DDoS Intrusion Detection Model for IoT Networks using Backpropagation Neural Network

Jasem Almotiri
Department of Computer Science, College of Computers, and information Technology
Taif University, Taif, Saudi Arabia

***Abstract*—In today's digital landscape, Internet of Things (IoT) networking has grown dramatically broad. The major feature of IoT network devices is their ability to connect to the internet and interact with it through data collecting and exchanging. Distributed Denial of Service (DDoS) is one form of cyber-attacks in which the hackers penetrate a single connection and then multiple machines are operating together to attack one target. The direct connectivity of IoT devices to the internet makes DDoS attacks worse and more dangerous. The more businesses adapted IoT networks to streamline the operations, the more allowing of DDoS intrusions at small and large scales to take place. Therefore, the intrusion detection module in the IoT networks is not optional in today's business environment. To achieve this objective, in this paper, an intelligent intrusion detection model is proposed to detect DDoS attacks in IoT networks. The intelligent model is a backpropagation neural network-based framework. The results are analyzed using different performance measures. The proposed model proves a detection rate of 99.46% and detection accuracy of 95.76% using the up-to-date benchmark CICDDoS2019 dataset. Furthermore, the proposed model has been compared with the most recent DDoS intrusion detection schemes and competitive performance is achieved.**

Keywords*—DDoS; backpropagation neural network; IoT network; intrusion detection; CICDDoS2019

I. INTRODUCTION

IoT networks are considered a considerable movement in the world of networking and data communication. This movement is majorly driven by the fifth-generation network (5G), which is expected to broaden todays' IoT functionalities. As the adaption of IoT networks is in a dramatic increase, IoT network is expected to be a huge growing market.

However, this growth is thwarted by DDoS attacks which are the most prevalent cyber threats. Some cybersecurity experts consider IoT networks as the major force behind DDoS attacks. These types of attacks have a distributive nature; in particular, they depend on one vulnerable device in the IoT network to create an opening for intruders to make use of any other IoT devices to drive a huge amount of traffic.

IoT platform security is embedded in the process of product development of IoT-based devices. However, when it comes to cybersecurity, no data processing unit is an island. From data production, data transmission, data processing, data visualization to data analysis and prediction, all of these major stages of IoT networking are considered open gates for intrusions and hackers to perform their attacks especially DDoS and Botnet attacks. The botnet is a collection of hijacked internet-connected devices that are used to execute a large-scale attack. The infected devices are controlled by attacks actors who they often cybercriminals and are used to perform particular malicious duties in an unobserved manner from the user. One of the major tasks to be done by a botnet is to generate malicious traffic for DDoS attacks.

DDoS attacks are carried out when multiple machines (devices) are operating together for sake of attacking one target device. Using control and command software, DDoS attacker avails of the weaknesses and security vulnerabilities of one IoT device to control several IoT network devices. Once the target device is in control, DDoS attack admits exponentially requests to be sent to the target device, which enhances the power of attack on one hand and boosts the difficulty of detecting attribution on the other hand, where due to the distributive nature of the DDoS attack, the original source of the attack becomes harder to be identified. DDoS attacks are easy to deal with in the short term, but it becomes very difficult in the long term. Due to the distributed nature of the DDoS attack, it is considered one of the most fatal enemies of the internet of things platform. In principle, if internet-connected devices get hacked, the DDoS malware could easily spread to other devices in the network [1], [2].

Although IoT has used a variety of protocols for security purposes, hackers and intruders developed more complex and intelligent techniques to fulfill their penetrations ending up with misalignment between the speed of IoT development and IoT cybersecurity. In the case of dealing with sensitive data processing and management through IoT platforms, there is a pressure to close the gap that caused an increased vulnerability, especially for crucial data where security becomes the single most cardinal factor that companies, and organizations consider when purchasing IoT products.

In response, a new generation of artificial intelligence-based intrusion detection techniques to address the limitations of the conventional intrusion detection techniques has emerged and adapted as a major building block of IoT security systems. Recently, the awareness to use the machine learning in general and artificial neural networks in particular to secure network traffic against DDoS attacks has increased rapidly. DDoS Detection systems based on artificial neural networks are such typical solutions to model and predict malicious behavior over network traffic flows and Backpropagation neural network are considered one of the powerful yet flexible supervised training algorithms.

In the context of supervised neural networks, backpropagation neural network is considered one of the powerful classifying and filtering engines in this field. Much of the power of backpropagation arises from the fact that the repeated composition of specific types of nonlinear functions boosts the abstract representation power of the neural network. Moreover, an efficient computation of the gradient at each layer can be highly performed by the backpropagation training algorithm. Therefore, backpropagation neural network can efficiently learn the abstract representation of the behavior of network flows in general, and the pattern of the malicious flows in particular.

Some researchers, in this field, deployed backpropagation in its original form to design and develop different schemes for intrusion detection systems. However, the majority of researchers tend to use the enhanced versions of backpropagation in order to address the shortcomings of the classical backpropagation algorithm in an attempt to lower the time and the computational overhead of the training phase or to remedy the convergence difficulties of the classical version. Other scenarios use a standard backpropagation neural network that heavily depends on a variety of features engineering techniques in prior.

These scenarios come at the expense of higher design complexity associated with new born issues related to time cost and resources. Furthermore, in the context of intrusion detection, the overall performance of many of the predictive models based on the enhanced versions do not exceed that can be achieved by the classical one.

With minimal neural network architecture, and without any features preprocessing, the main aim of this work is to prove that the conventional standard backpropagation can be used to build an accurate yet robust predictive model for DDoS attacks. We have achieved this by conducting the training using modern up-to date DDoS dataset and conducting rigorous testing analysis while the performance of the predictive model is benchmarked using highly indicative standard performance metrics.

The major improvements that we presented in this article are reviewed as follows:

- Propose an intelligent DDoS intrusion detection system that can predict DDoS malicious network traffic in IoT networks by exploiting the predictivity power of the standard backpropagation neural network.
- In this work, CICDDoS2019 dataset is used to verify that the proposed detection model applies to different types of DDoS malicious traffic flows.
- We have proved that using low-complexity standard version of backpropagation based neural network can achieve comparable detection performance even though no form of features engineering (such as feature weighting or features selection), was considered.

The rest of the paper is organized as follows: Section II presents related works; Section III describes our proposed DDoS intrusion detection methodology followed by the experimental performance evaluation in Section IV and Section V concludes this paper and suggests future directions.

II. Related Work

Exploiting the very basic form of backpropagation algorithm applied on multi-layered perceptron neural network, an intrusion detection model seeking a reduction in false alarm rate as a major performance aspect was proposed by [3] where it shows the intrusion recognition capability of the backpropagation algorithm despite the rudimentary framework.

Using internet packet traces as an experimental dataset, authors in [4] used the standard backpropagation neural network to thwart DoS and DDoS attacks in IoT environments.

As a precise and efficient classifier [5] used a standard backpropagation network to classify DDoS traffic after an initial judgment of the characteristics of the abnormal network traffic.

