction ratio of 82.5% by achieving 98.34% accuracy with ANOVA for XGBoost and helps in early detection of DDoS attacks on IoT devices.

Keywords DDoS · IoT · CICDDoS2019 dataset · *K*-NN · Random forest · Decision Tree · Chi-square · ANOVA

Abbreviations

DDoS	Distributed denial of service
DT	Decision tree
FNR	False-negative rate
FPR	False-positive rate
IDS	Intrusion detection system
IoT	Internet of Things
K-NN	*K*-Nearest neighbors
LDAP	Lightweight directory access protocol
LSTM	Long short-term memory
NaN	Not a number
NB	Naïve bayes
NetBIOS	Network basic input/output system
NTP	Network time protocol
RF	Random forest
TNR	True-negative rate
TPR	True-positive rate

✉ Vimal Gaur
vimalgaur@msit.in

[1] MMEC, Maharishi Markandeshwar Deemed to be University, Mullana Ambala, India

[2] Maharaja Surajmal Institute of Technology, Janakpuri, Delhi 110058, India

[3] Department of CSE, MMEC, Maharishi Markandeshwar (Deemed to be University), Mullana, Ambala 133207, India

1 Introduction

The Internet of Things (IoT) refers to a system of interrelated, internet-connected objects that are able to collect and share data over a wireless network without human intervention. It is inevitably and rapidly influencing our society. One cannot ignore the quantum of data that is being produced from different sources, e.g., web sources, digital media. The data produced might be in various structures like conventional, structured, heterogeneous, and streaming, which is quite an arduous task to manage [1]. According to a survey, one person can generate 1.7 megabytes in a second. With such a high rate of data generation, it has been analyzed that 90% of the data is unstructured as it may come in various forms and sizes [2]. According to a survey by International Data Corporation, the number of IoT devices could rise to 41.6 billion by 2025. As is evident from the data, the growth rate of connected devices is 127 devices per second [3–5]. This IoT explosion will provide hackers multiple prospects of opportunities. The larger the numbers of connected devices, the more likely botnets are to form

and result in more DDoS attacks [6]. These attacks behave passively as they don't steal any information but are flooded with packets that intercept the server. DDoS attacks can be broadly classified into: (i) Volume-based DDoS attacks, (ii) Protocol-based DDoS attacks, and (iii) Application layer DDoS attacks. Volume-based attacks depend on the volume of inbound traffic viz. UDP Floods, ICMP Floods**.** UDP Flood attack comes about when the attacker transmits a larger number of spoofed UDP packets to the server thereby consuming all the available bandwidth and bringing the server down. ICMP Flood Attack occurs when attackers send a sizeable amount of spoofed ICMP request packets to the server. ICMP packets exist in pairs as ICMP Echo request and ICMP Echo reply packets are sent to ensure device connectivity Protocol-Based DDoS Attacks belong to the transport layer and are also termed as layer 4 attacks. In SYN Flood offender sends TCP connection requests faster than the targeted machine can process them. The attacker exploits the resources of the server and makes it unresponsive. In Ping of Death, the victim machine is overwhelmed with packet size larger than it can handle. The application layer DDoS attacks slow down the server by sending a large number of spoofed HTTP GET requests to the server, as server treats these requests as authenticated requests and responds to all requests by sending HTTP POST. This means that a significantly larger amount of network resources are being allocated to the requester and thereby depleting the resources of the server and bringing it down [7].

This clearly gives an understanding that attackers are adopting innovative methods to exploit the system, therefore security of surroundings is paramount. The National Institute of Standards and Technology (NIST) reported that in 2017, American companies experienced losses of up to 65.5 billion dollars due to IT- related attacks and intrusions. Today, denial-of-service remains an immense threat to internet-dependent businesses and organizations, even after unwavering incessant efforts by security researchers and experts. Such attacks can cripple user' resources in a short span of time [8]. Over the years, the significance of security has also been a matter of great concern. One such type of attack is DDoS, which is most severe due to its impact. A large number of solutions are possible for detecting such attacks using machine learning and artificial intelligence. Intrusion detection and prevention systems, Firewalls and effective key management protocols are essential to achieve security of communication protocols and also to safeguard the support layer from unauthorized access. Many of these solutions generated a lot of alerts for threatening situations, which resulted in a false alarm rate [9, 10]. It is due to this fact, current research has shifted toward focusing on reducing false alarms and generating high detection rates. Security strategy should be maintained for IoT devices with the progress in number of technologies.

In addition, due care must be taken that incorporation of security parameters should not be delayed for later stages rather it should be considered in the early stages of strategy development of IoT devices. These days, for technology enhancements and IoT devices approaching billions, implementing security measures manually [11] has become a challenging task, so there is a need to automate this process. For automation of the operating environment of devices, the bluetooth firewalls must be properly enabled.

In addition, the severity of attack detection is of paramount importance for the smooth operation of IoT devices. In other words, IoT devices cannot operate without the support of legitimate preprocessing methods, neural network architectures and hyper-parameter optimizers [12, 13]. The above-mentioned issues are studied after analyzing the existing intrusion detection datasets, machine learning algorithms and optimizers to assess the datasets. This gives a clear understanding of the type of network traffic that helps the training model to properly label the training instances.

Although a commendable research has been done in identifying datasets, intrusion detection systems, DDoS attack types but the scope is limited so far. In our paper, we have tried to overcome all these issues. We proposed a hybrid methodology for early detection of DDoS attacks on IoT devices. The significant contributions of our paper are as follows:

a. The Train–Test Split is applied on the entire dataset, i.e., the 70% data in the dataset is used for the training and 30% is used for the testing. We have used CICDDos2019 Dataset which includes two types of attacks (exploitation-based and reflection-based), which is based on network flow features.
b. Machine learning classifiers are applied on this dataset and maximum value of performance parameters is recorded for full features.
c. Chi-square feature selection is applied on Machine learning Classifiers (RF, DT, KNN and XGBoost) as follows: First these classifiers are applied at an interval of '5,' a series of iterations are performed and the number of features is recorded for which maximum accuracy is achieved. To select an optimal window thereafter these classifiers are applied at an interval of '1,' e.g., when a series of iterations are performed at an interval of '5,' then maximum accuracy is achieved for 35 features. So, this implies in next step we have an optimal window to start iterations at an interval of '1' from 30 to 40 features and we will get the exact number of features for which accuracy is maximum.
d. The same set of iterations are performed for Extra Tree method and ANOVA.

e. XGBoost was found to give the highest accuracy with ANOVA feature selection of 98.34% with an 82.5% feature reduction ratio.

The remainder of the paper is structured as follows: In Sect. 2 we have identified the different methods used in detection of DDoS attacks. Furthermore, we have outlined the machine learning classifiers, feature selection methods and performance parameters in Sect. 3. Section 4 describes the environmental setup, dataset, data preprocessing steps to be used and thereafter methodology used is described. Results and discussions are described in Sect. 5. The conclusion as well as future directions are described in Sect. 6.

2 Literature Review

In this section, we discuss the methods used by various researchers to detect and compare DDoS attacks.

Salahuddin et al. [14] presented an anomaly detection system and found that most of the attacks have F1 scores above 99%. To analyze time windows on contrasting time-based features, Kitsune's feature extractor has been used. He further analyzed the complexity of the Auto-encoder by augmenting the number of parameters.

M.S.Elsayed et al. [15] proposed a new model.

DDoSNet applied on CICDDoS2019 with varying parameters (number of hidden layers, learning rates etc.). Hyperparameters tuning of the deep neural network (RNN with auto-encoder) in this model leads to superior performance parameters. Afterward, machine learning algorithms viz Decision Tree (DT), Naïve Bayes (NB), Booster, Random Forest (RF), SVM, and Logistic Regression (LR) were implemented and the proposed model gives the highest accuracy of 99% followed by LR and SVM.

Maranhao et al. [16] proposed a higher-order singular value decomposition feature extraction technique. Once the data are filtered through this technique, it has been forwarded to machine learning classifiers, namely Decision Tree (DT), Random Forest (RF) and Gradient Boosting. As a result, the new model (Gradient Boosting) outperforms other techniques with an accuracy value of 99.84%.

Filho et al. [17] proposed two methodologies for detection of distributed reflection denial-of-service (DrDOS) attacks in IoT on CICDDoS2019. In the first methodology, a hybrid IDS (Signature-based and Anomaly-based) have been proposed and in the second three deep learning models have been implemented to test accuracy values. The training accuracy and test accuracy come out to be 99.85% and 99.19%, respectively.

Shurman et al. [18] introduced two methodologies for detecting reflection-based attacks on IoT devices. In the first methodology, hybrid IDS and in the second Long Short Term Memory (LSTM) have been used. Hybrid IDS uses both signature-based and anomaly-based approaches. After incorporating the benefits of both approaches, three models of deep neural networks have been used for detecting attacks and final training accuracy comes out to be 99.85% and testing accuracy as 99.19%.

Jiabin Li et al. [19] realized a real-time volumetric detection scheme for DDoS attack detection in IoT using 1999 DARPA Intrusion Detection Evaluation Data Set, 2009 DARPA DDoS Dataset, and the CICDDoS2019 Evaluation Dataset. To fasten the process of finding the first true packet and evaluating the time performance of response, the author proposed the addition of temporal variables in calculating the rate of omitted positive samples. The less the value of this parameter (Temporal False Omission Rate), the more is the probability of finding attacking packets. This parameter value has been evaluated for all the datasets resulting in TFOR values less than 0.5% thereby identifying all volumetric DDoS attacks.

Yizhen Jia et al. [20] proposed a model for flow guard. In this model initially, flow filtration is done to select the most relevant features after that suspicious flow is gathered from the DDoS detection module to identify malicious data. Classification of flow can be done as flooding-based attacks, Benign Traffic, Slow Request/Response Attacks. A large dataset has been gathered by DDoS simulators BoNeSi and SlowHTTPTest and is combined with CICDDoS2019 Evaluation Dataset for evaluating the performance of the flow guard. Classification models LSTM, ID3, RF, NB, LR have been applied to simulate the flow behavior. LSTM has a high value of Precision, Recall, F1 Score as 99%.

Sharafaldin et al. [21] concluded that existing datasets are not capable of handling modern reflexive DDoS attacks, one of the most important reflexive DDoS attacks is the Network Time protocol. Networked devices/hosts receive time information from NTP protocol and the Time to Live field in IP header indicates the number of devices/hosts being used for generating request packets for sharing time-related information. The result of this attack is that it leads to an increased query to response ratio. Another reflexive attack is the Simple Service Discovery Protocol (SSDP), where the victim's infrastructure is affected by an increased amount of traffic which in turn makes web resources offline. The author included Network basic Input/output system (NetBIOS) protocol for monitoring all the processes and their respective sessions (TCP) since this is not supported by already existing datasets due to its unauthentic behavior and susceptibility to poison attacks. He further added datasets for UDP-Lag reflectors who hide the identity of attacker and the server is overloaded with fake requests (UDP Flooding) thereby not allowing a server to respond. The proposed dataset handles all the traffic between source and destination and also monitors heterogeneous data set. Further, on this new dataset, he

concluded ID3 (Iterative Dichotomiser generates a decision tree) as the best algorithm in terms of Accuracy, Specificity, F1 Measure and accuracy value comes out to be 78%.

Alsamiri and Alsubhi [22] segregated attack types and weights are assigned to these attack types since updating a network on regular basis leads to diversified attacks in IoT devices. Also, general common of attacks within a group is determined. The author divided the implementation into three phases. In the first phase, feature extraction is performed to shorten input data from 84 network traffic facilities to seven to get quick responses.

