Journal of Scientific & Industrial Research
Vol. 81, March 2022, pp. 276-286

DDoS Attack Detection Using Hybrid Machine Learning Based IDS Models

S Sumathi[1]*, R Rajesh[2] and N Karthikeyan[3]

[1]University V O C College of Engineering, Anna University, Thoothukudi 628 003, Tamilnadu, India
[2]Indian Institute of Technology Madras, Chennai 600 036, Tamilnadu, India
[3]Syed Ammal Engineering College, Ramanathapuram 623 502, Tamilnadu, India

Received 22 December 2021; revised 13 January 2022; accepted 15 February 2022

A kind of cyber-attack that severely paralyzes the victim server by injecting illegitimate packets of data is a DDoS attack, which is progressive in nature. Therefore its detection is a highly tedious task. Hence, IDS models are developed to detect this attack efficiently, based on machine learning algorithms such as C4.5, SVM, and KNN classifier algorithms and 10-fold cross validation techniques. The NSL-KDD bench mark dataset is employed to validate the models experimentally. A 10-fold cross validation technique is used to select the trend features, and ten trial runs are made to avoid biased output. The classic SVM classifier model reported better accuracy, but the precision and sensitivity of the C4.5 classifier algorithm are better than that of SVM and KNN models. In order to improve the performance of the machine learning based intrusion detection models, an attempt is made to feed the SVM and KNN based IDS model with the features selected by C4.5 classifier algorithm, and the obtained performance metric values are reported. It is evident from the results obtained that the hybrid combination of C4.5 with SVM out performed all other models discussed in this research with an accuracy of 0.9604.

Keywords: Distributed denial of service, K-nearest neighbors, NSL-KDD dataset, Support vector machine

Introduction

Cloud computing provides the software and hardware computing resources based on the user's demand and is paid based on usage. It is diverse in nature; various companies and organizations have their resource into a standard traffic system. It utilizes a dynamic scaling strategy to provide highly reliable and flexible service to the users. Instead of the client-server mechanism, the cloud uses a virtualization strategy that offers easy switching of servers into various virtual machines. The cloud has a significant advantage of common resource sharing. So, the users need not buy any third-party applications and can access the resource over the cloud at any time. The immense growth of information and technology has increased the online services provided by organizations of any type, service, size, and industry at consumers' doorsteps. Governments and enterprises migrated their whole or most IT infrastructure into the cloud.[1] Hence a survey is conducted in which various DDoS mitigation solutions are proposed based on computing models. The solution involves flow of multilevel information as well as resource management during the attack. The development of cloud computing and the Internet of Things (IoT) has facilitated the delivery of on-demand services for all users. It offers huge data storage facilities over the internet and can be accessed at any time and location globally.

The Cloud computing architecture has two major blocks: the front and back end. The user side is the front end with the client system with the applications to access the resource from the cloud, and the back end runs the cloud system of sharing resources. The system may have similar interfaces for all users or some domain-specific applications provided based on the user's requirement. The applications are processed and controlled by a dedicated central server that monitors the safe and secure operation in the network. The back end has a large storage system to store the user information as a backup, enabling the user to utilize it in any case of loss of information. Cloud computing architecture is divided into four layers, namely hardware, infrastructure, platform, and application layers public and private sectors are integrated into a common service provider; this diverse nature pays the way for illegal cyber-attacks such as Denial of Service (DoS), Distributed Denial of Service (DDoS), Drive-by attack, SQL injection attack, Man-in-the-Middle (MitM) attack and so on.

*Author for Correspondence
E-mail: sumock123@yahoo.com

The DDoS attack is a kind of severe intrusion intended to paralyze the victim in the network by sudden flooding of attack packets, introducing zombies to cause traffic congestion over the service. It is a big challenge to distinguish illegitimate traffic from legitimate traffic. The main aim of this research is the efficient detection of this attack using hybrid machine learning models. The combination of machine learning algorithms in detecting DDoS attacks is tested on the NSL-KDD dataset. A dataset is initially tested using C4.5, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) with a 10-fold cross validation technique. Further, C4.5 is combined with SVM along with10-fold cross validation technique to validate the dataset in detecting DDoS attacks. The paper is organized as follows: First section -Introduction, Second section-Related works, Third section-Materials and methods, Fourth section - Results and discussion, and Fifth section-Conclusions.

Review of Distributive Denial of Service Attacks

The DDoS uses DoS as the basic module. In a DDoS attack, an attacker aims to deplete network infrastructure, capacity, or compute resources by overwhelming it with requests.[2] In this, a discussion is made about the future of DDoS, commercial DDoS solutions as well as the role of machine learning methods in DDoS attack detection in a cloud environment. The botnets are types of agents flooding into the network, and they pay the way for other attacks such as DOS, phishing, spoofing, etc.

The typical architecture of the DDoS attack has five layers of the framework as shown in Fig 1. The intruders like spoofers, botnets, and eavesdroppers occupy the intruder layer. The layer occupied between the attacker and the victim is the master layer or handlers. The slaves or zombies are the additional layers between the attacker and victim.

In the scanning phase, many computing systems present over a network are scanned with the help of attack software. The exploitation phase recognizes the vulnerable hosts and notes the list of conceded hosts. The propagation phase scans the vulnerable host systems by the handlers and compromises them to act as zombie/Daemon. The zombie runs the special software that generates and floods the stream of illegitimate traffics into the target server. The three types of propagation that occurred in this phase are central source propagation, back chain propagation, and autonomous propagation. The attack phase

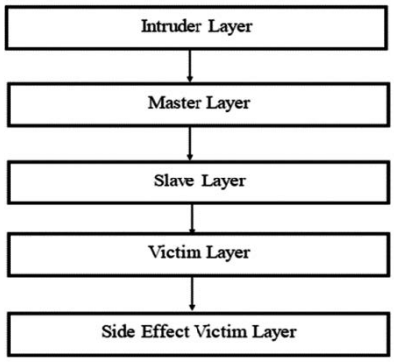

Fig. 1 — Architecture of DDoS attack detection

comprises of multiple zombies in the network, and they start to launch a coordinated attack over the victim system. The DDoS attacks are broadly classified into volume-based attacks, protocol-based attacks, and application layer-based attacks. SYN Flood Attack is one form of distributed denial of service attack that attains the handshake process of TCP.[3] In this, trained neural network and a novel binary fruit fly optimization algorithm is used in the prediction of SYN flood attack. The attacker continuously sends SYN packets to the server during the attack to open multiple half sessions, and the final ACK is not established. In an ICMP flooding attack, the victims are overwhelmed by ICMP echo requests.[4] In this, an attack detection system is designed and implemented for detecting DDoS flooding attacks like TCP, HTTP, UDP and ICMP flooding attacks in SDN network. The ICMP flood aimed to exhaust the bandwidth of the victim server, and this attack is otherwise termed as Smurf attack. DNS flooding attacks send a large number of nonexistent domain-name requests to the DNS, causing DNS server failure.[5] In this study, a Flow Differentiation Detector (FDD) is proposed which is deployed in the SDN controller Open Daylight to detect hybrid DDoS attacks with detection accuracy above 90%.