Using a backpropagation-based autoencoder, [6] designed a joint anomaly and signature-based DDoS intrusion detector implemented in the cloud. Based on a behavioral study that collected a variety of DDoS signatures in one database, the targeted traffic was first compared to the known DDoS attacks. If no matching has occurred, then the traffic fed to the backpropagation autoencoder to be classified into DDoS or benign. If a DDoS attack was detected, the signature of this attack was used to update the database of DDoS signatures.

Despite the vivid versatility of backpropagation in designing intelligent intrusion detection models/systems, it has some downsides such as local minima, slow convergence, and network paralysis. These downsides moving the wheel of extension around the axel of standard backpropagation were enhanced versions of backpropagation emerged and implemented for harder predictive tasks such as DDoS/DoS detection. As an example of using an enhanced version of backpropagation in the domain of intelligent intrusion detection, [7] presented an intrusion detection system composed of hybrid phases of misuse detection and anomaly detection by applying the Levenberg-Marquardt algorithm as a technique to optimize the backpropagation-based network that was used as the classification engine of the proposed predictive system.

Applying the same backpropagation customization, authors in [8] used multi-core technology to detect DDoS intrusions via neural networks. Multi-core uses one CPU that combines a couple or more independent cores into a single circuit. Their IP flow-based DDoS intrusion detection technique is built based on the idea that an IP packet holds information of the upper layer which can be exploited into special attributes representing the special characteristics of DDoS attacks in a well-posed manner. Then, attributes vectors are fed into a Levenberg-Marquardt based backpropagation neural network as input to be classified into DDoS or benign traffic.

Authors in [9] proposed a DDoS intrusion detection algorithm composed of two main phases: a training phase and a detection phase. In the training phase, a non-linear time series model called GARCH was used to evaluate the normal traffic

prediction, whereas a feed-forward backpropagation neural network was used as the benign/DDoS classifying engine.

For cloud security, based on backpropagation neural networks, authors in [10] proposed a DDoS detector model that offered a solution to tracing back a given cloud traceback for the sake of finding the source of the real attack. Moreover, they introduced a cloud protector built using a feedforward backpropagation neural network.

III. METHOD

As IoT networks are broad, the attack surface for this type of network is even broader. This has been reflected in various methodologies and techniques designed to fall under the umbrella of IoT network security. Public Key Infrastructure (PKI) authentication, securing Application Program Interface (API), and network intrusion detection systems, the lists long are just a few of the techniques IT leaders can use to combat the growing malicious penetrations rooted in vulnerable devices of IoT networks. As a rule of thumb, as the ways available for devices to be able to connect, the more ways attackers and hackers can intercept them.

Therefore, one of the most harmonious security techniques that can be deployed for IoT network security is the security gateways, which act as intermediary units between the IoT network devices and the outside world (internet, cloud computing, etc.). These units are equipped with processors chips with more computational capabilities, memory, and processing power rather than that exists in the IoT devices themselves. These additional features enable gateway units to run artificial neural network-based intelligent intrusion detection systems to ensure intruders cannot access the IoT network devices they connect.

Fig. 1 shows the general high-level framework of our proposed DDoS intrusion detection model in the context of IoT networking where at the heart of the IoT securing gateway lies the proposed intrusion detection model. Fig. 2 illustrates the flow pipeline of our proposed system, which is built using the backpropagation feedforward artificial neural network that trained, validated, and tested using a real benchmark CICDDoS2019 dataset [11]. As illustrated in Fig. 2, once the dataset is pre-processed in a way harmonious to the format accepted by the neural network, it is split into two datasets: training and testing. Then, a backpropagation neural network is trained to obtain the optimal parameters and meta parameters of the network structure. Afterward, the network is ready to be used as intelligent DDoS intrusion detection, where a new set of flow traffic (DDoS and normal traffic) is presented to the network, where it detects the DDoS attack traffic and raises an alarm. The following subsections elaborate each module of our proposed model shown in Fig. 2 in detail.

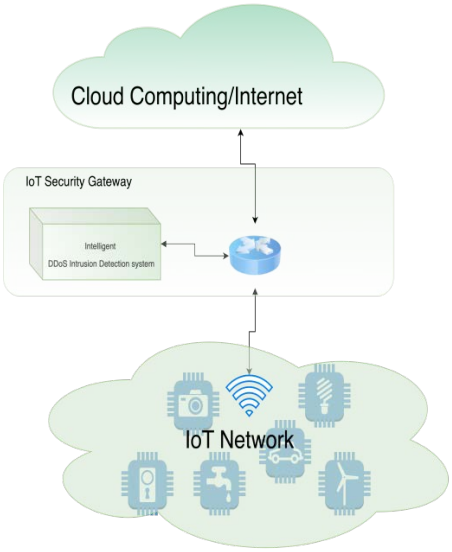


Fig. 1. High-Level Framework of the Proposed DDoS Intrusion Detection System in the Framework of IoT Networks.

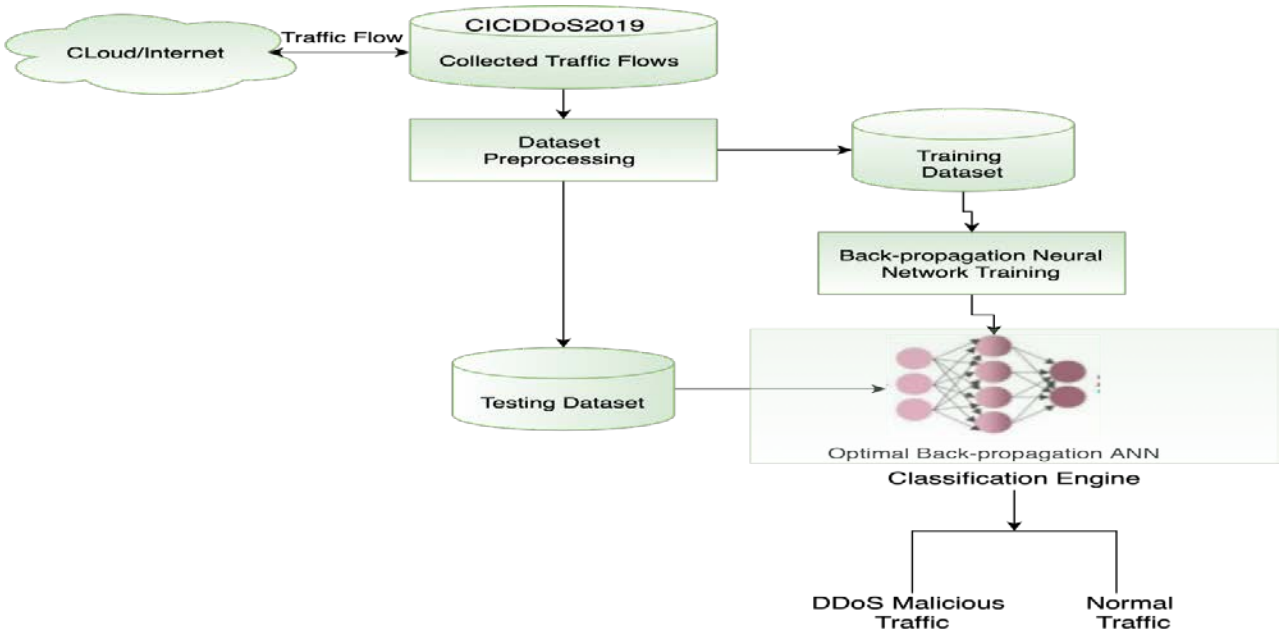


Fig. 2. Proposed DDoS Intrusion Detection System using Standard Backpropagation Neural Network.