A comprehensive analysis of ten attack types and seven Machine Learning algorithms for calculating *F*-Measure is done. Results show that ID3 has the highest *F*-measure value as 1.00 for DOS_TCP and the best feature of attacks is also identified. The Machine learning algorithms are applied to the attacks with the best features as part of the second phase. Results suggest that Adaboost gives the best performance (Accuracy, Precision, Recall, F-Measure) but the time required to execute it is almost 60 times more time as compared to Naïve Bayes. Later, RF resistors have been applied for amplitude reduction to run at better times and this has significantly enhanced the performance of both RF and NB.

Gurulakshmi and Nesarani [23] demarcates between usual and weird traffic using SVM algorithm and speculates weird traffic. The underlying approach works in phases, in the very first phase XOIC tool is used to produce traffic from numerous sources to a single destination. Since Traffic is of DDoS type, traffic (which is monitored) is nothing but a sequence of packets having source address, destination address, packet count. Further, to analyze the real instances of a packet and to save these instances for the future, an open-source tool named Wireshark has been used for obtaining packets and Feature selection has been applied to the packet count field which limits the amount of space required for computation and accuracy is determined. Results show that less featured set KNN has the highest accuracy.

Meidan et al. [24] used light gradient boosting machine (LGBM), deep neural network (DNN) and support vector machine (SVM) on network traffic data obtained from various commercial IoT. LGBM performed best. Grid Search has been used for hyper-parameter tuning of the number of leaves and the maximum number of iterations. A threshold is developed for different values of NZFP, and a suitable NZFP value is chosen for FPR = 0, negative effect on TPR.

Wehbi et al. [25] concluded that the prime reason behind network and application disruption is DDoS attacks since they lead to greater delays in transmission of a packet. Three approaches have been used for DDoS detection using machine leaning. Feature extraction helps in differentiating normal traffic from anomalous traffic. The author stated in approach 1 that stateful and stateless feature helps in achieving this to a greater extent. Further analysis suggests that stateful features consider all the flow characteristics of network traffic thus providing a higher level of accuracy as compared to stateless. After the classification of the dataset is done, *K*-NN, LSVM, NN, DT, RF classifiers have been used and *K*-NN gives the highest accuracy (of identifying) delete of 0.99%. Classification performance decreases due to the loss of a large amount of data so attention must be paid to legitimate traffic. In approach 2 statistical classification can be achieved through SDN network and thereby applying QDA, LDA, SVM, NB, KNN, RF, DT classifiers and obtaining a negligible false-positive rate of 0.3%. ANN when used in approach 3 gives a good detection accuracy.

Hosseini and Azizi [26] visualizes the flow of data in a network using the KNIME framework and reduce bias that occurs as a result of redundant data in a dataset, two DDoS datasets-NSL-KDD dataset and the other collected from different sources. Several machine learning algorithms (NB, RF, DT, MLP, KNN) have been used and Random Forest turns out to give the best results. These algorithms work in the batch mode since the size of data instances is unknown as data arrives in a streaming mode.

Alkasassbeh et al. [27] created a new dataset to handle network layer and application-layer attacks. Over time every dataset must be upgraded and streamlined to mitigate threats to Confidentiality, Integrity and Availability of security services. The author concluded the above findings by focusing on new attack types as Smurf, UDP Flood, HTTP Flood, SIDDoS thereafter machine learning algorithms were applied to this newly collected dataset and MLP turns out to give the highest accuracy rate.

Wang et al. [28] set forth dynamic methods (Sequential Backward Selection-Multi Layer Perceptron (SBS-MLP), Sequential Forward Selection-Multi Layer Perceptron (SFS-MLP), Clamping Technique Sequential Backward Selection (CTSBS)) for detection of DDoS attacks on NSL-KDD dataset and finds SBS-MLP as best. MLP is a type of feedforward Artificial Neural Network (ANN) and uses supervised learning technique (Table 1).

3 Machine Learning Classifiers and Feature Selection Methods

Machine learning is used to give ability to machine to take decisions similar to humans, as machine learn from experience. Machine learning and artificial intelligence are related to each other as intelligence to machine is given in artificial intelligence (AI). In machine learning, system learns from data as no instruction is given to system. System does this by extracting patterns in data to make predictions and decisions. To increase performance of machine, we need several machine learning algorithms which acts as a mechanism to provide training to machine. Two main requirements in

Table 1 Methodology used by different researchers in detection of DDoS attacks

S.No.	Author	Year	Dataset	Classification algorithms	Accuracy rate	Limitation
1	Salahuddin et al. [14]	Nov. 2020	CICDDOS2019	Time-based anomaly detection (an auto-encoder for DDoS detection)	99% < *F*1 score < 100%	This approach didn't used any feature selection algorithm, the results have been obtained with full feature set
2	Elsayed et al. [15]	Nov. 2020	CICDDOS2019	A new model has been proposed	99%	This accuracy is achieved with full feature set
				Naïve bayes (NB), Decision tree (DT), Booster, random forest (RF), SVM, and Logistic regression (LR)	NB = 57 %, DT = 77%, Booster = 84%, RF = 86%, SVM = 93%, LR = 95%	These values are less than our classifier algorithm with complete features
3	Maranhao et al. [16]	Oct. 2020	CICDDOS2019	Higher-order singular value decomposition technique for denoising	DT = 97.54% GB = 99.87% RF = 99.95%	This approach results in good denoising performance and didn't use any feature selection algorithm
				Decision tree, Random forest and Gradient boosting		
			CICIDS2017	Decision tree, Random forest and Gradient boosting	99.96% (RF)	
4	Filho et al. [17]	June 2020	CIC-DoS, CICIDS2017, CSE-CIC-IDS2018, Customized dataset	RFECV feature selection is used with RF, DT, LR, SGD, perceptron, AdaBoost. accuracy of MLA along with number of features is shown	RF = 99.6% (28), DT = 99.41% (25), LR = 97.23% (26), SGD = 96.94% (16), Perceptron = 93.72% (28), AdaBoost = 93.11% (7)	Very few DDoS attacks are taken into consideration. Mainly focus on DoS attacks
5	Shurman et al. [18]	Jan. 2020	CICDDoS2019	Three LSTM models have been proposed each with different layers	First model-train accuracy-92.05%, test accuracy-91.54% Second model-train accuracy-97.27%, test accuracy-96.74% Third model-train accuracy-99.85% Test accuracy-99.19%	Deep learning model have been applied on full feature set
6	Jiabin Li et al. [19]	Feb. 2020	1999 DARPA intrusion detection evaluation data set 2009 DARPA DDoS dataset CICDDoS2019 evaluation dataset	Designed a new indicator Temporal False Omission Rate (TFOR) for evaluating time performance of response	TPR = 100%, FPR < 3%, average temporal false omission rate = 0.3447%	This approach didn't used any feature selection algorithm, the results have been obtained with full feature set
7	Yizhen Jia et al. [20]	May 2020	CICDDoS2019	Proposed LSTM model and proposed CNN model	LSTM Model = 98.9%, CNN model = 99.9%	This accuracy is achieved with full feature set
8	Iman Sharafaldin et al. [21]	Oct. 2019	CICDDoS2019	ID3, RF, NB, LR	ID3 = 78%, RF = 77%, NB = 41% LR = 25% are precision values	All these values are with full features

Table 1 (continued)

S.No.	Author	Year	Dataset	Classification algorithms	Accuracy rate	Limitation
9	Alsamiri and Alsubhi [22]	Jan. 2019	Bot-IoT	Random forest regressor is used for feature selection on MLA	KNN = 99%, ID3 = 97%, RF = 97%, AdaBoost = 97%, QDA = 87%, MLP = 84%, NB = 79% All these results are obtained with 7 features	This is performed with older dataset, not covering all the types of DDoS attacks
10	Gurulakshmi and Nesarani [23]	May 2018	Packet capturing is done using Wireshark tool	SVM and KNN	92% (KNN)	Filtered packets are based on denial-of-service attacks. DDoS attacks are not taken into account
11	Meidan et al. [24]	Oct. 2020	Data has been collected from dedicated network (includes IoT and non-IoT devices)	LGBM, DNN and SVM. Hyperparameter tuning has been done	99.3% (LGBM)	Network traffic data has been gathered from commercial IoT devices
12	Wehbi et al. [25]	April 2019	Cleveland dataset on heart disease	NB, DT, ada boosting and multilayer feed forward neural network	59.73% (Ada Boost)	This dataset is based on diagnosis of heart disease
13	Hosseini and Azizi [26]	July 2019	NSL-KDD sataset	NB, RF, DT, MLP, KNN	98.8% (MLP)	All types of DDoS attacks are not considered in this dataset. Dataset is not based on network traffic analysis
			Dataset collected by NS2 simulator	NB, RF, DT, MLP, KNN	98.63% (MLP)	
14	Alkasassbeh et al. [27]	Feb. 2016	Dataset collected from NS2 simulator	RF, NB, MLP	98.63% (MLP)	All types of DDoS attacks are not considered in this dataset. Dataset is not based on network traffic analysis
15	Wang et al. [28]	Dec. 2020	NSL-KDD dataset	SBS-MLP, SFS-MLP, CTSBS	97.66% (SBS-MLP)	SBS-MLP achieves this accuracy with 31 features and 94.88% recall

training models using these algorithms are as follows: Training data and Testing data. In training step, classifier models are produced using various algorithms and in testing step models are evaluated for accuracy and other parameters. Machine learning algorithms when integrated with Feature selection algorithms reduces training time, increases system performance. Machine learning classifiers and feature selection methods that has been used in our paper are described below.

3.1 Machine Learning Classifiers

3.1.1 Decision Tree

It classifies instances by starting at the root node of the tree and moving until the leaf of the tree. It has four attributes, namely decision node, leaf node, branch, path. Apart from discussing patterns for consistent data set, they can also deal with inconsistent data since all the objects in a class have equal values of conditional probability [29]. Different decision tree algorithms (ID 3, C 4.5, CART) exist for handling categorical and numerical attributes. Further, CART can be used to perform regression and classification tasks. Different costs can be imposed when constructing a classification model using DT. Researchers have their own opinion for maximizing accuracy and minimizing costs [30] (Fig. 1).

3.1.2 Random Forest

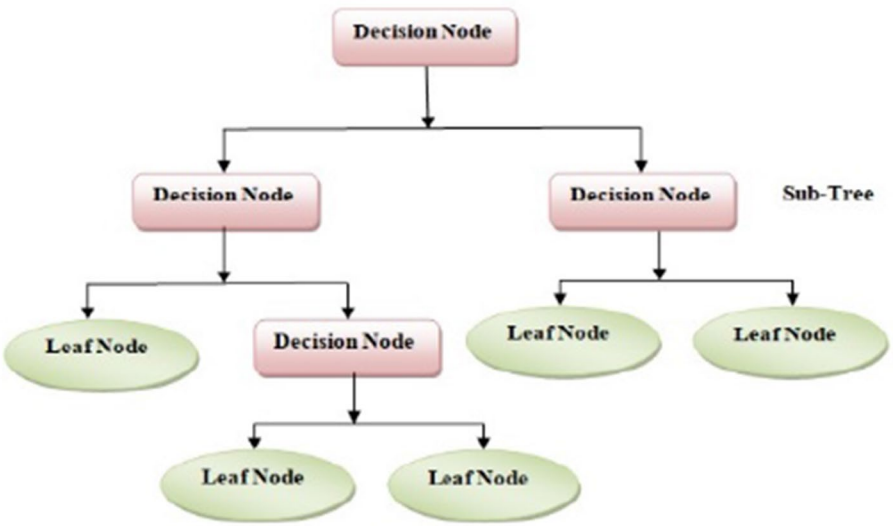
Random forest is a supervised learning method having an implicit feature selection mechanism that uses embedded methods (both filter and wrapper). The decision on the number of features that can be tested at any node of the tree facilitates feature selection. This choice of features at any node along with the number of trees in a forest make this ensemble learning a good choice for a particular dataset [31]. Learning using RF begins by selecting some set of trees and iteratively adding few more trees until a threshold value is reached. It is this threshold value that acts as a stopping criterion for further addition of new trees. Also, these trees work independently and collective output is obtained after combining all results. RF effectively handles missing values and also classifies multi-labels. Further, dynamic techniques have been proposed to remove skewness in the dataset which ultimately leads to class imbalance and an increase in the limitedness of classification techniques [32] (Fig. 2).

