Volume-11, Issue-07, July2022
JOURNAL OF COMPUTING TECHNOLOGIES (JCT)
International Journal
Page Number: 01-08

Machine Learning Based PRP Neural Network For IDS and Prevention

Jay Prakash Gawande,[1] *Prof. Darshna Rai,*[2]
[1]*M.Tech Student Cyber Security,* [2]*Assistant Professor,*
[1,2]*Department of Computer Science and Engineering*
[1,2]*School of Research & Technology (SORT), People's University, Bhopal, INDIA*
jaygawande0@gmail.com[1] *darshanarai@peoplesuniversity.edu.in*[2]

***Abstract*—Intrusion detection system (IDS) is regarded as the second line of defense against network threats. Cyber-attack indomitable problem between researchers. Recently March 2022 hackers used a DDoS attack to shut down the National Telecommunications Authority of the Marshall Islands. In this research work presented a PRP(Polak–Ribière–Polyak) with cascaded feed forward network for detection of DDoS cyber-attack. The PRP algorithm presents better learning efficiency as well as better accuracy. The proposed PRP algorithm shows better results as compared to other previous weight optimizer method based on machine learning as well as deep learning methods. For the implementation of proposed method use MATLAB 2020. This research work uses the Canadian Institute of Cyber Security (CICIDS2017) data set to perform the proposed methodology. The proposed method shows good results in terms of accuracy, precision, selectivity, sensitivity, and confusion matrix (C.M.). The presented method shows an accuracy of 98.60% and the other parameters are discussed in the simulation and result section.**

***Keywords*- Cyber-attack,PRP (Polak–Ribière–Polyak), Accuracy, Precision, Selectivity, Sensitivity, CICIDS2017, MATLAB 2020 and Confusion Matrix (C.M.). ,etc…**

I. INTRODUCTION

The The Internet has become an indispensable tool for human life in this epoch at every moment of life internet aids us. Hence to provide security for the internet become vital. In the proportion of advancement fear of unlawful activities has also been increasing rapidly. An attack on the basic pillars of security confidentiality, integrity, and availability is a sequence of activities having the aim of weakening computer network security. System attacks like external and internal attacks, attacks based on the network like a collection of information, Denial of Service by heavily requesting a particular target, and so on. There is no system which is made perfectly safe and secure because of few limitations, hence the attacker finally finds a loophole in the system to intrude, to analyze the network data for the probable intrusions (attacks), an IDS has become the principal component of computer security to bolster existing defenses.

Type of attacks

There are different type of attacks are happened in the cyber attack world. In the below section shows the different attacks.

A. Anomaly based intrusion detection system (AIDS)

This model is created by using machine learning statistical and knowledge-based methods, any difference between the model's behavior and observed behavior is considered as an anomaly.

B. Machine learning based IDS detection

Machine learning is broadly classified into supervised and unsupervised. Supervised counts on the significant information in labeled data, lack of labeled data is a limitation for this method.

1. **ANN -** It has the strong fitting ability and is capable of dealing with non-linear data, it is susceptible to becoming stuck in the local optimum, training is time taking via this approach, use of activation function and loss functions could be improvement measures.
2. **KNN** It applies to massive data, is very conducive for non-linear data, quickly trains the model, and is robust to noise, it takes a long testing time and is very sensitive to parameter k. It could reduce

comparison time by using trigonometric inequalities and optimized parameters by using PSO (Particle swarm optimization) [14], balancing of the dataset could be done by SMOTE(synthetic minority oversampling technique) [15].

3. **Naive Bayes** It can learn incrementally, robust to noise, On attribute-related data, its performance is not up to the mark, importing of latent variable could be done to relax the independent variable [16].
4. **SVM** It has strong generation capabilities and learns useful information from the small training set, It does not perform well on big data and is very sensitive to the kernel function. For further improvement, optimization can be done by using particle swarm optimization [17].
5. **Decision tree** It has strong interpretation and select features automatically, balancing of data with SMOTE and introduction of latent variables may improve the performance [18].
6. **K-Means** It is simple and has strong scalability and can be fitted into big data, it can be trained rapidly [19].

Ensemble and Hybrid classifiers Some classifiers are weak in performance and do not perform as expected hence better approach comes into the frame by joining weak classifiers, which gives far better results than earlier, this approach is called the ensembling of classifiers. Ensemble method trained various classifiers and then by voting final output is selected.