A botnet is a group of software and hardware agents that acts anonymously to attack the network and ultimately control the computing system of the victim. Botnets are often utilized for sending spam, stealing data, and performing DDoS attacks.[6] In this, best machine learning methods from supervised, unsupervised and regression learning is chosen to create an ensemble learning model for botnet detection in IOT with minimum feature requests. The kind of attack in which the hacker post requests that appear to be legitimate, but it exhausts maximum resources though it utilizes less bandwidth.

The slow DDoS attacks generally target the application layer in which the attacker sends bogus requests to the server impersonating as a legitimate request.[7] Here, an anomaly detection system is proposed to detect slow HTTP DDoS attack in the application layer. The zero-day attack is a kind of DDoS attack that includes vulnerabilities that are not yet been repaired.

Review of Intrusion Detection System

The severity and characteristics of cyber-attacks presented in the previous sections indicate that the hackers are very clever in causing severe damage to network resources without the victim's knowledge. Intrusion identification and mitigation are the serious research areas to be considered, and numerous detection and mitigation techniques have been developed in the past decades. In traditional networks, the hardware and software to detect and mitigate DDoS attacks are expensive and challenging to deploy.[8] In this study, based on flow entries random forest algorithm is used to classify the packets either as normal or attack. The average detection time is 36 ms whereas the average mitigation time is 1179 ms. The IDS systems are classified as signature-based IDS and anomaly-based IDS. The knowledge-based model identifies the abnormal behavior of the network based on the knowledge base of activity of a regular system, and any action that deviated from the normal behavior is flagged as an intrusion. The function of Finite State Machine, Description languages, and Expert systems are based on knowledge-based AIDS.

Related Works

The DDoS attack is a severe problem in cloud computing; the detection and mitigation of intrusion is a challenging task that will affect the functionality of the entire architecture. For this reason, numerous cyber-security measures have been carried out to protect the server from attackers or hackers. The traditional cyber-security methods failed to protect the server against several external unauthorized traffics. So, developing an Intrusion Detection System (IDS) in IoT architecture has become the most critical field of research.

Bhosale *et al.* (2020) have utilized five different classification algorithms to perform intrusion detection on KDD Container 99, and the essential attributes were selected by the feature selection algorithm.[9] In this study, using the performance metrics a comparative study is done on 5 different classifiers namely Naïve Bayes, CNN, SVM, ANN and KNN along with feature extraction technique using KDD container 99 dataset. The experimental results of the Navies Bayes, CNN, SVM, ANN, and KNN models were analyzed, and reported that these models could perform better intrusion classification accuracy than other statistical models. Singhal *et al*. (2020) have introduced Bigdata techniques to address the issue of DDoS attacks detection in the application layer.[10] A review has been done about various machine learning algorithms for DDoS attack detection and mitigation in the cloud computing environment. Roopak *et al*. (2020) have implemented a multi-objective optimization algorithm for feature selection, which has attained 99% system performance, and the total selected features reduced to 90% compared to other methods. Still, the framed multi-objective framework increases the system complexity.[11] Swami *et al*. (2020) have developed a Machine Learning techniques-based Intrusion defense system for Software-defined networking topology.[12] Various DDoS attack behaviors and machine learning approaches were discussed in detail to mitigate the issues. But in this study, the intrusion identification was not made. Dwivedi *et al.* (2020) have presented an evolutionary algorithm-based machine learning intrusion detection system (IDS).[13] A grasshopper optimization algorithm has been utilized to identify the trend features and fed into the developed classifier models such as SVM, decision tree, naive Bayes, and Multilayer Perceptron Neural Network models.

Machine learning classification algorithms for the DDoS attack detection method were developed by Hussain (2020).[(14)] The ML algorithms such as Bayesian Network (BayesNet), Bootstrap Aggregating (Bagging), KNN, Sequential Minimal Optimization (SMO), and Simple Logistic approaches were implemented, and overall performance was analyzed. The obtained performance confirmed that the KNN outperformed all others with better metric values. But in this work, the influence of feature selection was not reported. Doucette *et al.* (2020) have discussed feature selection mechanisms based on Robust PCA.[15] The RPCA was employed on ARMED classification strategy to identify the mischief traffics in DoS. In this work, the influence of hyper-parameter tuning was not done. Semi-supervised learning algorithms based on machine learning technique for intrusion classification in IoTs was developed by Rathore & Park (2018).[(16)] The experimental validation on the

NSL-KDD dataset reported 86.53% accuracy in a short time interval. In this work, the false alarm rate was not considered. Several algorithms were reviewed and discussed their identification ability. Ravi & Shalinie (2020) have presented a learning-driven method for DDoS attack classification and mitigation in the SDN cloud environment.[17] The experimental simulation results achieved 96.28% accuracy in DDoS classification. The effect of feature selection was not discussed in this work. Nesa *et al.* (2018) have differed on non-parametric sequence-based learning algorithms for outlier identification in an IoT environment.[18] The developed algorithm works on both event and error scenarios with a detection accuracy of 99.65% and 98.53%, respectively. This model has achieved better classification accuracy, but other metric analysis was not made. Naïve Bayes Mukherjee & Sharma (2012) proposed Feature Vitality Based Reduction Method in order to identify important reduced input features. Then an efficient classifier naive bayes is applied on reduced datasets for intrusion detection.[19] Selected reduced attributes gave better performance IDS that is efficient and effective for network intrusion detection.