A. Dataset Pre-Processing

To evaluate our backpropagation network-DDoS detection model, the real benchmark CICDDoS2019 dataset [11] is used. This dataset is composed of two types of DDoS attacks: reflection and exploitation in form of both malicious and normal network packets.

Even though the CICDDoS2019 dataset is major consists of numerical values, dataset preprocessing is an essential step to validate its suitability for both: as input to the neural network and for the classification task of the model. Fig. 3 illustrates the basic steps that applied to the raw CICDDoS2019 dataset which include: dataset cleansing, features selection, and dataset normalization.

Initially, the total number of raw dataset features is 81, in the dataset cleansing, we select the features that have zero values, which have no impact on the classification output of the neural network and dropped them out resulting in 22 unique features. Moreover, all dataset samples that contain infinity or misleading values have been removed as well. The label column of the dataset is converted into a numerical representation where 0 represents the benign traffic and 1 represents the DDoS one.

As the last step of dataset preprocessing, the dataset values are normalized using the standard min-max normalization [12], [13] as illustrated in (1):

$$\acute{v}_i = \frac{v_i - min_{DB}}{max_{DB} - min_{DB}} \cdot (new_{DB}^{max} - new_{DB}^{min}) + new_{DB}^{min} \quad (1)$$

where (1) represents the min-max normalization step that linearly transformed the raw database $DB = CICDDoS2019$ values $v_i \in DB$ into new values $\acute{v}_i$ in the range $[new_{DB}^{max}, new_{DB}^{min}]$.

B. Proposed Model

At the core of the proposed model, a backpropagation feedforward neural network is used to model the behavior of the flows of the network traffic. The basic annotated structure of the backpropagation neural network is shown in the simplest form of the feedforward network structure composed of input units at the left, any number of intermediary hidden layers, and a layer of output units at the right.

Connections from the second hidden layer to the last L hidden layers are hidden whereas the connections from higher layers to lower layers are forbidden. Backpropagation comprises two major phases: (1) Forward phase, and (2) Backward phases. In the forward phase, the outputs of all nodes are computed, the local derivatives of the nodes 'activation function relative to the net inputs are computed as well.

On the other hand, the main task of the backward phase is to aggregate the products of these local values over all paths from the nodes to the network outputs. In the forward phase, the components of an input training vector are fed into the neural network. This results in driving a forward cascade of computations across network layers using the current state of weights to yield the network output, which represents the predicted output of the network. Afterward, the predicted output is compared to the label associated with the training instance and the derivative of the loss function concerning the output is computed.

The derivative of the resulting loss is fed to the backward phase where it is used to compute the loss concerning the weights in all layers in the backward path.

As pictorially illustrated in Fig. 4, Let $\mathfrak{D} = \{y, d\}$ refers to the training dataset composed of the input-output (label) pairs and letx_j refers to the net input of thej^{th} unit where x_j is a linear combination function of the i^{th} weighted outputs y_i of i^{th} layer as in (2):

$$x_j = \sum_i^I y_i \, w_{ji} \quad (2)$$

Where

I : refers to the number of features of input vector (for our case, $I = 63$).

w_{ji}: is the weight of the connection between the layer (j) and the layer (i).

y_i : is the output of the layer (i).

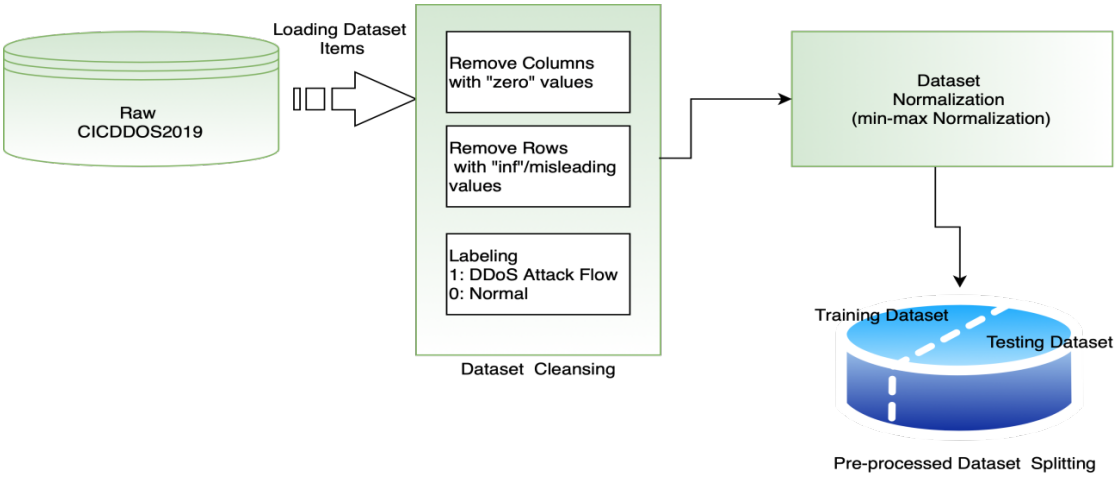


Fig. 3. CICDDoS2019 Dataset Pre-Processing Steps.

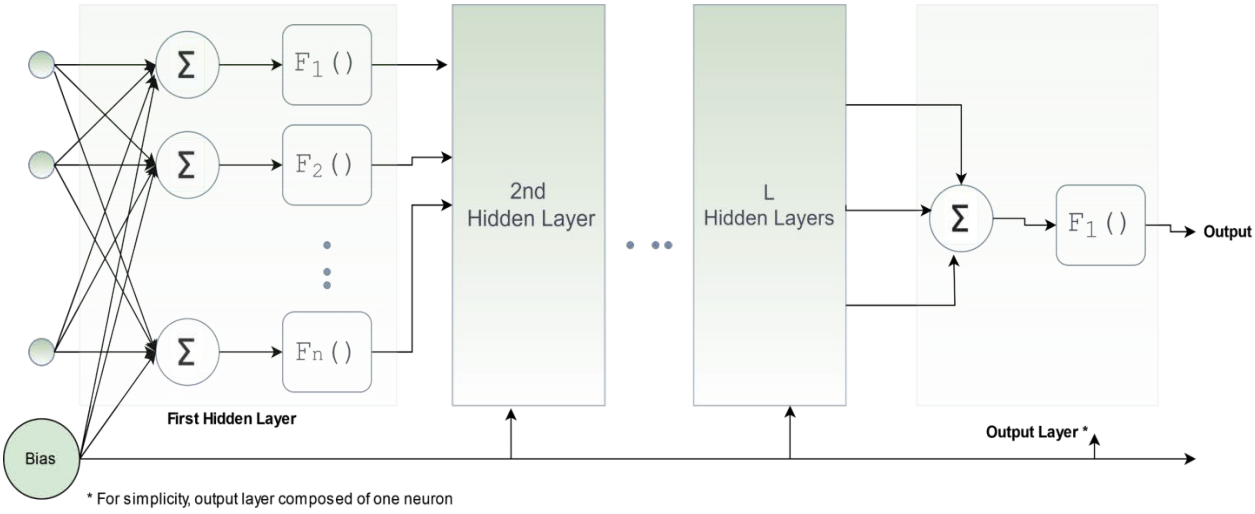


Fig. 4. Backpropagation Neural Network Layers Structure.

