Development of IDS using mining and machine learning techniques to estimate DoS malware

G. Revathy*, P. Sathish Kumar and
Velayutham Rajendran

Department of Electronics and Communication Engineering,
Vels Institute of Science, Technology & Advanced Studies (VISTAS),
Pallavaram, Chennai, 600117, India
Email: grevathy19@gmail.com
Email: sathish.se@velsuniv.ac.in
Email: drvrajen@gmail.com
*Corresponding author

Abstract: A denial of service is a main type of cyber security attack. Intrusion detection system techniques play a very important role for detecting and preventing mechanisms that eradicate the issues made by hackers in the network environment. In this research, we describe different data mining techniques which can be used to handle different kinds of network attacks. Three machine learning techniques are used for classification problems, such as decision tree classifier, gradient boosting classifier, K-nearest neighbour classifier, to find the metric values of false negative rate, accuracy, F-score and prediction time. We found that the decision tree classifier and voting classifier is the best method which has less prediction time and better accuracy of 99.86% and 99.9% which makes the model better along with greater performance. The result shows high accuracy level and less prediction time. Moreover, the relationships between existing approach and proposed approaches in terms of metrics are described.

Keywords: denial of service; DoS; machine learning techniques; MLT; statistical analysis; false negative rate; FNR; intrusion detection system; IDS.

Reference to this paper should be made as follows: Revathy, G., Kumar, P.S. and Rajendran, V. (2021) ‘Development of IDS using mining and machine learning techniques to estimate DoS malware’, *Int. J. Computational Science and Engineering*, Vol. 24, No. 3, pp.259–275.

Biographical notes: G. Revathy secured UG (BTech) in Information Technology from Anna University, Chennai India, in 2010. She received her PG (MTech) in Computer Science and Engineering from Anna University, Chennai, India in 2013. She is currently pursuing her PhD in Electronics and Communication at VISTAS in Chennai. Her research scrutiny involves machine learning, data mining, and wireless networks.

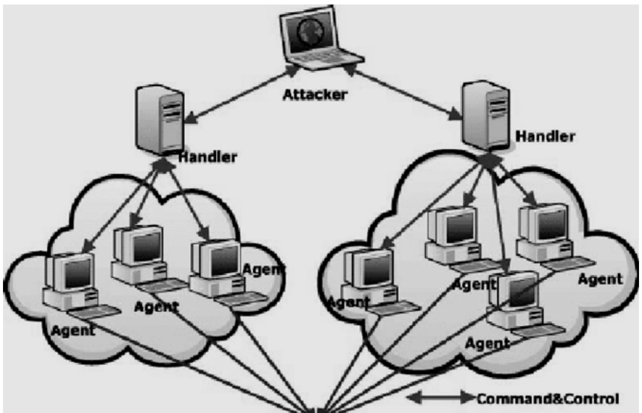
P. Sathish Kumar received his BE in Electronics and Communication Engineering from the Priyadharshini Engineering College, Vaniyambadi, Tamil Nadu India, in 2006, MTech degree in VLSI from the Sathyabama University, Chennai, in 2013 and completed PhD in Electronics and Communication in the VISTAS at Chennai. He has published over six papers in Scopus. His research interests include machine learning, embedded system, and low power VLSI.

Velayutham Rajendran finished his UG degree from Madurai Kamaraj University, received MTech degree from IISC Bangalore and awarded with PhD degree from Chiba University, Japan (1993). He was very closely linked with various institutions and organisations such as National Institute of Ocean Technology (Chennai), Indian Institute of Technology (Chennai), IIS (Bangalore), SSN College of Engineering (Chennai), etc. He is currently working as the Director and Professor in the Department of ECE in Vels Institute of Science, Technology and Advanced Studies (VISTAS), Pallavaram, Chennai. Moreover, he had published nearly 55 journal papers and 60 conference papers in national as well as international. He is has more than seven patents related to his work.

1 Introduction

An intrusion detection system (IDS) monitors either networks or other systems for inconsistent behaviours. The major attack which creates issues to the network is mainly the DoS attack. Mostly all the users work with internet hence it is very important to detect DoS attacks and prevent the network going down. A DoS attack is being used to tie up the resources of a website to prevent users needing to access the site from doing so. DoS attacks have developed into ‘distributed denial of service’ (DDoS) attacks that are more complicated and advanced. GitHub targeted code-hosting-service in 2018 was the largest attack ever reported. The major issues in the network are finding the hackers who make the network fault or slow down (network traffic). The best way to detect and identify DoS attack is network monitoring and statistical analysis of features in the datasets. Wani et al. (2020) performed for attacking mechanism using Tor hammer on cloud stack environment using machine learning (ML) algorithm such as RF, NB, SVM, decision tree and C4.5. Yusof et al. (2016) introduced early and fast detection of DoS attack through predictive analysis for classifying the status of the network which may be either malicious or normal. By evaluating WEKA dataset based on IDS, hybrid classifier involves K-nearest neighbour and support vector machine have been used for classifying, detecting, and predicting the DoS attack in the network. Khraisat et al. (2019) presented invasion techniques for detecting the intruders who hack the network environment. Bose and Mahapatra (2001) utilised intelligent retrieval approaches such as rule induction along with neural network that could be followed for analysing data and behaviours of patterns in the dataset and further for developing data mining (DM) applications. Mand and Reed (2012) surveyed that prohibiting the DoS attack by categorising into three junctures such as prevention, detection and response has lower detection rate (DR) and poor false positive rate (FPR). Figure 1 shows the basic DoS architecture. In DoS, the incoming data flooding of victims comes from several outlets. This effectively prevents the attack from stopping by blocking a single source.

Figure 1 Modus vivendi for DoS attack



This paper is structured as follows: Section 2 compares the related work of existing researchers done for detecting attacks. Section 3 describes the methodology used in the proposed work. Section 4 explains about the overview, architecture, algorithm of the proposed model. Section 5 describes the dataset description, separation of attack and normal dataset. Section 6 elaborates related to the experiment and result analysis of the proposed research work. Conclusions and discussion for future work are given in Section 7.

Nowadays, the complexity of DoS attack has enlarged significantly due to increase in attacks in the network when compared to previous years. The complexity of DoS attack can be summarised as follows:

- intruders try to weaken the system very easily
- the source attack quantity is extremely large when there are more attackers but with an individual intruder the traffic is less.

To resolve the above intricate dilemma, this survey focused on statistical analysis of detecting and finding DoS attacks in the system through training and validation phases in ML.

The main contributions of this paper are as follows:

- Basically, finding and removing unnecessary features makes training faster and also improving model accuracy in ML algorithms for high dimensional data.
- Hence, in this survey feature reduction (dataset comprises 41 features which are reduced to 26 features) is analysed by finding z-score value.
- We statistically analysed for feature reduction by calculating t-score value and also placing threshold value as ($\alpha = 0.05$).
- Fix the threshold value ($\alpha = 0.05$) to remove the null values present in dataset.
- If level of significance < α, then the condition is to reject null hypothesis. If level of significance > α, then condition is fail to reject null hypothesis.
- Detecting intruders in the network and eradicating the attack via training as well as testing process under ML with large amount of attack (DoS) dataset.
- Performing arithmetic investigation to detect abnormalities through measuring central tendency namely mean, median, and mode.
- The whole dataset divided into two phases: 80% for training phase and remaining 20% for testing phase.
- Also, 80% of training dataset divided as 1%, 10%, and 100% to check the accuracy comparison evaluated by different classifiers algorithms.
- Metrics such as accuracy, precision, recall, false negative rate (FNR) are evaluated. Based on highest classifier accuracy, the overall performance of the model is evaluated helpful for finding attacker who's hit the network system.
- We found that values of both t-score and P-value using statistical analysis and ML methods are similar.
- The first four features are selected for analysis via various ML algorithms like KNN, gradient boosting, and decision tree to save memory and reduce time taken. Finally the accuracy predictions for selected feature set and full feature set are similar for every algorithms.

- Training and testing applied in selected features also which is similar to full feature set.
- Moreover, another three algorithms DT, SVC and voting classifier are compared for finding full feature set and selected feature set.
- Our experimental results reveal that values calculated by selected feature set and full feature set are similar. So, we will better use selected feature set to save memory in case of huge database.

This algorithm has been implemented in hardware that makes it less complex for finding attacks in the system. Moreover, this paper would have been extended to the analysis mainly concentrated on hardware-friendly which are incorporated as overhead.

2 Related study

Many of the organisations such as IT companies are affected due to these kinds of attacks (data stealing by unauthorised users) in the networks. More researchers have been working on intrusion detection-based models for detecting malicious attack in the network environment over the last three decades. DM techniques are being used to identify anomalies, malware, or any types of attacks in a protected system environment (Salo et al., 2018). Yihunie et al. (2018) investigated the negative collision of both DoS and DDoS attack through evaluating detached attacks and many system attacks. Moreover, this survey pointed how DoS attacks produced harm to business productivity and also how to tackle these issues created by DoS attack. By that the network performance and response time of the server increased quickly. Alkasassbeh et al. (2016) demonstrated the detection of DoS attack using DM techniques. This paper explains about IDS which is the main solution for finding any attack or anomaly. Detection of attacks is found using classification techniques such as naïve Bayes, random forest algorithm and multilayer perceptron (MLP). This investigation shows that MLP achieved the high accuracy rate of 98.63%. Panda and Patra (2009) evaluated the ML techniques for detecting anomalies in the network. DM algorithms such as random forest, AdaBoost and naïve Bayes have been used for finding the DR and low false alarm rate (FAR) to build an efficient model.