Materials and Methods

Review of C4.5 Classifier Algorithm

The C4.5 is a tree-based classifier algorithm. Decision tree validation is used to see the accuracy of the algorithm model C 4.5 in determining the exemplary Teacher using software RapidMiner.[20] It adopts a normalized gain information technique to split the attributes such that the corresponding subset is stored in each node of the tree. It also adds the advantage of managing missing datasets in the continuous input set. The tree construction mechanism keeps the input data in sub-nodes based on the test criteria. The input dataset is partitioned during the iteration steps based on the attribute test result. The tree growing process is stopped when the same attributes are processed, or the entire data set belongs to the same class. The tree may continue to grow even after the stopping criteria are attained such a condition is called overfitting, so the tree should be pruned to avoid oversizing of the tree. The process of removing the unwanted branches is called pruning. For every iteration, the entropy and the information gain is calculated as follows:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Value\ (A)} \frac{|S_v|}{|S|} Entropy(S_v) \quad \ldots (1)$$

$$Entropy(S) = \sum_{i=1}^{c} -P_i\ log_2\ P_i \quad \ldots (2)$$

where, for all attributes A, $S_v$is the subset of S with attribute value$v$. $P = (p_1, p_2, \ldots, p_n$, the probability distribution and p_j is the set of possible attributes.

The algorithmic steps of C4.5 algorithm as follows:

Step-1: Set the root node with original input dataset. Continue steps 2 to 8 until stopping criteria is attained.

Step-2: The entropy and information gain for all attributes are calculated.

Step-3: The entropy and information gain value is utilized to select the attribute with small entropy and high gain value.

Step-4: Create a tree with a decision node based on the best attribute selected in the previous step.

Step-5: The input dataset is split into subsets based on the selected attributes in the step 4.

Step-6: Steps 1 to 5 are followed for all created subsets until the stopping criteria are attained.

The Pseudo of C4.5 algorithm is as follows:

Inputs: input training samples, target, and non-target attributes
Output: a decision tree
Start,
Create an empty node,
if ($T \leftarrow 0$) →node of failure value
end if
if ($T \leftarrow C$)→node of target attribute value
end if
if (R is empty) →node of majority attribute
end if
while (stopping criteria)
{
The attribute of highest gain ratio value is chosen
node $N$is labeled with chosen attribute
For all test attribute {
Split the input sample $T \rightarrow T_1, T_2 \ldots . T_n$
If ($T_i$is empty) {
leaf node of the majority class in input sample;
}
else {
the target value attained is attached to the leaf node

Return the tree.

Review of K-Nearest Neighbor (KNN) classifier algorithm

K-Nearest Neighbor Algorithm is a supervised learning algorithm that utilizes the feature similarity concept to identify the closeness of incoming data points with the training dataset commonly employed

to perform classification models. In this, the training occurrences are not processed but stored to take a decision on testing samples based on the stored instances. For the given test instance, the most similar instances or the nearest instances are to be estimated. The K-NN classification method has been qualified online and real-time to find user behaviour data coordinating to a specific user group containing the relationship between the similarity of many users and target users from a huge amount of data.[21,22] The algorithmic steps are described as follows:

Step 1: The training and testing dataset is loaded into the model

Step 2: The K value is defined; it represents the number of nearest points near the testing instant.

Step 3: For all testing instants, do steps 4 to 7.

Step 4: The distance between the testing data instant and the training data instances are calculated by the Euclidean method.

Step 5: Sort the distance value attained in Step 4: Then, choose the top K rows from the sorted array.

Step 6: The test point class is assigned based on the most frequent types attained in step 5.

Step 7: Terminate the process.

Review of Support Vector Machine (SVM) algorithm

The SVM classifier algorithm constructs a hyperplane to separate two classes of relevant data; the typical classifier model is shown in Fig. 2 for n-dimensional data where n-1 hyper planes are constructed. The boundary lines are built along with the hyperplane to contemplate the relevant data into the class. The minimum distance between the support vectors should be maximized, which means to attain better classification accuracy, the margin should be maximized. The non-linear dataset suffers from linear separation, so kernel tricks are used to separate the data by constructing non-linear decision boundaries. Support vectors are the closest data points to the hyperplane. The hyper plane is the decision plane in which the data points are classified. Margin is the distance of the closest data points from the decision boundary. In differentiating lung cancer and COPD from controls, Support Vector Machine (SVM) with 3-fold cross-validation outperformed all other classifiers with an accuracy of 92.3% in cross-validation.[23] In the discrimination of lung cancer from controls, the k-nearest neighbors gave an acceptable accuracy, sensitivity, and specificity of 91.3%, 84.4%, and 94.4% respectively. The support vector machine gave better results for COPD discrimination from controls with 90.9% accuracy, 81.6% sensitivity, and 95.8% specificity.[24]

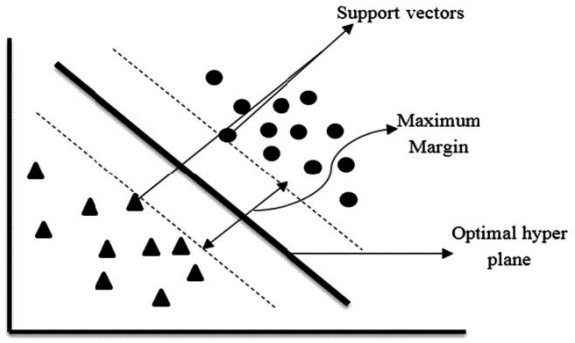

Fig. 2 — Classic SVM Classifier model

For the given training data $\{(x_1, y_1), (x_2, y_2) \ldots (x_i, y_i)\}_{i=1}^{N}$, where $x_i (x_i \in R^t)$ denoted the input vector of the first $i$training samples, $x_i = \left[x_i^1, x_i^2, \ldots, x_i^t\right]^T$and $y_i \in (-1, +1)$ is the corresponding output class.