3.1.3 KNN

The major challenge these days is the handling of large datasets, KNN [33] uses prediction methodology for finding k-nearest neighbors (KNN) by calculating distance using an enhanced distance algorithm. We will see with the following steps how it works-:

1. Initially load all the training and training data.
2. Select the nearest data point by selecting the appropriate integer value for *k*.
3. For every data point in the test data perform the following steps-:
   a. Using any distance method find the distance between test data and each row of training data.
   b. Sort these distances in ascending order.
   c. From this sorted array, chose top k rows.
   d. Assign a class to the test point, after choosing a frequent class of rows.

Fig. 1 Decision tree



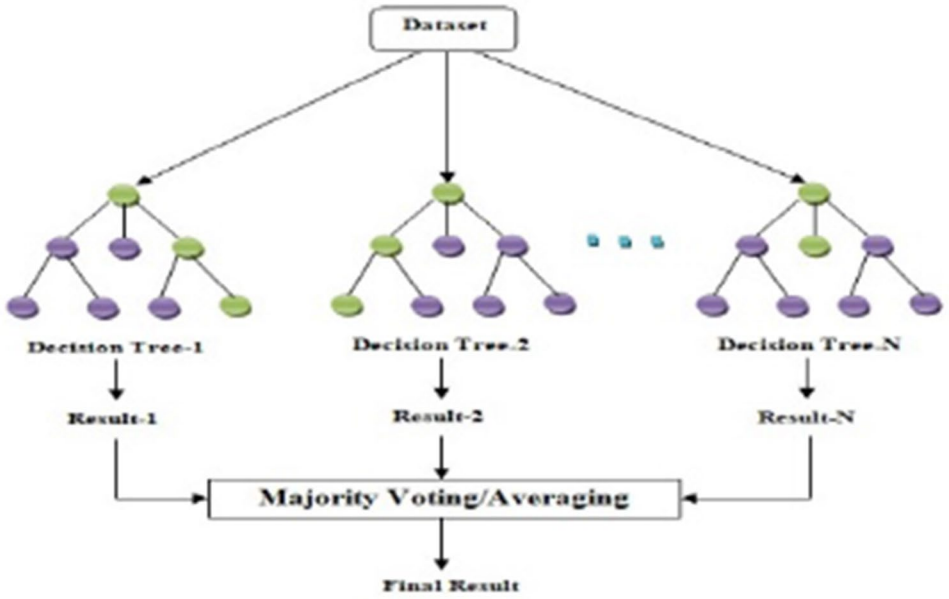


Fig. 2 Random forest

3.1.4 XGBoost (eXtreme Gradient Boosting)

XGBoost is a tree-based algorithm based on sequential ensembling. It is capable of dealing with issues like accuracy, losses, variances that occur in the results. A fault prediction scheme has been proposed [34] for detecting features that are the root cause of a fault and thereby achieving high accuracy and low positive rate. A sequence of steps involved in XGBoost are-:

1. Initially a Decision tree is formed.
2. Then, Bagging is performed to make predictions by analyzing the majority decision.
3. Forest is constructed using Random Forest since trees are independent.
4. Boosting is done to identify losses.
5. Gradient boosting is done to overfit a dataset quickly.
6. Parallel tree boosting (GBDT, GBM) is provided by XGBoost.

3.2 Feature Selection Methods

Categorization of feature selection methods can be done as Filter method, Wrapper method, and Embedded method [35]. The filter method selects features according to a threshold value and criteria decided by the user. Feature importance can be calculated by treating features individually and also by evaluating the entire feature space. The filter method is capable of accepting continuous and categorical features and can generate categorical and continuous responses. The main drawback of filter methods is that a strong correlation exists between independent variables. This independence between variables must be taken into consideration before training any model. A series of an example of wrapper methods exists for dealing with these issues thereby giving high accuracy. The process of the wrapper method includes searching for a subset of features, building a machine learning model, evaluating model performance. The above process is repeated until the desired condition is met. This process of repetition makes these wrapper methods very expensive. To cope up with the disadvantages of the filter method and wrapper method, an embedded approach is proposed. This purpose is achieved by applying various combinations of filter and wrapper methods thereby increasing accuracy.

3.2.1 Feature Selection Using Chi-square Algorithm

It is a filter method for evaluating features using statistical methods thereby making the model less prone to overfitting. The input variable and response variable are categorical (Nominal, Ordinal, Boolean) in nature. It is used for observing the dependency between features and response. More is the chi-square value, more is the dependence of features on response [36].

3.2.2 Feature Selection Using Extra Tree Classifier Algorithm

Extra Randomized Trees employ bagging and random subspace, so a random value is chosen for top-down splitting. The bias–variance trade-off is very well explained with extremely randomized trees. An increase invariance can be achieved by increasing the number of trees in this ensemble method. When the variables are inappropriate Extra Trees are expected to work faster as compared to random Forest. But with high dimensional datasets, Extra- Tree becomes worse [37].

3.2.3 Feature Selection Using ANOVA (Analysis of Variance) Algorithm

ANOVA (Analysis of variance) is a filter method used for evaluating the performance of features when data is normally distributed [38]. This concludes that feature importance plays an important role in identifying performance improvement. In it, input variables are numerical (integer and float) and response variables are categorical in nature. The features which are performing well in isolation will not necessarily give good classification results. ANOVA has its applications in many areas viz engineering, commercial, medicine, bioinformatics, text mining.

When machine learning is used some sort of knowledge base is required, which is then used in the detection process using any detection model and finally feedback is provided through some appropriate feedback mechanism. For this two sets of data are required: training and testing. In the training step, evaluation of different classification models are done using an understanding of the various algorithms and in the testing step performance measures like Accuracy, Precision, True-Positive Rate, True-Negative Rate, False-Positive Rate, False-Negative Rate, Specificity of these classification models have been evaluated to check which performs best. Based upon values of performance measures, decision can be taken as to which classification model is to be chosen. A good algorithm should have a high value of true positive but a low value of false positive and false negative.

3.3 Performance Parameters

Various performance parameters viz. Accuracy, Precision, Recall, Specificity, False-Positive Rate, False-Negative Rate and F1 Score, are used to evaluate the performance of classifiers are discussed in this section. Before discussing these parameters we will discuss True-Positive, True-Negative, False-Positive, and False-Negative. True-Positive (TP) is when a positive instance is predicted as positive. False-Positive (FP) is when a negative instance is predicted as positive. True-Negative (TN) is when a negative instance is predicted as negative. False Negative (FN) **is** when a positive instance is predicted as negative.

3.3.1 Accuracy

Accuracy can be measured as the correct predictions out of total predictions performed.

$$\text{Accuracy} = \frac{\text{True positives} + \text{True Negatives}}{\text{True Positives} + \text{False positives} + \text{False Negatives} + \text{True Negatives}} \quad (1)$$

3.3.2 Precision

Precision measures the ability of a system to produce only relevant results.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (2)$$

3.3.3 Recall or Sensitivity or True-Positive Rate (TPR)

Recall measures the ability of a system to produce all relevant results.

$$\text{TPR} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (3)$$

3.3.4 Specificity or True-Negative Rate (TNR)

Specificity measures the ability of a system to make correct negative predictions.

$$\text{TNR} = \frac{\text{True Negatives}}{\text{True Negatives} + \text{False Positives}} \quad (4)$$

3.3.5 False-Positive Rate (FPR)

False-Positive Rate measures the probability of a system to raise false alarm.

$$\text{FPR} = \frac{\text{False Postives}}{\text{False Positives} + \text{True Negatives}} \quad (5)$$

3.3.6 False-Negative Rate (FNR) or Miss Rate

False-Negative Rate measures the probability of a system to miss true positives.

$$\text{FNR} = \frac{\text{False Negatives}}{\text{False Negatives} + \text{True Positives}} \quad (6)$$

3.3.7 F1 Scores or F-Measure

F-Measure calculates the harmonic mean of precision and recall.

$$F1\ \text{Score} = 2 * \frac{(\text{Recall} * \text{Precision})}{\text{Recall} + \text{Precision}} \quad (7)$$

4 Experiments

This section demonstrates the hybrid methodology adopted in this paper and is fractionated into four subsections. First, section illustrates the environmental setup used in implementation. Next, a brief description of the dataset used throughout the paper is presented. Next section presents the preprocessing steps to handle missing values and outliers. Methodology to be used is presented in last section. The suggested methodology performance results were analyzed using statistical metrics as discussed in last section.

4.1 Environmental Setup

We carried out our analysis on a cloud-based environment called Google Colab. Google Colaboratory is a free online cloud-based Jupyter notebook environment. This environment helps to train our machine learning models on CPUs, GPUs, and TPUs. Colab is a hosted Jupyter notebook service that provides free access to computing resources with no additional requirement for setup. The virtual machine on the google colab run on Intel(R) Xeon(R) CPU @ 2.30 GHz with a CPU speed of 2299.998 MHZ and a cache size of 46,080 KB. It provided a Ram of size 12.72 GB and a disk size of 107.77 GB. Our paper is implemented using Python 3.7.10. The Scikit learn 0.22.2 has been used for the implantation of machine learning algorithms and Tensorflow 2.4.1 has been used for the implementation of neural networks.

4.2 Dataset

In this paper, we obtained dataset provided by the Canadian Institute for Cybersecurity (CIC) for DDoS evaluation, namely, CICDDoS2019 [21]. This dataset is publicly available (PCAP and CSV files), and is completely labeled dataset with millions of labeled instances (legitimate and DDoS attack). The labels for Training set are mentioned as: BENIGN, DrDoS_LDAP, DrDoS_MSSQL, DrDoS_NetBIOS, DrDoS_UDP, Syn, UDP_Lag, and WebDDoS. Testing set labels are mentioned as: BENIGN, LDAP, MSSQL, NetBIOS, UDP, Syn, UDP_Lag. Dataset is extracted using CICFlowMeter tools and categorized into training day and testing day. The attacks along with the time of their occurrence in these two days are listed below in Table 2. Both datasets contain 86 columns and one column for categorical data which is used to divide the remaining data into groups.

Table 2 DDoS attack types and timings

Days	Attacks	Attack times
First day (testing day)	PortMap	9:43–9:51
	NetBIOS	10:00–10:09
	LDAP	10:21–10:30
	MSSQL	10:33–10:42
	UDP	10:53–11:03
	UDP-Lag	11:14–11:24
	SYN	11:28–17:35
Second day (training day)	NTP	10:35–10:45
	DNS	10:52–11:05
	LDAP	11:22–11:32
	MSSQL	11:36–11:45
	NetBIOS	11:50–12:00
	SNMP	12:12–12:23
	SSDP	12:27–12:37
	UDP	12:45–13:09
	UDP-Lag	13:11–13:15
	WebDDoS	13:18–13:29
	SYN	13:29–13:34
	TFTP	13:35–17:15

From this dataset, we used only 79 features, eliminating circumstantial features such as SourceIP, Source Port, DestinationIP, Destination Port, Protocol, FlowID and Timestamp. Features of training day and testing day are listed below- (Table 3):

4.3 Data Preprocessing

The dataset consisted of one file for each attack category. Seven attack categories are in testing day and twelve in training day. The feature set for each attack category comprises of various features about network state. The data for all the attack categories are combined as the feature set is similar for each of them and the resultant dataset is randomized. The data are transformed for training after Exploratory Data Analysis and outliers and missing values treatment.

