A HYBRID APPROACH FOR CYBER SECURITY: IMPROVED INTRUSION DETECTION SYSTEM USING ANN-SVM

Abhishek Kajal[1]
Guru Jambheshwar University of Science and Technology,
Hisar (Haryana) – 125001, India
abhishekkajal82@gmail.com

Sunil Kumar Nandal[2]
Guru Jambheshwar University of Science and Technology,
Hisar (Haryana) – 125001, India
nandal_sunil@yahoo.co.in

Abstract: The instances of cyber attacks are increasing as of exponential hike in online processing amid COVID-19 pandemic scenario that significantly compromising huge number of confidential data as well. With the rising security demands, the detection of cyber-attacks has emerged as a promising field that offers wider scope for the application developers. In the similar scenario, authors have proposed an improved detection system that could effectively detect attacks as DDoS, malware etc. Initially, Genetic Algorithm is implemented for the feature section and reducing the size of data thereby reducing the computational work load of classifiers. In second phase, Discrete Wavelet Transform (DWT) with Artificial Bee colony (ABC) is used to divides the data into four categories and also filters out the irrelevant features. In later stages, Artificial Neural Network is employed whose results are refined by Support Vector Machine (SVM). The novelty of the work lies in the precise detection of the malicious behaviour of the nodes. This ANN-SVM hybrid approach enhanced the classification efficiency of the proposed system in identifying the cyber-attacks. The simulation study over 1000 rounds reflects the performance of the proposed system in terms of higher precision, recall and f-measure in comparison to the existing works dedicated to deal with DDoS attacks. Also, the proposed work demonstrated better outcomes in terms of average True Positive rate and False Positive rate.

Keywords: Intrusion Detection System, Genetic Algorithm, Discrete Wavelet Transform with Artificial Bee colony, Artificial Neural Network, Support Vector Machine.

1. Introduction

Nowadays cloud computing is growing rapidly, making people more dependent on computer networks than ever before. At the same time, the number of information security violations has sharply increased. Therefore, the end to end security is extremely important. This includes ensuring the security and reliability of network devices, as well as highly effective protection against various network attacks that create vulnerabilities in installed security protocols. The Intrusion Detection System (IDS) is considered one of the essential tools to monitor malicious activities continuously and detect threat that could compromise the integrity, privacy, or availability of the network. In this modern era, everyone has the access of internet either through smart phone or through laptop or tab, and it is supposed that the internet services should be available 27 ×7 without any connection failure [Agarwal and Hussain, (2018)]. Cloud inaccessibility can be the result of a failure of a cloud infrastructure component and occurrence of malicious activities such as distributed denial of service (DDoS) [Patel et al., (2013)]. Before detecting such attacks, one should know about the basic nature of such attacks.

1.1 DDoS attacks

Today is the world of internet and DDoS is one of the most affecting threats in online computing environment. If we have enough resources, only then one can defeat DDoS attacks. However, the client-server and peer system do not have enough resources to defeat them. Therefore, cloud system needs a protection system to detect DDoS at prior stage [Daffu and Kaur, (2016)]. DDoS attacker tried to affect network in its different forms. The basic nature of DDoS attacker is to flood the network with a large number of packets and then exhaust the network resources. Mainly DDoS attacker attacks on an individual cloud user as they have limited resources. But it can be possible to detect DDoS attacker and then prevent the network from data loss. The attacker saturates the main server of the communicating system by sending fake packets requests, and hence the normal/ genuine user is not been responded by the main server. This results in overloading of server [Somani et al., (2017)]. The DDoS attack scenario in which main server is controlled by the attackers is shown in Figure 1.

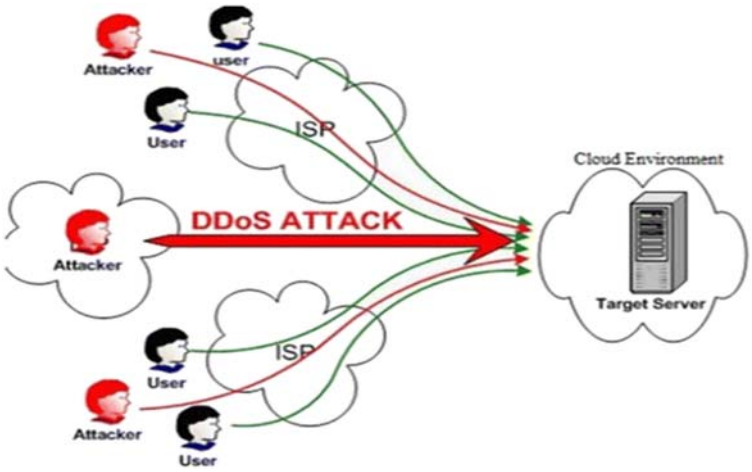

Figure 1: DDoS Attack [Kesavmoorthy and Sounder, (2019)]

Such cyber attack arises due to the failure in the protection mechanisms utilised. Among various security control measures, IDS is an important tool of the defence mechanism to protect cloud networks from damage. IDS monitor the flow of information in the network, and based on that decision has been taken whether the incoming packet is normal or malicious [Agarwal and Hussain, (2018)]. The purpose of IDS is to warn or notify the system that certain malicious activities have occurred. Depending upon the collecting nature of data from the incoming message, the IDS is categorized into two types (i) HIDS (Host-based intrusion detection system) and (ii) NIDS (Network-based intrusion detection system). HIDS detect the attacker, by analysing the collected data by the operating System (OS) and the action performed in response to the data. In NIDS, the classification between the attacker and normal node is performed based on the information analysed, which is gathered from the network packets. According to the way of detecting attacks, the IDS are categorized into two types one is knowledge based and other is anomaly-based system. The anomaly-based method identify attacker by comparing the action performed by the recent user to the standard user. In signature-based approach, the ongoing data traffic is monitored by different activities and compared with the rule set [Deshpande et al., (2018); Chiba et al. (2019)].

In this research article, we presented an IDS system by using the advantages of nature inspired algorithm with ML approach. The major advantage of this designed approach is to detect multiple malicious behaviour of node and hence enhance efficiency of by proposed work. The remaining article is organised as: Related work is the field of securing network against cyber attacks as DDoS is presented in section 2. In next section 3, the proposed work with workflow is elaborated. In next section 4, experimental results and theirs comparative analysis are presented. In last section 5, the conclusion of the work followed by references is presented.