The optimal hyperplane equation is given by,

$$w^T.x + b = 0 \quad \ldots (3)$$

Construct other two planes running parallel to the optimal one of equal distance, each of two planes is constructed close to the data points of separate classes such that no points appear in between the parallel planes, the equation of parallel hyperplanes are given by,

$$w^T.x + b = +1 \text{ and } w^T.x + b = -1 \quad \ldots (4)$$

The hyperplane is defined such that it minimizes the equation, $\frac{1}{2}\|w\|^2$that satisfies the following constraints,

$$y_i(w^t.x_i + b) \geq 1 \forall i = 1,2,\ldots N \quad \ldots (5)$$

On introducing Lagrange multipliers, the above convex optimization equation is represented as,

$$L(w, b, \alpha) = \frac{1}{2}\|w^2\| - \sum_{i=1}^{N} \alpha_i [y_i(w^t.x_i + b) - 1] \quad \ldots (6)$$

To be minimized with respect to w and $b$subject to the constraints$\alpha_i \geq 0, \forall i = 1,2,\ldots N$ presented as,

$$\left.\begin{array}{l} w = \sum_{i=1}^{N} y_i \alpha_i x_{i`} \\ \sum_{i=1}^{N} y_i \alpha_i = 0 \end{array}\right\} \quad \ldots (7)$$

On substituting the above constraints in (4), the dual optimization problem is given by,

$$\min_{\alpha \in R^m} Q(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j x_j^i x_j - \sum_{i=1}^{N}\alpha_i \quad \ldots (8)$$

Subject to the constraints,

$$\sum_{i=1}^{N}\alpha_i y_i = 0, \alpha_i \geq 0 \forall i = 1,2,..N \quad \ldots (9)$$

The linearly separable classification is represented by $sign(w^t.x + b)$

If the data is nonlinear then a slack variable ξ_i is introduced in the Eq. (5),

$$y_i(w^t.x_i + b) \geq (1 - \xi_i) \quad \text{and } \xi_i \geq 0 \forall i = 1,2,..N \quad \ldots (10)$$

The optimal separating plane is given by,

$$\min_{(w,\xi)} L(w,\xi) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \quad \ldots (11)$$

Subject to, $\left.\begin{matrix} y_i(w^t.x_t + b) \geq 1 - \xi_i & for 1 \leq i \leq N \\ \xi_i \geq 0 & for 1 \leq i \leq N \end{matrix}\right\}$ … (12)

On introducing Lagrange Multipliers,

$$L(w,b,\alpha,\xi,\beta) = \frac{1}{2}\|w^2\| - \sum_{i=1}^{N}\alpha_i[y_i(w^t.x_i + b) - 1 + \xi_i] - \sum_{i=1}^{m}\beta_i\xi_i \quad \ldots (13)$$

On solving the above equation the dual optimization problem is attained by,

$$\min_{\alpha \in R^m} Q(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j x_j^i x_j - \sum_{i=1}^{N}\alpha_i \quad \ldots (14)$$

With constraints,

$$\sum_{i=1}^{N}\alpha_i y_i = 0, 0 \leq \alpha_i \leq C \forall i = 1,2,..N \quad \ldots (15)$$

And solution of $w = \sum_{i=1}^{N}\alpha_i y_i x_i$, the decision function of $sign(w^t.x + b)$. So far the linear classifier in input space is discussed, but when it becomes inappropriate to consider linear separating problem then the input data should be initially transferred into high dimensional space by the transformation : $\varphi: R^t \rightarrow R^n$ then equation (12) is written as,

$$\min_{\alpha \in R^m} Q(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \varphi(x_i)\varphi(x_j) - \sum_{i=1}^{N}\alpha_i \quad \ldots (16)$$

The decision function, $sign(w^t\varphi(x) + b)$, where $w = \sum_{i=1}^{N} y_i\alpha_i\varphi(x_i)$.The function $K(x,x') = \varphi(x)'\varphi(x')$ represents the kernel function, now the Eq. (16) is represented as

$$\min_{\alpha \in R^m} Q(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i,x_j) - \sum_{i=1}^{N}\alpha_i \quad \ldots (17)$$

Subject to the constraints, $\sum_{i=1}^{N}\alpha_i y_i = 0, 0 \leq \alpha_i \leq C \forall i = 1,2,..N$. With the decision function $sign(\sum_{i=1}^{N} w^t\varphi(x) + b)$.

The Kernel Function needs to be defined explicitly. The Radial basis function is employed in thisstudy and is expressed as,

$$K(x,x') = exp(-\delta\|a - b\|^2) \quad \ldots (18)$$

Experimental Modeling of the Machine Learning-based IDS Model

The machine learning-based intrusion detection models are experimentally validated by experimental analysis made on benchmark datasets in MATLAB R2014a environment and executed in Intel Duo Core2 Processor with 2 GB Ram of speed 2.27 GHz. The experimental modeling of the machine learning-based IDS system is depicted in Fig. 3.

Data Preprocessing using Min-Max Normalization Method

The block diagram of the proposed intrusion detection model is shown in Fig 3. The first step in the development of the intrusion detection model is the dataset construction.Following this, the K-fold cross validation technique and the classifier model are used to classify the attack as normal or malicious. The NSL-KDD dataset is considered in this study, and the dataset consists of 125973 single connection training vectors of 41 features and 22544 testing vectors, as

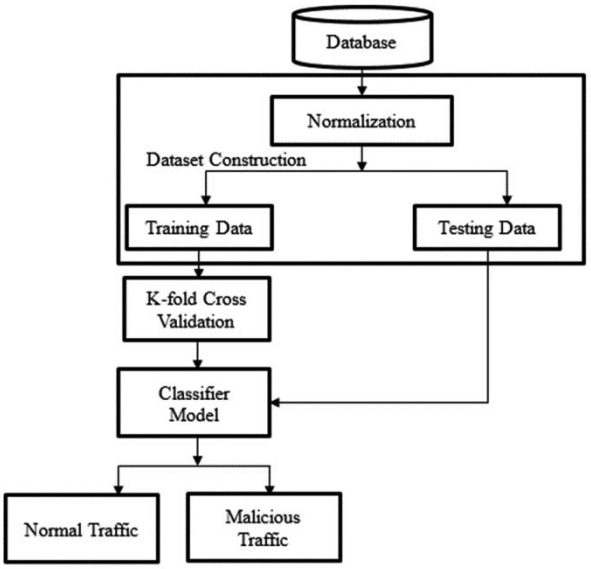

Fig. 3 — Block diagram representation of the machine learning based intrusion model

depicted in Table 1. Data preprocessing is essential to convert this dataset into an understandable format. Data preprocessing can be done by data cleaning, data transformation, and data reduction. This research chooses the min-max normalization method under the data transformation technique. In this, the feature of minimum value is assigned with the value 0, whereas the maximum value feature is assigned with the value of 1. All other values are assigned with values between 0 and 1. Each variable in the given database is initially encoded to numeric data then they are normalized to the range [0, 1] to eliminate the scale difference effect. The min-max normalization method is used to normalize the extracted features such that the training dataset comprises [0, 1]. The normalized input data is expressed as,

$$I_i^{'} = \left(\frac{I_i - I_{\min}}{I_{\max} - I_{\min}} \right) \left(I_{\max}^{'} - I_{\min}^{i} \right) + I_{\min}^{'} \quad \ldots (19)$$

where I_i is original input data, $I_{\min}$ is the minimum input value, $I_{\max}$ is the maximum input value, $I_{\max}^{'}$ is the maximum target value, $I_{\min}^{'}$ is the minimum target value.