4.4 Methodology

4.4.1 Chi-square Feature Selection

Chi-square does not work for negative values, so we replaced these values with NaN or Null values. Next, these values have been replaced with the mean of each column (Fig. 3). Also, it has been found that no records are available for WebDDoS attacks in the test dataset. So we dropped all the records of the WebDDoS attack from the training dataset which reduced the total numbers of records. Subsequently, a Standard Scaler was used to fetch all the values in the

Table 3 Features of training day and testing day dataset

S. No.	Features	S. No.	Features 2
1	FlowID	44	BwdPackets/s
2	SourceIP	45	MinPacketLength
3	SourcePort	46	MaxPacketLength
4	DestinationIP	47	PacketLengthMean
5	DestinationPort	48	PacketLengthStd
6	Protocol	49	PacketLengthVariance
7	Timestamp	50	FINFlagCount
8	FlowDuration	51	SYNFlagCount
9	TotalFwdPackets	52	RSTFlagCount
10	TotalBackwardPackets	53	PSHFlagCount
11	TotalLengthofFwdPackets	54	ACKFlagCount
12	TotalLengthofBwdPackets	55	URGFlagCountT
13	FwdPacketLengthMax	56	CWEFlagCount
14	FwdPacketLengthMin	57	ECEFlagCount
15	FwdPacketLengthMean	58	Down/UpRatio
16	FwdPacketLengthStd	59	AveragePacketSize
17	BwdPacketLengthMax	60	AvgFwdSegmentSize
18	BwdPacketLengthMin	61	AvgBwdSegmentSize
19	BwdPacketLengthMean	62	FwdHeaderLength.1
20	BwdPacketLengthStd	63	FwdAvgBytes/Bulk
21	FlowBytes/s	64	FwdAvgPackets/Bulk
22	FlowPackets/s	65	FwdAvgBulkRate
23	FlowIATMean	66	BwdAvgBytes/Bulk
24	FlowIATStd	67	BwdAvgPackets/Bulk
25	FlowIATMax	68	BwdAvgBulkRate
26	FlowIATMin	69	SubflowFwdPackets
27	FwdIATTotal	70	SubflowFwdBytes
28	FwdIATMean	71	SubflowBwdPackets
29	FwdIATStd	72	SubflowBwdBytes
30	FwdIATMax	73	Init_Win_bytes_forward
31	FwdIATMin	74	Init_Win_bytes_backward
32	BwdIATTotal	75	act_data_pkt_fwd
33	BwdIATMean	76	min_seg_size_forward
34	BwdIATStd	77	ActiveMean
35	BwdIATMax	78	ActiveStd
36	BwdIATMin	79	ActiveMax
37	FwdPSHFlags	80	ActiveMin
38	BwdPSHFlags	81	IdleMean
39	FwdURGFlags	82	IdleStd
40	BwdURGFlags	83	IdleMax
41	FwdHeaderLength	84	IdleMin
42	BwdHeaderLength	85	SimillarHTTP
43	FwdPackets/s	86	Inbound

dataset on the same scale before training the model. The same data preprocessing steps were applied to the test dataset. The parameters to be used by each classifier has been listed below:

The parameters for the classifiers in the proposed hybrid methodology are listed below (Table 4):-

As shown in the table below accuracy of Random Forest, Decision Tree, KNN, XGBoost have been calculated for series of features with an interval of '5' in between viz 5, 10, 15,…,0.60 (Table 5).

After performing iterations using chi-square, it has been analyzed that RF gives maximum accuracy of 91.55% with 35 features, Decision Tree gives 91.53% with 40 features, KNN gives 90.42% with 55 features and 90.41% with 40 features. Finally, XGBoost gives 92.60% with 50 features.

Various machine learning algorithms are used to analyze the data with a varied number of features. In order to check the optimal number of features required to achieve the highest accuracy without compromising the response time, various features are selected in incremental numbers with an interval of '5' between consecutive features. These features are selected using the chi-square feature selection algorithm. The reduction in the number of features using RF, DT, KNN and XGBoost is 59%, 53.5%, 53.5% and 42%, respectively. The aforementioned findings are represented below using Fig. 4.

To evaluate the performance, extensive experiments have been conducted and an optimal window is selected. After checking accuracy at an interval of '5,' it has been analyzed that maximum accuracy for XGBoost is achieved with 50 features.

The maximum accuracy for chi-square is obtained for 35 features. To optimize the accuracy further, iterations are run within the range of 30 to 40 features, starting with 35 and gradually increasing and decreasing the count of features by an interval of '1' viz 34, 33, 32, etc. This means an optimal window is selected to find the exact number of features, thereafter these features are varied according to this window and a classifier algorithm is applied on these features and accuracy is calculated and stored. When the same procedure is applied to RF, DT, KNN and XGBoost, the results are the same. As a conclusion, XGBoost integrated with chi-square gives 92.6% accuracy. The feature scores obtained as a result of applying chi-square are listed below in Table 6.

The top 20 features obtained using chi-square feature selection are shown below in Fig. 5.

4.4.2 Extra Tree Feature Selection

Like the different algorithms checked for accuracy by using the chi-square feature selection method, the same process is repeated for the other feature selection methods like

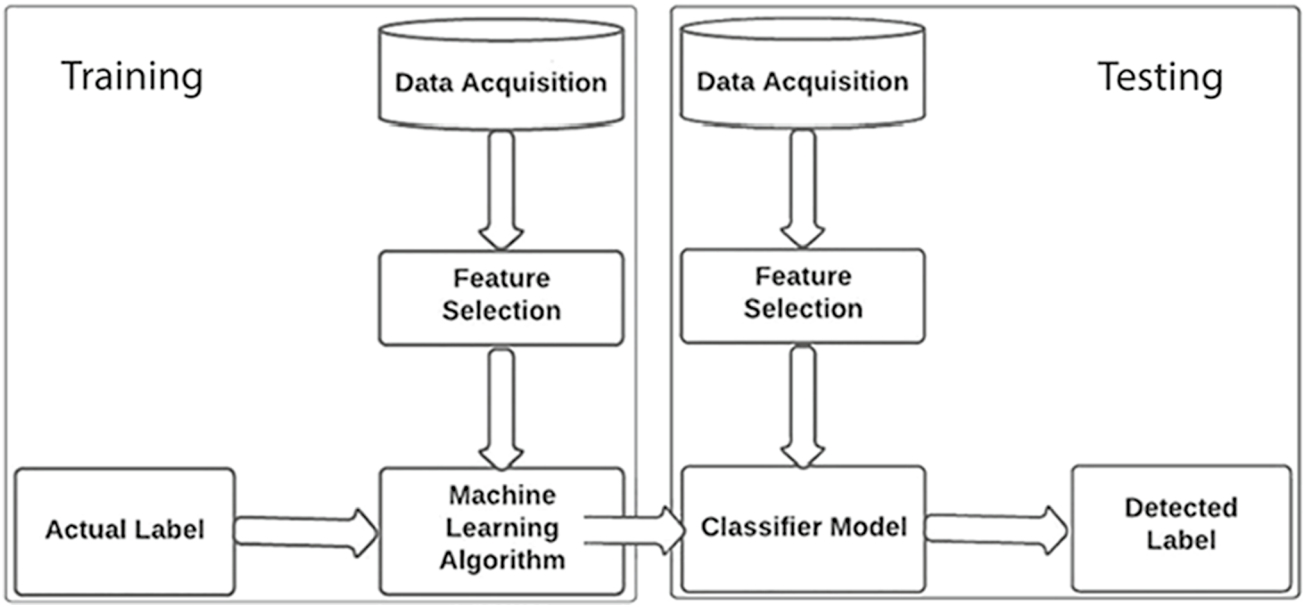


Fig. 3 Hybrid methodology for DDoS attack detection

Table 4 Parameters of proposed hybrid methodology

Method	Parameter	Setting	Method	Parameter	Setting
Random forest	n_estimator	100	XGBoost	max_depth	3
	criterion	Gini		learning_rate	0.1
	max_depth	3		n_estimators	100
	class_weight	Balanced		Silent	True
	Bootstrap	True		Objective	binary:logistic
	random_state	2		Booster	gbtree
	warm_start	True		n_jobs	1
Decision tree	Criterion	Gini		min_child_weight	1
	Splitter	Best		reg_lambda	1
	min_samples_split	2		scale_pos_weight	1
	min_samples_leaf	1		base_score	0.5
KNN	n_neighbors	5		subsample	1
	Weights	Uniform		colsample_bytree	1
	Algorithm	Auto			
	leaf_size	30			
	Metric	Minkowski			

Table 5 Accuracy of classifiers on applying chi-square feature selection algorithm with an interval of '5' in between

No. of features	Random forest	Decision tree	KNN	XGBoost	No. of features	Random forest	Decision tree	KNN	XGBoost
5	0.895919308	0.895581805	0.888424101	0.901233526	35	0.915526962	0.911521847	0.903849112	0.913049802
10	0.899454207	0.886929104	0.89303785	0.907090908	40	0.912503131	0.915379849	0.90410487	0.913905844
15	0.896934861	0.884754284	0.885720187	0.904481036	45	0.886037028	0.918379849	0.904196714	0.914161602
20	0.896945408	0.876171894	0.882688885	0.901876876	50	0.89564904	0.784230463	0.904196714	0.92604775
25	0.897026266	0.891494074	0.883684671	0.907330846	55	0.904152769	0.907005216	0.904251205	0.925708937
30	0.900990952	0.893246147	0.885004329	0.890269337	60	0.909756591	0.903339793	0.904251205	0.926065961