Norouzian and Merati (2011) introduced a novel classification method known as MLP neural network for finding attacks. The experimental analysis shows the accuracy rate of 90.78% through MLP with two hidden layers. Kaur et al. (2017) explained the classification of detection approaches against the DDoS attack. Comparisons of anomaly-based, signature-based, and hybrid-based were depicted. Manso et al. (2019) detected the DoS attack in the system earlier as well as alleviating the attacks based on software defined network infrastructure. Karimazad and Faraahi (2011) proposed an analysis of network traffic using anomaly-based detection. In UCLA dataset, neural network algorithm such as a radial-based function (RBF) was proposed in this work achieving an accuracy of 96%. Chakraborty et al. (2019) explains how DoS attacks occur in the system, how such attacks are harmful to the system, and ways to protect the system from that kind of attack. Some other cyber security issues happened in the network via botnet. Hung and Sun (2018) proposed novel work for distinguishing possible botnet through evaluating the number of flows in network area, extracting patterns to application layer, finally scrutinising P2P botnet and also HTTP protocol-based botnet under training and testing phases of ML approach.

Almutairi et al. (2020) offered techniques for botnet communiqué interchange comprising HTTP, IRC, P2P, and DNS using IP fluxing employed in both network region and host region. The introduced algorithm carried out pre-processing technique to mine the feature mainly for differentiate legal behaviour from illegal (botnet) behaviour. Callegari et al. (2018) intended a new technique namely kernel-principle component analysis algorithm for predicting network abnormalities that defeats some disadvantages of classical PCA-based approach whereas enhancing performance in finding network abnormalities.

Harikumar et al. (2017) identified an association rule that allows important features from high dimensional data for developing apriori algorithm. Based on apriori algorithm, the feature selection can be done using entropy and also information gain which leads to time complexity.

Yu et al. (2019) proposed an intelligent bee colony algorithm for detecting DDos attack in the system based on the inspiration of abnormality extraction, and a traffic reduction algorithm was used to reduce traffic in the network environment. This joint mechanism detects the quality of DDos attack data stream to enhance data flow accuracy detection very effective and efficient.

Zareapoor and Shamsolmoali (2018) detected DDoS attacks in the system based on cloud computing. The packet features which demonstrate the DDoS attacks in traffic were discussed and two levels of filtering to detect the attack were proposed. For performance evaluation, processing time and detection accuracy were the metrics utilised.

Khatak and Maini (2016) discussed basic information of DoS attack, i.e., ping of death, SYN flooding, smurf, teardrop, buffer over flow or everlasting denial of services which are harmful to the system hardware. Techniques utilised to recover from these kinds of attacks include routers, firewalls, and replacing systems hardware in case of permanent denial of service.

2.1 *Methods for detecting attacks*

IDS are the monitoring and detecting devices that can detect any anomaly or attacks in the system or network environment. The scope of the IDS is to detect anomaly or attack in the system by getting alarm when an intrusion is detected. Hence the exploratory outcome shows that performance is based on better DR, attack type, FPR and accuracy (Khatak and Maini, 2016; Alkasassbeh et al., 2016; Siris and Papagalou, 2006). Thakare and Kaur (2017) exposed multivariate correlation analysis (MCA) approach mined the geometric correlation (detecting DoS attack) between network traffic. KDD cup dataset is helpful to examine the effectiveness of the network system. To realise this, the IDS monitor all traffic (incoming and outgoing) in the network. IDS have three methods for finding the attacks in the system or in any network environment. They are:

a *Signature-based detection:* This detection is depicted to ascertain the identified attacks by using their signature. Basically, this detection provides the solution for every issue in the network security. One drawback is incompetence to detect the newly found malicious or previously found unknown attacks. Moreover, the novel type of anomaly cannot be found if the signature is not notified. Reshamwala and Mahajan (2012) discussed how to defeat the weakness of signature-based IDS in detecting novel attacks. In this experiment, the researcher possessed statistical analysis using KDD Cup 99 for better performance. The identification of known attacks which have been preloaded in the IDS database is an efficient process and much more successful. The database is frequently updated to increase the performance of the detection.

b *Anomaly-based detection:* In this method, comparing the activities of current user with the predefined users to detect the behaviour of anomalies. This detection is very effective against zero-day attack without any updating to the system. Norouzian and Merati (2011) discussed the finding of attacks through anomaly detection which works only on online data.

c *Hybrid-based detection:* To overcome the disadvantages of using single system, combining two methods for detecting intrusion is much better.

2.2 How to examine DoS attack in the system

We are utilising IDS KDD Cup dataset for detecting and classifying the dataset into two types namely DoS malicious and standard type using DM algorithms namely K-nearest neighbour, decision tree and gradient boosting. DoS malicious are represented by these following species of attacks such as Neptune (SYN flood), land, teardrop, Smurf and back. Some of the DoS attack types and their out-turn are provided in Table 1.

Table 1 DoS attack types and out-turn

S. no.	*Type of attack*	*Service of the attack*	*Out-turn*
1	Neptune	TCP	Delay in service for more than one port
2	Land	HTTP	Stop the system
3	Teardrop	Echo	Bootstrap the system
4	Smurf	ICMP	Slow down the system
5	Back	HTTP	Server reply will be slow

3 Methodology in proposed work

Methodology is the major confront which is to prevail the selection of feature set that is scarce to classify standard (normal) data from abnormality and maintain the size of feature set to bare minimum. Milenkoski et al. (2015) elucidated design space into organising workload, metrics and methods for finding malicious in the system. This researcher scrutinised the IDS via methods and techniques correlated with every part of the design space. In some other work detection of DoS attack in Wi-Fi (802.11) protocol is through determination of significant threshold (Milliken et al., 2013). Currently, the following methods are used to estimate the malware in denial of service.

1 DM technique

2 ML technique

3 statistical analysis.

3.1 DM technique

DM refers to isolating or ‘mining’ knowledge from huge amounts of data. Fayyad (1997) describes about DM which is also known as ‘knowledge discovery’. The scope of DM is to extricate information from a whole data and reconstruct it into a usable as well as understandable format or pattern. DM can be viewed as a result of the natural evolution of statistical analysis. In some conditions, IDS are not able to detect the anomaly in the network. These kinds of attacks are typically found through the ML mechanisms such as supervised, unsupervised, etc. that should be implemented in IDSs to prevent and detect these attacks without waiting for any kind of updates in the network.

DM process includes the following process:

- recognise the resource dataset
- select the corresponding data points from the dataset that need to be scrutinise
- extract the relevant information from the data
- find the key values from the extracted dataset
- understand and describe the outcome.

For finding patterns, DM approach can be utilised in an existing dataset which are applicable only in limited area. In this survey, huge amount of data samples in the dataset for detecting and eradicate DoS attack twisted by intruders, so we get into ML approaches that are suitable for large dataset.

3.2 ML technique

MLT is mainly for implementing IDS to discover attacks in the system or in any network. Much like ML, a computer can learn how to determine whether an activity is anomaly or normal which can be used for intrusion detection. Based on their learning strategies, ML techniques can be divided into three main classes shown in Figure 2.

a Supervised

Supervised learning is the most popular paradigm in ML technique. It is also described as task-oriented since the input and output are labelled in the dataset. Pervez and Farid (2014) examines that the feature selection process removes the extraneous data from the input dataset which generates higher classification accuracy.

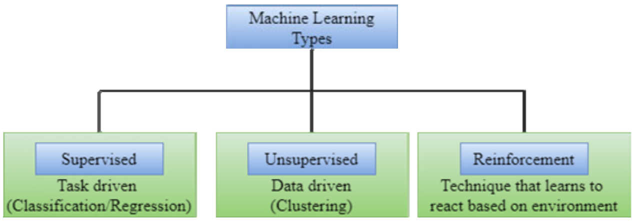
b Unsupervised

Unsupervised learning is the reverse of supervised learning. Here the features are not labelled, instead our algorithm fed into data and many tools to understand the properties of data which makes the newly structured data.

c Reinforcement

Reinforcement learning is a behaviour driven class having the relationship between presence and absence of labels (supervised and unsupervised).

Figure 2 ML – types (see online version for colours)


Machine Learning Types
Supervised
Task driven (Classification/Regression)
Unsupervised
Data driven (Clustering)
Reinforcement
Technique that learns to react based on environment

3.3 Statistical analysis

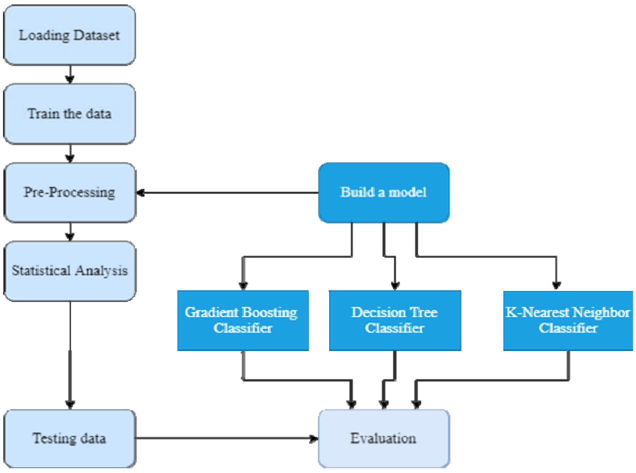
A statistical analysis is mainly used to analyse the data datasets such as representing data, understanding few interactions between variables, metrics or features in the dataset. Usage of numerical and statistical models is to obtain a general understanding of the data to make predictions through ML approaches.

4 Proposed work

The proposed work of this survey mainly insists DM algorithms like GB, DT and KNN, undergoes feature reduction, as well as training and testing method to identify attacks under ML strategy, finally statistical analysis can be done through performance of metrics evaluation. The proposed work consists of the steps as illustrated in Figure 3 which can be summarised as follows:

- the IDS dataset has been gathered from MIT Lincoln laboratory for my research work to classify the datasets as malicious or standard
- data preparation, where loading of data into particular location and prepare it for utilise in training phase
- data pre-processing has four phases namely data cleaning, amalgamation, dropping and transformation for making the process simple and easier
- through analysing the dataset statistically, the model can be evaluated via ML algorithm involving gradient boosting, decision tree and K-nearest neighbour to enhance the overall performance by calculating accuracy metric
- testing the model and finally validate the outcome to realise the better performance for KDD dataset.