As shown in Fig. 4, all neurons are given biases by introducing an extra input line with no input but of a constant output (always has a value of 1) and it treated as an extra weighted input connection. Each neuron in the network has a real valued nonlinear function of its total input as in (3):

$$y_j(x) = \frac{(1-e^{-\alpha x_j})}{(1+e^{-\alpha x_j})} \quad (3)$$

$where$:

Δw : represents the updated value of the weight vector $\boldsymbol{w}$.

γ: is a positive learning step (learning rate) parameter.

The target is to find a set of weights parameters generated the network that is same as (or sufficiently close to) the desired output, the network model has the given set of weights parameters which used to make predictions and the differences between those predictions and the actual outputs are computed as error values. Typically, we seek to minimize the error of the function given by in (4):

$$x_j = \sum_i^I y_i \, w_{ji} \quad (4)$$

Where, E is the error function, d is the index over target outputs, j is an index over the output units of the output layer.

Gradient decent of the error function ∇E lies at the heart of backpropagation, where it seeks to change the weights parameters through multiple evaluations where the optimization algorithm ($\nabla E = 0$) navigates down the gradient of error function till the minimum is found.

IV. Experimental Results and Discussion

This section analyzes the performance of the proposed DDoS intrusion detection framework using backpropagation-based predictive model. This system is implemented, and experimentation is performed on Intel(R) Core (TM) i7-4500U CPU @ 1.80GHz and 2.40 GHz with 8GB of RAM and 64-bit Windows operating system. For more deeper experimental analysis and more controllability, we use MATrix LABratory (MATLAB)®2021b to build the back propagation neural network-based model from scratch where the available built-in neural network toolbox has not been used.

A. Performance Metrics

For purpose of model analysis, we used a subset of CICDDoS2019 dataset [11] we conducted many simulation trials, where the training dataset volume almost equals the volume testing dataset. In order to evaluate the detection performance of the system, we first established the confusion matrix, as shown in Table I. Then, the confusion matrices of the training and testing results are obtained. Based on the confusion matrices we considered the performance metrics that include Accuracy, Precision, F1-measure, False Positive Rate (FPR), Recall, Mathew Correlation coefficient and Kappa coefficients as in (5)-(11).

$$\text{Accuracy (Acc)} = \frac{(TP+TN)}{(TP+TN+FP+FN)} \quad (5)$$

$$\text{Precision (Pr)} = \frac{TP}{TP+FP} \quad (6)$$

$$\text{Mathew Correlation Coefficient (Mcc)} = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (7)$$

$$\text{Kappa Coefficient (K)} = \frac{Obs^{agree}-Expect^{agree}}{1-Expect^{agree}} \quad (8)$$

Where

$$Obs^{agree} = ACC = \frac{(TP+TN)}{(TP+TN+FP+FN)},$$

$$Expect^{agree} = \frac{A+B}{(TP+TN+FP+FN)},$$

TABLE I. Confusion Matrix

	Predicted	
Actual	Normal	DDoS
Normal	TN	FP
DDoS	FN	TP

$$A = \frac{(TP + FN)(TP + FP)}{(TP + TN + FP + FN)},$$

$$B = \frac{(FP + TN)(FN + TN)}{(TP + TN + FP + FN)}$$

$$\text{FPR} = \frac{FP}{FP+TN} \quad (9)$$

$$\text{F-Measure (F1)} = \frac{2}{\frac{1}{Recall}+\frac{1}{Pr}} \quad (10)$$

$$\text{Recall (Sensitivity)} = \frac{TP}{TP+FN} \quad (11)$$

where, False Positive (FP) and False Negatives (FN) refers to misclassified events. In contrary, True Positive (TP) and True Negative (TN) refers to the events that are correctly predicted by the model, i.e., if the model predicts normal events as normal, then, it is recorded as TN, whereas, if the model predicts the DDoS attack traffic flow as a DDoS malicious attack traffic, then it is recorded as TP.

B. Results

In this section we performed a series of experiments to determine the optimal network model architecture of the proposed backpropagation-based model along with varying many networks' hyper-parameters.

Afterwards, we performed another set of experiments that analyze the impacts of varying specific network parameters on the detection performance of the proposed intrusion detection model. We used four layers backpropagation neural network with two hidden layers of 64 neurons with $tanh(x) = (\frac{2}{1+exp(-\alpha x)} - 1)$ as activation function, where the optimal value of the sigmoid slope α parameter is to be determined through the experimental analysis.

Table II shows the architecture of this network whereas Table III shows the optimal values of hyper-parameters of the backpropagation neural network used in our proposed intrusion detection model. Dataset was normalized via min-max normalization with [0.5, +0.5] range and then have been split into training and testing sub-datasets with normal/DDoS distribution as shown in Fig. 5.

Although in the real-time applications, the number of DDoS attack traffic is much less than the normal one, for a robust intrusion detection system, we used a training dataset composed of about 50% DDoS attacks and the same scenario was used for test dataset as can be noted from Fig. 5.

Table IV and Table V show the confusion matrices of both training and testing phases respectively, meanwhile, Table VI represents the detection performance of both stages in terms of performance metrics listed in (5)-(11).

From Table IV, Table V, TPR and TNR are high whereas FPR and FNR are low. Moreover, besides the high detection performance shown in Table VI, it is noticeable that there is a subtle difference in the detection performance between the training and testing phases, therefore, our backpropagation model is not underfit or overfit.