k-Fold Cross Validation Technique

The k-fold cross validation divides the input dataset into k-fold groups, where k-1 groups are treated as a training group and the remaining one is employed as testing groups. On adopting-fold cross validation each sample is trained for k-1 times and tested for 1-time. In this research, 10-fold cross validation is employed over the training data samples, and testing data samples are utilized as an unknown dataset to model to demonstrate its classification ability.

For the considered classifier model, the training dataset is partitioned into ten groups and employed 10-fold cross validation methodology to train the dataset. The essential features are selected; feature selection is a critical strategy for machine learning algorithms. The vital components are identified during every fold of k-fold training, and the model is fed with the dataset of trend features. The next phase of training is testing; the trained classifier model is tested with the unknown dataset to model its effectiveness through specific performance metric results. The developed models are experimentally validated by Accuracy, Precision, Sensitivity, Selectivity, F1 Score, and AUC based on the confusion matrix table developed as shown in Table 2.

Table 1 — Number of Records of NSL-KDD Dataset

Dataset	Number of Records		
	Normal	Abnormal	Total
Training	67343 (53%)	58,630 (47%)	125973
Testing	9711 (43%)	12833 (57%)	22544

Table 2 — Confusion Matrix for the machine learning-based IDS

Actual Class	True Outcome: Intrusion Identified	
	P (Malicious Traffic)	N (Normal Traffic)
P (Malicious Traffic)	TP (Malicious traffic identified as Malicious)	FP (Malicious traffic identified as Normal)
N (Normal Traffic)	FN (Normal traffic identified as Malicious)	TN (Normal Traffic Identified as Normal)

Results and Discussion

The machine learning algorithms are successfully trained with the training dataset of 10-fold cross validation. During the process of training, the essential features are identified for each fold and presented in Table 3. The graph between the frequency of selected features during the 10-fold cross validation and their corresponding accuracy is plotted in Fig. 4, and the accuracy has achieved to better value on feeding the models with the feature subset that has a frequency of occurrence of 8 and above value, so the only the features that have a frequency of occurrence above eight is considered to train the model and the corresponding performances are investigated.

The frequency of occurrence of all selected features by the machine learning based IDS models are presented in Table 4. The C4.5 classifier model selected total of 18 features at the end of 10-fold CV but F3 (service) occurred for three times and F6, F7,

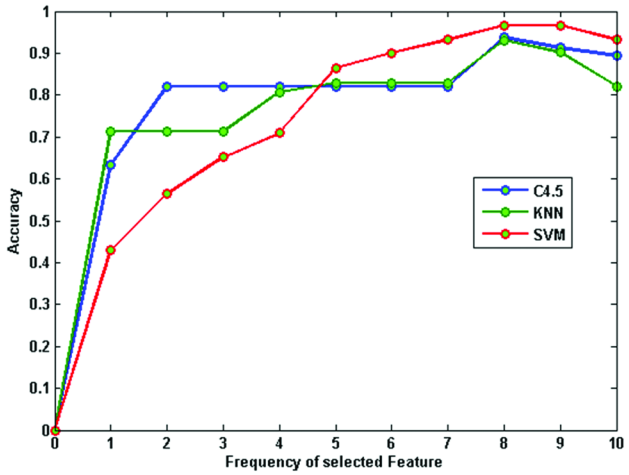

Fig. 4 — Plot Represents the Accuracy for Selected Features

Table 3 — Feature Selection and frequency by the machine learning-based IDS models

Fold	Selected Features	Fold	Selected Features
C4.5 IDS model with 10-fold cross validation technique			
#1	F3, F5, F8, F10, F12, F23, F25, F29, F30,F35, F36, F37, F38	#6	F4, F5, F8, F10, F12, F23, F25, F29, F30, F36, F37, F41
#2	F4,F5,F6,F8,F10,F11,F12,F23,F25,F29,30,F35,F36,F37	#7	F4, F5, F8, F10, F12, F23, F25, F29, F30, F35, F36
#3	F3, F4, F5, F8, F10, F12, F23, F25, F29, F30, F35, F36, F37	#8	F4, F5, F8, F10, F12, F23, F25, F29, F30, F35, F36, F37, F38
#4	F3, F4, F5, F7, F8, F10, F12, F23, F25, F29, F30, F35, F36	#9	F4, F5, F8, F10, F12, F23, F25, F29, F30, F35, F36, F37
#5	F4,F5,F8,F10,F12,F23,F25,F29,F30,F35,F36,F37	#10	F4, F5, F8, F10, F12, F23, F25, F29, F30, F35, F36,F37
KNN IDS model with 10-fold cross validation technique			
#1	F4, F5, F8, F10, F12, F23, F29, F30, F33, F35, F36, F37, F39	#6	F4, F5, F6, F8, F10, F12, F23, F25, F26, F29, F30, F33, F35,F36, F37,F39
#2	F4, F5, F8, F10, F12, F23, F25, F26, F29, F30, F33, F35, F36, F37, F39	#7	F4, F5, F8, F10, F12, F23, F25, F26, F29, F30, F35, F36, F37, F38, F39
#3	F4, F5, F6, F8, F10, F12, F25, F29, F30, F33, F35, F36, F37, F38, F39	#8	F4, F5, F8, F10, F12, F23, F25, F26, F29, F30, F33, F35, F36, F37, F39
#4	F4, F5, F6, F8, F10, F12, F23, F25, F26, F29, F30, F33, F35, F36, F39	#9	F4, F5, F8, F10, F12, F23, F25,F26, F29, F30, F35, F36, F37, F39
#5	F4, F5, F6, F8, F10, F12, F23, F25, F26, 29, F30, F33, F35, F36, F37, F38, F39	#10	F4, F5, F8, F10, F12, F23, F25, F26, F29, F30, F35, F36, F37 ,F39
SVM IDS model with 10-fold cross validation technique			
#1	F4, F5, F8, F10, F12, F23, F25, F26, F 29, F30, F35, F36, F37, F38	#6	F4, F5, F8, F10, F12, F23, F29, F30, F35, F36, F37
#2	F4, F5, F8, F10, F12, F23, F26, F29, F30, F35, F36, F39	#7	F4, F5, F8, F10, F12, F23, F26, F29, F30, F35, F37, F38
#3	F4, F5, F8, F10, F12, F23, F26, F29, F30, F34, F35, F36, F38	#8	F4, F5, F8, F10, F12, F23, F25, F29, F30, F35, F36, F37, F39
#4	F4, F5, F8, F10, F12, F23, F26, F29, F30, F35, F36, F38	#9	F4, F5, F8, F10, F12, F23, F26, F29, F30, F35, F36
#5	F4, F5, F8, F10, F12, F23, F26, F29, F30, F35, F36, F38	#10	F4, F5, F8, F10, F12, F23, F25, F29, F30, F35, F36