Figure 3 Overview of the proposed work (see online version for colours)



The following are the issues going to be solved via proposed work:

- detecting attackers/intruders in the network and introducing possible ways to eradicate the attacks via training and validation part using gradient boosting, decision tree, and KNN ML algorithms to separate normal data and malicious data
- ML algorithms are used for mentioning the attack data, normal data by vertical representation in Figures 5–14.

4.1 Metrics calculation in the proposed work

In this paper, four metrics are calculated during statistical analysis while detecting the malware which is attack or normal. The metrics are as follows:

- *Accuracy:* Accuracy is defined as the number of predicted values calculated from total number of predictions. It is the proportion of FP and FN against the detection of DoS attack. Accuracy is calculated using the formula.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}} \quad (1)$$

- *FNR:* FNR is used to minimise the attack in the network. High FNR denotes the threshold value is high. FNR is calculated using the following formula.

$$\text{FNR} = \frac{\text{FN}}{\text{TP} + \text{TN}} \quad (2)$$

where TP is true positive and FN is false negative.

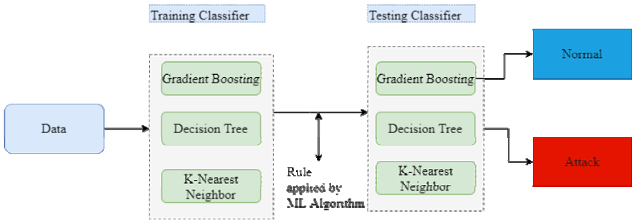
- *Prediction time:* Prediction time is defined as the time to predict the future values based on values already found through statistical analysis.
- *F-score:* F-score is the harmonic mean of both the recall and precision. In statistical analysis, F-score represents the measure of test accuracy.

$$\text{F-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3)$$

4.2 *Proposed architecture*

The proposed architecture describes the detection of DoS attack in the network using statistical analysis by selecting the features such as source host, destination host, service host, duration, logged in, wrong fragment, destination bytes, source bytes which are listed in the evaluation of mean median and modes (Figure 4). Exploring the dataset, then data pre-processing, training classifier develop the model for the dataset such as gradient boosting, decision tree, and K-nearest neighbour classifiers are used, then testing classifier shows the results of detecting the dataset whether it is attack or normal.

Figure 4 Architecture of the proposed model (see online version for colours)



4.3 *Algorithm*

Yusof et al. (2016) describes detection of DoS attack using MLT such as hybrid of support vector machine and K-nearest neighbour to classify the network status, Mukkamala and Sung (2003) describes detecting DoS attack by SVM classifier. Tsai et al. (2009) highlighted a single hybrid and ensemble classifier. In this paper, the researcher has implemented three ML algorithms which are mainly used for solving the classification problem, and hence to identify the malware in the network environment. The algorithm for decision tree classifier is provided.

```
Algorithm for Decision Tree classifier:
Input: Build the training dataset Di= {($X_1$, $C_1$),…($X_N$,$C_N$)}
X=($X_1$…$X_N$) new instances to be classified Output: finding
accuracy specifies normal.
Step 1: For each labelled instance ($X_i$, $C_i$) do
Step 2: If X has an unknown system call
          then
            X is abnormal;
            Else then
Step 3: For each process $D_i$in training data do
          Calculate $S_{im}$(X, Di):
            If $S_{im}$(X, Di)= 1.0
        then
            X is normal
            and Exit;
Step 4: Order $S_{im}$(X, $D_i$) from lowest to highest (i= 1,….N);
        Locate K biggest scores of $S_{im}$(X,D);
        Pick the Decision instances to X: $D^k$x;
        Allocate to x the most frequent class in $D^k$x;
Step 5: Calculate $S_{im}$Avg for Decision tree; If
        $S_{im}$Avg>threshold then
          X is normal;
            Else then
            X is abnormal;
          END.
```

Note: Sim refers to simulation.

In ML, the classifier algorithms which are used to solve the classification problem are:

- gradient boosting classifier
- decision tree classifier
- K-nearest neighbour classifier.

Basically, when do we say a model is good? Yes, based on the performance after evaluation we decide the model (good or worst). Performance is mainly based on higher accuracy and lower prediction time. From the above DM techniques, both KNN and decision tree classifier show higher accuracy of 99.86% but the prediction time is shorter in decision tree classifier which scores better performance to the model.

In addition to these algorithms, this survey focused on comparisons of decision tree algorithm, support vector classifier and voting classifier to calculate the accuracy for enhancing the performance of the model. The experimental results are given in Section 6.

Support vector classifier is one of the supervised learning models mainly utilised for classification issue occurs in the network. Herein, the DoS attack classification needs some classification algorithm to isolate the attack from normal. Voting classifier is a ML classification model that trains on an ensemble of several models and forecasts the outcome founded on maximum possibility of chosen data is decided as the final output.

5 Dataset description

Some of the researchers have used NSL-KDD dataset (Ingre and Yadav, 2015; Nadiammai and Hemalatha, 2014; Zargari and Voorhis, 2012), WEKA dataset (Yusof et al., 2016) and DARPA dataset (Lee et al., 1999; Mukkamala and Sung, 2003; Yu and Hao, 2007) for categorising the attack type and standard type in the network environment. In this research, the dataset is from MIT Lincoln laboratory and a DoS attack test has been performed.

a Exploring dataset

Dataset consists of total number of records, attack data, normal data, number of features and percentage of attack are shown in Table 2.

Table 2 Dataset description

Total no. of records	*No. of features*	*No. of attacks*	*No. of normal records*	*% of attacks*
22,544	38	2,799	19,745	12.40%

The constant features in the datasets are [‘urgent’, ‘land’, ‘num_shells’, ‘num_outbound_cmds’, ‘is_host_login’], so that it should be dropped which is shown in Table 3.

Table 3 Eliminate the irrelevant features

Duration	*Protocol type*	*Service*	*Flag*	*Attacks type*
0	TCP	Private	REJ	Other

b Dataset preparation

Transmission control protocol (TCP) is used for reliable communication and error checked. SF flag identifies the connection either normal or error. Attack type denotes either normal or anomaly.

c Conversion of categorical to numeric values

Kddcup99 datasets have categorical values, but ML algorithms process numerical values best. First the dataset converts categorical variables into numerical values. One hot encoding scheme is mainly for transfer into numerical values from categorical by setting 1/0 values into new column of each category.

5.1 Z-score (numerical features) calculation

Z-score is used for standardising scores on the same scale by dividing a score’s deviation by the standard deviation in a dataset. The number of standard deviations in which a given data point is from the mean can be calculated by means of Z-score. The result is a standard score. The Z-score dataset is shown in Table 4.

Table 4 Z-score

	Duration	*(Source bytes)* *src_bytes*	*(Destination bytes)* *dst_bytes*	…	*flag_SF*	*flag_SH*
0	–0.155534	–0.021988	–0.096896	…	–1.392705	–0.056997
1	–0.155534	–0.021988	–0.096896	…	–1.392705	–0.056997
2	–0.154113	0.005473	–0.096896	…	0.718027	–0.056997
3	–0.155534	–0.021946	–0.096896	…	0.718027	–0.056997
4	–0.154823	–0.021988	–0.096189	…	–1.392705	–0.056997
5	–0.155534	–0.021423	0.587166	…	0.718027	–0.056997
6	–0.155534	–0.019826	–0.078657	…	0.718027	–0.056997
7	–0.155534	–0.021715	–0.088696	…	0.718027	–0.056997
8	–0.155534	–0.021296	–0.074887	…	0.718027	–0.056997
9	–0.155534	–0.021933	–0.089497	…	0.718027	–0.056997
10	–0.155534	–0.021988	–0.096896	…	0.718027	–0.056997
11	–0.155534	–0.020685	–0.081344	…	0.718027	–0.056997
12	–0.155534	–0.021988	–0.096896	…	–1.392705	–0.056997
13	–0.155534	–0.021988	–0.096896	…	–1.392705	–0.056997
14	–0.12924	–0.020353	17.067107	…	0.718027	–0.056997
15	–0.155534	–0.021248	0.073236	…	0.718027	–0.056997
16	–0.155534	–0.021538	–0.065839	…	0.718027	–0.056997
17	–0.155534	–0.021468	0.001601	…	0.718027	–0.056997
18	–0.155534	–0.021893	–0.094822	…	0.718027	–0.056997
19	–0.155534	–0.021988	–0.096896	...	–1.392705	–0.056997

5.2 *Identification of deviations in the features between attack and normal datasets using t-test value*

A t-test is the inferential statistics used to identify the significant difference between the mean of two groups, which may be associated in some features. We normally specify the level of probability or alpha level to be 0.05. The result of test statistic value is to be compared with critical values to know if our result falls within the acceptable alpha level of 0.05. Based on statistical analysis, the significant features are identified between normal and attack dataset using t-test. The attack and normal datasets are shown in Table 5 and Table 6 respectively. T-test value can be identified through Table 7.