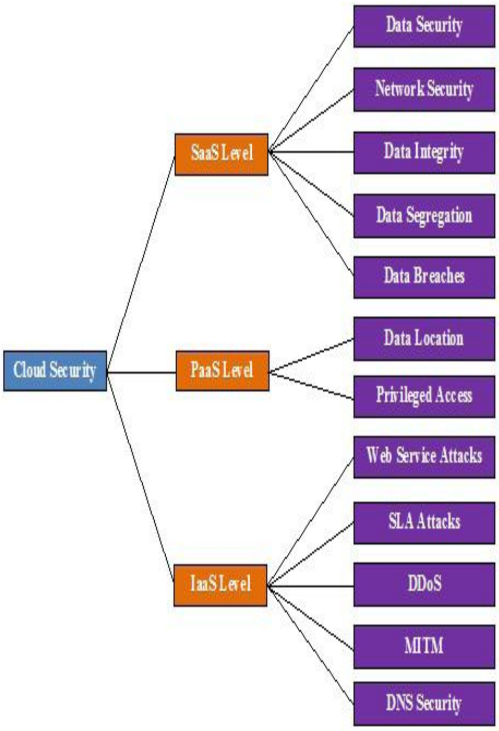


Figure. 1 – Type of Attacks in Clouds

II. LITERATURE REVIEW

In In this chapter discuss the various previous work projected by different researcher. Additionally, discuss the comparison of various analysis work.

Akgun, et.al. (2022).In this paper, we provide a new intrusion detection system that is based on deep learning models for DDoS attacks. We used CIC- DDoS 2019 dataset, which contains 12 classes, including a benign class. We tested various deep learning models such as DNN, CNN, and LSTM for various units per layer. We also improved the system using pre processing techniques such as feature elimination and selsection where we selected 40 important features out of 88. We obtained a new homogeneous data set by selecting an equal number of samples from each attack type with random subset selection. Afterward, we removed duplicate records to obtain a clean, non-repetitive data set which most relevant studies ignored. In this respect, this study presents two new data sets to the literature that directly affect the performance of the training processes produced from the CIC-DDoS 2019 data set. Finally, we applied min- max normalization processes to examine their effect on the performance. Therefore we produced applicable data by obtaining a normalized set containing an equal number of pre-processed samples from each attack type without duplicates [1].**Ullah, S., et.al. (2022)**.The extensive growth of smart vehicularnetworks has opened up several doors for cyber criminals. Attacks on intra-vehicle networks can cause deaths and severe accidents. This research proposes a hybrid DL-based model for intrusion detection in IoV. The presented scheme contains a hybrid combination of LSTM and GRU that reduces the training and response time. The performance of the proposed approach was evaluated by conducting extensive experiments on a combined dataset of CIC DoS 2016, CICIDS 2017, and CSE-CIC-IDS 2018, and car-hacking datasets. The experimental results demonstrate thatthe proposed model achieves 99.5% accuracy for the combined DDoS dataset and 99.9% for the car hacking dataset, respectively [2].**Saghezchi, et.al. (2022).** In this paper, we applied ML for detecting DDoS attacks in Industry 4.0 CPPSs. Authors exported network traffic traces (PCAP files) from a real-world large-scale semiconductor production factory and employed 11 different semi-supervised, unsupervised, and supervised ML algorithms for anomaly detection in network traffic flows. The simulation results showed that supervised learning algorithms outperformed both unsupervised and semisupervised ones. In particular, DT, RF, and K-NN detected DDoS attacks with Accuracy =Recall = 0.999, Precision = 0.999, and FPR = 0.001.However, the two applied unsupervised algorithms (K-Means and EM) also showeda very good performance (Accuracy = 0.95, Recall > 0.9, Precision > 0.9, and FPR < 0.09),although their performance decreased significantly when the PCA algorithm was applied(even with 95% variance retain). This is an interesting finding, since unlike supervised learning, unsupervised learning does not require data labelling which is a tedious task in practice and needs a significant amount of human effort and intervention [3].**Mighan, et.al (2021)**-In this researcher work, the literature reveals that as use of internet increases, intrusion detection is considered as an important security issue. Therefore, the main goal of this paper is to address scalable network intrusion detection in big data framework. Besides, one has to keep in mind that detection accuracy, time and cost are of importance, as well. Thus, the present study tried to use deep learning methods to improve diagnostic accuracy, decrease the error rate, improve prediction speed and save time and cost. This presented work a hybrid SAE–SVM scheme for a fast and efficient cyber security intrusion detection system. In the proposed system, a stacked auto encoder network was used as a feature extraction method and SVM as the classifier. The deep network platform outperformed other feature extraction methods. Also, the performance of the proposed framework was evaluated using the big data processing tool of Apache Spark and machine learning algorithms. We examined the performance of the proposed SAE–SVM scheme by reducing the 42-dimensional ISCX dataset to approximately 75% of its original size and then classified the reduced data by SVM in Spark [4]. **Snehi, J., et.al (2021)-** In this research work, The defense is the critical element of the computer system, and the most challenging issues are detecting the intrusion attacks. The IDS is the most critical cyber-security factor which can detect intrusion before, during, and after an attack. This works provides an overall IDS benchmarking which quantifies different IDS properties, types of anomaly-based IDS that are deployed in different environments or platforms, and comparison among them based on methods used, their details, and advantages of each method. In analyzed the different IDS techniques based on anomaly and various issues associated with anomaly-based IDSs. They addressed global environments for intrusion detection and framework for behavioral or anomaly-based intrusion detection systems and discussed the challenges facing anomaly-based IDSs. After reviewing the various anomaly-based IDS techniques, Then analyzed that successful detection rates could not be achieved by a single technique. To lower the false prediction rate and decreased the complexity of the process, an efficient automated hybrid technique is suggested for achieving accurate detection rates to enhance anomaly detection [5]. **Liu, L., etr.al (2020)**- In this presented work, As network intrusion continues to evolve, the pressure on network intrusion detection is also increasing. In particular, the problems caused by imbalanced network traf c make it difficult for intrusion detection systems to predict the distribution of malicious attacks, making cyberspace security face a considerable threat. This presented a novel Difcult Set Sampling Technique (DSSTE) algorithm, which enables the classification model to strengthen imbalanced network data learning. A targeted increase in the number of minority samples that need to be learned can reduce the imbalance of network trafc and strengthen the minority's learning under challenging samples to improve the classification accuracy. Then six classical classification methods in machine learning and deep learning and combined them with other sampling techniques. Experiments show that our method can accurately determine the samples that need to be expanded in the imbalanced network= trafc and improve the attack recognition more effectively [6].**Sarhan, et.al (2020)**- In this researcher works, They community with four new NIDS datasets using Net Flow features. These datasets are to be used in ML-based NIDS training and evaluation stages. The datasets are showing positive results by achieving similar binary-class detection performance compared to the complete set of their respective original datasets. However, the NF-ToN-IoT and NF-CSE-CIC-IDS2018 datasets were inefficient when conducting multi-class detection experiments. Further feature analysis is required to identify the strength of each Net Flow feature, and how these datasets can be improved by adding key features from the original datasets to aid in the detection of missed attack types [7].

