http://ijea.jctjournals.com/ ISSN(ONLINE):2320 – 0804

Volume-11, Issue-05, May, 2022

International Journal of Engineering Associates (IJEA)

Page Number: 01-06

A DDoS Attack Detection Using Machine Leaning Approach

[1]Charulika Yadav *(M.Tech Scholar),*[2] Prof. Sumit Sharma *(Head of department)*

[1,2]Department of Computer Science and Engineering

[1,2] Vaishnavi Institute of Technology and Science, Bhopal(M.P), INDIA

[1]charulikay@gmail.com

ABSTRACT – Distributed Denial of Service(D-DoS) attacks exhaust a specified system's computational and communication capabilities, prohibiting it from providing regular services to authorized users. Within the scope of this research, an improved feature selected-based network was provided for the purpose of effective DDoS intrusion identification. MATLAB 2020 software will be used for the actualization of the situation to improve. This study is based on the MATLAB programming language, which is widely used in academic and industry study designs. The R2020 MATLAB environment has been used to create and simulate the suggested technique. Different kinds of distributed denial of service attacks might well be found on the internet. The Canadian Institute of Cyber security is the subject of this study endeavor (CICIDS2017). The 80 parameters that were used for classification in this collection of data are as follows: According to the tools that were used, the flow records are labeled as 'Slowloris,' 'Slowhttptest,' 'Hulk,' and 'Begian,' with the benign traffic being the only exception. In terms of accuracy, precision, selectivity, sensitivity, and specificity, as well as the confusion matrix, the suggested technique produces positive results (C.M.). The accuracy achieved by the approach described here is 99.912 percent.

Keywords—Distributed Denial of Service(D-DoS), Features Selection, Cyber Security, Artificial Neural Networks, Support Vector Machine (SVM), and Machine Learning.

I. INTRODUCTION

Distributed DoS (DDoS) attack detection is the primary goal of the first approach. There are many ways to detect anomalies, but the most common is to look for behavior or information that doesn't fit into the developed model. In many applications, abnormalities, abnormal values, the dissonance of observation, exceptions, aberrations, surprise, peculiarities, or contaminants are commonly referred to as abnormalities, anomalies, or anomalies [4]. A well-planned training program with plenty of data will tell the DDoS detection which actions are reasonable and which are malicious in this strategy instead of pre-installing traditional activities into the DDoS detection. Any anomaly detection method must take into account the quality of the data being input. It's common for general input data to be a collection of the information instance (also referred to as a record, point, vector, pattern, case, sample, observation, entity). On the one hand, this methodology allows for the exploration of new species of attacks; on the other hand, it can lead to inaccurate judgments, such as raising an alarm when the network is operating normally or ignoring an attack because it considers the attack to be traditional activity.

An IDS Associate can be built using signature-based detection, which relies on a cognitive component. As an outcome of the signature-based detection methodology's effectiveness in spotting known threats through the signatures of determined events, it is extremely helpful [6]. However, this method can accurately report and defend against known attacks, but the drawback is that it has a limited effect on new attack designs, as well as the cognitive material must be modified frequently to ensure that the IDS has smart performance. To create the IDS, the main method is selected in this thesis. The main problem with using a cloud platform is that the IDS may overload some of the cloud's already-busy nodes, reducing its ability to detect suspicious activity. When it comes to IDS deployment, it's important that the IDS doesn't consume too many resources, but it also needs to identify attacks quickly. The idea of giving the transmitted IDS the ability to change its layout based on data about how resources are being used across the cloud is intriguing. Furthermore, an IDS system must be able to detect and respond to unknown (new) cloud-based attacks. Although it's going to be a lot more appropriate, anomaly detection is going to require a lot of resources [7, 8]. It is, therefore, necessary to strike the right balance between satisfying cloud customers and delivering low-cost intrusion detection at the same time.

In the above section I discuss the introduction part of proposed research work, discuss the cyber-attack and D-DoS attacks. In the section II discuss DoS attack related work. Next section III discusses the proposed methodology these were presented by different researchers. Finally, describe the DDoS assault detection mechanism that was presented. Section IV discusses the result and discussion. Last but not least discuss the conclusion in section VI.