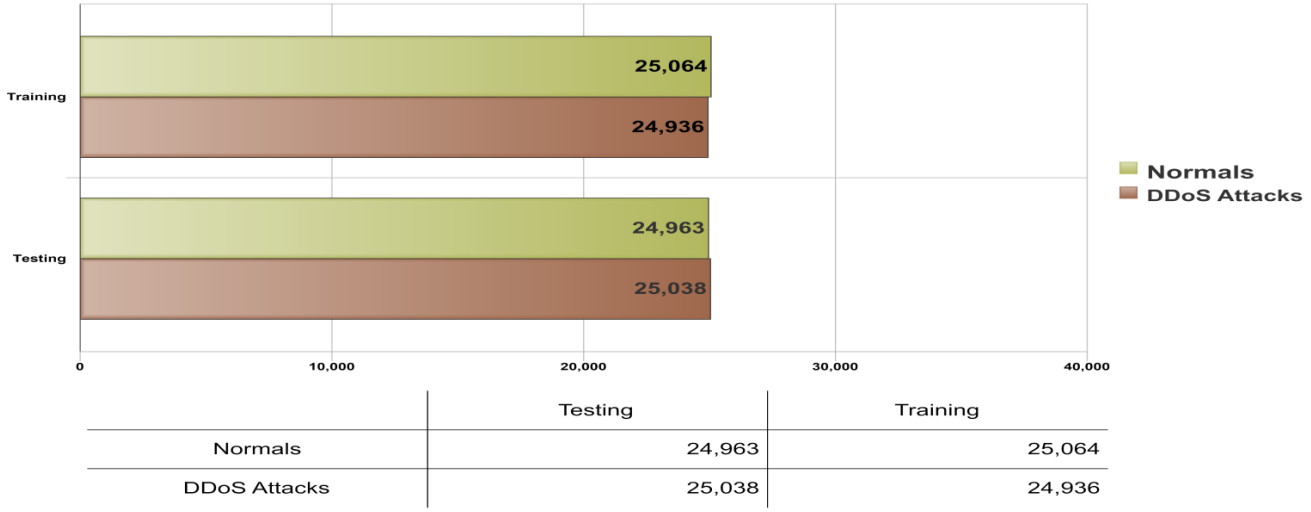
In terms of True positive rate, True negative rate, False positive rate, and False negative rate illustrated in (12)-(15), Table VII summarizes a comparison between our approach and a recent work proposed by Liu et al. [14].

TABLE II. NETWORK LAYERS ARCHITECTURE

Layer	Neurons number	Activation Function
Input	64	none
Hidden #1	44	Tanh
Hidden #2	20	Tanh
Output	1	Tanh

TABLE III. HYPER-PARAMETERS OF NETWORK MODEL

Parameter	Value
Learning rate γ	-10
Number of Epochs	4000
Training batch size	50,000
Loss Function	Mean Square Error (MSE)



	Testing	Training
Normals	24,963	25,064
DDoS Attacks	25,038	24,936

Fig. 5. Frequency of DDoS Intrusions for Training and Testing Datasets for the 2-Labels (Normal/DDoS) Scenario.

TABLE IV. CONFUSION MATRIX OF TRAINING PHASE

	Predicted	
Actual	Normal	DDoS
Normal	23092	1893
DDoS	134	24881

TABLE V. CONFUSION MATRIX OF TESTING PHASE

	Predicted	
Actual	Normal	DDoS
Normal	23022	1984
DDoS	136	24859

TABLE VI. AVERAGE DETECTION PERFORMANCE OF THE PROPOSED BACKPROPAGATION NN-BASED DDOS INTRUSION DETECTION MODEL

Performance Metric	**Training Phase**	**Testing Phase**
Accuracy	0.9595	0.9576
Detection Rate (Recall)	0.9946	0.9945
Specificity	0.9242	0.9207
Precision	0.9293	0.9261
FPR	0.0758	0.0793
F1-Score	0.9609	0.9591
MCC Coefficient	0.9212	0.9177
Kappa Coefficient	0.9189	0.9152

$$True\ Positive\ Rate = \frac{TP}{TP+FN} \quad (12)$$

$$True\ Negative\ Rate = \frac{TN}{TN+FP} \quad (13)$$

$$False\ Positive\ Rate = \frac{FP}{FP+TN} \quad (14)$$

$$False\ Negative\ Rate = \frac{FN}{FN+TP} \quad (15)$$

As can be noted from Table VII, detection performances achieved by our proposed BP-based model were higher compared to those achieved by the different ML-based listed in Table VII. Although Liu *et al.* [14] had considered only the results of the neural network and disregarded the other approaches due to the unacceptable results, our proposed model still has higher performance. On the other hand, In contrary to the simple preprocessing of neural network inputs required by our proposed model, Liu *et al.* [14] used the Fast Fourier Transform (FFT) coefficients and the information entropy as input features to the input layer of the neural network which entails extra computational overhead and preprocessing of the network traffic before the detection takes place.

Furthermore, we compare our proposed DDoS intrusion detection model with various related Machine Learning approaches as shown in Table VIII.

TABLE VII. THE AVERAGE PERFORMANCE OF OUR PROPOSED BACKPROPAGATION ANN MODEL IN TERMS OF FPR, FNR, TPR, TNR METRICS

	Year	**FPR**	**FNR**	**TPR**	**TNR**
Random Forest [14]	2021	0.844	0.0001	0.999	0.156
Gaussian Naive Bayes [14]	2021	1.0	0.0	1.0	0.0
Neural Network [14]	2021	0.0222	0.0069	0.9930	0.9777
Proposed BP ANN-Model	-	0.0793	0.0054	0.9946	0.9207

TABLE VIII. AVERAGE PERFORMANCE EVALUATION OF THE PROPOSED MODEL WITH OTHER CLASSICAL ML-BASED TECHNIQUES

Machine Learning-based Method	**Year**	**Acc**	**Recall**	**F1 Score**	**Precision**
ID3 [11]	2019	-	0.65	0.69	0.78
Random Forest(RF) [11]	2019	-	0.56	0.62	0.77
Naive Bayes (NB) [11]	2019	-	0.11	0.05	0.41
Multinomial Logistic Regression (LR) [11]	2019	-	0.02	0.04	0.25
Bandwidth Control Mechanism + Extreme Gradient Boosting Algorithm (XGBoost) [15]	2020	0.997	1.000	1.0000	1.0000
Logistic Regression [15],[16]	2020	0.8000	0.8000	0.8000	0.8500
Naive Bayes [15],[17]	2020	0.7700	0.7700	0.7600	0.8400
ID3 [15],[18]	2020	0.9850	0.9990	0.9900	0.9900
Random Forest [15],[19]	2020	0.9855	0.9900	0.9900	0.9900
Autoencoder [20]	2021	0.8945	-	-	-
Restricted Boltzmann [20]	2021	0.5651	-	-	-
K-means [20]	2021	0.7538	-	-	-
Expectation-Minimization (EM) [20]	2021	0.7096	-	-	-
Proposed Model	-	0.9576	0.9946	0.9591	0.9261

As shown in Table VIII The obtained results show that the detection performance of our proposed BP-based DDoS predictive model is far superior to that had been reported by the authors of the CICDDoS2019 dataset [11] using ID3, RF, NB, and LR algorithms owing to the nonlinear modeling power of the backpropagation neural network. The same findings can be noticed if the detection performance of our proposed BP-model is compared to Autoencoder, Restricted Boltzmann, K-means, and Expectation-minimization machine learning algorithms adopted by [20].