2. Related Work

Different types of intruder detection methods used by number of scholars in the network technology. The detection methods depend upon the ability of the network to detect malicious activity from any of the three sides (victim, source and the network side). These mechanisms used soft computing approach, machine learning approach a statistical approach. The main goal of each approach is to detect attack and protect network from malicious or unauthorized user [He et al., (2016)]. [Zargar et al., (2013)] have discussed the Hidden Semi Markov based model for the detection of DDoS attack in the network. At the same time, information theory mechanism is also used for the threat detection, which is very complex to implement. [Kajal et al, (2020)] demonstrated a multi threat cyber detection system to detect the various cyber attacks by using machine learning approach as a classifier and genetic algorithm to reduce the feature set size. [Kushwah et al., (2017)] have designed a secure cloud system against DDoS attack using “black hole optimization” approach in combination to ANN. Experiment has been performed on MATLAB “NSL-KDD” dataset. Total 23 number of classes (1 for normal and remaining contains different types of attacks) has been used to train and test the network. 96.30% of detection accuracy has been analysed for the designed system. [Kajal et al, (2019)] presented hybrid algorithm architecture of swarm intelligence based artificial bee colony with neural network to derive evaluation parameters to detect network intruders. [Tsai et al., (2009)] have tested IDS by various machine learning techniques by aiming on developing single, hybrid and group of classifiers. Also compared with various datasets. [Lima et al., (2019)] have presented machine learning based DoS detection model in addition to signature-based feature extracted approach. Initially, the data for normal as well as for DDoS signature has been extracted and then pass to the machine learning algorithm for training process. Finally, the performance has been observed in terms of accuracy rate, false alarm rate, precision, sampling rate. [Watson et al., (2015)] have designed a real time detection model by using support Vector Machine (SVM) for automatic detection of DoS attack with an accuracy of 90%. [Hosseini and Mehrdad, (2019)] have used multiple classifiers like naïve Bayes, multilayer perceptron, random forest, Decision Tree (DT),

and k-nearest neighbours (K-NN) to detect DDoS attack in network. Among all, random forest classifier provides better results. [Velliangiri et al., (2020)] have investigated DDoS attack using deep learning classifier. Log file has been created by grouping the log information collected from the cloud users. This information is then used to train the classifier with minimum processing time, which is possible by using Bhattacharya distance measure. Elephant Herd Optimisation (EHO) in addition to Taylor series has been developed to detect DDoS attack with maximum accuracy of 83%.

3. Proposed Work

In this research automatic intrusion detection System (IDS) is proposed that integrates nature inspired and machine learning algorithms. The work flow of the proposed work is further divided in following parts.

i. The first part covers the pre-processing of the features with the implementation of Genetic Algorithm (GA) to confer a selection criterion for feature section of the training datasets. It results into two categories, namely, malicious and normal.

ii. In the second part, wavelet transform is applied that further divides the data into four categories that are further processed using Artificial Bee Colony (ABC) to filter out the irrelevant features. As a result, the data get processed and divided into training and classification sets using a 70:30 scenario.

iii. The last part involves the implementation of Artificial Neural Network (ANN) and Support Vector Machine (SVM) classifiers. Here, the research has been conducted to recognise distinct attacks. The overall architecture of proposed IDS system is shown in Figure 2.

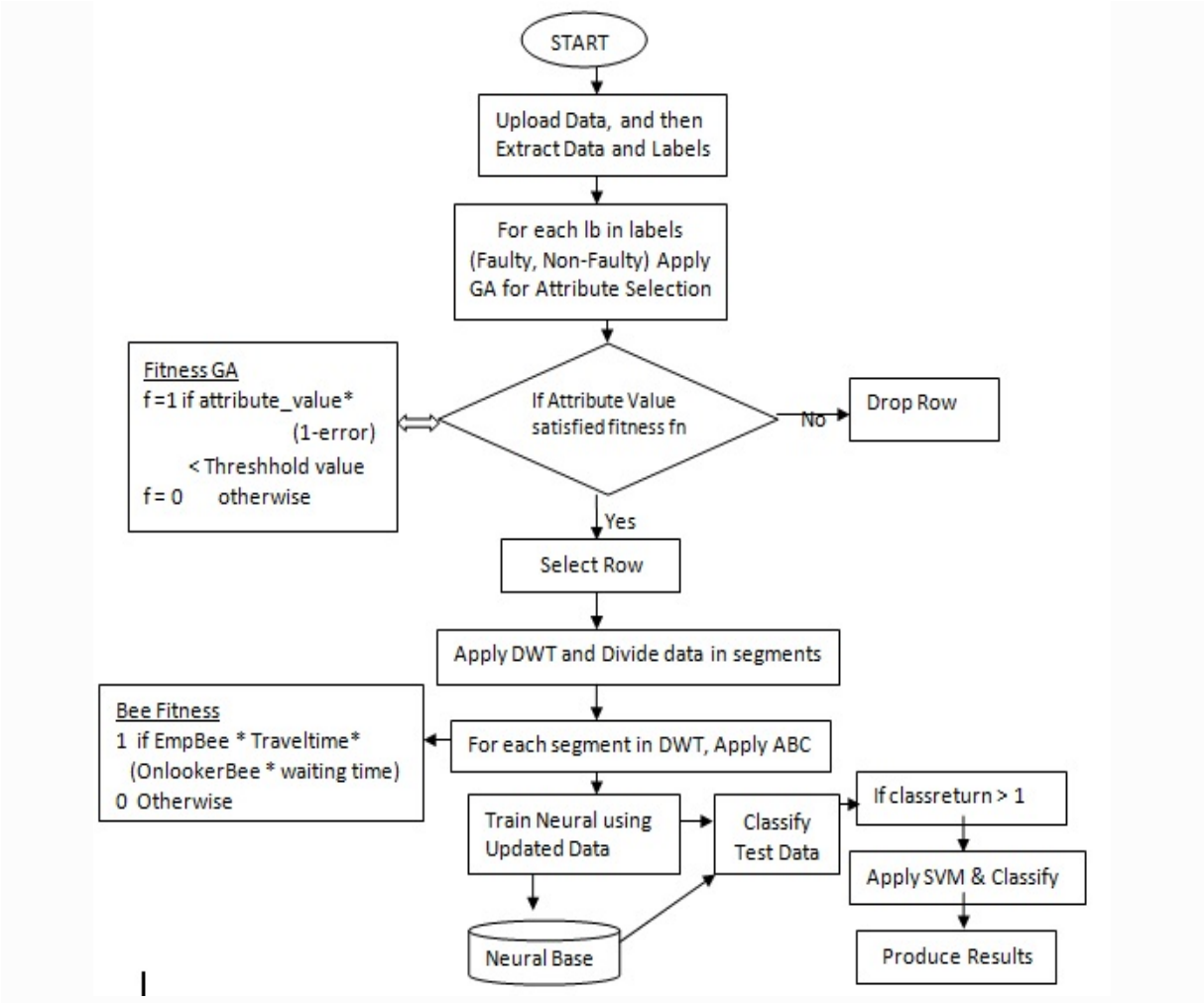

Figure 2. Proposed Workflow

3.1 Upload dataset

Initially, the dataset is collected from KDD cup dataset. The data consists of two different categories namely attack and normal. This dataset is used in training as well as in validation process. The dataset contains millions of records that include around 14 features that are being created by each connection. The data is available in different forms and also been considered as standard dataset to detect intruder in the network.