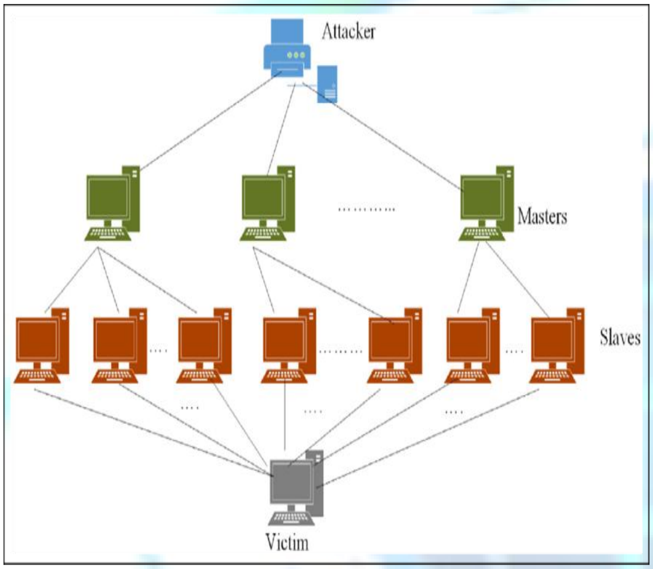


Fig. 1 Structure of DDOS Attacks

II. RELATED WORK

The literature review on DDoS attacks is discussed in this section. Throughout the previous decade, various researchers presented various works on DDoS attacks.

Snehi_et.al,(2021), In this research work authors presented analysed the most devastating DDoS and IoT-DDoS attacks, as well as the elements of today's Cyber- Physical Device, architectural features, as well as security problems. A layer between perception as well as the cloud, Fog Computing has been suggested as a method of improving performance and performing the assigned duties on behalf of Cloud. DDoS/IoT-DDoS detection and reduction have been analysed. Next but not least, uncertainty, as well as gap evaluation, was carried, as well as the narrow down method was used to identify general gaps or vulnerabilities in the possible solutions. Using that research study, they've attempted to summarise the vulnerability analysis that can serve as a foundation for future DDoS/IoTDDoS defense solutions for future technologists and researchers. DDoS attacks are just one of several cybersecurity risks that vulnerability research examines. Provided a wide range of options [01]. Jia_et.al,(2020), In this research work investigator Virupakshar_et.al, (2020), In this research work.The researchers reviewed 70 publications from high-profile journals. Some 47% of researchers utilised data theory approaches, 42percent utilised machine learning strategies, and 20% used Artificial Neural Networks to detect DDoS attacks in SDN. Additionally, they've provided details on various security mechanisms so that other researchers can better understand the current situation. For SDN-enabled systems, operators remain a primary target for attackers [02]. Dong_et.al,(2019), In this research work DDoS attacks are becoming more prevalent in Analysts who say SDN and cloud computing environments pose the greatest risk. DDoS attacks and their detection in SDN and cloud computing are discussed at this moment by researchers. Because SDN could be a target of a DDoS attack, they look into how DDoS attacks are introduced on SDN as well as possible solutions. For DDoS attacks, researchers also discuss how to create exploratory conditions and then use simulation tools in SDN and cloud computing environments. Many unsolved issues in this area are examined, such as how to mitigate DDoS attacks in an SDN and cloud computing setting. Despite the never-ending research in this area, some issues remain unresolved. Future research should focus on this issue. They suggest the following lines of investigation for future work: Attacks against the SDN Even though SDN's unified regulation is its most attractive feature, it can also be a single point of failure when subjected to a Distributed Denial of Service (DDoS) attack. Traditional anomaly detection methods are having a hard time dealing with the growing number of DDoS attacks. As a result, SDN research is focusing on big data analysis and location advancements for DDoS attacks. SDN and NFV are mutually beneficial, but they are not dependent on each other. It is SDN's role to empower NFV. Because the logic for virtualization technology runs on a controller rather than on physical circuits, SDN helps to automate the network by allowing users to make approach-based decisions about how to coordinate system traffic flows [03]. Li_ et.al, (2018, September), This research work, presented, DDoS attack detection system based on PCA feature reduction as well as RNN prediction. Utilizing the KDD dataset, researchers compared our PCA-RNN detection method with several other methods for detecting DDoS attacks. Our PCA-RNN technique's experimental results show that it has improved detection accuracy, performance, as well as applicability[09]. Dayal, et. al (2017, January), In this research work, Researchers presented an attack model to identify and classify various DDoS attack scenarios in SDN. The hyenae attack tool was used to carry out a variety of DDoS attacks in an SDN environment utilising a few of the most popular conventional DDoS attack methods. Despite the fact that volumetric attacks have a significant effect on the research plane, they do not have a significant impact on the controller. During the attack phase, the effect is clearly visible. Protocol exploitation threats, on the other hand, have little impact on network traffic. They focus on consuming other device resources, such as the TCAM, logical port, etc. Immediately after the attack, and immediately after the attack, controllers might be seriously impacted.. TCP SYN flood as well as HTTP flood attacks can in reality bring down the control system[10]. Buragohain, Chaitanya, et.al. (IEEE 2016), In this research work, presented and tested the proposed architecture of Flow Trapp. An SDN-based structure is provided for the detection and mitigation of both high- and low-rate DDoS attacks in data centers by this system. Incoming attack traffic can be classified using the optimization technique, which compares it to an application-specific tuple of genuine flow traffic. Remediation is used when a malicious user is discovered to transfer attack traffic regularly, rather than blocking the location at the outset. Architecture is implemented using SDN innovations such as OpenFlow and flow statistics collectors such as Flow To enhance the effectiveness of FlowTrApp, the OpenFlow controller, as well as the sFlow-RT application, can share the burden of detecting and mitigating DDoS attacks. Our method outperforms an existing QoS-based method in terms of performance[12].Zhao, T., Lo, et.al (2015, August).In this research work, Hadoop, as well as HBase, were used to design an effective DDoS detection method that can identify threats quickly. The first step was to create a Hadoop as well as HBase cluster to process a massive unorganized set of data. Once the neural network model for DDoS detection was created as well as six training data were used to train the neural network model, it was able to identify DDoS attacks. Three parts are used to demonstrate how well-suited the skilled neural network is for the task[16].