F11, F38, F41 are appeared only one time throughout the entire training process is completed. The model outperformed with better performance metric values for the selected 12 features as shown in Table 5 such as F4 (flag), F5 (src_bytes), F8 (wrong_fragment), F10 (hot), F12 (logged_in), F23 (Count), F25 (serror_rate), F29 (same_srv_ rate), F30 (diff_srv_ rate), F35 (dst_host_diff_ srv_rate), F36, (dst_host_ same_src_port_rate), F37 (dst_host_srv_diff_host_ rate). The KNN reported total of 17 features at the end of 10-fold CV, but F6 appeared for 4 times, F33 (dst_host_srv_rate) occurred 7 times, F38 appeared for 3 times so neglecting these three features out of 15 features, F4 (flag), F5 (src_bytes), F8 (wrong_fragment), F10 (hot), F12 (logged_in), F23 (Count), F25 (serror_rate), F26 (srv_serror_rate), F29 (same_srv_rate), F30 (diff_srv_rate), F35 (dst_host_ diff_srv_rate), F36 (dst_host_ same_src_port_rate), F37 (dst_host_srv_diff_ host_rate), F39 (dst_host_ srv_serror_rate) are the features selected to demonstrate the performance of the KNN IDS model as shown in Table 6.

The performance of SVM model outperformed with the 10 selected features of F4 (flag), F5 (src_bytes), F8 (wrong_ fragment), F10 (hot), F12 (logged_in), F23 (Count), F29 (same_srv_rate), F30 (diff_srv_ rate), F35 (dst_host_diff_srv_rate), F36 (dst_host_same_src_port_rate) as shown in Table 7. The model reported 16 features at the end of 10 fold CV, but the features F34 occurred for one time, F39 appeared for two times, F38 have appeared for five times, F25 appeared for three times, F26 appeared for seven times, F37 occurred for four times, and so these features are neglected for test dataset validation. The optimally selected feature subsets of testing data are fed into the machine learning-based IDS models and validated for their effectiveness. To avoid the biased output, the models are made to run for ten trial runs, and the average of the obtained performance metric results is depicted in Fig 5. It is observed from the

Table 4 — Frequency of selected features between the machine learning based intrusion detection models

Feature	C4.5	KNN	SVM
F3	3	0	0
F4	**9**	**10**	**10**
F5	**10**	**10**	**10**
F6	1	4	0
F7	1	0	0
F8	**10**	**10**	**10**
F10	**10**	**10**	**10**
F11	1	0	0
F12	**10**	**10**	**10**
F23	**10**	**9**	**10**
F25	**10**	**9**	3
F26	0	**8**	7
F29	**10**	**10**	**10**
F30	**10**	**10**	**10**
F33	0	**7**	0
F34	0	0	1
F35	**9**	**10**	**10**
F36	**10**	**10**	**9**
F37	**8**	**9**	4
F38	1	3	5
F39	0	**10**	2
F40	0	0	0
F41	1	0	0
Total Number of Selected Features	12	14	10

Table 5 — Selected Features - Performance Metric of C4.5

No.	Selected Features	No.	Selected Features
1	F4 (flag)	2	F5 (src_bytes)
3	F8 (wrong_fragment)	4	F10 (hot)
5	F12 (logged_in)	6	F23 (Count)
7	F25 (serror_rate)	8	F29 (same_srv_rate)
9	F30 (diff_srv_rate)	10	F35 (dst_host_diff_srv_rate)
11	F36(dst_host_same_ src_port_rate)	12	F37 (dst_host_srv_diff_host_rate)

Table 6 — Selected Features - Performance Metric of KNN

No.	Selected Features	No.	Selected Features
1	F4 (flag)	2	F5 (src_bytes)
3	F8 (wrong_fragment)	4	F10 (hot)
5	F12 (logged_in)	6	F23 (Count),
7	F25 (serror_rate)	8	F26 (srv_serror_rate)
9	F29 (same_srv_rate)	10	F30 (diff_srv_rate)
11	F35 (dst_host_diff_srv_rate)	12	F36 (dst_host_same_src_port_rate)
13	F37 (dst_host_srv_diff_host_ rate)	14	F39 (dst_host_srv_serror_rate)

table that the SVM classifier model outperformed other models with better classification accuracy, sensitivity, and F1 score, whereas the precision and specificity results are better for C4.5 as compared to

Table 7 — Selected Features - Performance Metric of the SVM

No.	Selected Features	No.	Selected Features
1	F4 (flag)	2	F5 (src_bytes)
3	F8 (wrong_fragment)	4	F10 (hot)
5	F12 (logged_in)	6	F23 (count)
7	F29 (same_srv_rate)	8	F30 (diff_srv_rate)
9	F35 (dst_host_diff_srv_rate)	10	F36 (dst_host_same_src_port_rate)