3.2 Apply GA for feature row selection

Since the data is available in unstructured form, therefore, needs to be refined. For this, we used Genetic Algorithm (GA), as a feature selection approach. To select appropriate features from the large dataset is one of the most widely used methods initiated by machine learning technique to filter irrelevant features and select the suitable feature among the main features that in turns improve the efficiency of the machine learning algorithm [(Osanaiye et al., (2016)]. In ML process, the selection and the construction of features from the available dataset is a very complex and time-consuming process. These attributes are summarized, integrated or split to produce features from the original data [(Kannan et al., (2015)]. In general, it is impossible to conduct an all-round search in order to find the most relevant features. Hence, feature selection becomes a challenging issue in machine learning and plays a very important role in various applications such as classification and regression. Here, for feature selection DWT with ABC algorithm is used. Generally, there are a lot of functions; many of them are either useless or lead to a reduction in learning accuracy. Eradicating such functions not only improves accuracy, but also reduces computational complexity [Jović et al., (2015)].

3.2.1 Genetic Algorithm (GA)

Genetic Algorithm is a well-known method that randomizes the generation of character strings. By introducing GA operators, the best fittest string out of the population would be selected for an optimal solution [Kannan et al., (2012)]. GA is a computational tool that is used in a variety of applications. It is very effective to solve complex problem and provide an optimal solution, especially when each objective function is varied and having numerous of local optima point. GA can simulate the search in the evolutionary stage and are mostly used to resolve feature selection problems. The simplified nature of Genetic Algorithm drives to the emergence of evolutionary process, which further starts with a randomly generated population of individuals (chromosomes) that are being specified by a probability distribution. The probability distribution is generally uniform, which updates population at a stage called generation. In every iteration process, multiple individuals are selected on the basis of fitness function. The updated individuals are then responsible for the generation of new population. The main function performed by GA includes:

- **Selection:** The selection is related to the probabilistic persistence of the fittest, that is, a more suitable chromosome is selected for survival, where fitness indicates a comparable indicator of the ability of the chromosome to determine the present problem.
- **Crossover:** Using CO operator, a gene selected randomly along chromosome string, and then the genes appear after that point is exchanged.
- **Mutation:** This action takes place to find a new and better solution by increasing the randomness. This is the possibility that a certain bit in the chromosome will be flicked. The working flow of GA is written below:

Algorithm 1: Genetic Algorithm for Attributes Selection

Input: Extracted Data Attributes (DA)

Output: Selected Row Data with Best Attributes (BA)

Start attributes selection

Set fitness function of the GA using given equation

$$Fit\ Fun = \begin{cases} 1; & if\ Selected_{Attributes} \times (1 - Error) < Threshold_{Attributes} \\ 0; & Otherwise \end{cases}$$

Where, $Selected_{Attributes}$: is selected attributes from DA and $Threshold_{Attributes}$ is the threshold value attributes and it is the average of all attributes in the DA.

Initialize GA parameters – Iterations (T)
– Population Size (P)
– Crossover function
– Mutation function
– Selection function ($Selected_{Attributes}$ and $Threshold_{Attributes}$)

Compute, 'R' as rows, 'C' as column of DA

BA = []

For i = 1 → R

For j = 1 → C

$P_{Val} = \sum_{i=1}^{P} \text{Img (i)}$

$Threshold_{Val} = \frac{\sum_{i=1}^{P} \text{Img (i)}}{Length\ of\ Img}$

$Fit\ Fun = Fit\ Fun\ (P_{Val},\ Threshold_{Val}))$

If *Fit Fun* satisfy

BA = GA (Fit Fun, Initialize Parameters) // Select perform

Else

BA = Reject the Attributes

End

End

End

Returns: BA as a selection of Best Attributes from DA

End

About 70 % of the total data has been passed to the GA that selects each row of the dataset using equation (1).

$$F_s = \begin{matrix} 1 & if\ (1-e) \times fs\ \times ft \\ 0 & Otherwise \end{matrix} \quad (1)$$

Extract features from three different data sets are used as feature vectors, with 0 and 1 representing the attacker and non-attacker, respectively. The available CSV data set has been applied to GA to reduce the size of the feature set by selecting the best optimized features and create a feature set. The optimized feature set is then used as input data for DWT.

3.3 Apply Discrete Wavelet Transform (DWT) in addition to Artificial Bee Colony (ABC)

The DWT is a very efficient tool used for data processing. The main role of which is to decompose data into distinct elements specifically in the frequency domain. DWT can be applied in one dimensional (1D) form that decomposed input data into two components (L, H). This is possible by using Low pass filter (LPF) and High Pass Filter (HPF). In case of two-dimensional DWT (2-D DWT), the data is broken into four components (LL, LH, HH, HL). In this research, DWT is used to segment the selected data using GA into small segments using 2D DWT. Thereafter, apply ABC on the segmented data to obtain the optimized dataset. Here, ABC is used to optimize or to select the best feature dataset.

[Karaboga, (2011)] proposed ABC to express the foraging behaviour of bees. The bees group mainly includes employed, scout and onlookers bees. Bees that may going in search food source would be employed bee and the bees that are waiting for the best food in the dancing area is known as onlooker bees.

The role of scouts is to conduct random search to find new sources of food. The upper half of this algorithm is artificial bees, while lower half is of onlookers. A probabilistic solution of the optimization problem is to find by location of source of food, and the quantity of nectar of the source of food, which is related to quality of related solution, and being determined by. Since this algorithm works for both local and global search, therefore, it delivered solutions with better efficiency [Hajimirzaei et al., (2019)]. The designed fitness function for ABC is given by equation (2).

$$fit = \begin{Bmatrix} 1\ if\ Empbee \times Travel\ time \times (Onlooker\ Bee \times Waiting\ Time \\ 0 \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad Otherwise \end{Bmatrix} \quad (2)$$

The probability to which better food source is selected by the onlooker bee is given by equation (3).

$$p_b = \frac{fit_b}{\sum fit_n} \quad (3)$$

Where n is the population size

To determine a new solution (V_{ij}), in neighbourhood of the old solution (u_{ij}), equation (4) can be used.

$$V_{ij} = u_{ij} + \varphi_{ij}(u_{ij} - u_{kj}) \quad (4)$$

Where

i,j,k→randomly selected indexes

φ_{ij}→ random function having values [-1,1].

After getting optimized values, ANN is trained by linking the optimized values to its weight and bias function.

3.4 Artificial Neural Network (Training and Classification)

ANN is a mathematical method that functions in the same way as that of human brain. The machine learning techniques is of two types supervised and unsupervised. In supervised approach, input as well as target output need to be defined. ANN is a supervised method that learns from a set of training data, and then compares the incoming data to the target sample. If any error appears at the output, that error value is passed to the hidden layer to modify the weight matrix. The general structure of ANN is shown in Figure 3. Each virtual machine can be used to simulate several nodes in the neural network, so that multiple VMs in the can be monitored by ANN. The following process shows the working of ANN algorithm [Pandeeswari and Ganesh, (2016)].

As shown in Figure 3, {X=X1, X2, X3……Xn} are the input presented at the input layer of ANN model. Theta (θ) is the weight function used for modification of the hidden later weight value. If the classification rate of the test data is below 2%, then apply SVM to train as well as to test the data. Otherwise, evaluate the performance parameters.