III. PROPOSED METHODOLOGY

This section discusses the proposed solution for the detection and identification of DDOS attacks on clouds.

A. Proposed Work

In this section discuss the proposed method. The key objective of a Distributed Denial of Service (D-DoS) attack is to compile multiple systems across. The Internet with agents and form botnets of networks. In the system model, the recognition module is compared to other strategies that use RBF networks with PSO-optimized training. For the detection of D-DoS attack in this proposed research work implemented artificial neural network-based a modified cascaded feed forward neural with improved regression approach.

B, Bayesian regularization back propagation neural network (BR-BPNN)

In this section, a back propagation neural network (BPNN) along with the Bayesian regularization learning algorithm is described. The background theory on BPNN along with theBayesian regularization is given in Appendix A. A more detailed discussion can be foundin. BR-BPNN is utilized to achieve better generalization and minimal over-fitting for the trained networks [27, 28].

Consider a neural network with training data set D having nt input and target vector pairs in the network model, i.e

$$D = \{(u_{1,}t_{o1}), (u_{2,}t_{o2}), \dots, (u_{nt,}t_{ont})\} \qquad (1)$$

For each input (u) to the network, the difference between target output (t_o) and predicted output (α_o) is computed as error e.

$$F(\overline{w}) = E_D = \sum_{i=1}^{nt}(ei)2 = \sum_{i=1}^{nt}(t_{oi} = a_{oi})T(t_{oi} - a_{oi}), \qquad (2)$$

where w denotes the vector of size K containing all the weights and biases of the network.

In order to generalize the neural network, the performance index of Eq. (6) is modified using a regularization method. A penalty term is added to the performance index F(w).

$$F(\overline{w}) = \mu\overline{w}^T\overline{w} + vE_D = \mu E_w + vE_{D,} \qquad (3)$$

where μ and v are the regularization parameters and E_w represents the sum of the squared network weights (SSW). Considering the network weights $\overline{w}$ as random variables, the aim is to choose the weights that maximize the posterior probability distribution of the weights $P(\overline{w}|D, \mu, v, M_N)$ given a certain data D. According to Bayes' rule [27], the posterior distribution of the weights depends on the likelihood function $P(\overline{w}|D, \mu, v, M_N)$ the prior density $P(\overline{w}|\mu, M_N)$, and the normalization factor $P(\overline{w}|D, \mu, V, M_N)$ for a particular neural network model M_N and can be evaluated from

$$P(\overline{w}|D, \mu, v, = M_N \frac{P(D|w, v, M_N)\, P(w|\mu M_N)}{P\ (D\backslash\mu, V, M_N)}) \qquad (4)$$

Considering that the noise in the training set has a Gaussian distribution, the likelihood function is given by

$$P\ (D|\overline{w}, \mu, v, M_N\ ) = \frac{\exp\ (-vE_D)}{Z_D(v)} \qquad (5)$$

Where ZD = (π/v) Q/2 and Q = n_t × N^{n1}.