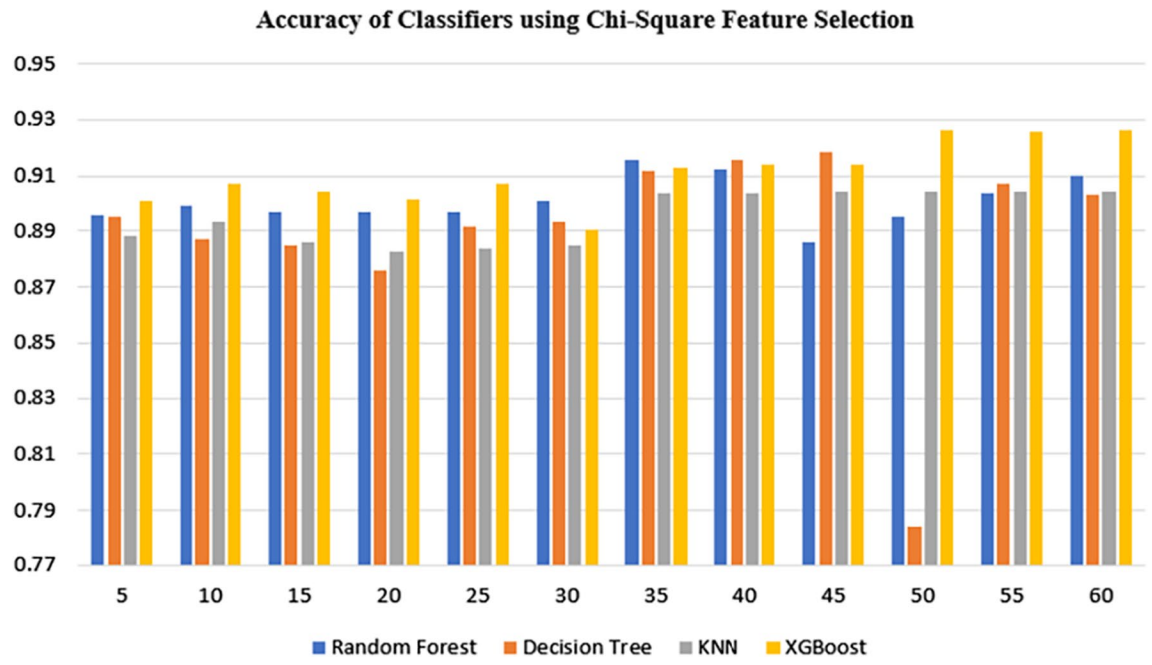


Fig. 4 Accuracy of RF, DT, KNN, XGBoost using chi-square feature selection

Extra Tree and ANOVA. Starting below with Extra Tree (Table 7):-

Features are selected using the Extra Tree algorithm and it has been analyzed that maximum accuracy achieved with RF is 91.79%, with DT 91.78%, with KNN 91.44%, with XGBoost 92.78% when the number of features is 45, 15, 40 and 30 respectively. Thus it can be concluded that using the Extra Tree Classifier there is a nearly 47.7% reduction in the number of features with the Random Forest algorithm (Table 8). Similarly, there is an 82.56%, 53.5% and 65% reduction in the number of features with Decision Tree, KNN and XGBoost respectively. Once results with an interval of '5' in between have been showcased, the accuracy of all the classifiers with Extra Tree feature selection algorithm on selected features with an interval of '1' in between have been calculated and results show the same values as above (Figs. 6, 7). This fine-grained analysis of the features showcased that XGBoost achieves the highest accuracy of 92.78% with 30 features. Below is mentioned the feature importance scores using the Extra Tree classifier in Table 6.

4.4.3 ANOVA Feature Selection

After performing series of iterations using chi-square and Extra Tree, the next step is to couple ANOVA with all the machine learning classifiers and results obtained are shown below (Table 9):

The above results can be more clearly shown using the following Fig. 8.

Feature are selected using ANOVA algorithm and it has been analyzed that maximum accuracy achieved with RF is 91.91%, with DT 91.39%, with KNN 91.37%, with XGBoost 98.347% when the number of features is 55, 35, 15 and 15, respectively. From this, a reduction in the percentage of features can be calculated viz 36% for RF, 59.3% for DT, 82.56% for KNN and 82.56% for XGBoost. So, it can be clearly shown that XGBoost gives the highest accuracy with ANOVA of about 98.34%. When the above procedure is repeated with an interval of '1,' these classifiers give the same accuracy. The scores obtained can be shown in Table 10.

Even after performing a series of iterations with respect to peak value(s) of different classifier(s) using feature selection algorithm(s) like chi-square, Extra Tree and ANOVA it has been found that XGBoost ranked at 1st position among all in terms of all performance parameters.

Ultimately, it can be concluded that when XGBoost is integrated with ANOVA, it performs best in terms of accuracy among all the other classifiers being used in this study (Fig. 9).

5 Results and Discussions

The proposed hybrid methodology has been evaluated utilizing the aforementioned dataset, environmental setup, and performance parameters. Results show that XGBoost achieves maximum accuracy with all the feature selection algorithms. XGBoost achieves an accuracy of 92.67% with

Table 6 Feature scores obtained on applying chi-square feature selection

Feature name	Score	Feature name	Score
FwdIATTotal	1.05404E+14	Init_Win_bytes_backward	4,503,621,415
FlowDuration	1.0537E+14	TotalLengthofBwdPackets	3,798,714,311
FwdHeaderLength	7.33991E+13	BwdPacketLengthMax	405,880,844.5
FwdHeaderLength.1	7.33991E+13	AveragePacketSize	395,389,509.7
IdleMax	2.88093E+13	MinPacketLength	236,104,745.3
FwdIATMax	2.81079E+13	FwdPacketLengthMin	235,663,971.6
FlowIATMax	2.80925E+13	PacketLengthMean	222,105,013.1
BwdHeaderLength	2.24785E+13	FwdPacketLengthMean	220,548,660.4
IdleMean	1.94039E+13	AvgFwdSegmentSize	220,548,660.4
min_seg_size_forward	1.62776E+13	FwdPacketLengthMax	203,683,750.8
IdleMin	1.36408E+13	MaxPacketLength	201,285,123.3
FwdIATStd	1.08069E+13	BwdPacketLengthMean	132,250,405.7
FlowIATStd	1.05294E+13	AvgBwdSegmentSize	132,250,405.7
FwdIATMean	8.34179E+12	BwdPacketLengthStd	109,926,046.3
BwdIATTotal	8.33243E+12	TotalFwdPackets	85,534,210.17
FlowIATMean	7.63298E+12	SubflowFwdPackets	85,534,210.17
BwdIATMax	7.50676E+12	BwdPacketLengthMin	49,424,968.79
IdleStd	7.07762E+12	PacketLengthStd	16,652,554.54
BwdIATStd	4.11175E+12	FwdPacketLengthStd	15,523,384.56
BwdIATMean	2.40222E+12	TotalBackwardPackets	2,936,373.285
FwdPackets/s	1.31244E+12	SubflowBwdPackets	2,936,373.285
ActiveMax	1.29511E+11	BwdIATMin	2,874,809.424
ActiveStd	58,684,045,271	act_data_pkt_fwd	1,911,365.036
ActiveMin	37,179,401,935	ACKFlagCount	1,312,346.786
ActiveMean	35,814,808,965	Down/UpRatio	725,494.5269
FlowIATMin	29,576,874,975	URGFlagCount	540,413.6588
PacketLengthVariance	27,278,421,280	CWEFlagCount	250,819.4381
FwdIATMin	26,470,247,143	FwdPSHFlags	206,903.7792
Init_Win_bytes_forward	7,763,650,061	RSTFlagCount	206,903.7792
SubflowBwdBytes	3,798,714,311	Inbound	7834.999233
BwdPackets/s	2,779,140,585	SYNFlagCount	1505.562997
TotalLengthofFwdPackets	575,112,219.2		

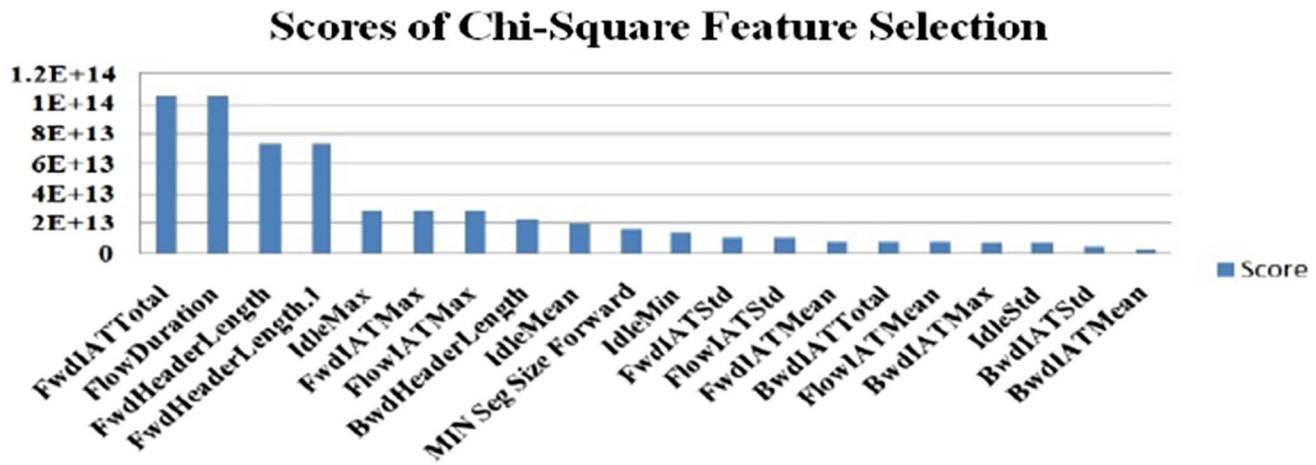


Fig. 5 Top 20 features using chi-square feature selection

Table 7 Accuracy of classifiers with extra tree feature selection algorithm on selected features with an interval of '5' in between

No. of features	Random forest	Decision tree	KNN	XGBoost	No. of features	Random forest	Decision tree	KNN	XGBoost
5	0.7512364	0.824703	0.89581647	0.751279448	35	0.9000839	0.9129922	0.913710291	0.927828827
10	0.8133302	0.8515405	0.912267588	0.909228376	40	0.9108851	0.913005	0.914477564	0.927853875
15	0.8682588	0.9178156	0.914057893	0.926700328	45	0.9179958	0.9131843	0.904188804	0.9278341
20	0.8729112	0.8989976	0.914129522	0.927071221	50	0.9158957	0.9139964	0.903218067	0.92786574
25	0.8849661	0.8997003	0.913728308	0.927636349	55	0.9180063	0.913865	0.904610672	0.927911443
30	0.8955497	0.9018668	0.913942318	0.927895623	60	0.9146489	0.9091559	0.904615946	0.927263699

Table 8 Features importance scores using extra tree classifier

Features	Score	Features	Score
MinPacketLength	0.118972635	FlowIATMin	0.001799
FwdPackets/s	0.10258459	FwdIATMin	0.001698
FlowDuration	0.090207574	URGFlagCount	0.001075
FwdIATTotal	0.07554805	IdleStd	0.000789
FwdPacketLengthMin	0.069687387	BwdPackets/s	0.000751
FwdPacketLengthMax	0.060570344	CWEFlagCount	0.000549
AvgFwdSegmentSize	0.058871874	Down/UpRatio	0.00052
MaxPacketLength	0.05799913	BwdPacketLengthMax	0.000512
PacketLengthMean	0.045163164	BwdIATMax	0.000482
FlowIATStd	0.045119961	BwdIATMean	0.000461
AveragePacketSize	0.038586145	Init_Win_bytes_backward	0.000417
PacketLengthStd	0.030278608	BwdHeaderLength	0.000362
FwdPacketLengthMean	0.029100966	FwdPSHFlags	0.00036
FwdHeaderLength.1	0.016190193	SubflowBwdPackets	0.0003
PacketLengthVariance	0.012958047	TotalBackwardPackets	0.0003
TotalFwdPackets	0.011696497	RSTFlagCount	0.000256
ACKFlagCount	0.011624844	BwdPacketLengthStd	0.000222
min_seg_size_forward	0.01089065	BwdIATMin	0.000215
FwdHeaderLength	0.01053693	AvgBwdSegmentSize	0.000156
Init_Win_bytes_forward	0.009126009	ActiveMax	0.000135
FlowIATMean	0.008937864	SubflowBwdBytes	0.000104
FwdIATStd	0.008576617	ActiveMean	0.000101
SubflowFwdPackets	0.008180575	ActiveMin	9.38E–05
FlowIATMax	0.00806697	BwdPacketLengthMin	7.7E–05
FwdIATMax	0.005969064	SYNFlagCount	6.03E–05
Inbound	0.005962699	TotalLengthofBwdPackets	5.99E–05
FwdIATMean	0.005867086	ActiveStd	5.34E–05
BwdIATTotal	0.005317947	BwdPacketLengthMean	5.23E–05
TotalLengthofFwdPackets	0.004848316	BwdPSHFlags	0
FwdPacketLengthStd	0.004057587	FwdURGFlags	0
act_data_pkt_fwd	0.00380867	BwdURGFlags	0
IdleMin	0.003276722	FINFlagCount	0
BwdIATStd	0.00270199	PSHFlagCount	0
SubflowFwdBytes	0.002389513	ECEFlagCount	0
IdleMean	0.002223116	FwdAvgBytes/Bulk	0
IdleMax	0.00214157		