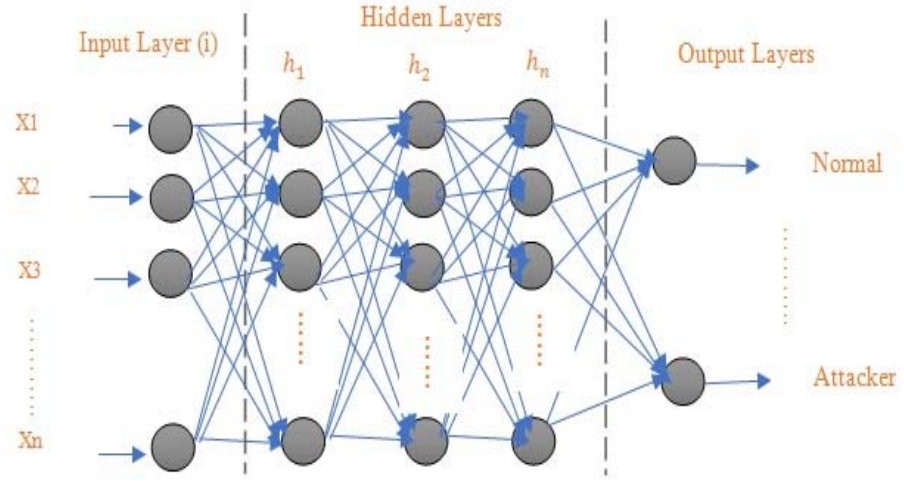

Figure 3: ANN Structure

After following above process, ANN is trained as per attacker (DDoS, malware and spoofing), and normal nodes and stored into the database. During testing the performance of the proposed work, if the classification rate of the ANN structure is less than 2 %, then apply Support Vector Machine (SVM). Otherwise calculate performance parameters. The dual machine learning approach helps to increase the classification rate of the designed IDS system for cloud environment. The overall workflow is shown in Figure 2. The working of SVM is described in the following section.

3.5 Support Vector Machine (SVM)

In the design process, the training data is used to impose the test set on the model, and the calculation error of the model is applied to the training and test inputs to adjust the model or train with high accuracy. After designing the model and accurately reaching the appropriate model based on the input training and testing, that the model is ready to distinguish attacker and normal node. Otherwise, the design process should be corrected. The enhanced task SVM data classification is based on linearity. Linear division data has attempted to select lines with more consistent margins. Usually, the QP method known as the method of solving the problem is used to solve the equation to find the best line of the data [(Sakr et al., (2019)].

The important task of this is to find the most appropriate line or hyper plane partitioning the data into two separate classes. SVM is classifier that receives the data at the input and generates such a dividing line. The vector representing the line is evaluated as V, and is represented by equation (5):

$$||V|| = \sqrt{V_n^2 + V_2^2 + \ldots\ldots + V_n^2} \quad (5)$$

Where, $||V||$ represent the length of the vector, $V = (V_1, V_2, \ldots\ldots, V_n)$ which is also termed as the norm of the vector. The direction of the vector $X = (X_1, X_2)$ is represented by ω in the equation (6).

$$\omega = \left(\frac{X_1}{||X||}, \frac{X_2}{||X||}\right) \quad (6)$$

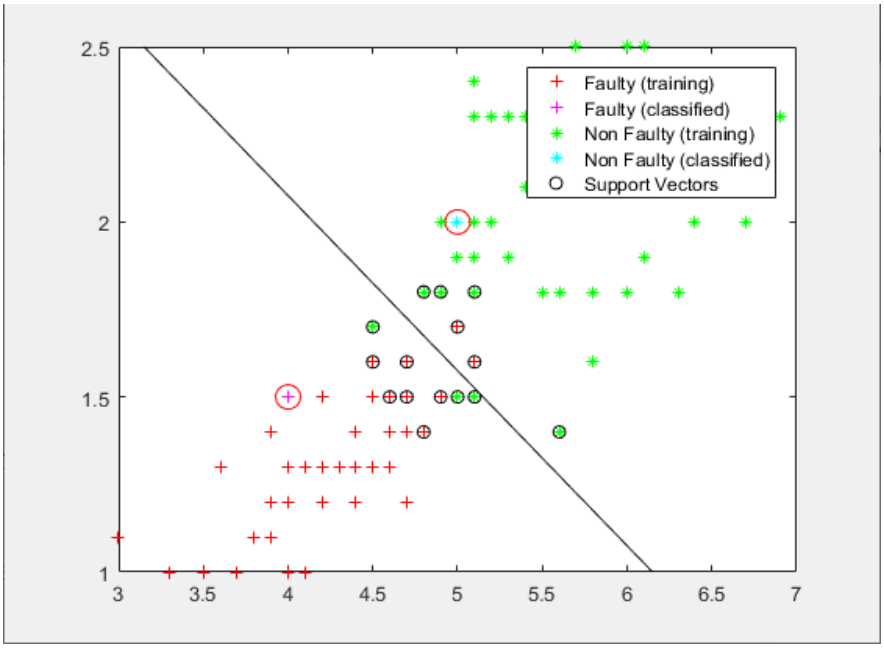

Figure 4 Classification using SVM

Figure 4 shows the classification of attacks using SVM algorithm. The graph shows the classified attacker node represented by the plus sign encircled by support vector, while non-faulty nodes are represented by the blue star. The general equations used by SVM are given below.

$$\cos(\theta) = \frac{X_1}{||X||} \quad \text{and} \quad \cos(\phi) = \frac{X_2}{||X||} \tag{7}$$

Evaluating equation (6) and equation (7), we observed that $\omega = (\cos(\theta), \cos(\phi))$ (8)

The hyperplane that is drawn to separate linearly separable data in SVM is defined by the line function as given by equation (9):

$$y = Ax + K \tag{9}$$

Where, K is the constant term and x and y are the coordinates. While modifying x with $X_1$and y withX_2, the above life function get modified by equation (10):

$$AX_1 - X_2 + K = 0 \tag{10}$$

If we compare $V = (X_1, X_2)$ and $\omega = (A, -1)$, the equation (10) can be rewritten as

$$\omega.X + K = 0 \tag{11}$$

Here, equation (19) represents the equation for hyperplane that is derived for two dimensional vectors. However, it has been observed that it works perfectly for 'n' number of dimensions as well. Further, this hyperplane equation obtained is used for making predictions using hypothesis function $Hypo_{fun}$ as follows:

$$if\ \omega.x + K \geq 0\ ; \quad Hypo_{fun}(X_i) = +1\ , \tag{12}$$

$$if\ \omega.x + K\ < 0\ ; \quad Hypo_{fun}(X_i) = -1, \tag{13}$$

Using equation (12) and (13) data that lie above the hyperplane will be classified as +1 class and that lie below the hyperplane will be classified as -1 class. The above equations show the ability to separate the data with a high level of precision.