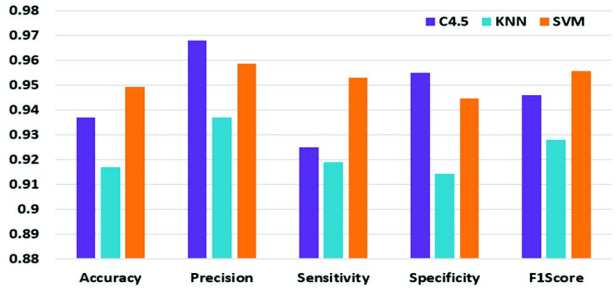

Fig. 5 — Performance analysis of machine learning-based intrusion detection IDS models

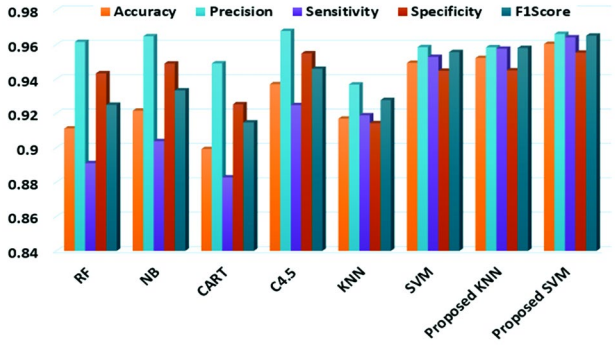

Fig. 6 — Comparison of the machine learning based intrusion detection systems with existing systems

SVM based IDS model. So, an attempt is made to combine the individual qualities of this algorithm, and a hybrid model is developed.

The selected attributes of the C4.5 classifier algorithm are fed into the SVM and KNN classifier algorithm, and the corresponding performances are analyzed. It is clear that the performance of the SVM classifier algorithm is enhanced by feeding the output feature subset of the C4.5 classifier strategy; the obtained results for ten trial runs are plotted in Fig 5. The performance analysis is plotted in Fig 6. The proposed strategy considerably improves the models' true positive rate and true negative rate compared to the classic individual algorithms. From Table 8, it is clear that the machine learning-based intrusion detection models obtained better classification metric

Table 8 — Comparative analysis made with the existing Models in the literature

Model Under Study	Accuracy	Precision	Sensitivity	Specificity	**F1Score**
Logistic regresssion Gohil & Kumar (2020)[25]	0.7934	0.8555	0.7999	0.0154	0.7888
Decision tree Muthamil *et al.* (2021)[26]	0.7888	0.6555	0.7222	0.0144	0.7111
Sequential Minimal optimization (2019)[27]	0.9563	0.9666	0.9666	0.0600	0.0963
Principle Component analysis +Naive Bayes Bagyalakshmi & Samundeeswari (2020)[28]	0.8721	0.9562	0.8561	0.9358	0.9034
Naive Bayes Pranto *et al.* (2022) [29]	0.8711	0.8033	0.9099	0.1555	0.8533
SVM with eight significant features Tonkal *et al.* (2021) [30]	0.8148	0.8134	0.8287	0.7895	0,8523
SVC Ahuja *et al.* (2021)[31]	0.8583	0.8579	0,8746	0.8404	0.8661
C4.5	0.9370	0.9680	0.9248	0.9549	0.9459
KNN	0.9170	0.9368	0.9189	0.9143	0.9278
SVM	0.9494	0.9585	0.9528	0.9448	0.9557
KNN IDS model with 10 fold-cross valiation technique	0.9523	0.9585	0.9576	0.9450	0.9581
SVM with 10 fold-cross valiation technique	0.9604	0.9662	0.9642	0.9553	0.9652

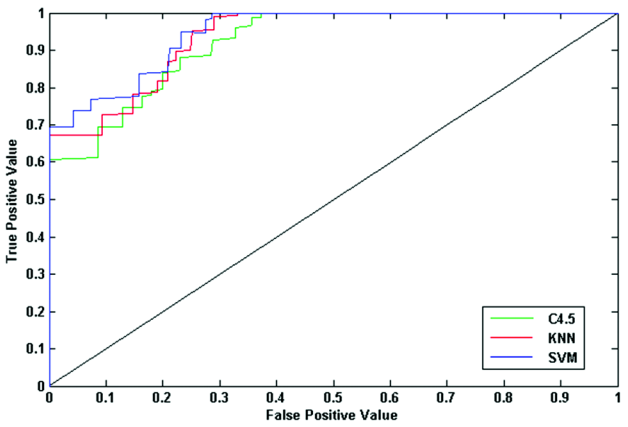

Fig.7 — Receiver Operating Characteristics of the machine learning based detection systems

measures than other existing models in literature, such as Random Forest, Naïve Bayes, and CART methodology. The machine learning-based intrusion detection models are trained by 10 fold cross validation. Based on the number of times of occurrence of features, the optimal feature subset is framed and utilized to test the model. The model is validated with an independent dataset by the specified performance metric measures and compared with the performance of existing models in the works of literature. From Fig 7, it is clear that SVM has better ROC value than C4.5 and KNN. Based on the study, it was analyzed that the hybridizing of two algorithms can improve the performance of the model by combining their individual qualities than the conventional algorithms themselves. So, the hybrid combination of C4.5 classifier algorithm with SVM and KNN models and the obtained results demonstrated its significance, and also from the investigation made it is demonstrated that the SVM based classifier model outperformed all other models with better intrusion detection performance with minimal number of feature subset as compared to other models in the literatures under comparison.

Conclusions

This research paper discussed various machine learning algorithms employed to design the Intrusion Detection System. The machine learning based intrusion detection models were trained by 10 folds cross validation. The optimal feature subset was framed based on the frequency of occurrence of features, which has been utilized to test the model. The machine learning based intrusion detection model is validated with an independent dataset by utilizing the specified performance metric measures and compared with the performance of existing models. The integration of these two algorithms may get the advantages of both algorithms, which leads to providing better results than the conventional algorithms. So, the hybrid combination of the C4.5 classifier algorithm with SVM and KNN models and the results obtained confirmed that the SVM based classifier model outperformed all other models with better intrusion detection performance and a minimal number of feature subset as compared to other models.

References

1 Somani G, Gaur M S, Sanghi D, Conti M & Buyya R, DDoS attacks in cloud computing: Issues, taxonomy, and future directions, *Comput Commun*, **107** (2017) 30–48.

2 Bhardwaj A, Mangat V, Vig R, Halder S & Conti M, Distributed denial of service attacks in cloud: State-of-the-art of scientific and commercial solutions, *Comput Sci Rev*, **39** (2021) 100332.