Similarly, assuming a Gaussian distribution for the network weights, the prior probability density $P\ (\overline{w}|\mu, M_N)$ is given as:

$$P(\overline{w}|\mu, M_N) = \frac{\exp(-\mu E_w)}{Z_w(\mu)} \qquad (6)$$

Where Z_w = (π/α) K/2

The posterior probability with the network weights _x0016_w can then be expressed as :

$$P(\overline{w}|D, \mu, v, M_N) = \frac{\exp(-\mu E_w - vE_D)}{Z_F(\mu,v)} = \frac{\exp(-F(w))}{Z_F(\mu,v)} \qquad (7)$$

Where $Z_F(\mu, v) = Z_D(v)Z_w(\mu)$ is the normalization factor.

$$P(\mu, v|D, M_N) = \frac{P(D|\mu,v,M_N)\, P(\mu,v|M_N)}{P(D|M_N)} \qquad (8)$$

where $P(\mu, v|M_N)$ denotes the assumed uniform prior density for the parameters μ and v. From Eq. (12), it is evident that maximizing the likelihood function $P(D|\mu, v, M_N)$ eventually maximizes the posterior probability $P(\mu, v|D, M_N)$.

$$\mu^* = \frac{\gamma}{2E_w(\overline{w}^*)} \text{and } v^* = \frac{Q-\gamma}{2E_D\overline{w}^*)} \qquad (9)$$

where γ signifies the “number" of effective parameters exhausted in minimizing the error function

$$\gamma = K - \mu^* tr(H^*)^{-1} \text{ or } 0 \leq \gamma \leq K \qquad (10)$$

$$H^* \approx J^T J \qquad (11)$$

where J is the Jacobian matrix formed by the first derivatives of the network errors e with respect to network weights wij . In (14), tr(.) denotes the trace operator. The normalization factor $z_F(\mu, v)$can then be approximated as

$$Z_F(\mu, v) \approx (2\pi)^{K/2} (\det(H^*))^{-1/2} \exp(-F(\overline{w}^*)) \qquad (12)$$

$$\overline{w}^{k+1} = \overline{w}^k - [J^T J + \lambda I]^{-1} J^T e, \qquad (13)$$

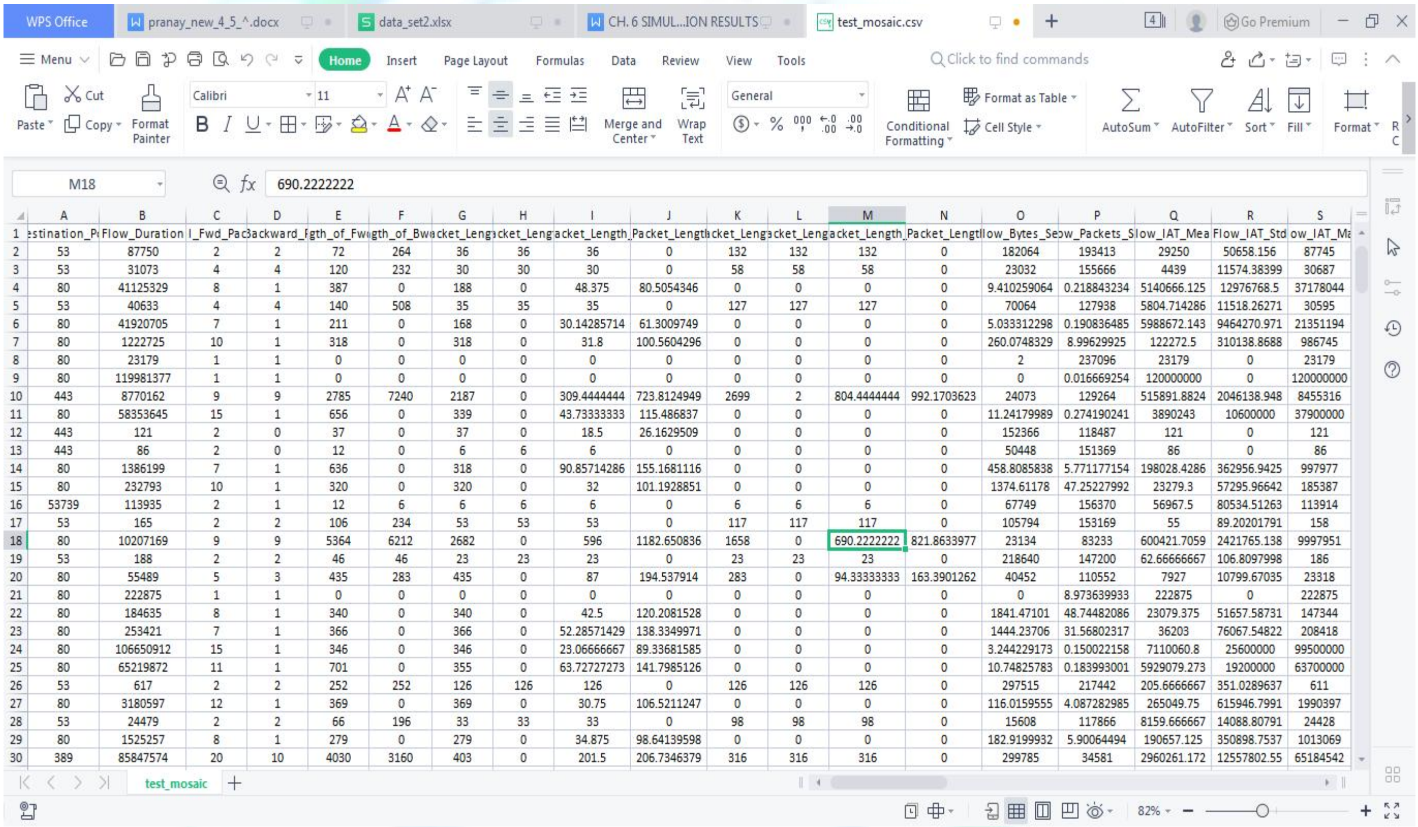
where λ denotes the Levenberg's damping factor and $J^T e$ is the error gradient, which need to be close to zero at end of the training.