In this context, an instrumentally high precision (99.2%) work was also proposed by [Almseidin et al., (2017)] who had evaluated the efficacy of various machine learning approaches for intrusion detection. His work established that Random Forest outperformed the other classifiers, namely, Multi-layer Perceptron, Naïve Bayes and Bayes Network with intrusion detection accuracy of 91.9%, 91.2% and 90.7% and precision of 97.8%, 98.8% and 99.2%, respectively. The steps followed by the best studied classifier, Random Forest Algorithm, in Almseidin et al. work are as follows:

1. Select $R \rightarrow random\ features$ from total features $\rightarrow M$
2. Where, $R \ll M$

Build decision trees with selected subsets

3. Calculate number of Nodes $\rightarrow$ $Node_n$ using best split point
4. Split node $\rightarrow$ sub_{nodes} using best split

5. Repeat steps 1 to 4 until $Node_n \rightarrow 1$
6. Repeat steps 1 to 5 to Build Forest with number of trees
7. Return the best feature by random selection process

Using above selection process, the study established the fact that if the implementation of a single classifier could reach an accuracy of 93.77% and precision of 99.2% then, the ensemble of various classifiers could further increase the intrusion detection accuracy to a higher level. Therefore, in the present work, authors have proposed the combination of high performance ANN and SVM classifiers. The Classification algorithm designed using dual techniques ANN with SVM is written below.

Algorithm 2: SVM with ANN

Input: Optimized Best Attributes as a training data (TD) of route node with route (FR), Cat as labelled target in terms of DDoS attack and N as a carrier neurons

Output: Optimized Route (OR)

Start

To select best route from FR, ANN is used

Index = Find index of N_{PROP} in FR

If index of route is normal then

OR(i) = FR (index)

Else

Mark as faulty route

End

Call and set the ANN using TD

Set, IDS-NET = NEWFF (TD, Cat, N)

IDS-NET = TRAIN (IDS-NET, T, G)

Current Sensor Nodes, N_C = Properties of current node in IDS Network

Sensor Nodes Characteristics = SVMCLASSIFY (IDS-NET, N_C)

If Sensor Nodes Characteristics is valid then

OR = Validated

Else

OR = Need Correction or mark as DDoS attack

If transmission not occurs with time period

Marked as DDoS Attacks

Else attacker founded in the FR

Marked as Malware Attacks

Else

Marked as Spoofing Attacks

End

End

Returns: OR as an Optimized and Validated Route

End

4. Experimental Result and Analysis

The proposed work is analysed against KKD cup 99 dataset, comprising of DDoS attack instances and a normal. Evaluation metrics of precision, recall and f-measure are employed to demonstrate the efficiency of the proposed work over 1000 simulation rounds. The parameters are calculated as follows:

$$Precision = \frac{T_P}{T_P + F_P} \quad (14)$$

$$Recall = \frac{T_P}{T_P + F_N} \quad (15)$$

$$F_{measure} = 2\left(\frac{Precision * Recall}{Precision + Recall}\right) \quad (16)$$

$$Accuracy\ or\ Detection\ Rate = \frac{T_P + T_N}{T_P + T_N + F_P + F_N} \quad (17)$$

Where, T_P represents True Positives, T_N represents True Negatives, F_P represents False Positives and F_N represents False Negatives observed during analysis. The parametric values of the performance parameter are listed in Table 1 against simulations varying from 10 to 2000 simulation rounds comparing attack and normal scenarios.

Table 1 Performance parameters

Number of Simulation Rounds	Precision		Recall		F-measure	
	DDoS	Normal	DDoS	Normal	DDoS	Normal
10	0.954	0.968	0.849	0.887	0.898	0.926
20	0.968	0.971	0.863	0.901	0.912	0.935
50	0.978	0.979	0.871	0.924	0.921	0.951
100	0.981	0.985	0.889	0.937	0.933	0.960
200	0.985	0.988	0.919	0.957	0.951	0.972
500	0.993	0.994	0.927	0.975	0.959	0.984
1000	0.994	0.996	0.943	0.985	0.968	0.990
1500	0.995	0.997	0.951	0.991	0.973	0.994
2000	0.996	0.997	0.961	0.997	0.978	0.997

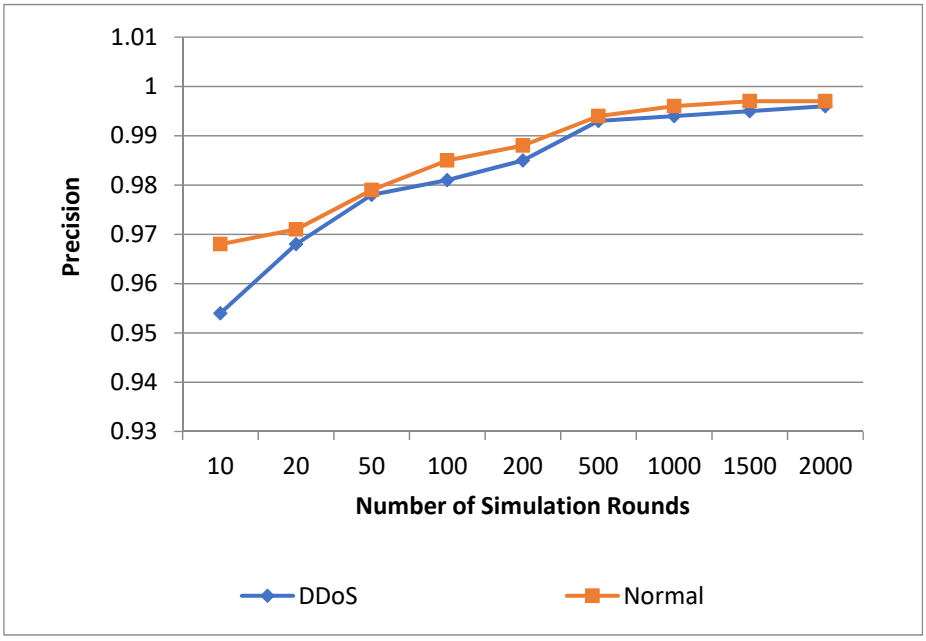

Figure 5 Precision Analysis

Precision analysis is further plotted in Figure 5 with simulation rounds along X-axis against the observed precision values for DDoS and Normal scenarios along Y-axis. It is observed that with the increase in the number of simulation rounds, the precision of the proposed work also increases. However, very small variation is observed after increase in the simulation rounds from 1000 onwards. On an average, graph shows an average precision of 0.982 under DDoS attack which is very near to the average precision of 0.923 using normal dataset. There is hardly a difference of 0.344% observed between attack and the normal case which reflects the strength of the proposed work.

Recall values of both scenarios are summarized in Table 1 that is also plotted in Figure 6. Recall is used to reflect the retrieval of the most relevant results. In the present, cases it is observed that recall demonstrated by the proposed work increases as the number of simulations are increased to 2000. In presence of DDoS attack, average recall of 0.908 is observed while it is 0.95 in absence of attack. In other words, 4.23% difference is observed between average recalls, under DDoS and normal scenarios.