3 Nagaraju V, Raaza A, Rajendran V & Ravikumar D, Deep learning binary fruit fly algorithm for identifying SYN flood attack from TCP/IP, *Mater Today Proc,* 22, 2021.

4 Jose A S, Nair L R, Paul V, Towards detecting flooding DDOS attacks over software defined networks using machine learning technique, *Rev GEINTEC*, **11(4)** (2021) 3837–3865.

5 Chen Y H, Lai Y C, Zhou K Z, Identifying hybrid DDoS attacks in deterministic machine-to-machine networks on a per-deterministic-flow basis, *Micromachines*, **12(9)** (2021) 1019

6 Rezaei A, Using ensemble learning technique for detecting botnet on IoT, *SN Comput Sci*, **2(3)** (2021) 1–4.

7 Muraleedharan N, Janet B, A flow-based anomaly detection system for slow DDoS attack on HTTP, *CASB* (2021) 29–45.

8 Nurwarsito H & Nadhif M F, DDoS Attack Early Detection and Mitigation System on SDN using Random Forest Algorithm and Ryu Framework, in *Proc 8th Int Conf Comput Commun Eng (ICCCE)* 2021, 178–183

9 Bhosale K S, Nenova M & Iliev G, Intrusion detection in communication networks using different classifiers, in *Techno-Societal 2018* (Springer, Cham) 2020, 19–28

10 Singhal S, Medeira P A, Singhal P & Khorajiya M, Detection of application layer DDoS attacks using big data technologies, *J Discret Math Sci*, **23(2)** (2020) 563–571

11 Roopak M, Tian G Y & Chambers J, Multi-objective-based feature selection for DDoS attack detection in IoT networks, *IET Networks*, **9(3)** (2020) 120–127

12 Swami R, Dave M & Ranga V, DDoS attacks and defense mechanisms using machine learning techniques for SDN, in *Research Anthology on Combating Denial-of-Service Attacks* (IGI Global) 2021, 248–264

13 Dwivedi S, Vardhan M & Tripathi S, Defense against distributed DoS attack detection by using intelligent evolutionary algorithm, *Int J Comput Appl*, (2020), DOI: 10.1080/1206212X.2020.1720951.

14 Hussain Y S, Network Intrusion Detection for Distributed Denial-of-Service (DDoS) Attacks using Machine Learning Classification Techniques (2020), http://hdl.handle.net/1828/11679.

15 Doucette C, Broderick-Sander R, Toll B, Helsinger A, Soule N, Pal P, Zhou C & Paffenroth R, A robust principal component analysis approach to DoS-related network anomaly detection, *Proc SPIE 11417, Cyber Sensing 2020*, 114170B (27 April 2020); https://doi.org/10.1117/12.2562774.

16 Rathore S, Park J H, Semi-supervised learning based distributed attack detection framework for IoT, *Appl Soft Comput*, **72** (2018) 79–89.

17 Ravi N, Shalinie S M. Learning-driven detection and mitigation of DDoS attack in IoT via SDN-cloud architecture, *IEEE Internet of Things* Journal, **7(4)** (2020), 3559–3570.

18 Nesa N, Ghosh T & Banerjee I. Non-parametric sequence-based learning approach for outlier detection in IoT, *Future Gener Comput* Syst, **82** (2018) 412–21.

19 Mukherjee S & Sharma N. Intrusion detection using naive Bayes classifier with feature reduction, *Procedia Technology*, **4** (2012) 119–128.

20 Siahaan H, Mawengkang H, Efendi S, Wanto A, Windarto A P. Application of classification method C4. 5 on selection of exemplary teachers, *J Phy*: *Conf Ser*, **1235(1)** (2019) 012005.

21 Patro S G, Mishra B K, Panda S K, Kumar R, Long H V, Taniar D & Priyadarshini I, A hybrid action-related K-nearest neighbour (HAR-KNN) approach for recommendation systems, *IEEE Access*, **8** (2020) 90978–90991.

22 Sumathi S & Rajesh R, Comparative study on TCP SYN flood DDoS attack detection: A machine learning algorithm based approach, *WSEAS Trans Syst Control*, **16(1)** (2021) 584–591.

23 Binson V A, Subramoniam M & Mathew L, Discrimination of COPD and lung cancer from controls through breath analysis using a self-developed e-nose, *J Breath Res*, **15(4)** (2021) 04600.

24 VA B, Subramoniam M & Mathew L, Noninvasive detection of COPD and Lung Cancer through breath analysis using MOS Sensor array based e-nose, *Expert Rev Mo Diagn*, **21(11)** (2021) 1223–1233.

25 Gohil M & Kumar S, Evaluation of classification algorithms for distributed denial of service attack detection, in *AIKE* (IEEE) 2020, 138–141

26 Sudar K M, Beulah M, Deepalakshmi P, Nagaraj P & Chinnasamy P, Detection of Distributed Denial of Service Attacks in SDN using Machine learning techniques, in *IEEE Int Conf Comput Commun Informat (ICCCI)*, 27 Jan 2021, 2021, 1–5, doi: 10.1109/ICCCI50826.2021.9402517.

27 Das S, Mahfouz A M, Venugopal D & Shiva S, DDoS intrusion detection through machine learning ensemble, in *IEEE 19th Int Conf Softw Qual, Reliab Secur Compan (QRS-C)*, 2019, 471–477, doi: 10.1109/QRS-C.2019.00090.

28 Bagyalakshmi C & Samundeeswari E S, DDoS attack classification on cloud environment using machine learning techniques with different feature selection methods, *Int J*, **9(5)** (2020).

29 Pranto M B, Ratul M H, Rahman M M, Diya I J & Zahir Z B, Performance of machine learning techniques in anomaly detection with basic feature selection strategy-a network intrusion detection system, *J Adv Inf Technol*, **13(1)** (2022).

30 Tonkal Ö, Polat H, Başaran E, Cömert Z & Kocaoğlu R, Machine Learning Approach Equipped with Neighbourhood Component Analysis for DDoS Attack Detection in Software-Defined Networking, *Electronics*, **10(11)** (2021) 1227.

31 Ahuja N, Singal G, Mukhopadhyay D & Kumar N, Automated DDOS attack detection in software defined networking, *J Netw Comput Appl*, **187(** 2021) 103–108.