IV. RESULT AND DISCUSSION

In this section describing out the implementation detail and designing issues for our proposed research work.

A. Data set

The Canadian Institute of Cyber security (CICIDS2017) Intrusion Detection Evaluation Dataset is utilised for design training and evaluation [18]. Numerous threats, such as DDoS as well as botnet activity are documented in the report. We used the DoS data set as the basis for our classification model in this study. There are 84 variables in each flow record in the CICIDS 2017 dataset, which is in comma-separated (CSV) format.



690.2222222

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	estination_P	Flow_Duration	l_Fwd_Pac	3ackward_	gth_of_Fw	gth_of_Bw	acket_Leng	cket_Leng	acket_Length_	Packet_Length	cket_Leng	cket_Leng	acket_Length_	Packet_Length	low_Bytes_Se	ow_Packets_S	low_IAT_Mea	Flow_IAT_Std	ow_IAT_Ma
2	53	87750	2	2	72	264	36	36	36	0	132	132	132	0	182064	193413	29250	50658.156	87745
3	53	31073	4	4	120	232	30	30	30	0	58	58	58	0	23032	155666	4439	11574.38399	30687
4	80	41125329	8	1	387	0	188	0	48.375	80.5054346	0	0	0	0	9.410259064	0.218843234	5140666.125	12976768.5	37178044
5	53	40633	4	4	140	508	35	35	35	0	127	127	127	0	70064	127938	5804.714286	11518.26271	30595
6	80	41920705	7	1	211	0	168	0	30.14285714	61.3009749	0	0	0	0	5.033312298	0.190836485	5988672.143	9464270.971	21351194
7	80	1222725	10	1	318	0	318	0	31.8	100.5604296	0	0	0	0	260.0748329	8.99629925	122272.5	310138.8688	986745
8	80	23179	1	1	0	0	0	0	0	0	0	0	0	0	2	237096	23179	0	23179
9	80	119981377	1	1	0	0	0	0	0	0	0	0	0	0	0	0.016669254	120000000	0	120000000
10	443	8770162	9	9	2785	7240	2187	0	309.4444444	723.8124949	2699	2	804.4444444	992.1703623	24073	129264	515891.8824	2046138.948	8455316
11	80	58353645	15	1	656	0	339	0	43.73333333	115.486837	0	0	0	0	11.24179989	0.274190241	3890243	10600000	37900000
12	443	121	2	0	37	0	37	0	18.5	26.1629509	0	0	0	0	152366	118487	121	0	121
13	443	86	2	0	12	0	6	6	6	0	0	0	0	0	50448	151369	86	0	86
14	80	1386199	7	1	636	0	318	0	90.85714286	155.1681116	0	0	0	0	458.8085838	5.771177154	198028.4286	362956.9425	997977
15	80	232793	10	1	320	0	320	0	32	101.1928851	0	0	0	0	1374.61178	47.25227992	23279.3	57295.96642	185387
16	53739	113935	2	1	12	6	6	6	6	0	6	6	6	0	67749	156370	56967.5	80534.51263	113914
17	53	165	2	2	106	234	53	53	53	0	117	117	117	0	105794	153169	55	89.20201791	158
18	80	10207169	9	9	5364	6212	2682	0	596	1182.650836	1658	0	690.2222222	821.8633977	23134	83233	600421.7059	2421765.138	9997951
19	53	188	2	2	46	46	23	23	23	0	23	23	23	0	218640	147200	62.66666667	106.8097998	186
20	80	55489	5	3	435	283	435	0	87	194.537914	283	0	94.33333333	163.3901262	40452	110552	7927	10799.67035	23318
21	80	222875	1	1	0	0	0	0	0	0	0	0	0	0	0	8.973639933	222875	0	222875
22	80	184635	8	1	340	0	340	0	42.5	120.2081528	0	0	0	0	1841.47101	48.74482086	23079.375	51657.58731	147344
23	80	253421	7	1	366	0	366	0	52.28571429	138.3349971	0	0	0	0	1444.23706	31.56802317	36203	76067.54822	208418
24	80	106650912	15	1	346	0	346	0	23.06666667	89.33681585	0	0	0	0	3.244229173	0.150022158	7110060.8	25600000	99500000
25	80	65219872	11	1	701	0	355	0	63.72727273	141.7985126	0	0	0	0	10.74825783	0.183993001	5929079.273	19200000	63700000
26	53	617	2	2	252	252	126	126	126	0	126	126	126	0	297515	217442	205.6666667	351.0289637	611
27	80	3180597	12	1	369	0	369	0	30.75	106.5211247	0	0	0	0	116.0159555	4.087282985	265049.75	615946.7991	1990397
28	53	24479	2	2	66	196	33	33	33	0	98	98	98	0	15608	117866	8159.666667	14088.80791	24428
29	80	1525257	8	1	279	0	279	0	34.875	98.64139598	0	0	0	0	182.9199932	5.90064494	190657.125	350898.7537	1013069
30	389	85847574	20	10	4030	3160	403	0	201.5	206.7346379	316	316	316	0	299785	34581	2960261.172	12557802.55	65184542