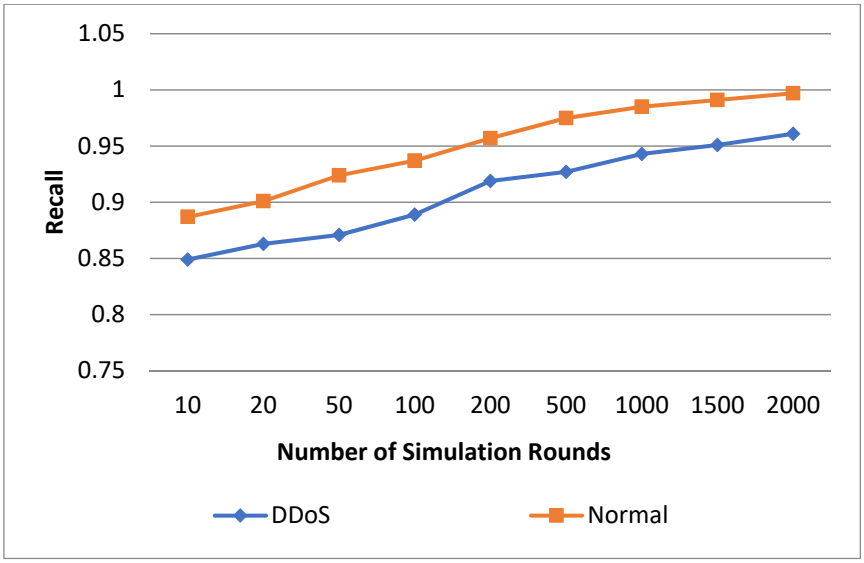

Figure 6 Recall Analysis

Further, evaluation is performed in terms of f-measure using the employed KDD dataset. Figure 7 compares the f-measure calculated using two cases. F-measure is a harmonic mean of precision and recall and is calculated to reflect the possibility of retrieving the most relevant outcomes. It is observed that f-measure of 0.978 and 0.997 is observed for the last simulation round while employing DDoS attack and normal datasets, respectively. The results show an average f-measure of 0.944 and 0.968 against DDoS attack and normal scenarios with an average difference of 2.4%. In other words, f-measure analysis shows that attacks could be effectively defended with increased simulation rounds using proposed work

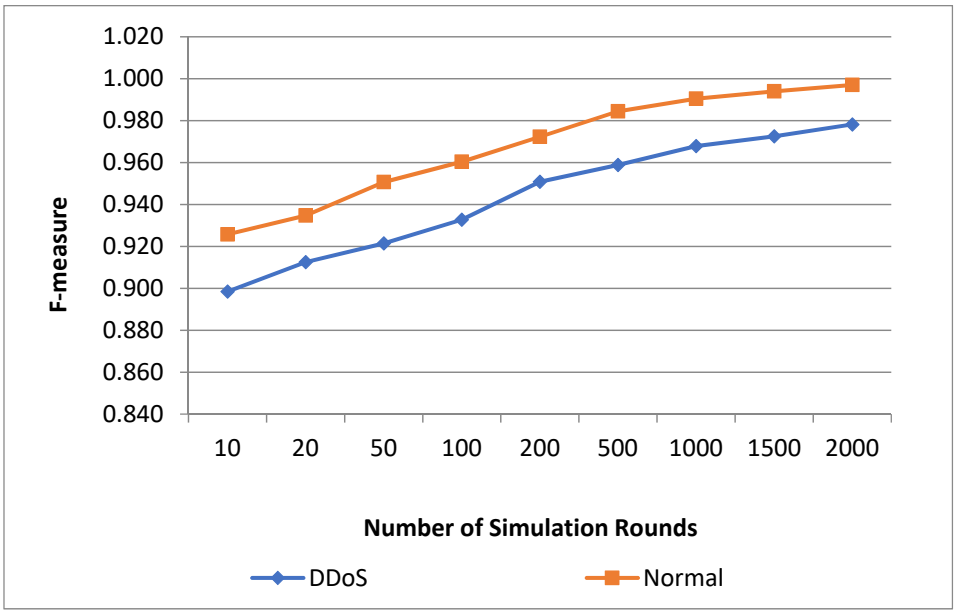

Figure 7 F-measure Analysis

Table 2 compares the TPR, FPR and detection rate of the proposed work against variation in the number of nodes from 50 to 600. The table reflects the sensitivity and accuracy of intrusion detections in terms of True Positive Rate and Detection Rate, respectively.

Table 2 TPR, FPR and Detection Rate for proposed intrusion detection system

Number of Nodes	TPR	FPR	Detection Rate
50	0.8547	0.2300	0.9555
100	0.9125	0.2015	0.9762
200	0.9235	0.1985	0.9824
300	0.9312	0.1877	0.9859
400	0.9411	0.1722	0.9873
500	0.9455	0.1685	0.9945
600	0.9562	0.1422	0.9956

Figure 8 shows that TPR raises from 0.854 to 0.956, FPR decreases from 0.23 to 0.142 and detection rate increases from 0.955 to 0.995 when the number of nodes is varied from 50 to 600. This reflects that hike in number of nodes causes an average change of 10.15%, 8.78% and 4.01% in case of TPR, FPR and detection rate of the proposed work.