Table 5 Attack dataset (DoS)

	Duration	*Source bytes*	*Destination bytes*	*(Flag value)* *flag_SF*	*(Flag value)* *flag_SH*
0	–0.15553	–0.020888	–0.096896	0.718027	–0.056997
1	0.416546	0.140761	–0.096849	–1.392705	–0.056997
2	–0.15553	0.093373	0.294926	0.718027	–0.056997
3	5.123238	–0.021988	–0.094822	0.718027	–0.056997
4	5.595115	–0.021988	–0.096189	0.718027	–0.056997

Table 6 Normal dataset

	Duration	*Source bytes*	*Destination bytes*	*(Flag value)* *flag_SF*	*(Flag value)* *flag_SH*
0	–0.15553	–0.021988	–0.096896	–1.392705	–0.056997
1	–0.15553	–0.021988	–0.096896	–1.392705	–0.056997
2	–0.15411	0.005473	–0.096896	0.718027	–0.056997
3	–0.15553	–0.021946	–0.096896	0.718027	–0.056997
4	–0.15482	–0.021988	–0.096189	–1.392705	–0.056997

Table 7 t-score

	service_ecr_i	*Icmp-protocol*	*flag_RSTR*
p-value	0	0	0
T-score/t-test value (squared)	5,891.076	3,557.0347	3,302.3981
	Duration	*Service login*	*service_daytime*
p-value	0	0.0425	0.0462
T-score/t-test value (squared)	2,965.4009	4.1167	3.9745

Median, mode and mean (compare the attack/normal)

- *Median:* Median is defined as the middle number in the given sample after arranging the numbers in ascending order.
- *Mode:* Mode is defined as the number that occurs more often in the given sample data.
- *Mean:* Mean is defined as the average of all numbers in the sample. It is sometimes called as arithmetic mean (AM).

Table 8 shows all the features in the dataset which compares the median, mode and mean between attack and normal type.

Table 8 Features (median, mode and mean)

Features	*Description*
Median ()	Median (middle value) of the data sample
Mode ()	Most common value (single mode) of nominal or discrete data
Mean ()	Arithmetic mean ('average') of data

6 Experiment and result (statistical) analysis

Tavallaee et al. (2009) found that the subset of features manifested better performance that are carried out through statistical analysis on corrected dataset using KDD dataset. The graphical representation of each and every feature in the dataset is shown and is classified as normal or attack type (Figure 5 to Figure 14). The features are represented in the graphical view which mentioned the attack and normal type. Red colour indicates 'with attack' and blue indicates 'without attack'.

Figure 5 Duration (see online version for colours)

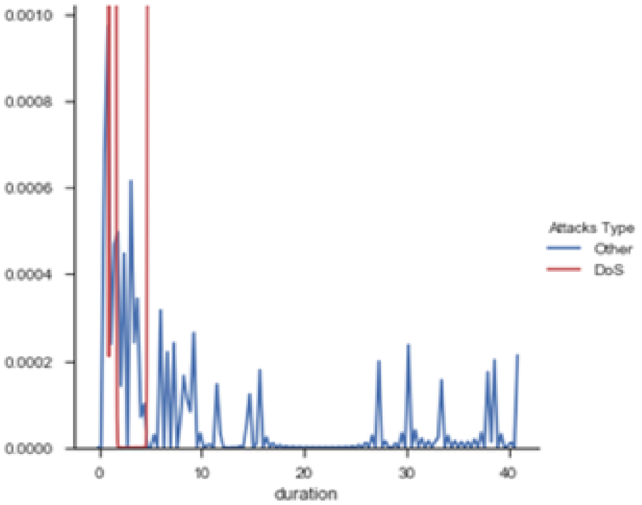


Figure 5 explains about the feature 'duration' and represents the length (number of seconds) of the correlation which isolates normal data as well as attack data.

Figure 6 Logged in (see online version for colours)

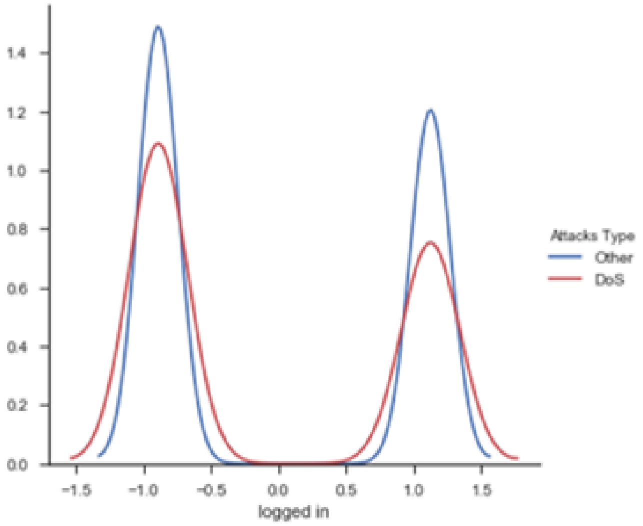
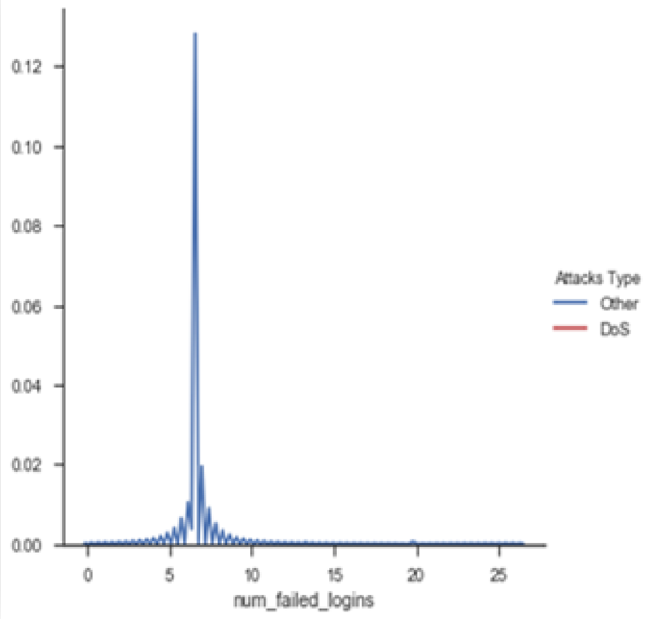


Figure 6 defined regarding ‘logged in’ feature specified through digits 1’s and 0’s. If successfully logged in, then it points 1, otherwise the data represents 0.

The numbers of times legitimate users login failed can be identified entirely have shown in Figure 7.

Figure 7 Number of failed logins (see online version for colours)



The number of wrong fragments occurs in the network while identifying attack can be shown in Figure 8.

Basically, root shell refers to administrative access while running some script in the system. At that time, finding separation of normal and attack data. If root shell is obtained, then the data represents 1 otherwise it represents 0 shown in Figure 9.

Figure 8 Wrong fragment (see online version for colours)

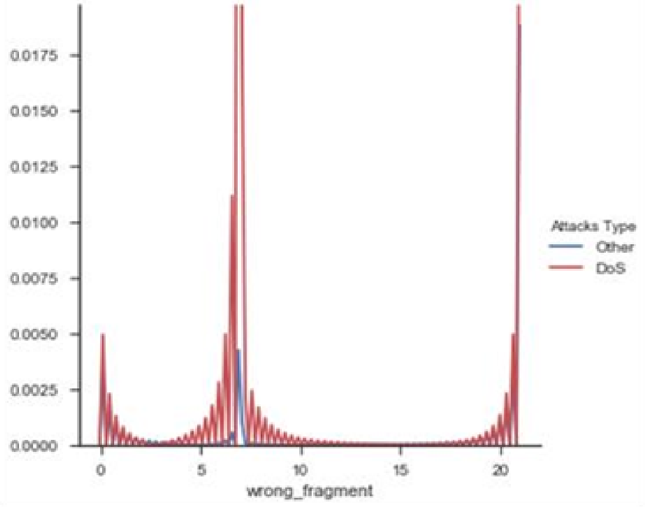
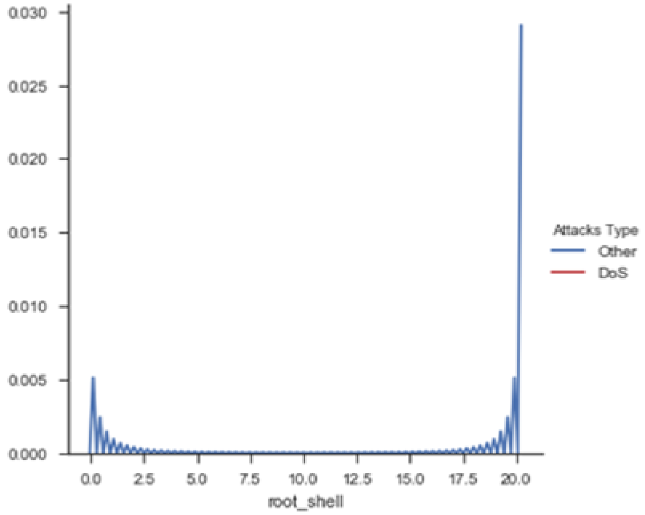
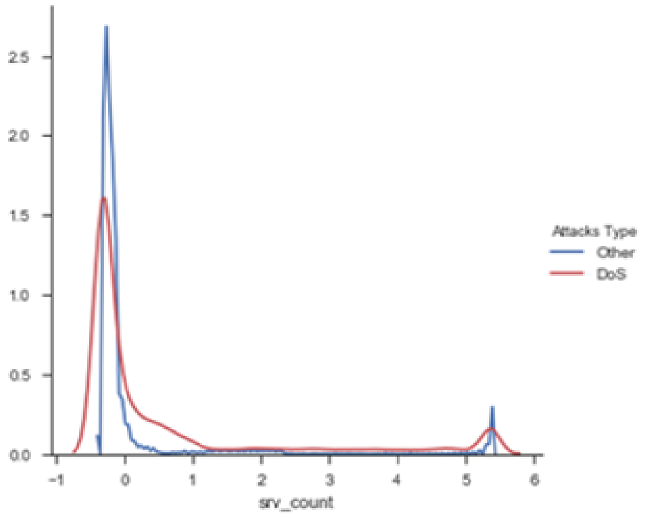


Figure 9 Root shell (see online version for colours)



The service count indicates the number of network service on the destination namely http, and telnet for calculating attack while service occurs shown in Figure 10.

Figure 10 Service count (see online version for colours)



1, if login may be a ‘guest’ login while system running, otherwise it represent 0 can be shown in Figure 11.