Even though ID3, RF, NB, and LR algorithms were re-generated by [15], they have achieved superior detection performance to that had been achieved by [11]. Authors in [15] attributed their higher results to the performed preprocessing steps they adapted. However, in comparing our results to that reported by [15] for LR and NB-based algorithms, we can conclude that the influence of the pre-processing is limited because our BP-model that processed by the simple classical pre-processing steps was able to achieve superior results in comparison with these algorithms.

A regular BP-based detection model and the detection performance of [15] built is based on sequential steps of preprocessing and a combination of the bandwidth control mechanism and Extreme gradient boosting algorithm.

C. Experiments on Different Network Parameters Tunning

Since backpropagation network is a gradient-descent based learning algorithm, the first steps that come into consideration while building the network architecture is the initial state of the network as well as the network parameters tunning of the network in order to converging to the optimal minima of loss function gradient in least number of epochs.

Therefore, in the following subsections, we investigate the influence of the required number of Epochs, learning rate γ and sigmoid slope α on the detection performance of the system in training and testing phases and then analyze for the most appropriate values of these parameters.

It is worthy to mention that the backpropagation network is highly sensitive to the initial state of their weight's metrices, we have turned the initial weights to random values in the range $-0.285 \leq w^{ini} \leq -1.06$ before we are conducting these experiments. Otherwise, improper weights initialization can drive the network to saturate at a static accuracy threshold and stuck in a static local minimum.

1) Impact of increasing number of epochs: To examine the effect of increasing epochs number of the training phase on the detection performance in the prediction phase of the proposed BP-based model in terms of accuracy, recall, FPR, Precision, Kappa and MCC coefficient. First, we set the span of epoch to be from 200 to 9000 and fix other parameters such as learning rate, initial weights, and sigmoid slope, then a comparative analysis between the impact of epochs number on the detection performance on both training and testing phases is conducted as illustrated in Fig. 6. and Fig. 7.

As illustrated in Fig. 6 and Fig. 7 the predictive behavior of the proposed model in training and testing phases is almost the same, which ensures that our model is not underfitting or overfitting. To emphasize this behavior further and to ensure the stability and robustness of the prediction performance, the difference of accuracy performance between these phases are zoomed in as illustrated in Fig. 8 where it can be noted that the differential behavior decreases steadily to less than 0.1e-3 and in unison manner as the number of the training epochs increases.

As can be noted from Fig. 6, the accuracy of detection increases steadily as the training epochs increases until it hits $Epoch = 2000$, where beyond this value, the rate of change in detection is saturated and cannot be traced without zooming in effect. At $Epoch = 4500$, the accuracy behavior shows a temporary tenuous decrease; however, it is not exhibited by the Recall and FPR performance behavior. It is clear that by epoch 4000, almost accuracy, recall, and FPR performances start to converge quickly to a steady-state behavior, therefore 4000 is a sufficient number of training epochs for sake of optimal performance. Although FPR, as shown in Fig. 6, reaches the minimum at $Epochs = 2000$, the accuracy and sensitivity of the predictive model do not show the same characteristic, and as a compromised solution, Epochs = 2000 was not adapted.

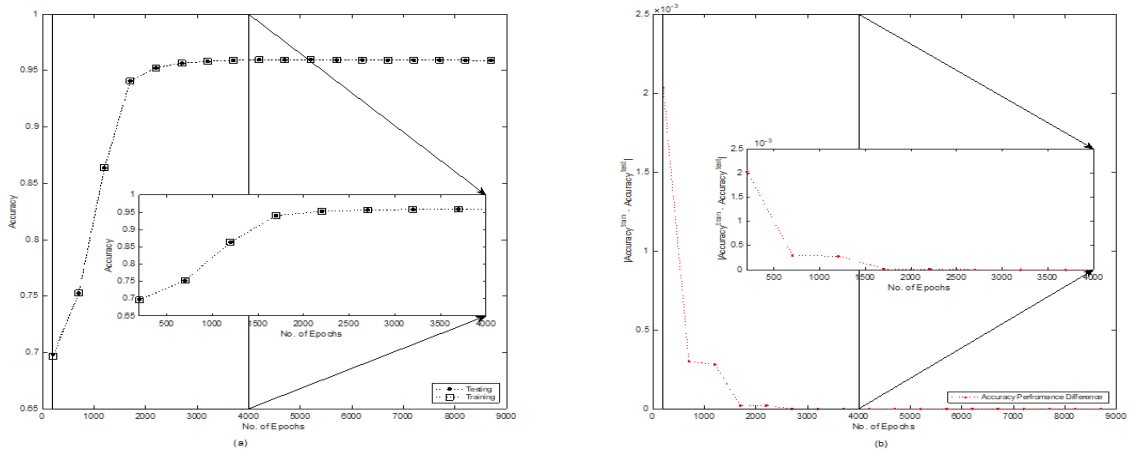


Fig. 6. (a) Accuracy Performance of the Proposed Model in Training and Testing Phases Versus Number of Training Epochs Zoomed in from Epoch No = 4000 to 8000. (b) Difference between the Accuracy Detection of Training and Testing Phases Zoomed in from Epoch No =4000 to 8000.

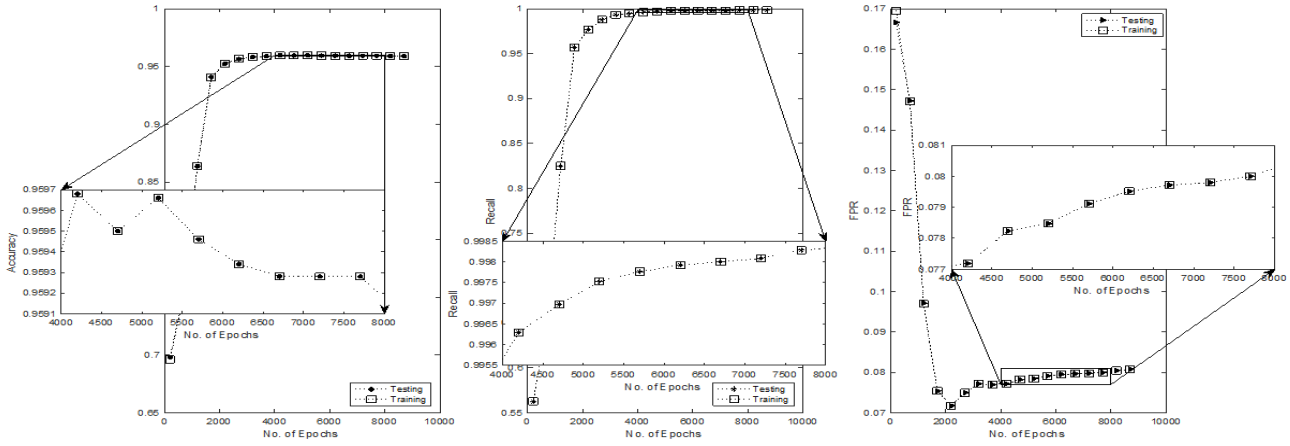


Fig. 7. Accuracy, Recall (Sensitivity), and FPR Performance of the Proposed Model in Training and Testing Phases Versus Number of Training Epochs Zoomed in from Epoch No = 4000 to 8000.