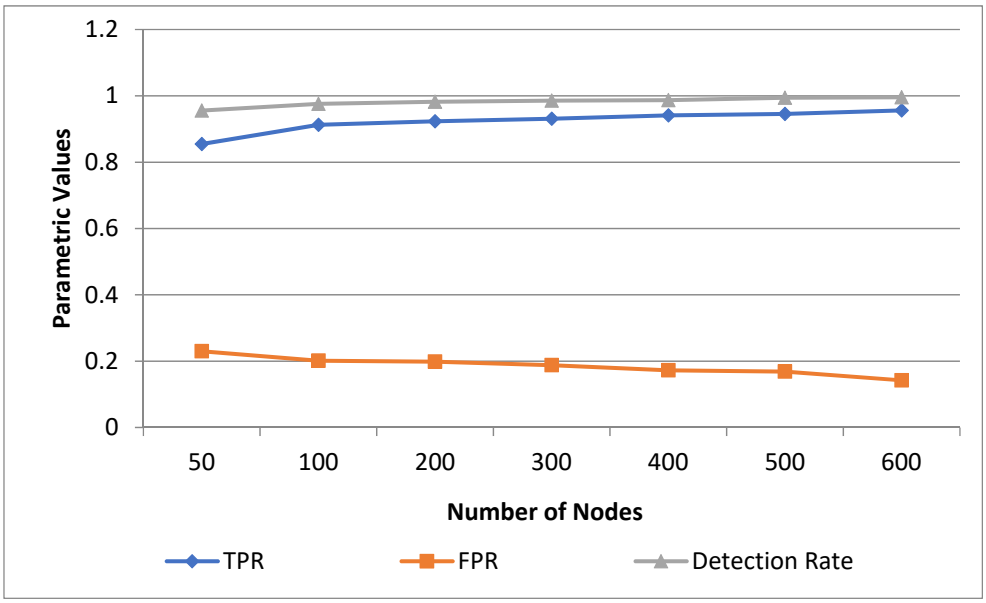

Figure 8 TPR, FPR and Detection Rate for proposed work

4.1 Comparison Result and Analysis

The effectiveness of proposed attack detection system is also evaluated against three existing works of [Velliangiri et al., (2020)], [Almseidin et al., (2017)] and [Alsharafat et al., (2013)] who have addressed intrusion detection against DDoS attack using KDD dataset. Table 3 summarizes and Figure 8 compares the detection accuracy of the proposed work against the existing works. Recently, [Velliangiri et al., (2020)] had proposed Taylor Elephant Herd Optimisation based Deep Belief Network to detect DDoS attacks and achieved a classification accuracy of only 83%. Further, [Almseidin et al., (2017)] had implemented Random forest classifier to achieve a detection accuracy of 93.77% while [Alsharafat et al., (2013)] work comprising of eXtended Classifier System with Artificial Neural Network (ANN-XCS) demonstrated a detection accuracy of 98.1% against the proposed work that have implemented the optimization techniques with combination of Support Vector Machine and Artificial Neural Networks to achieve a detection accuracy of 98.24%.

Table 3 Accuracy Comparison

Reference Work	Technique Used	Accuracy
Proposed Work	SVM with ANN	98.247
Velliangiri et al., (2020)	TEHO-DBN	83
Almseidin et al., (2017)	Random Forest	93.77
Alsharafat et al., (2013)	ANN-XCS	98.1

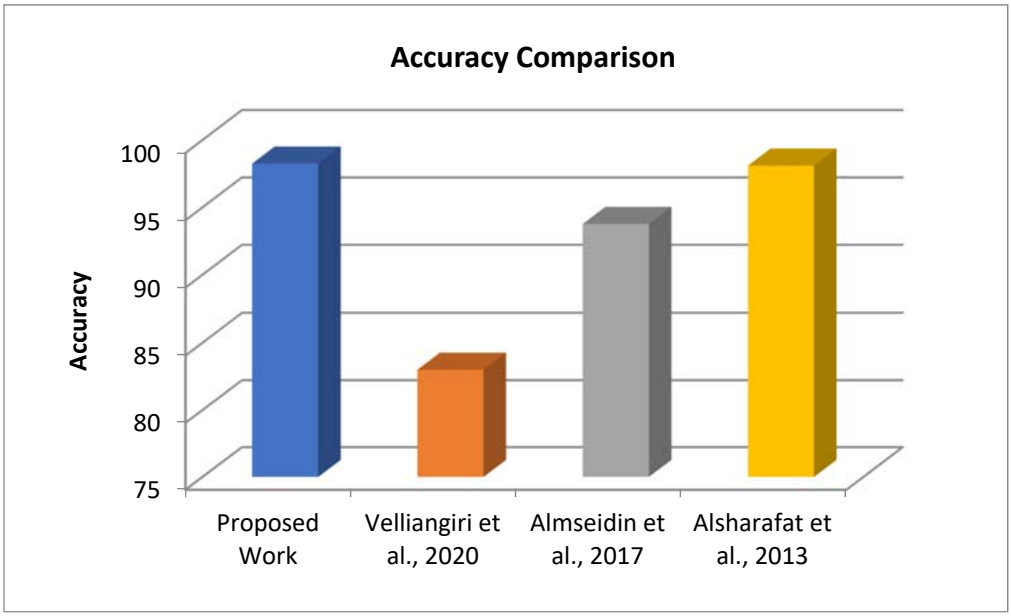

Figure 8 Accuracy comparison

The proposed work has implemented hybrid of ANN and SVM that is reflected in the higher detection accuracy of the proposed work and proved to be 15.247%, 4.477% and 0.147% better than the existing works of [Velliangiri et al., (2020)], [Almseidin et al., (2017)] and [Alsharafat et al., (2013)] respectively, in detecting DDoS attack.

Table 4 summarizes the comparison of precision of DDoS attack detection against the existing work of [Velliangiri et al., (2020)] and [Almseidin et al., (2017)] It is observed that the proposed system could successfully defend DDoS attack with an enhanced precision of 99.6% in comparison to the precision of 89.6% by Velliangiri et al. and 99.2% by Almseidin et al. work. This means that proposed work proved to be 10% better than Velliangiri et al. work and 0.4% better than Almseidin et al. work in terms of precision for intrusion detection in case of DDoS attack.

Table 4 Precision Comparison

Reference Work	Technique Used	Precision
Proposed Work	SVM with ANN	99.6
Velliangiri et al., (2020)	TEHO-DBN	89.6
Almseidin et al., (2017)	Random forest	99.2

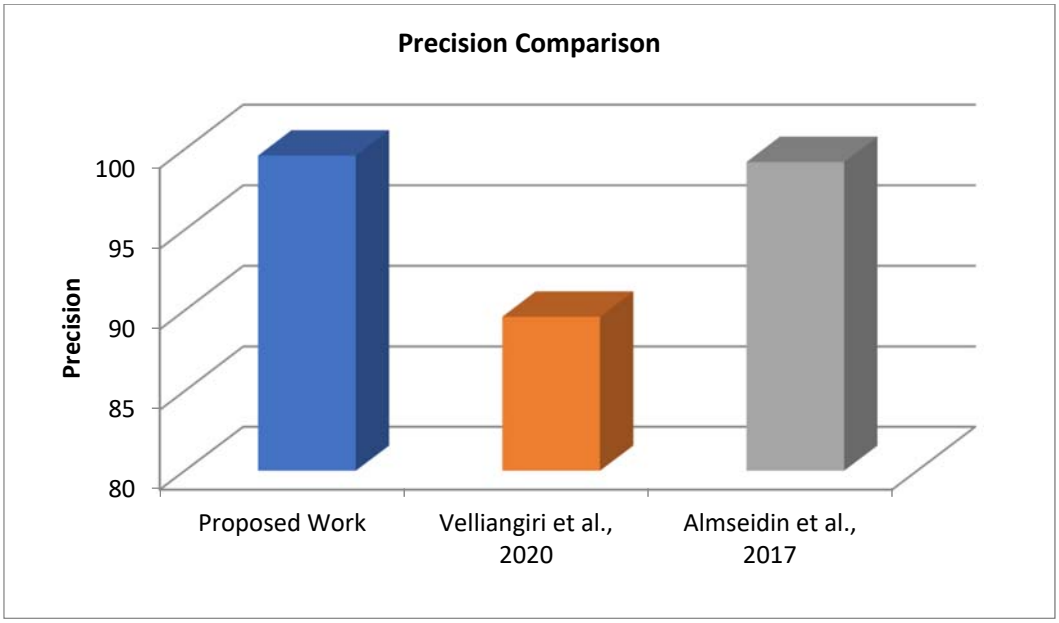

Figure 9 Precision comparison

The precision and accuracy of intrusion detection comparison together demonstrates the effectiveness of the proposed work in case of DDoS attack. The proposed work demonstrated the enhanced performance because of the implementation of an appropriate selection method for data features followed by the training of the network with dual classifiers ANN and SVM.