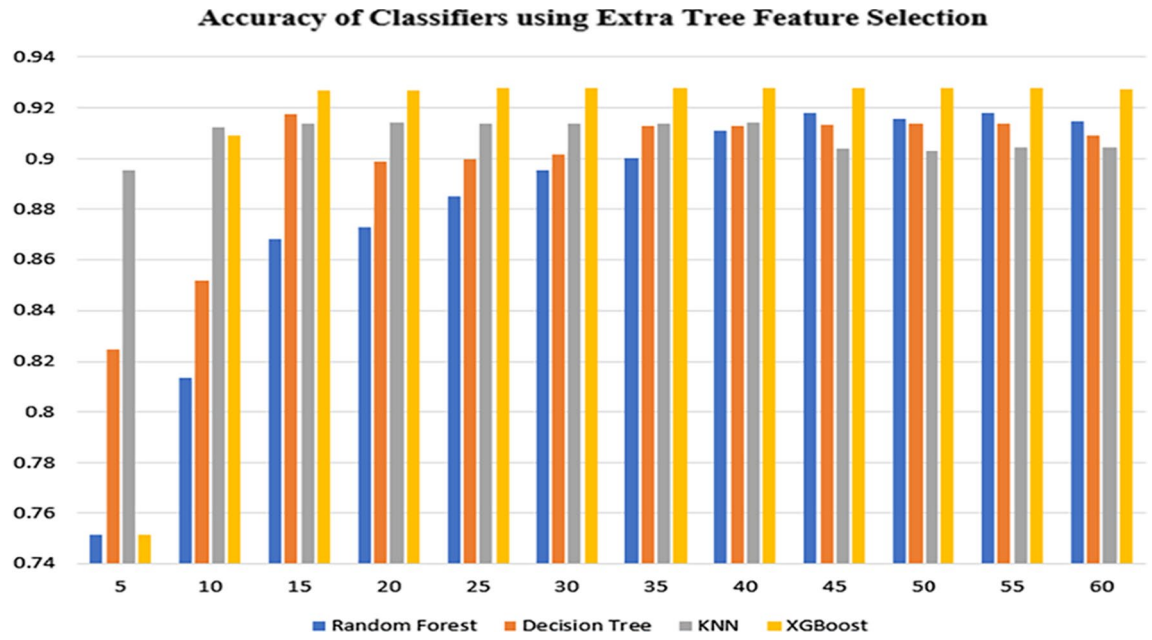


Fig. 6 Accuracy of RF, DT, KNN, XGBoost using extra tree feature selection

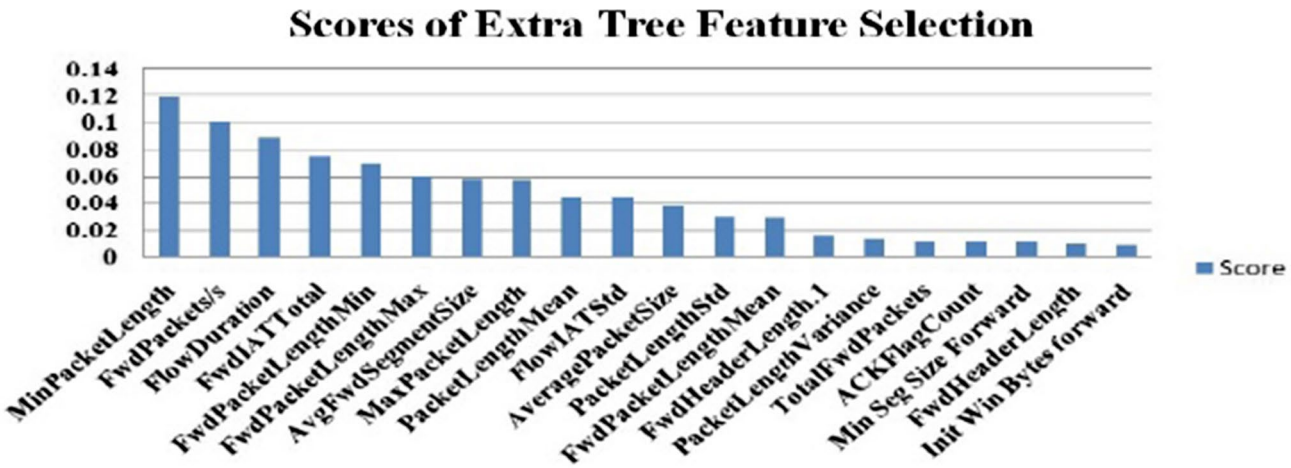


Fig. 7 Top 20 features using extra tree feature selection

Table 9 Accuracy of classifiers with ANOVA feature selection algorithm on selected features with an interval of '5' in between

No. of features	Random forest	Decision tree	KNN	XGBoost	No. of features	Random forest	Decision tree	KNN	XGBoost
5	0.7525943	0.7489992	0.7499541	0.75383	35	0.9125304	0.9139645	0.9125572	0.64799
10	0.8824424	0.8904737	0.8969766	0.96980	40	0.9115227	0.913848	0.9125572	0.64800
15	0.8905708	0.9134244	0.9137481	0.98347	45	0.9175752	0.9136701	0.9071946	0.64800
20	0.8936395	0.8865116	0.9128736	0.98012	50	0.9069309	0.9139537	0.9029935	0.76578
25	0.9135868	0.9126629	0.912678	0.64745	55	0.9191502	0.9139041	0.9042244	0.918678
30	0.9068976	0.8091940	0.9124662	0.64788	60	0.9185758	0.9066554	0.9046265	0.952761

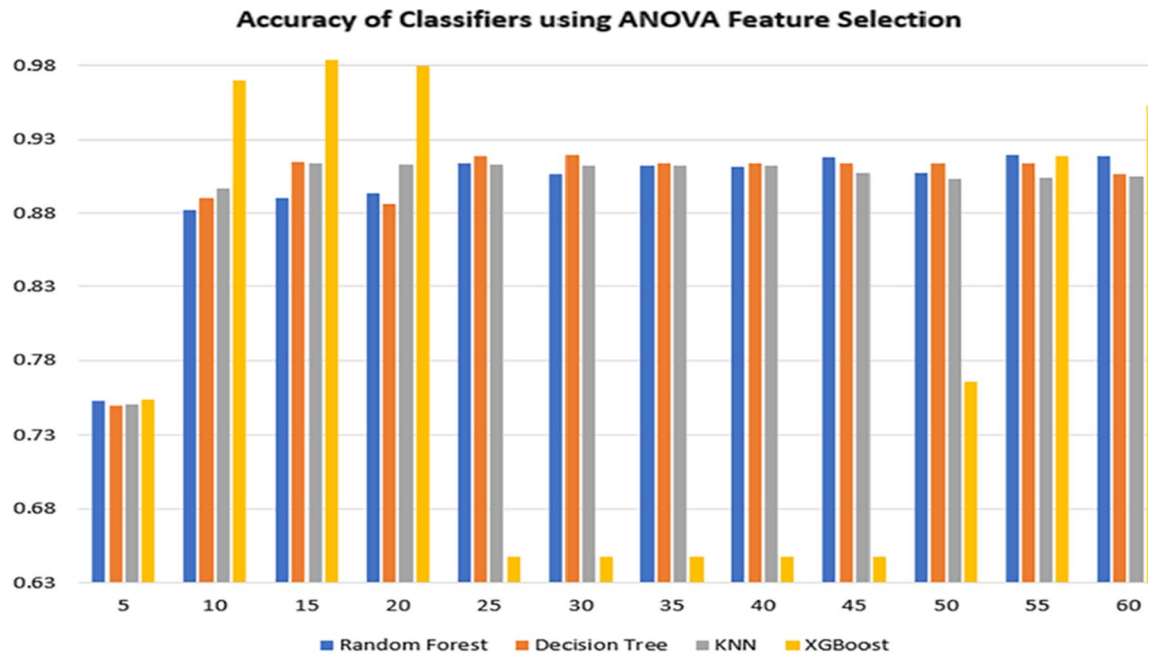


Fig. 8 Accuracy of RF, DT, KNN, XGBoost using ANOVA feature selection

chi-square for 50 features (42% feature reduction ratio), an accuracy of 92.78% with Extra Tree for 30 features (65% feature reduction ratio) and 98.34% with ANOVA for 15 features (82.5% feature reduction ratio). As seen, the best performance was achieved when XGBoost is coupled with ANOVA with Accuracy = 98.347%, Precision = 99%, Recall = 99%, *F*-1 Score = 99%. This section discusses the results obtained from numerical simulations. The results obtained after an association of feature selection algorithms with machine learning classifiers are depicted in Table 11.

From Table 11, we could see that all the feature selection algorithms have excellent values (the best could achieve 98.34% accuracy for XGBoost when coupled with ANOVA and the least could be 90.410% when KNN is coupled with chi-square). As can be seen, Random forest when checked with all features gives maximum accuracy as 90.234%. But with chi-square it gives a maximum accuracy of 91.552% with only 35 features, 91.799% with 45 features for Extra Tree and 91.915% with 55 features for ANOVA feature selection as shown in Fig. 10. Next, Decision Tree with all features gives 90.412%, when coupled with chi-square gives 91.537%, for 40 features, with Extra Tree gives 91.781% for 15 features and 91.396% for ANOVA with 35 features as shown in Fig. 11. Next series of iterations are performed with KNN, KNN with all features gives 90.459%, KNN with chi-square gives 90.410% for 40 features, KNN with Extra Tree gives 91.447% for 20 features and with ANOVA gives 91.287% for 15 features (Fig. 12). When XGBoost is coupled with different feature selection algorithms it outperforms all other classifiers as shown in Fig. 13. The maximum accuracy achieved by XGBoost is when it is coupled with ANOVA is 98.347% with 15 features. This clearly gives a high feature reduction rate of 82.5%. Among the different feature selection algorithm(s) viz. chi-square, Extra tree and ANOVA being used with classifiers viz RF, DT, KNN and XGBoost, it has been found that XGBoost gives the highest value for all the performance parameters.