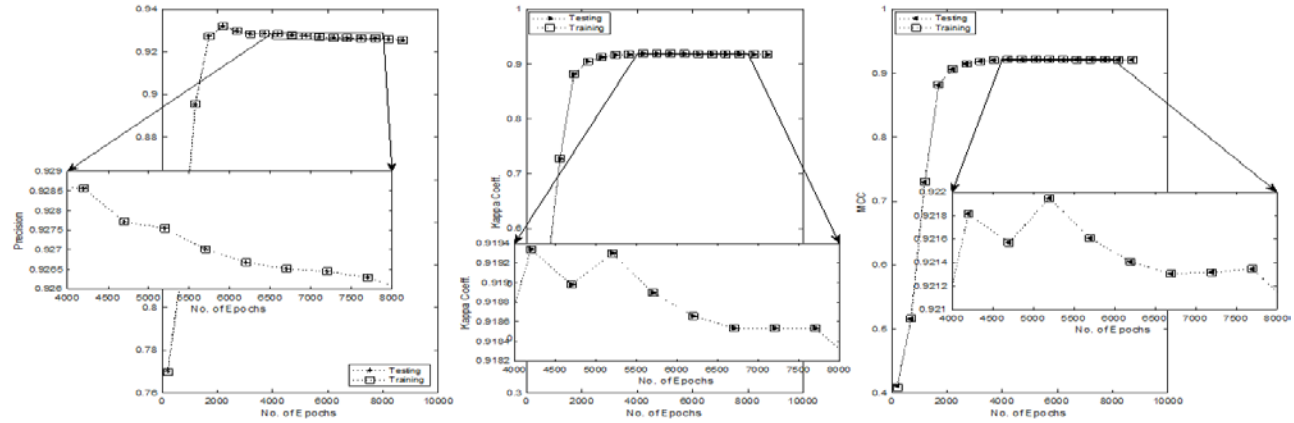


Fig. 8. Precision, Kappa and MCC Coefficients based Performance of the Proposed Model in Training and Testing Phases Versus Number of Training Epochs Zoomed in from Epoch No = 4000 to 8000.

Epochs number of 4000 may appear as a large number of training iterations that required for the neural network to reach a steady detection performance which insinuates a computational challenge appertains to the running time required to train the network. However, due to the simple and efficient network architecture, the entire training phase takes less than five minutes (299 seconds at rate of 13 seconds per epoch). On the other hand, the prediction stage of the system is more time-crucial in comparison to the training phase. Most of the computational overhead is front-loaded during the training phase, the prediction process for 50,000 traffic flows takes less than 0.068 second (in a rate of 1e-6 second per each traffic flow), which is considered as highly computational efficient.

2) The variation of sigmoid slope experiment: In this experiment, the slope of sigmoid function (activation function used in all network neurons) was changed in the range $0 \leq \alpha \leq 1$ and the detection accuracy, sensitivity and FPR metrics were recorded.

As shown in Fig. 9, the performance of the system shows a noticeable enhancement as α parameter is increased from 0.1 to 0.2, however, as the value of α transcends 0.2, the system shows a degradation in terms of accuracy and FPR. Furthermore, as the value of α transcends 0.5, the network fails to converge.

On the other hand, even though detection rate shows an enhancement as value of α transcends 0.2, it is unnoticeable and in comparison, to the FPR-based behavior, it cannot be adapted. Thus, based on the experiment we have adapted $\alpha = 0.2$.

3) The variation of learning rate experiment: In this experiment, the effect of learning rate on the system performance is investigated. As shown in Fig. 10, the detection performance is decremental as the learning rate exceeds $\gamma > -10$, whereas, system performance, for all metrics, shows almost a saturated behavior against increasing learning rate in the range $-30 \leq \gamma \leq -10$. Therefore, $\gamma = -10$, was adapted.

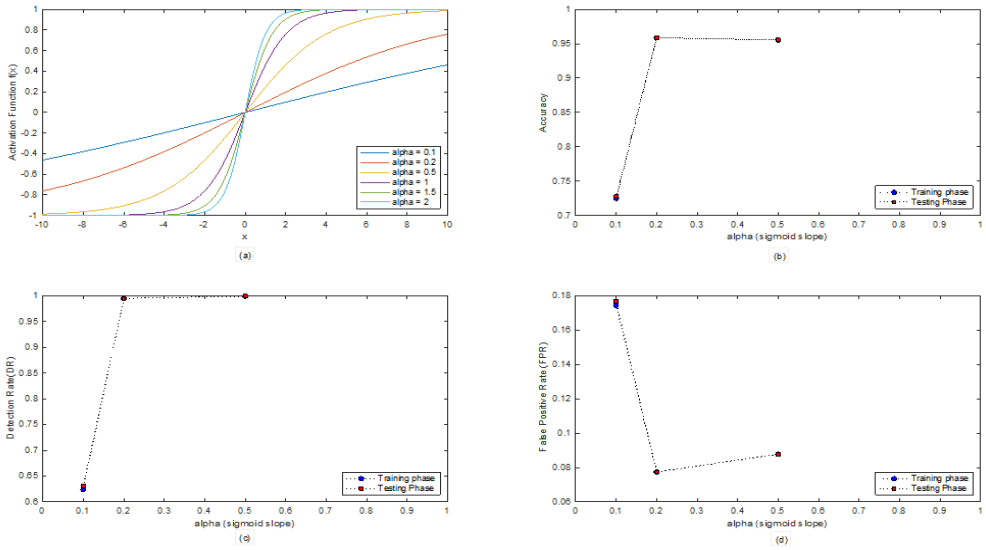


Fig. 9. System Performance (in Terms of Accuracy, Detection Rate and FPR) Under different Values of Sigmoid Slope α. (a) Sigmoid Activation Function Profiles for Different Values of Slope α. (b) Detection Accuracy versus Changing Sigmoid Slope α Parameter. (c) Detection Rate versus Changing Sigmoid Slope α Parameter. (d) FPR versus Changing Sigmoid Slope α Parameter.