5. Conclusion

In the present work, the challenge of detecting and defending most common cyber attack DDoS has been addressed. Authors have implemented Genetic Algorithm based architecture for feature extraction of various types of attacks followed by ANN and SVM hybrid based detection to propose an enhanced cyber-attack detection system. The effectiveness of this proposed work is reflected in the terms of performance parameters namely, recall, precision and f-measure while comparing them against normal data and existing works. High precision of 0.996 with detection accuracy of 98.247% is achieved in case of DDoS attack compared to existing studies. In other words, proposed work demonstrated 10% and 0.4% better precision against existing works dedicated for intrusion detection under DDoS attack using KDD dataset. Hence, higher precision value reflects the efficiency of the proposed architecture in offing security against cyber attacks. Other parameters that have been taken to demonstrate the efficiency of this work for intrusion detection are TPR, FPR and accuracy. The proposed work shows better performance in terms of observed parameters against the existing works.

References

[1] Agarwal, N.; Hussain, S. Z. (2018): A Closer Look at Intrusion Detection System for Web Applications. Security and Communication Networks, 2018, pp 1-27.
[2] Almseidin, M.; Alzubi, M.; Kovacs, S.; Alkasassbeh, M. (2017): Evaluation of machine learning algorithms for intrusion detection system. In: IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY), IEEE 2017, pp. 277-282.
[3] Alsharafat, W. (2013): Applying Artificial Neural Network and eXtended Classifier System for Network Intrusion Detection. International Arab Journal of Information Technology 10(3)
[4] Chiba, Z.; Abghour, N.; Moussaid, K.; Rida, M. (2019): Intelligent approach to build a Deep Neural Network based IDS using combination of machine learning algorithms." Computers & Security 86, pp. 291-317.
[5] Daffu, P.; Kaur, A. (2016): Mitigation of DDoS attacks in cloud computing. In: 5th International Conference on Wireless Networks and Embedded Systems (WECON), IEEE 2016, pp. 1-5.
[6] Deshpande, P.; Sharma, S. C.; Peddoju, S. K.; Junaid. S. (2018): HIDS: A host based intrusion detection system for cloud computing environment. International Journal of System Assurance Engineering and Management 9(3), pp. 567-576.
[7] Hajimirzaei, B.; Navimipour, N. J. (2019): Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm. ICT Express 5(1), pp. 56-59.
[8] He, X.; Dai, H.; Ning, P. (2016): Faster learning and adaptation in security games by exploiting information asymmetry. IEEE Transactions on Signal Processing 64(13), pp. 3429-3443.
[9] Hosseini, S.; Azizi, M. (2019): The hybrid technique for DDoS detection with supervised learning algorithms." Computer Networks 158, pp. 35-45.
[10] Jovic, A.; Brkic, K.; Bogunovic, N. (2015): A review of feature selection methods with applications. In: 38th international convention on information and communication technology, electronics and microelectronics (MIPRO), IEEE 2015, pp. 1200-1205.
[11] Kajal, A.; Nandal, S. K. (2019): A Hybrid Algorithm using neural network and artificial bee colony for cyber security threats. International Journal of Innovative Technology and Exploring Engineering, 8(12), pp. 1-6.
[12] Kajal, A.; Nandal, S. K. (2020): Cyber Security against DDoS, Malware, Spoofing attacks using Machine Learning with Genetic Algorithm. International Journal of Advanced Science and Technology, 29(5), pp. 5388-5400.

[13] Kannan, A.; Venkatesan, K. G.; Stagkopoulou, A.; Li, S.; Krishnan, S.; Rahman, A. (2015): A novel cloud intrusion detection system using feature selection and classification. International Journal of Intelligent Information Technologies 11(4), pp. 1-15.
[14] Kannan, A.; Maguire Jr, G. Q.; Sharma, A.; Schoo, P. (2012): Genetic algorithm based feature selection algorithm for effective intrusion detection in cloud networks. In: IEEE 12th International Conference on Data Mining Workshops, pp. 416-423.
[15] Karaboga, D.; Ozturk, C. (2011): Hybrid artificial bee colony algorithm for neural network training. In: Proceedings of 2011 IEEE Congress on Evolutionary Computation (CEC), pp. 84-88.
[16] Kesavamoorthy, R.; Soundar, K. R. (2019): Swarm intelligence based autonomous DDoS attack detection and defense using multi agent system. Cluster Computing 22(4), pp. 9469-9476.
[17] Kushwah, G. S.; Ali, S. T. (2017): Detecting DDoS attacks in cloud computing using ANN and black hole optimization. In: 2nd International Conference on Telecommunication and Networks (TEL-NET), IEEE 2017, pp. 1-5.
[18] Lima F.; de, F. S.; Silveira, F. A.; Junior, A. D. M. B.; Solar, G. V.; Silveira, L. F. (2019): Smart detection: an online approach for DoS/DDoS attack detection using machine learning. Security and Communication Networks 2019.
[19] Osanaiye, O.; Cai, H.; Choo, K. K. R.; Dehghantanha, A.; Xu, Z.; Dlodlo, M. (2016): Ensemble-based multi-filter feature selection method for DDoS detection. EURASIP Journal on Wireless Communications and Networking 2016(1), 130.
[20] Pandeeswari, N.; Kumar, G. (2016): Anomaly detection system in cloud environment using fuzzy clustering based ANN. Mobile Networks and Applications 21(3), pp. 494-505.
[21] Patel, A.; Taghavi, M.; Bakhtiyari, K; Junior, J. C. (2013): An intrusion detection and prevention system in cloud computing: A systematic review. Journal of network and computer applications 36(1), pp. 25-41.
[22] Sakr, M. M.; Tawfeeq, M. A.; El-Sisi, A. B. (2019): Network intrusion detection system based PSO-SVM for cloud computing. International Journal of Computer Network and Information Security 10(2), 22.
[23] Somani, G.; Gaur, M. S.; Sanghi, D.; Conti, M.; Buyya, R. (2017): DDoS attacks in cloud computing: Issues, taxonomy, and future directions. Computer Communications 107, pp. 30-48.
[24] Tsai C. F.; Hsu, Y. F.; Lin, C. Y.; Lin, W. Y. (2009): Intrusion detection by machine learning: A review. Expert Systems with Applications, 36(10), pp. 11994–12000.
[25] Velliangiri, S.; Karthikeyan, P.; Kumar, V. (2020): Detection of distributed denial of service attack in cloud computing using the optimization-based deep networks. Journal of Experimental & Theoretical Artificial Intelligence, pp. 1-20.
[26] Watson, M. R.; Marnerides, A. K.; Mauthe, A.; Hutchison, D. (2015): Malware detection in cloud computing infrastructures. IEEE Transactions on Dependable and Secure Computing 13(2), pp. 192-205.
[27] Zargar, S. T.; Joshi, J.; Tipper, D. (2013): A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. IEEE communications surveys & tutorials 15(4), pp. 2046-2069.