Fig. 2 Data Set In Excel

B. Result Parameters

The strategy described here examines a variety of outcome characteristics. Here are the variables you'll want to keep an eye on.

True Positive (T.P.)

A true positive is an event in which the model accurately predicts the positive class. When an experiment sees a positive, and the prediction was correct, it is a true positive.

False Negative (F.N.)

A test result that incorrectly suggests that a condition does not hold is known as a false negative error. When a test result wrongly suggests the absence of a disorder, a negative test occurs.

False Positive (F.P.)

A false positive occurs when the algorithm forecasts the positive class inaccurately. Mistakes in binary classification results in wrongly diagnosing a disorder as a false positive.

True Negative (T.N.)

A real negative is a result in which the model correctly predicts the negative class of outcomes.

Accuracy(ACC)

In the plant decease detection task, a detected as a decease is a true positive (TP) whereas a real negative (TN) is a non-effected leaf of plant detected. When it comes to false negatives (FN), the afflicted leaves are the culprit.

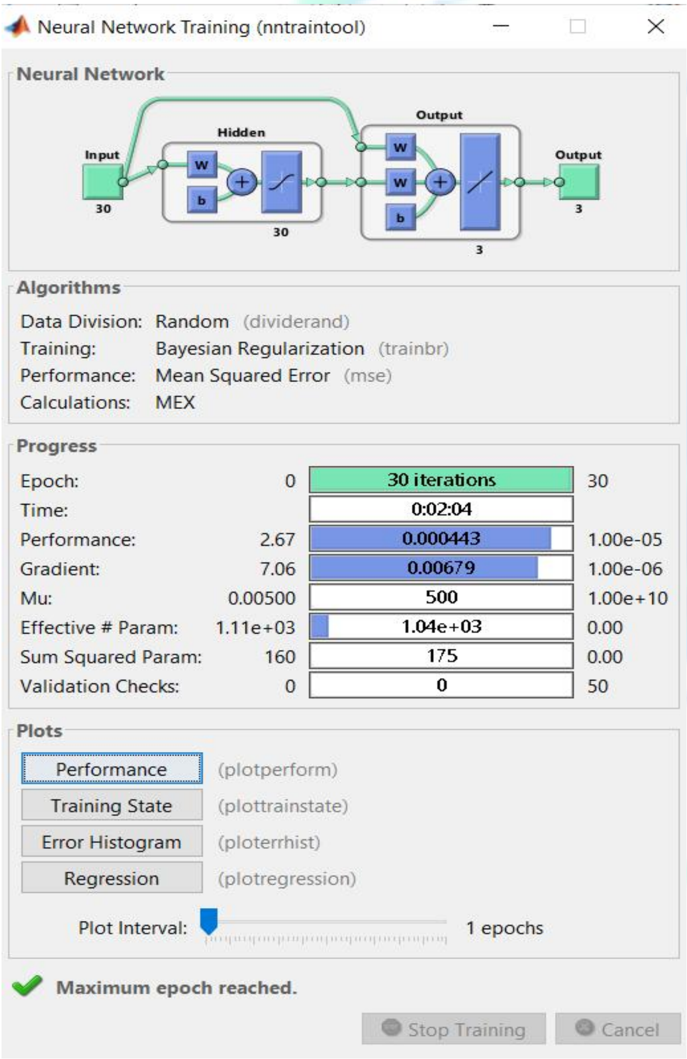
$$\text{Accuracy} = (TP + TN) / S \qquad (14)$$