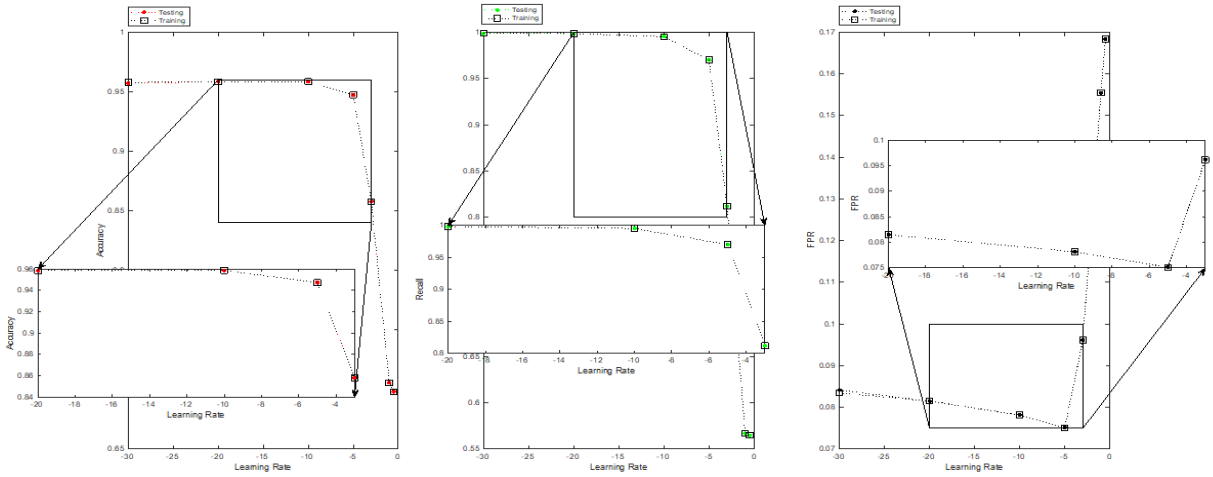


Fig. 10. Accuracy, Recall (Sensitivity), and FPR Performance of the Proposed Model in Training and Testing Phases Under Different Values of Learning Rate Parameter γ.

V. Conclusion

In this paper, we have proposed a backpropagation neural network-based methodology for DDoS attacks detection in IoT networks. CICDDoS2019 dataset has comprehensive categories of reflective DDoS attacks that have been considered, so our scheme uses this dataset for model training and evaluation. In contrast to many machines learning-based DDoS intrusion detection models that adapt numerous preprocessing steps and multiple stages and hybrid types of machine learning algorithms to attain high detection performance, our proposed model requires a simple preprocessing step used the standard backpropagation neural network only as a detection engine where we have achieved competitive detection performance. Results show that our model achieves a recall of 0.9946 and accuracy and FPR of 0.9576 and 0.0793 respectively. In our experimental results, we have conducted extensive comparisons with other up-to-date DDoS intrusion detection schemes, and we examined the effect of changing epoch parameter on the overall performance of the backpropagation neural network, however, for further detection performance amelioration, tunning other hyperparameters and examining their impact can be offered as a future work where different approaches such as Bayesian optimization and Random search can be utilized for this purpose.

References

[1] CISA, "Understanding Denial-of-Service Attacks," Cybersecurity and Infrastructure Security Agency. [Online]. Available: https://www.cisa.gov/uscert/ncas/tips/ST04-015.

[2] A. Dahiya and B. Gupta, "How IoT is Making DDoS Attacks More Dangerous?," Insights2Techinfo. [Online]. Available:

https://insights2techinfo.com/how-iot-is-making-ddos-attacks-more-dangerous/.

[3] M. darkaie and R. Tavoli, "Providing a method to reduce the false alarm rate in network intrusion detection systems using the multilayer perceptron technique and backpropagation algorithm," in 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), Tehran, Iran, Feb. 2019, pp. 001–006. doi: 10.1109/KBEI.2019.8735024.

[4] E. Hodo et al., "Threat analysis of IoT networks using artificial neural network intrusion detection system," in 2016 International Symposium on Networks, Computers and Communications (ISNCC), Yasmine Hammamet, Tunisia, May 2016, pp. 1–6. doi: 10.1109/ISNCC.2016.7746067.

[5] X. Wu and Y. Chen, "Validation of Chaos Hypothesis in NADA and Improved DDoS Detection Algorithm," IEEE Commun. Lett., vol. 17, no. 12, pp. 2396–2399, Dec. 2013, doi: 10.1109/LCOMM.2013.102913.130932.

[6] S. Alzahrani and L. Hong, "Detection of Distributed Denial of Service (DDoS) Attacks Using Artificial Intelligence on Cloud," in 2018 IEEE World Congress on Services (SERVICES), San Francisco, CA, Jul. 2018, pp. 35–36. doi: 10.1109/SERVICES.2018.00031.

[7] Ming-Qing Ling and Wei-Wei Liu, "Research on intrusion detection systems based on Levenberg-Marquardt algorithm," in 2008 International Conference on Machine Learning and Cybernetics, Kunming, China, Jul. 2008, pp. 3684–3688. doi: 10.1109/ICMLC.2008.4621045.

[8] Dongqi Wang, Zhu yufu, and Jia Jie, "A multi-core based DDoS detection method," in 2010 3rd International Conference on Computer Science and Information Technology, Chengdu, China, Jul. 2010, pp. 115–118. doi: 10.1109/ICCSIT.2010.5564969.

[9] O. P. Badve, B. B. Gupta, S. Yamaguchi, and Z. Gou, "DDoS detection and filtering technique in cloud environment using GARCH model," in 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), Osaka, Japan, Oct. 2015, pp. 584–586. doi: 10.1109/GCCE.2015.7398603.

[10] B. Joshi, A. S. Vijayan, and B. K. Joshi, "Securing cloud computing environment against DDoS attacks," in 2012 International Conference on Computer Communication and Informatics, Coimbatore, India, Jan. 2012, pp. 1–5. doi: 10.1109/ICCCI.2012.6158817.

[11] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," 2019, pp. 1–8.

[12] Data Mining. Elsevier, 2012. doi: 10.1016/C2009-0-61819-5.

[13] A Machine-Learning Approach to Phishing Detection and Defense. Elsevier, 2015. doi: 10.1016/C2014-0-03762-8.

[14] Z. Liu, C. Hu, and C. Shan, "Riemannian manifold on stream data: Fourier transform and entropy-based DDoS attacks detection method," Computers & Security, vol. 109, p. 102392, Oct. 2021, doi: 10.1016/j.cose.2021.102392.

[15] H. A. Alamri and V. Thayananthan, "Bandwidth Control Mechanism and Extreme Gradient Boosting Algorithm for Protecting Software-Defined Networks Against DDoS Attacks," IEEE Access, vol. 8, pp. 194269–194288, 2020, doi: 10.1109/ACCESS.2020.3033942.

[16] S. Menard, Logistic regression: From introductory to advanced concepts and applications. Sage, 2010.

[17] H. Zhang, "Exploring conditions for the optimality of naive Bayes," International Journal of Pattern Recognition and Artificial Intelligence, vol. 19, no. 02, pp. 183–198, 2005.

[18] H. H. Patel and P. Prajapati, "Study and analysis of decision tree based classification algorithms," International Journal of Computer Sciences and Engineering, vol. 6, no. 10, pp. 74–78, 2018.

[19] M. Denil, D. Matheson, and N. De Freitas, "Narrowing the gap: Random forests in theory and in practice," 2014, pp. 665–673.

[20] V. Odumuyiwa and R. Alabi, "DDOS Detection on Internet of Things Using Unsupervised Algorithms," Journal of Cyber Security and Mobility, pp. 569–592, 2021.