Figure 11 Guest login (see online version for colours)

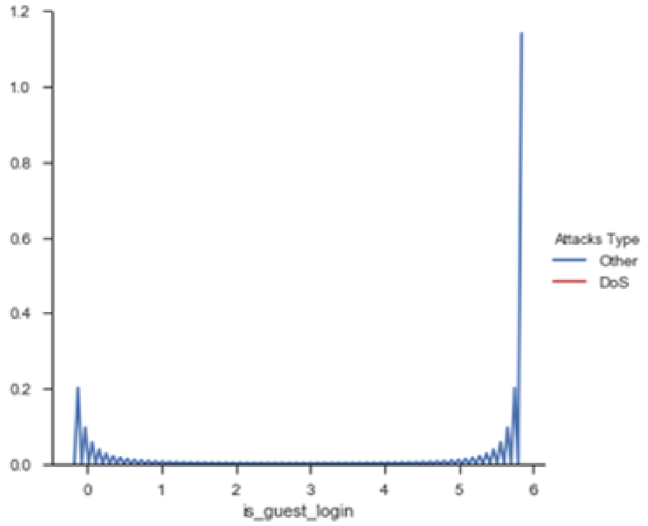


Figure 12 Hot (see online version for colours)

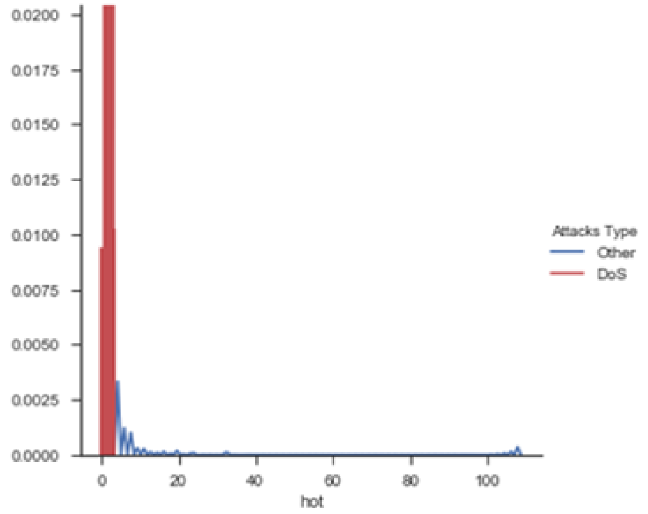


Figure 13 Service error rate (see online version for colours)

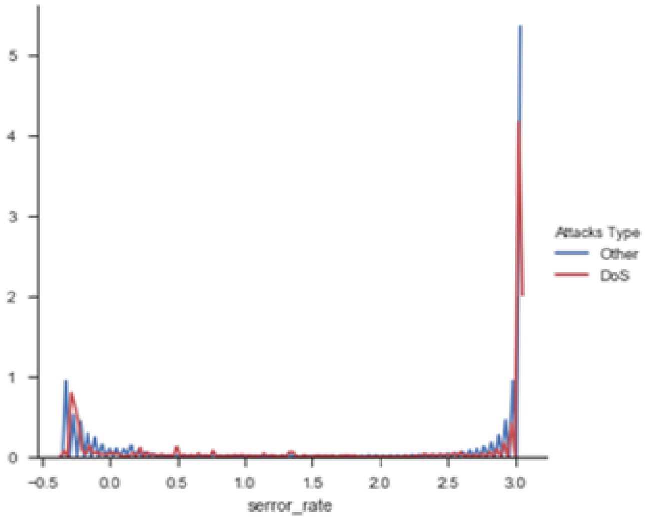
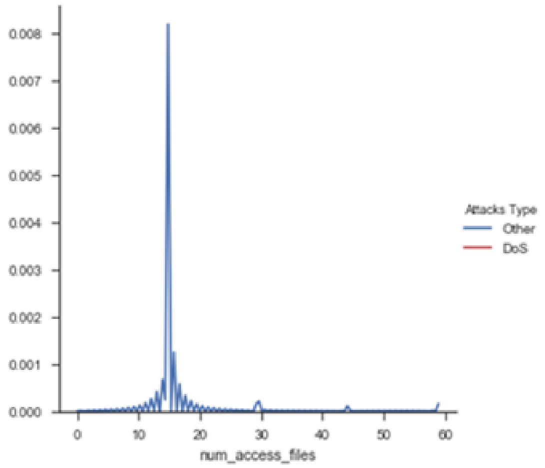


Figure 12 indicates the number of hot indicators while unauthorised users enter the system can be evaluated by isolating the attack and normal data.

Service error rate represents the percentage of connections in the network that are ‘SYN’ errors for finding DOS attack while SYN flood error occurs can be shown in Figure 13.

The total number of operations on file accessing for continuous type can be determined to specify attack data (Figure 14).

Figure 14 Number of access files (see online version for colours)



6.1 *Feature (full feature and selected feature) evaluation using ML algorithm*

Table 9 and Table 10 show the result of each metrics such as (FNR, accuracy, F-score and prediction time) of different algorithms with full and selected feature set. In order to evaluate the performance of proposed classifier, the KDD Cup 99 datasets has been divided into three sets of data 1%, 10% and 100% of dataset.

By analysing Table 11, the decision tree classifier model shows better accuracy with 99.8% and the prediction time is less (2 ms) thus achieving better performance for detecting the attack in the network. Hence, decision tree model achieved better results.

6.2 *Significant feature evaluation using SVC and voting classifier*

The experiments are conducted using KDD Cup DoS attack dataset for classifying attack and normal data extended to some other classifiers for comparison, namely decision tree, support vector, and voting classifier. The experimental outcomes show which voting classifier generates better accuracy of 99.9% that enhances the overall model performance.

Figure 17 explains the metrics comparison of training and testing samples between three classifiers such as decision tree, SVC and voting classifier. Based on accuracy score as 99.9% in testing phase, the voting classifier achieves better performance.

Table 9 Finding median, mode, mean to classify standard or attack (malware)

	(Attack) median	*(Normal) median*	*(Attack) mode*	*(Normal) mode*	*(Normal) mean*	*(Attack) mean*
Duration	–0.155534	–0.155534	–0.155534	–0.155534	–0.128375	0.905599
Destination bytes	–0.096896	–0.092513	–0.096896	–0.096896	0.006598	–0.046545
Wrong fragment	–0.059104	–0.059104	–0.059104	–0.059104	–0.014352	0.101247
Hot	–0.113521	–0.113521	–0.113521	–0.113521	–0.020402	0.143924
Failed login numbers	–0.143999	–0.143999	–0.143999	–0.143999	0.020413	–0.143999
logged in	–0.890373	–0.890373	–0.890373	–0.890373	0.009659	–0.06814
Root_shell	–0.049453	–0.049453	–0.049453	–0.049453	0.00701	–0.049453
Number of access files	–0.052318	–0.052318	–0.052318	–0.052318	0.007416	–0.052318
Guest login	–0.171071	–0.171071	–0.171071	–0.171071	0.024251	–0.171071
Service count	–0.237191	–0.293333	–0.338246	–0.338246	–0.082504	0.582006
Service error rate	–0.348468	–0.348468	–0.348468	–0.348468	–0.031221	0.220242
Service serror_rate	–0.34739	–0.34739	–0.34739	–0.34739	–0.031179	0.219943
Rerror rate	–0.573079	–0.573079	–0.573079	–0.573079	0.011616	–0.081943
Service rerror_rate	–0.565054	–0.565054	–0.565054	–0.565054	0.012223	–0.086222
Same service rate	0.629488	0.629488	0.629488	0.629488	–0.087018	0.613854
Different service rate	–0.363035	–0.363035	–0.363035	–0.363035	0.044485	–0.313814
Service different host rate	–0.386963	–0.386963	–0.386963	–0.386963	0.037516	–0.26465
Destination host count	0.650093	0.650093	0.650093	0.650093	–0.080653	0.56895
Destination host service rate	0.932618	–0.239311	1.022079	1.022079	–0.078314	0.552449
Destination host same service rate	0.852184	0.622657	0.89809	0.89809	–0.067671	0.477372
Destination host different service rate	–0.364909	–0.319601	–0.410217	–0.410217	0.01584	–0.111742
Destination host service source port rate	–0.431856	–0.431856	–0.431856	–0.431856	–0.046655	0.329122
Destination host service destination port rate	–0.22998	–0.22998	–0.22998	–0.22998	0.021184	–0.149441
Destination host serror rate	–0.358118	–0.358118	–0.358118	–0.358118	–0.010564	0.074519
Destination host service serror rate	–0.35275	–0.35275	–0.35275	–0.35275	–0.02149	0.151594
Destination host rerror rate	–0.421943	–0.602719	–0.602719	–0.602719	0.009151	–0.064553
Destination host service rerror rate	–0.540537	–0.565483	–0.565483	–0.565483	0.036619	–0.258324
Flag value SH	0.718027	0.718027	0.718027	0.718027	0.008855	–0.062469
Duration	–0.056997	–0.056997	–0.056997	–0.056997	0.00808	–0.056997

Table 10 Evaluation of full feature set

	Gradient boosting classifier			*Decision tree classifier*			*K-neighbour classifier*		
	0	*1*	*2*	*0*	*1*	*2*	*0*	*1*	*2*
FNR (testing value)	0.16	0.03048	0.00762	0.14286	0.02667	0.00381	0.29905	0.01714	0
FNR (training value)	0.05405	0	0	0.05405	0	0	0.24324	0	0
Testing (accuracy)	0.97028	0.99556	0.99867	0.97317	0.99357	0.99845	0.95387	0.99556	0.99956
Training (accuracy)	0.98667	1	1	0.98667	1	1	0.95333	0.99333	1
F-score (test value)	0.86811	0.98073	0.99428	0.88149	0.97241	0.99335	0.77966	0.98099	0.9981
F-score (training value)	0.94595	1	1	0.94595	1	1	0.8	0.97368	1
Prediction time (testing)(s)	0.009	0.01	0.01	0.003	0.005	0.002	0.178	0.897	12.845
Prediction time (training)(s)	0.091	0.422	4.589	0.003	0.013	0.15	0.002	0.007	1.466