The accuracy is the ratio of addition of number of correct production (TP+TN) and total number of production (TP + TN + FP + FN) .

$$\text{Acc} = \frac{TP+TN}{TP+TN+FP+FN} \qquad (15)$$

C. Simulation Outcomes

There's a neural network (NN) experiment depicted in the fig.(4.4). A total of thirty input features are used in this algorithm. Methods outlined in the suggested methodology chapter are used to attain these properties.



Neural Network Training (nntraintool)

Neural Network

Algorithms

Data Division: Random (dividerand)
Training: Bayesian Regularization (trainbr)
Performance: Mean Squared Error (mse)
Calculations: MEX

Progress

Epoch:	0	30 iterations	30
Time:		0:02:04	
Performance:	2.67	0.000443	1.00e-05
Gradient:	7.06	0.00679	1.00e-06
Mu:	0.00500	500	1.00e+10
Effective # Param:	1.11e+03	1.04e+03	0.00
Sum Squared Param:	160	175	0.00
Validation Checks:	0	0	50

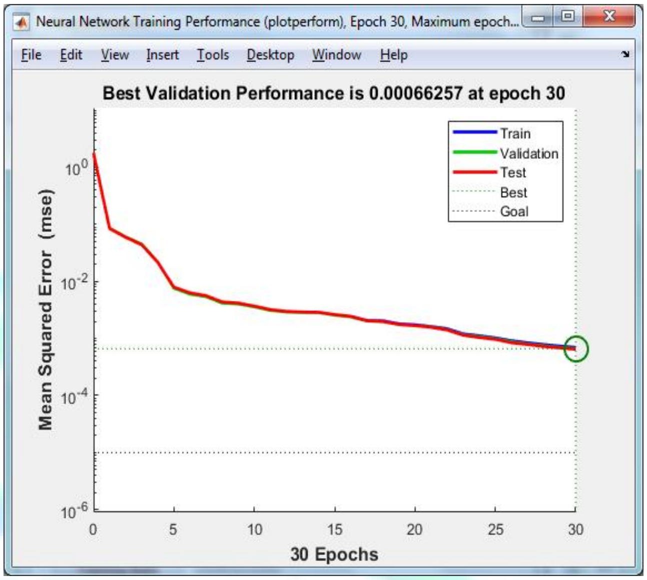
Plots

(plotperform)
(plottrainstate)
(ploterrhist)
(plotregression)

Plot Interval: 1 epochs

✓ Maximum epoch reached.

Fig. 3 Shows the cascaded Feed forward NN Network Outcome



Neural Network Training Performance (plotperform), Epoch 30, Maximum epoch...

Figure 4. Output of training validation performance

The mean square error of 0.00006257, which is extremely low, indicates that the suggested model's training method does not meet the requirements.

Table I. Experimental Results 1

acc_hybrid	99.9585
acc	99.952
TP	159289 132280 55132
FN	3 113 48
FP	38 49 77
TN	187535 214423 291608

V. CONCLUTION

In this research paper presented, The most of this work is analysis the various attacks of cloud computing, additionally discuss the various attacks on clouds and issues with cloud computing. In the last few year cloud computing is increases speedily and its application on different sectors. Each and every thing having two faces, one is positive and second is negative, cloud computing security threats are increases day to day.Network security relies heavily on intrusion detection technologies. The most difficult part of DDoS defense is weeding out the extraneous as well as unnecessary properties. As a result of a denial of service (DDoS) attack, a targeted system is unable to provide regular services to its legitimate customers. This proposed work presented a modified feature selected-based neural network for efficient DDoS attack detection.