6 Conclusion and Future Scope

In this paper, we have proposed a hybrid methodology for the early detection of DDoS attacks on newly released dataset. There are umpteen allied works on different datasets, but they don't comprise the extensive types of reflective DDoS attacks. In the CICDDoS2019 dataset comprehensive categories of attacks have been taken into account, so our methodology uses this dataset for training and evaluation. Firstly accuracy of all the classifiers has been noticed with full features and the performance of classifiers has been analyzed without any feature selection. Results show that XGBoost achieves an accuracy of 96.677%. Meanwhile, extensive iterations have been carried out using chi-square, Extra Tree and ANOVA and selection of important features have been done to reduce data dimensionality. After extensive iterations, it has been concluded that XGBoost coupled with ANOVA attain 98.374% accuracy for 15 features (82.5% feature reduction rate). By selecting critical features, this hybrid methodology can quickly detect DDoS attacks on IoT

Table 10 Features importance scores using ANOVA

Feature	Score	Feature	Score
ACKFlagCount	2,158,231.991	BwdPacketLengthMean	40,209.12051
AveragePacketSize	1,048,916.983	AvgBwdSegmentSize	40,209.12051
MinPacketLength	1,045,759.437	BwdPacketLengthMin	37,871.73339
FwdPacketLengthMin	1,022,063.152	FwdPSHFlags	33,721.42622
PacketLengthMean	1,002,141.614	RSTFlagCount	33,721.42622
FwdPacketLengthMean	995,327.0919	BwdPacketLengthMax	32,568.34659
AvgFwdSegmentSize	995,327.0919	BwdPacketLengthStd	23,913.52589
FlowDuration	916,093.2099	TotalBackwardPackets	21,532.06727
FwdIATTotal	916,068.6688	SubflowBwdPackets	21,532.06727
FwdPacketLengthMax	894,651.7018	Init_Win_bytes_backward	21,146.66543
MaxPacketLength	701,664.0653	BwdIATMean	21,035.21156
FwdIATMean	594,422.3036	BwdIATMax	20,978.39563
FlowIATMean	538,712.1519	BwdIATStd	20,799.68325
FlowIATMax	529,777.6775	BwdIATTotal	20,405.31633
FwdIATMax	529,776.2715	PacketLengthVariance	17,390.72385
Inbound	524,967.2349	BwdIATMin	10,596.68689
IdleMax	506,747.0373	ActiveMax	4253.084394
FwdIATStd	446,675.4695	ActiveStd	4031.080793
FlowIATStd	443,849.697	TotalLengthofBwdPackets	2506.559744
IdleMean	429,568.8473	SubflowBwdBytes	2506.559744
FwdPacketLengthStd	298,771.6343	min_seg_size_forward	2324.884787
IdleMin	283,231.9947	FwdHeaderLength	2062.83135
IdleStd	270,164.9871	FwdHeaderLength.1	2062.83135
Init_Win_bytes_forward	247,379.3179	BwdPackets/s	2055.499312
FwdPackets/s	232,927.1246	ActiveMean	1656.669655
PacketLengthStd	123,851	BwdHeaderLength	1525.970667
URGFlagCount	114,200.9308	ActiveMin	846.4567376
Down/UpRatio	98,682.2633	FlowIATMin	822.0258926
act_data_pkt_fwd	79,519.8853	FwdIATMin	694.5011845
TotalLengthofFwdPackets	77,522.90346	TotalFwdPackets	404.9195776
SubflowFwdBytes	77,522.90346	SubflowFwdPackets	404.9195776
CWEFlagCount	42,187.91937	SYNFlagCount	215.2984095

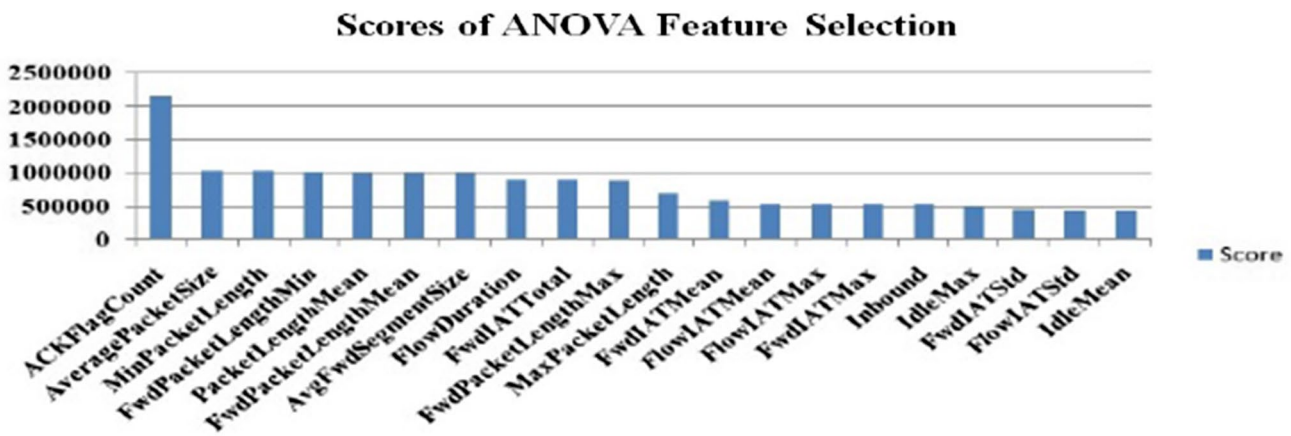


Fig. 9 Top 20 features using ANOVA feature selection

Table 11 Performance of classifiers on CICDDoS2019 dataset

Classifier	Feature selection method	No. of features	Accuracy	Precision	Recall or TPR	Specificity or TNR	FPR	FNR	*F*1-score
Random forest	All features	79	90.234	0.98	0.91	0.99	0.00	0.13	0.94
	Chi-square	35	91.552	0.99	0.92	0.98	0.01	0.17	0.95
	Extra tree	45	91.799	0.99	0.92	0.99	0.00	0.13	0.95
	ANOVA	55	91.915	0.98	0.94	0.99	0.00	0.13	0.94
Decision tree	All features	79	90.412	0.99	0.91	0.98	0.01	0.16	0.94
	Chi-square	40	91.537	0.99	0.91	0.98	0.01	0.16	0.94
	Extra tree	15	91.781	0.99	0.92	0.98	0.01	0.14	0.94
	ANOVA	35	91.396	0.99	0.95	0.98	0.01	0.15	0.95
KNN	All features	79	90.459	0.98	0.90	0.96	0.04	0.19	0.93
	Chi-square	40	90.410	0.98	0.90	0.97	0.04	0.19	0.93
	Extra tree	20	91.447	0.98	0.91	0.93	0.04	0.19	0.94
	ANOVA	15	91.287	0.98	0.95	0.94	0.04	0.19	0.94
XGBoost	All features	79	96.677	0.99	0.93	0.99	0	0.14	0.95
	Chi-square	50	92.604	0.99	0.92	0.99	0	0.14	0.95
	Extra tree	30	92.789	0/98	0.93	0.99	0	0.14	0.94
	ANOVA	15	98.347	0.98	0.98	0.99	0	0.15	0.98

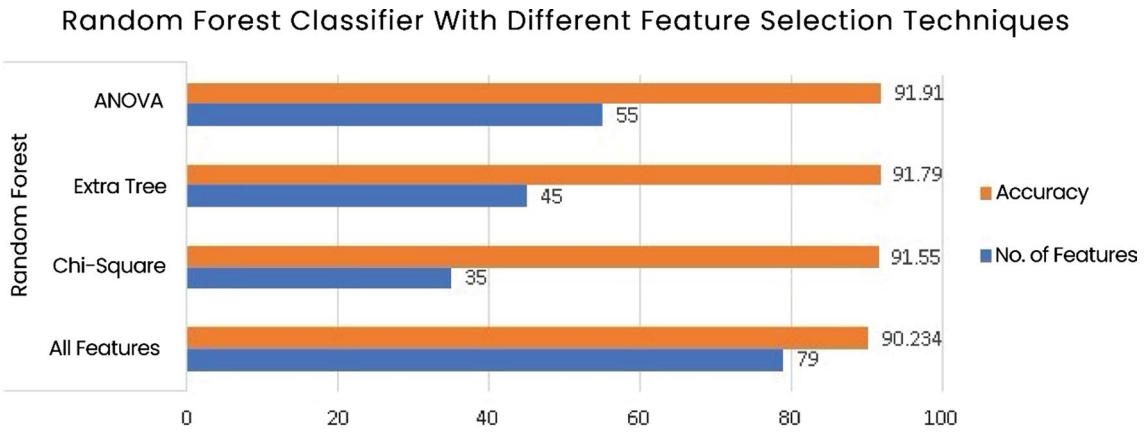


Fig. 10 Accuracy of random forest without and with feature selection

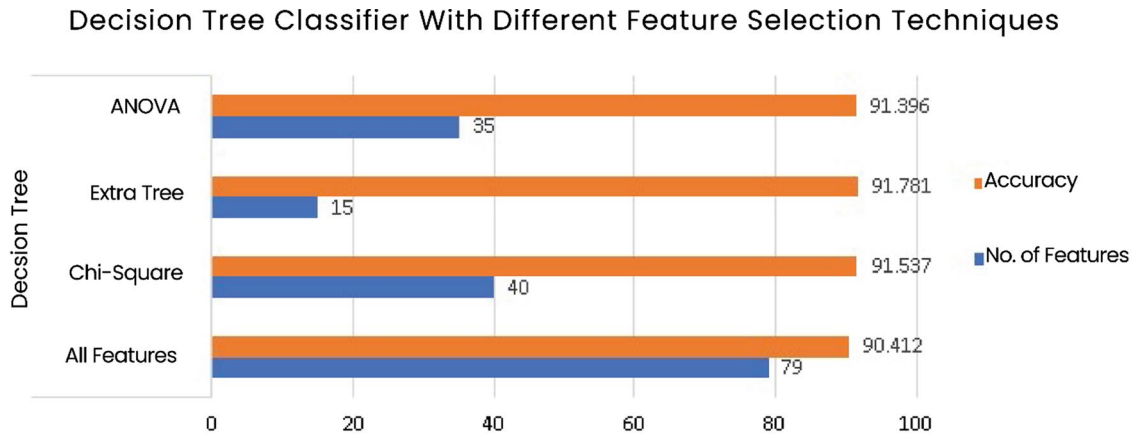


Fig. 11 Accuracy of decision tree without and with feature selection

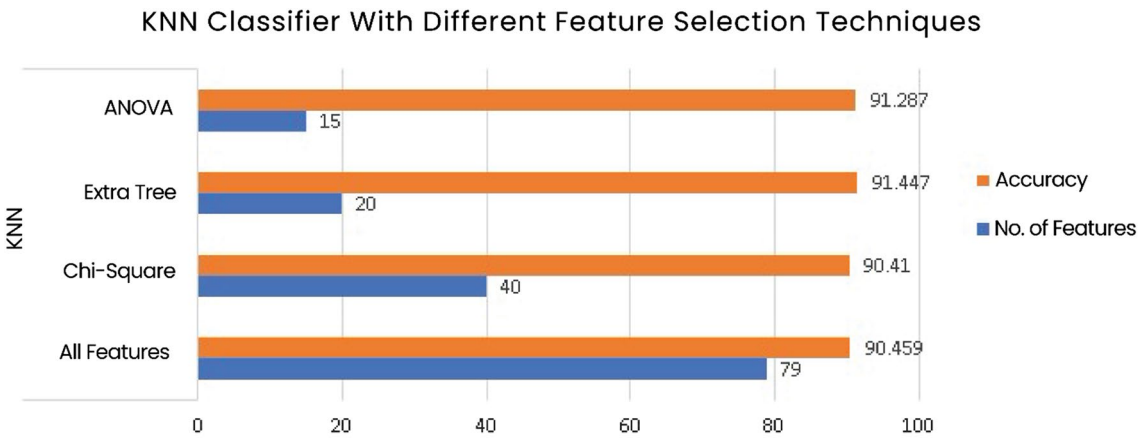


Fig. 12 Accuracy of KNN without and with feature selection

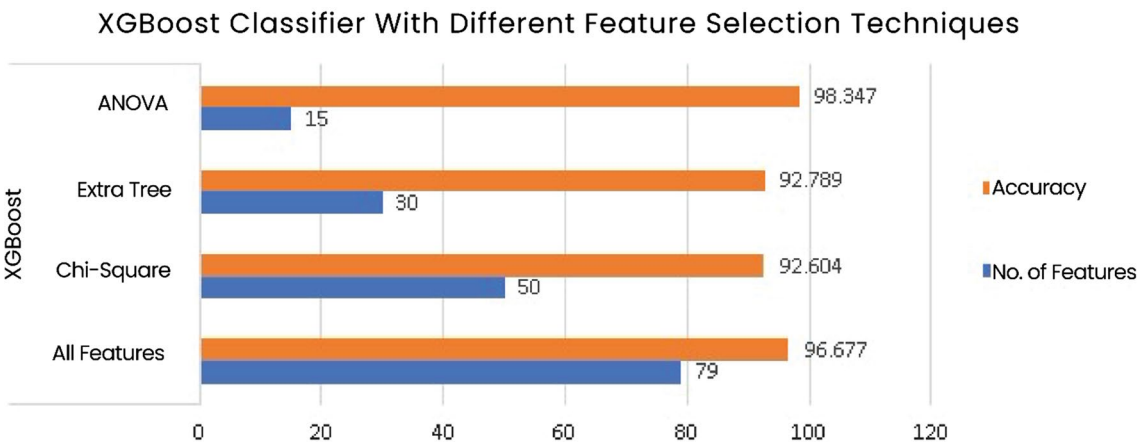


Fig. 13 Accuracy of XGBoost without and with feature selection

devices. To further ameliorate performance parameters, tuning of hyper parameters can be offered as future work. Since hyper-parameter optimization deals with issues of overfitting and underfitting, thereby leads to increase in accuracy. As a future work, different methods (Random Search, Grid Search and Bayesian Optimization) can be employed for tuning.