Table 11 Evaluation of selected feature set

	Gradient boosting classifier			*Decision tree classifier*			*K-neighbour classifier*		
	0	*1*	*2*	*0*	*1*	*2*	*0*	*1*	*2*
FNR (testing value)	0.142857	0.009524	0.007619	0.184762	0.030476	0	0.299048	0.017143	0.001905
FNR (training value)	0.054054	0	0	0.054054	0	0	0.243243	0	0
Testing (accuracy)	0.978709	0.996895	0.997782	0.972721	0.995121	0.998669	0.954092	0.995786	0.999335
Training (accuracy)	0.986667	1	1	0.99	1	1	0.953333	0.996667	1
F-score (test value)	0.903614	0.986717	0.990494	0.874362	0.978846	0.994318	0.780488	0.981922	0.997146
F-score (training value)	0.945946	1	1	0.958904	1	1	0.8	0.986667	1
Prediction time (testing)(s)	0.011	0.011	0.009	0.002	0.003	0.002	0.126	0.578	7.062
Prediction time (training)(s)	0.091	0.322	3.411	0.003	0.016	0.124	0.003	0.005	0.563

Table 12 Dataset evaluation using another three different classifiers

	Decision tree classifier			*Support vector classifier*			*Voting classifier*		
	0	*1*	*2*	*0*	*1*	*2*	*0*	*1*	*2*
FNR (testing value)	0.009998	0.043010	0.488122	0.011003	0.164040	5.129849	0.022979	0.225061	5.717496
FNR (training value)	0.012005	0.011586	0.009105	0.117028	0.363096	0.872220	0.317253	0.507123	1.099283
Testing (accuracy)	0.990000	1.000000	1.000000	0.943333	1.000000	0.996667	0.946667	1.000000	1.000000
Training (accuracy)	0.973387	0.995121	0.998669	0.954979	0.996230	0.998891	0.954092	0.995121	0.999778
F-score (test value)	0.958904	1.000000	1.000000	0.721311	1.000000	0.986667	0.733333	1.000000	1.000000
F-score (training value)	0.877551	0.978846	0.994318	0.769056	0.983763	0.995261	0.759582	0.978641	0.999049
Prediction time (testing)(s)	0.054054	0.000000	0.000000	0.405405	0.000000	0.000000	0.405405	0.000000	0.000000
Prediction time (training)(s)	0.180952	0.030476	0.000000	0.356190	0.019048	0.000000	0.377143	0.040000	0.000000

Notes: Here,

0 represents 1% training and testing dataset.
1 represents 10% training and testing dataset.
2 means 100% training and testing dataset.

Figure 15 Full feature set (see online version for colours)

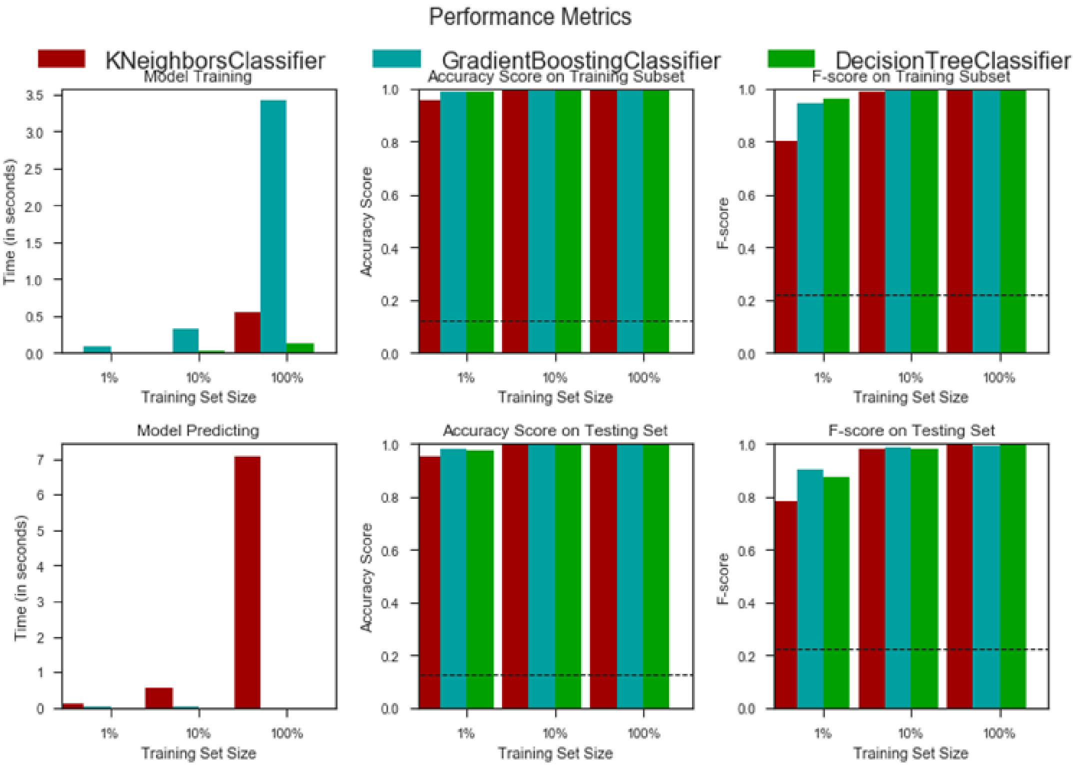


Figure 16 Selected feature set (see online version for colours)

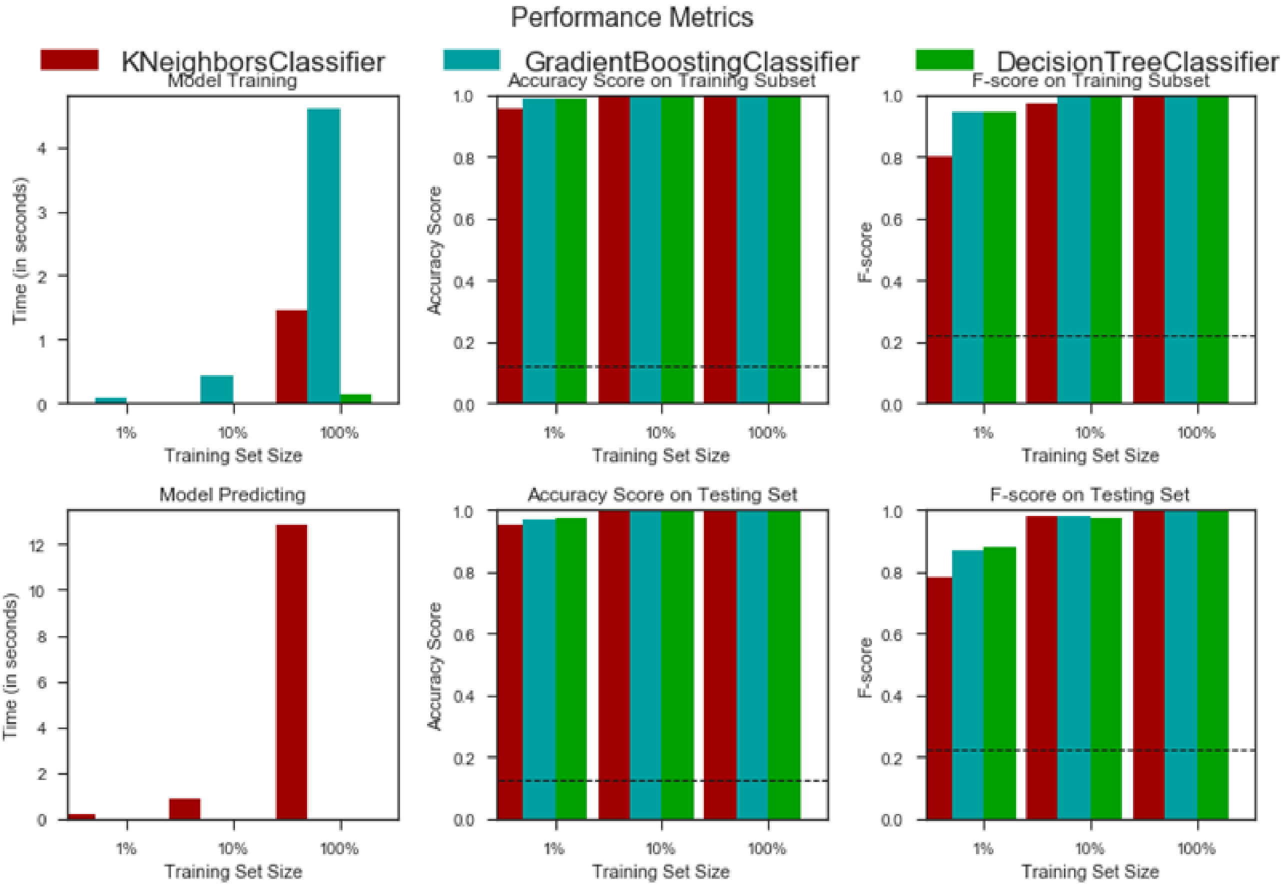


Figure 17 Significant features using SVC and voting (see online version for colours)