REFERENCES

[1] Snehi, Manish, and Abhinav Bhandari. "Vulnerability retrospection of security solutions for software-defined Cyber–Physical System against DDoS and IoT-DDoS attacks." Computer Science Review 40 (2021): 100371.

[2] Jia, Yizhen, et al. "Flowguard: An intelligent edge defense mechanism against IoT DDoS attacks." IEEE Internet of Things Journal 7.10 (2020): 9552-9562.

[3] Dong, Shi, Khushnood Abbas, and Raj Jain. "A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments." IEEE Access 7 (2019): 80813-80828.

[4] Agrawal, Neha, and Shashikala Tapaswi. "Defense mechanisms against DDoS attacks in a cloud computing environment: State-of-the-art and research challenges." IEEE Communications Surveys & Tutorials 21.4 (2019): 3769-3795.

[5] Wang, An, et al. "Delving into internet DDoS attacks by botnets: characterization and analysis." IEEE/ACM Transactions on Networking 26.6 (2018): 2843-2855.

[6] Yang, Kun, Junjie Zhang, Yang Xu, and Jonathan Chao. "Ddos attacks detection with autoencoder." In NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium, pp. 1-9. IEEE, 2020.

[7] Singh, Jagdeep, and Sunny Behal. "Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions." Computer Science Review 37 (2020): 100279.

[8] Wani, Abdul Raoof, Q. P. Rana, U. Saxena, and Nitin Pandey. "Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques." In 2019 Amity International conference on artificial intelligence (AICAI), pp. 870-875. IEEE, 2019.

[9] Li, Qian, Linhai Meng, Yuan Zhang, and Jinyao Yan. "DDoS attacks detection using machine learning algorithms." In International Forum on Digital TV and Wireless Multimedia Communications, pp. 205-216. Springer, Singapore, 2018.

[10] Dayal, Neelam, and Shashank Srivastava. "Analyzing behavior of DDoS attacks to identify DDoS detection features in SDN." In 2017 9th International Conference on Communication Systems and Networks (COMSNETS), pp. 274-281. IEEE, 2017.

[11] Hsieh, Chang-Jung, and Ting-Yuan Chan. "Detection DDoS attacks based on neural-network using Apache Spark." In 2016 international conference on applied system innovation (ICASI), pp. 1-4. IEEE, 2016.

[12] Buragohain, Chaitanya, and Nabajyoti Medhi. "FlowTrApp: An SDN based architecture for DDoS attack detection and mitigation in data centers." In 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), pp. 519-524. IEEE, 2016.

[13] Xiao, Peng, Zhiyang Li, Heng Qi, Wenyu Qu, and Haisheng Yu. "An efficient ddos detection with bloom filter in sdn." In 2016 IEEE Trustcom/BigDataSE/ISPA, pp. 1-6. IEEE, 2016.

[14] Yadav, Satyajit, and Selvakumar Subramanian. "Detection of Application Layer DDoS attack by feature learning using Stacked AutoEncoder." In 2016 international conference on computational techniques in information and communication technologies (icctict), pp. 361-366. IEEE, 2016.

[15] Wang, Rui, Zhiping Jia, and Lei Ju. "An entropy-based distributed DDoS detection mechanism in software-defined networking." In 2015 IEEE Trustcom/BigDataSE/ISPA, vol. 1, pp. 310-317. IEEE, 2015.

[16] Zhao, Teng, Dan Chia-Tien Lo, and Kai Qian. "A neural-network based DDoS detection system using hadoop and HBase." In 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems, pp. 1326-1331. IEEE, 2015.

[17] Yadav, Satyajit, and S. Selvakumar. "Detection of application layer DDoS attack by modeling user behavior using logistic regression." In 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)(Trends and Future Directions), pp. 1-6. IEEE, 2015.

[18] Balkanli, Eray, A. Nur Zincir-Heywood, and Malcolm I. Heywood. "Feature selection for robust backscatter DDoS detection." In 2015 IEEE 40th Local Computer Networks Conference Workshops (LCN Workshops), pp. 611-618. IEEE, 2015.

[19] Badve, Omkar P., Brij B. Gupta, Shingo Yamaguchi, and Zhaolong Gou. "DDoS detection and filtering technique in cloud environment using GARCH model." In 2015 IEEE 4th Global Conference on Consumer Electronics (GCCE), pp. 584-586. IEEE, 2015.