References

1. Mahjabin, T.; Xiao, Y.; Sun, G.; Jiang, W.: A survey of distributed denial-of-service attack, prevention, and mitigation techniques. Int. J. Distrib. Sens. N. **13**(12), 1–33 (2017). https://doi.org/10.1177/1550147717741463
2. Brasilino, L.R.; Swamy, M.: Mitigating DDoS Flooding Attacks against IoT using Custom Hardware Modules. In: Sixth International Conference on Internet of Things: Systems, Management and Security (IOTSMS), Granada, Spain, 22–25 October2019, pp.58–64. Granada, Spain: IEEE
3. Grammatikis, P.I.R.; Sarigiannidis, P.G.; Moscholios, I.D.: Securing the Internet of Things: challenges, threats and solutions. Internet Things **5**, 41–70 (2019)
4. Bodeia, C.; Chessaa, S.; Gallettab, L.: Measuring security in IoT communications. Theor. Comput. Sci. **764**(1), 100–124 (2019). https://doi.org/10.1016/j.tcs.2018.12.002
5. Ray, P.: A survey on Internet of Things architectures. J. King Saud. Univ. Comp. Info. Sci. **30**(3), 291–319 (2018). https://doi.org/10.1016/j.jksuci.2016.10.003
6. Siegel, J.E.; Kumar, S.; Sarma, S.E.: The future internet of things: secure, efficient, and model-based. IEEE Internet Things J. **5**(4), 2386–2398 (2017). https://doi.org/10.1109/JIOT.2017.2755620
7. Munshi, A.; Alqarni, N.A.; Almalki, N.A.: DDOS Attack on IoT Devices. In: 3rd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 19–21 March2020, pp. 1–5. Riyadh, Saudi Arabia: IEEE
8. Kim, M.: Supervised learning-based DDoS attacks detection: tuning. ETRI J. **41**(5), 560–573 (2019). https://doi.org/10.4218/etrij.2019-0156
9. Alzubi, O.; Alzubi, J.; Tedmori, S.; Rashaideh, H.; Almomani, O.: Consensus-based combining method for classifier ensembles. Int. Arab. J. Inf. Technol. **15**(1), 76–86 (2018)
10. Alzubi, O.A.; Alzubi, J.A.; Alweshah, M.; Qiqieh, I.; Shami, S.A.; Ramachandran, M.: An optimal pruning algorithm of classifier ensembles: dynamic programming approach. Neural. Comput. Appl. **32**(5), 16091–16107 (2020). https://doi.org/10.1007/s00521-020-04761-6
11. Babu, M.V.; Alzubi, J.A.; Sekaran, R.; Patan, R.; Ramachandran, M.; Gupta, D.: An improved IDAF-FIT clustering based ASLPP-RR routing with secure data aggregation in wireless sensor network. Mob. Netw. Appl. (2020). https://doi.org/10.1007/s11036-020-01664-7
12. Alzubi, J.A.: Bipolar fully recurrent deep structured neural learning based attack detection for securing industrial sensor networks. T. Emerg. Telecommun. T. (2020). https://doi.org/10.1002/ett.4069
13. Alzubi, J.: Optimal classifier ensemble design based on cooperative game theory. Res. J. Appl. Sci. **11**(12), 1336–1343 (2015). https://doi.org/10.19026/rjaset.11.2241
14. Salahuddin, M.A.; Bari, M.F.; Alameddine, H.A.; Pourahmadi, V.; Boutaba, R.: Time Based Anomaly Detection using Autoencoder. In: International Conference on Network and Service Management, Izmir, Turkey, 2–6 November2020, pp.1–9. Izmir, Turkey: IEEE
15. Elsayed, M.S.; Khac, N.A.L.; Dev, S.; Jurcut, A.D.: DDoSNet: A Deep-Learning Model for detecting network attacks. In: 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Cork, Ireland, 31 August-03 September2020, pp.391–396. Cork, Ireland: IEEE
16. Maranhao, J.P.A.; Costa, J.P.C.L.D.; Freitas, E.P.D.; Javidi, E.; Junior, R.T.D.S.: Error-robust distributed denial of service attack detection based on an average common feature extraction technique. Sensors **20**(20), 5845–5866 (2020). https://doi.org/10.3390/s20205845
17. Silveria, F.A.F.; Junior, A.D.M.B.; Vargas-Solar, G.; Silveria, L.F.: Smart Detection: an online approach for DoS/DDoS attack detection using machine learning. Secur. Commun. Netw. (2019). https://doi.org/10.1155/2019/1574749
18. Shurman, M.; Khrais, R.; Yateem, A.: DoS and DDoS attack detection using deep learning and IDS. Int. Arab J. Inf. Technol. **17**(4A), 655–661 (2020). https://doi.org/10.34028/iajit/17/4A/10
19. Li, J.; Liu, M.; Xue, Z.; Fan, X.; He, X.: Rtvd: a real-time volumetric detection scheme for ddos in the internet of things. IEEE Access **8**, 36191–36201 (2020). https://doi.org/10.1109/ACCESS.2020.2974293
20. Jia, Y.; Zhong, F.; Alrawais, A.; Gong, B.; Cheng, X.: Flowguard: an intelligent edge defense mechanism against IoT DDoS attacks. IEEE Internet Things J. **7**(10), 9552–9562 (2020). https://doi.org/10.1109/ACCESS.2020.2974293
21. Sharafaldin, I.; Lashkari, A.H.; Hakak, S.; Ghorbani, A.A.: Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In: 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, pp. 1–8, 1–3 October2019, Chennai, India: IEEE
22. Alsamiri, J.; Alsubhi, K.: Internet of things cyber attacks detection using machine learning. Int. J. Adv. Comput. Sci. Appl. **10**(12), 627–634 (2019). https://doi.org/10.14569/IJACSA.2019.0101280
23. Gurulakshmi, A.K.: Analysis of IoT Bots against DDOS attack using Machine Learning Algorithm. In: Proceedings of the 2nd International Conference on Trends in Electronics and Informatics (ICOEI 2018), Tirunelveli, India, pp. 1052–1057, 11–12 May 2018, Tirunelveli, India: IEEE. https://doi.org/10.1109/ICOEI.2018.8553896
24. Meidan, Y.; Sachidananda, V.; Peng, H.; Sagron, R.; Elovici, Y.; Shabtai, A.: A novel approach for detecting vulnerable IoT devices connected behind a home NAT. Comput. Secur. **97**, 101968–101991 (2020). https://doi.org/10.1016/j.cose.2020.101968-101991
25. Wehbi, K.; Hong, L.; Al-salah, T.; Bhutta, A.A.: A Survey on Machine Learning Based Detection on DDoS Attacks for IoT Systems. In: 2019 SoutheastCon, Huntsville, AL, USA, pp. 1–6, 11–14 April2019, AL, USA: IEEE. https://doi.org/10.1109/SoutheastCon42311.2019.9020468
26. Hosseini, S.; Azizi, M.: The hybrid technique for DDoS detection with supervised learning algorithms. Comput. Netw. **158**, 35–45 (2019). https://doi.org/10.1016/j.comnet.2019.04.027
27. Alkasassbeh, M.; Hassanat, A.B.; Naymat, G.A.; Almseidin, M.: Detecting distributed denial of service attacks using data mining techniques. Int. J. Adv. Comput. Sci. Appl. **7**(1), 436–445 (2016). https://doi.org/10.14569/IJACSA.2016.070159
28. Wang, M.; Lu, Y.; Qin, J.: A dynamic MLP-based DDoS attack detection method using feature selection and feedback. Comput. Secur. **88**, 101645–101659 (2020). https://doi.org/10.1016/j.cose.2019.101645
29. Al Hamad, M.; Zeki, A.M.: Accuracy vs. cost in decision trees: A survey. In: 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT),

Sakhier, Bahrain, pp. 1–4, 18–20 November 2020, Sakhier, Bahrain: IEEE. https://doi.org/10.1109/3ICT.2018.8855780
30. Azad, M.; Moshkov, M.: Classification and Optimization of Decision Trees for Inconsistent Decision Tables Represented as MVD tables. In: Proceedings of the Federated Conferenc e on Computer Science and Information Systems, Lodz, Poland, pp. 31–38, 13–16 September 2015, Lodz, Poland. IEEE. https://doi.org/10.15439/2015F231
31. Rani, P.; Kumar, R.; Jain, A.: Multistage model for accurate prediction of missing values using imputation methods in heart disease dataset. In: Raj, J.S.; Iliyasu, A.M.; Bestak, R.; Baig, Z.A. (Eds.) Innovative Data Communication Technologies and Application, pp. 637–653. Springer, Singapore (2021). https://doi.org/10.1007/978-981-15-9651-3_53
32. Rani, P.; Kumar, R.; Ahmed, N.M.S.; Jain, A.: A decision support system for heart disease prediction based upon machine learning. J. Reliab. Intell. Environ. (2021). https://doi.org/10.1007/s40860-021-00133-6
33. Xue, H.; Wang, P.: An Improved Sample Mean KNN Algorithm Based on LDA. In: 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, pp. 266–270, 24–25 August 2019, Hangzhou, China. https://doi.org/10.1109/IHMSC.2019.00068
34. Zhang, C.; Wang, D.; Song, C.; Wang, L.; Song, J.; Guan, L.; Zhang, M.: Interpretable learning algorithm based on XGboost for fault prediction in optical network. In: 2020 Optical Fiber Communications Conference and Exhibition (OFC), San Diego, CA, USA pp. 1–3, 8–12 March 2020, San Diego, CA, USA IEEE
35. Sadique, K.M.; Rahmani, R.; Johannesson, P.: Towards security on internet of things: applications and challenges in technology. Proc. Comput. Sci. **141**, 199–206 (2018). https://doi.org/10.1016/j.procs.2018.10.168
36. Sharma, D.: Implementing Chi-Square method and even mirroring for cryptography of speech signal using Matlab. In: International Conference on Next Generation Computing Technologies (NGCT), Dehradun, India pp. 394–397, 4–5 September2015, Dehradun, India. IEEE. https://doi.org/10.1109/NGCT.2015.7375148
37. Alsariera, Y.A.; Adeyemo, V.E.; Balogun, A.O.; Alazzawi, A.K.: AI meta-learners and extra-trees algorithm for the detection of phishing websites. IEEE Access **8**, 142532–142542 (2020). https://doi.org/10.1109/ACCESS.2020.3013699
38. Pena, M.; Alvarez, X.; Jadán, D.; Lucero, P.; Barragán, M.; Guamán, R.; Sánchez, V.; Cerrada, M.: ANOVA and cluster distance based contributions for feature empirical analysis to fault diagnosis in rotating machinery. In: International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC), Shanghai, China pp. 69–74, 16–18 August2017, Shanghai, China IEEE. https://doi.org/10.1109/SDPC.2017.23