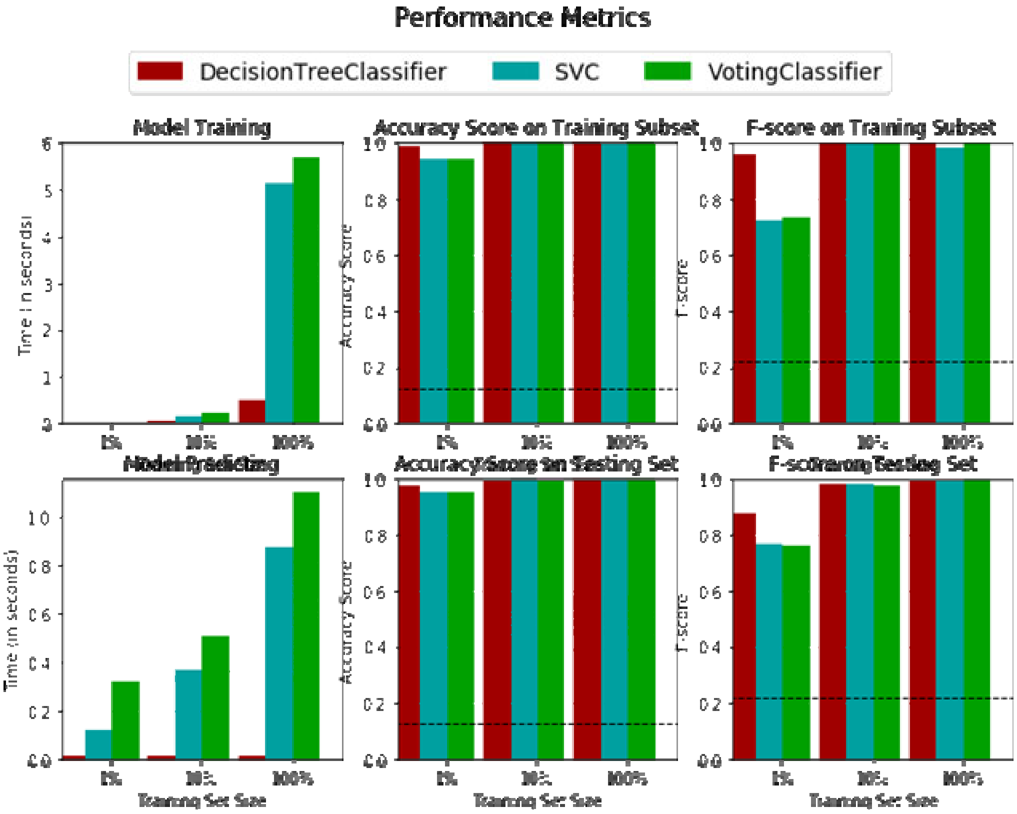
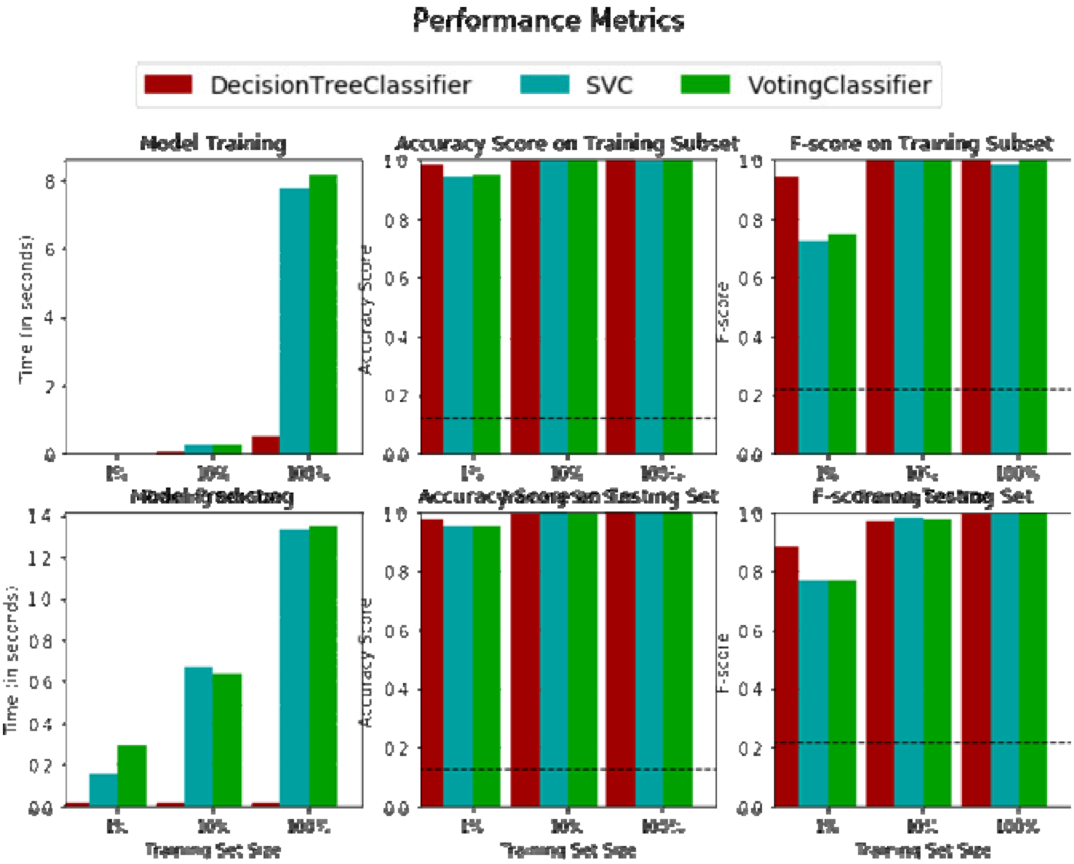


Figure 18 Full feature set using SVC and voting classifiers (see online version for colours)



The significant features extracted from the sample dataset for splitting normal and malicious data in a very good manner via decision tree, SVC and voting classifier are depicted in Figure 18.

6.3 *Allusive study of existing and proposed work*

Norouzian and Merati (2011) shows the experimental results with an accuracy rate of 90.78%. Karimazad and Faraahi (2011) shows the results with an accuracy rate of 96% for detecting the attack in the network. Alkasassbeh et al. (2016) explains the detection of DoS attack and the result shows that MLP produce higher accuracy rate of 98.63%. The comparison between existing and proposed work tabulation is shown in Table 13.

Hence, in this survey training and testing data using supervised learning algorithm can have the ability to detect the attacks in the network and eliminate the intruders in the system via statistical analysis as well as DM algorithms. Here X-axis refers the existing algorithm used by different researchers and Y-axis denotes accuracy rate.

Table 13 Allusive study of existing and proposed work

Author name	*Algorithm used for attack classification*	*Accuracy rate*
Norouzian and Merati (2011)	MLP	90.78%
Karimazad and Faraahi (2011)	RBF	96%
Alkasassbeh et al. (2016)	MLP, naïve Bayes, random forest	98.63%
Zareapoor and Shamsolmoali (2018)	PART algorithm	97%
Proposed work	Gradient boosting, decision tree, K-nearest neighbour	99.86%
Proposed work 2	Decision tree, SVC, voting classifier	99.9%

6.4 *Allusive table of performance survey in IDS by using various perspectives used by different researchers*

Table 14 is an allusive study which elaborates the performance of IDS perspectives investigated based on metrics such as accuracy, DR, FNR, FAR and FPR. The proposed work mainly focused on metrics such as accuracy, FNR, F-score and precision for achieving better performance by detecting attacks in the network.

Table 14 Allusive table of performance survey through metrics calculation

Researcher	*Algorithm*	*Datasets*	*Metrics calculated*				
			FNR/FAR	*Accuracy (%)*	*DR*	*F-score*	*FPR*
Wani et al. (2020)	K-means		-	0.958	-	0.959	-
	Decision tree		-	0.942	-	0.96	-
	RF		-	0.997	-	0.998	-
	SVM		-	0.976	-	0.996	-
	Naive Bayes		-	0.98	-	0.826	-
	C4.5		-	0.99	-	0.988	-
Yusof et al. (2016)	KNN	WEKA	-	96.6	0.26	0.97	-
	SVM		-	96.5	0.23	0.96	-
	K-means		-	96.7	0.20	0.97	-
	Naive Bayes		-	92.9	0.52	0.97	-
	Fuzzy C-mean		-	98.7	0.15	0.99	-
	Decision tree		-	95.6	0.25	0.96	-
Mand and Reed (2012)	Signature	-	-	-	Fast	-	Low
	Anomaly		-	-	Varies	-	High
Ingre and Yadav (2015)	Levenberg-Marquardt (binary class)	NSL-KDD	-	81.2	-	-	3.23
	Quasi Newton back propagation (five class)		-	79.9	-	-	-
Karimazad and Faraahi (2011)	UCLA		-	98.2	-	-	-
	Simulated network		-	96.5	-	-	-
Alkasassbeh et al. (2016)	MLP	Novel dataset	-	98.63	-	-	-
	Naive Bayes		-	96.91	-	-	-
	Random forest		-	98.02	-	-	-
Panda and Patra (2009)	Random forest	KDD Cup 1999	1.2		91.7	76.37	1.72
	AdaBoost		1.43		91.4	94.8	4.97
	Naïve Bayes		0.02	-	99	99.2	26
Mukkamala and Sung (2003)	Support vector machine	DARPA	-	99.25	-	-	-
Nadiammai and Hemalatha (2014)	Efficient data adapted algorithm		0.18	98.12	-	-	-
Kaur et al. (2017)	Anomaly-based	KDD Cup	High		Medium		High
	Signature-based		Medium		High (known attacks)		Very low
	Hybrid		Low		High		Low
Thakare and Kaur (2017)	Multivariate correlation analysis	KDD Cup 99	-	High	-	-	-
Pervez and Farid (2014)	Support vector machine	NSL-KDD Cup 99	-	91 (3 features) 99 (36 features)	-	-	-
Aggarwala and Sharmab (2015)	Basic	KDD	3.22	-	80.78	-	-
	Content		3.47	-	79.42	-	-
	Traffic		3.22	-	78.59	-	-
	Host		3.22	-	76.54	-	-
Harikumar et al. (2017)	Apriori	KDD Cup	-	90	-	-	-
Callegari et al. (2018)	K-PCA	Real data	-	-	Low	-	-
Yu and Hao (2007)	Multi objective generic algorithm	DARPA	-	-	99.95	-	0.04
	E-IMOGA		-	-	99.98	-	0.03
Zargari and Voorhis (2012)	Random forest algorithm	NSL-KDD	-	-	High	-	-

7 Conclusions and future scope

In an application of IDS, decision tree classifier performs with better accuracy and shorter prediction time compared to other machine learning techniques described in Salo et al. (2018) and Zareapoor and Shamsolmoali (2018). In this paper, 2,799 DoS attacks found out of 22,544 total sample data whereas normal data was 19,745 and the percentage of attacks is 12.4%. Even though, fewer attacks available could hack the system condescendingly. So, finding least amount of attack can be done via predictive analysis in this projected work. For 100% training data and testing, the metric values of FNR, accuracy, F-score and prediction time for decision tree classifier are 0, 99.9%, 0.9943 and 2 ms respectively. From these results the F-score was greater in KNN classifier however its prediction time was longer compared to all other classifiers.

This survey undergoes training and testing method using supervised machine learning algorithm for detecting DoS attack, finally carry out correlation whatever the attacks founded in the network which might be either normal or malicious. Hence the decision tree classifier achieves a shorter prediction time and also improved F-score and accuracy (99.8%) (Table 11). Moreover, another experimental outcome reveals that voting classifier generates high accuracy rate as 99.9% by training and testing phase through supervised machine learning algorithm. Hence, future scope will work even in unidentified situation or DoS attack not defined in any time, the method has to detect the attack using unsupervised machine learning approach namely clustering is a major challenge for preventing the cyber security issues. If these statistical analyses can be implemented in hardware, we could have ability to find the real-time detection of intruders, which will be supportive, compassionate to industries like IT companies, banking and insurance, with intention of serving in cyber security domain.

References

Aggarwala, P. and Sharmab, S.K. (2015) 'Analysis of KDD dataset attributes – class wise for intrusion detection', *Procedia Computer Science*, Vol. 57, pp.842–851.

Alkasassbeh, M., Al-Naymat, G., Hassanat, A.B.A. and Almseidin, M. (2016) 'Detecting distributed denial of service attacks using data mining techniques', *International Journal of Advanced Computer Science and Applications (IJACSA)*, Vol. 7, No. 1, pp.436–445.

Almutairi, S., Mahfoudh, S., Almutairi, S. and Alowibdi, J.S. (2020) 'Hybrid botnet detection based on host and network analysis', *Journal of Computer Networks and Communications*.

Bose, I. and Mahapatra, R.K. (2001) 'Business data mining a machine learning perspective', *Information & Management*, Vol. 39, No. 3, pp.211–225.

Callegari, C., Donatini, L., Giordano, S. and Pagano, M. (2018) 'Improving stability of PCA-based network anomaly detection by means of kernel-PCA', *Int. J. Computational Science and Engineering*, Vol. 16, No. 1, pp.9–16.

Chakraborty, S., Kumar, P. and Sinha, B. (2019) 'A study on DDoS attacks danger and its prevention', *IJRAR*, Vol. 6, No. 2, pp.10–15.

Fayyad, U. (1997) 'Knowledge discovery in databases: an overview', *Relational Data Mining*, pp.28–47.

Harikumar, S., Dilipkumar, D.U. and Kaimal, M.R. (2017) 'Efficient attribute selection strategies for association rule mining in high dimensional data', *Int. J. Computational Science and Engineering*, Vol. 15, Nos. 3/4, pp.201–213.

Hung, C-H. and Sun, H-M. (2018) 'A botnet detection system based on machine-learning using flow-based features', *The Twelfth International Conference on Emerging Security Information, Systems and Technologies, SECURWARE 2018*.

Ingre, B. and Yadav, A. (2015) 'Performance analysis of NSL-KDD dataset using ANN', *2015 International Conference on Signal Processing and Communication Engineering Systems*, Guntur, pp.92–96.

Karimazad, R. and Faraahi, A. (2011) 'An anomaly-based method for DDoS attacks detection using RBF neural networks', in *2011 International Conference on Network and Electronics Engineering, IPCSIT*, Vol. 11.

Kaur, P., Kumar, M. and Bhandari, A. (2017) 'A review of detection approaches for distributed denial of service attacks', *Systems Science & Control Engineering: An Open Access Journal*.

Khatak, A. and Maini, R. (2016) 'Comparative analysis of different denial of service attacks', *The International Journal of Engineering And Science (IJES)*, Vol. 5, No. 5, pp.32–35.

Khraisat, A., Gondal, I., Vamplew, P. and Kamruzzaman, J. (2019) 'Survey of intrusion detection systems: techniques, datasets and challenges', *Cyber Security*, Vol. 2, No. 1, Springer.

Lee, W., Stolfo, S.J. and Mok, K.W. (1999) 'A data mining framework for building intrusion detection models', in *Proceedings of the 1999 IEEE Symposium on Security and Privacy*.

Mand, A. and Reed, M.J. (2012) 'Methodologies for detecting dos/ddos attacks against network servers', in *The Seventh International Conference on Systems and Networks Communications, ICSNC Semi Markov Models*.

Manso, P., Moura, J. and Serrão, C. (2019) 'SDN-based intrusion detection system for early detection and mitigation of DDoS attacks', *Journal of Information*, Vol. 10, No. 3, pp.1–17.

Milenkoski, A., Vieira, M., Kounev, S., Avritzer, A. and Payne, B.D. (2015) 'Evaluating computer intrusion detection systems: a survey of common practices', *ACM Comput. Surv.*, Vol. 48, No. 1, pp.1–41.

Milliken, J., Selis, V., Yap, K.M. and Marshall, A. (2013) 'Impact of metric selection on wireless deauthentication DoS attack performance', *IEEE Wireless Communications Letters*, Vol. 2, No. 5, pp.571–574 [online] https://doi.org/10.1109/WCL.2013.072513.13042.

Mukkamala, S. and Sung, A.H. (2003) 'Detecting denial of service attacks using support vector machines', *The IEEE International Conference on Fuzzy Systems*,

Nadiammai, G.V. and Hemalatha, M. (2014) 'Effective approach toward intrusion detection system using data mining techniques', *Egyptian Informatics Journal*, Vol. 15, No. 1, pp.37–50.

Norouzian, M.R. and Merati, S. (2011) 'Classifying attacks in a network intrusion detection system based on artificial neural networks', in *2011 13th International Conference on Advanced Communication Technology (ICACT)*, IEEE, pp.868–873.

Panda, M. and Patra, M.R. (2009) 'Evaluating machine learning algorithms for detecting network intrusions', *International Journal of Recent Trends in Engineering*, May, Vol. 1, No. 1, pp.472–477.

Pervez, M.S. and Farid, D.M. (2014) 'Feature selection and Intrusion detection classification in NSL-KDD cup 99 dataset employing SVMs', *The 8th International Conference on Software Knowledge Information Management and Applications (SKIMA 2014)*, Dhaka, pp.1–6.

Reshamwala, A. and Mahajan, S. (2012) 'Prediction of DoS attack sequences', *Proceedings – 2012 International Conference on Communication, Information and Computing Technology, ICCICT*, pp.1–5 [online] https://doi.org/10.1109/ICCICT.2012.6398148.

Salo, F., Injadat, M.n., Nassif, A.B., Shami, A. and Essex, A. (2018) 'Data mining techniques in intrusion detection systems: a systematic literature review', *IEEE*, Vol. 6, pp.56046–56058.

Siris, V.A. and Papagalou, F. (2006) 'Application of anomaly detection algorithms for detecting SYN flooding attacks', *Computer Communications*, Vol. 29, No. 9, pp.1433–1442.

Tavallaee, M,. Bagheri, E., Lu, W. and Ghorbani, A. (2009) 'A detailed analysis of KDD Cup 99 dataset', in *Proc. 2nd IEEE Computer Intelligence Security Defence Appl.*, pp.1–6.

Thakare, S.S. and Kaur, P. (2017) 'Denial-of-service attack detection system', *Proc. - 1st Int. Conf. Intell. Syst. Inf. Manag. ICISIM 2017*, January. August, pp.281–285.

Tsai, C.F., Hsu, Y-F., Lin, C-Y. and Lin, W-Y. (2009) 'Intrusion detection by machine learning: a review', *Expert Syst. with Appl.*, Vol. 36, No. 10, pp.11994–12000..

Wani, A.R., Rana, Q.P. and Pandey, N. (2020) 'Machine learning solutions for analysis and detection of DDoS attacks in cloud computing environment', *International Journal of Engineering and Advanced Technology (IJEAT)*, February, Vol. 9, No. 3, pp.2205–2209, ISSN: 2249–8958.

Yihunie, F., Odeh, A. and Abdelfattah, E. (2018) 'Analysis of ping of death DoS and DDoS attacks', *2018 IEEE Long Isl. Syst. Appl. Technol. Conf. LISAT 2018*, May, pp.1–4.

Yu, X., Han, D., Du, Z. and Tian, Q. (2019) 'Design of DDoS attack detection system based on intelligent bee colony algorithm', *Int. J. Computational Science and Engineering*, Vol. 19, No. 2, pp.223–231.

Yu, Y. and Hao, H. (2007) 'An ensemble approach to intrusion detection system based on improved multi-objective genetic algorithm', *Journal of Software*, June, Vol. 18, No. 6, pp.1369–1378.

Yusof, A.R., Udzir, N.I. and Selamat, A. (2016) 'An evaluation on KNN-SVM algorithm for detection and prediction of DDoS attack', *29th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, IEA/AIE 2016*, 2–3 August 2016, p.34.

Zareapoor, M. and Shamsolmoali, P. (2018) 'Advance DDOS detection and mitigation technique for securing cloud', *Int. J. Computational Science and Engineering*, Vol. 16, No. 3, pp.303–310.

Zargari, S. and Voorhis, D. (2012) 'Feature selection in the corrected KDD-dataset', *2012 Third International Conference on Emerging Intelligent Data and Web Technologies*, Bucharest, pp.174–180.