III. PROPOSED METHODOLOGY

A. Distributed Denial of Service (DDoS)

Internet resources and services are made unavailable to their intended users by denial of service (DoS) attacks. Flooding the victim machine with external communication requests is a common DoS attack tactic, and it renders the device unable to reply to valid traffic.

B. 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.

C. Training On*q*-Polak–Ribière–Polyak conjugate gradient algorithm

This section discusses the proposed solution for the detection and identification of DDOS attacks on clouds. **On*q*-Polak–Ribière–Polyak conjugate gradient algorithm** Consider the following unconstrained nonlinear optimization problem:

$$(P) \min f(x) \qquad (3.1)$$

Where $f : R^n \rightarrow$ R is a continuously q-differentiable function. The numerical optimization algorithms of general objective functions differ mainly in generating the search directions. In the conjugate gradient algorithms, a sequence of iterates is generated with a given starting point $x^0 \in R^n$ by the following schema:

$$x^{k+1} = x^k + p^k,\ p^k = a_k d^k_{q^k}, \tag{3.2}$$

for all $k \geq 0$, where x^k is the current iterate, $d^k_{q^k}$ is a descent direction of f at$x^k$and$a_k > 0$ isthe step-length. Note that the descent direction $d^k q_k$ =-$g^k q_k$ leads to the q-steepest descentmethod. In the case q^k approaches,

$$(1,1,\ldots\ldots.,1)^T \tag{3.3}$$

as $k \longrightarrow \infty$, the method reduces to the classical steepest descent method [7]. The search direction d^k $q$is guaranteed to have a descent direction due to the following:

$$(g^k_{qk})Td^k q_k < 0. \tag{3.4}$$

The directions $d^k q_k$ are generated in the light of classical conjugate direction methods as:

$$a_k d^k_{q^k} = \begin{cases} -g^k q_k k = 0 \\ -g^k q_k + \beta_k q^{-PRP} d^{k-1} q^{k-1,}\text{k} \geq 1, \end{cases} \tag{3.5}$$

where $\beta_k{}^{q-PRP} \in$ R is modified from a scalar quantity β^k in the PRP method and presented as follows:

$$\beta_K{}^{q-PRP} = \frac{(g^k_{q^k})^T (g^{k-1}_{q^{k-1}})}{\left\| g^{k-1}_{q\,k-1} \right\|^2}. \tag{3.6}$$

Trainingof D-DoS attack detection

In the machine learning process, training is an important part of the proposed attack detection. For the training first required the data set of previous attack. For the implementation of proposed method, we use (Canadian Institute of Cybersecurity (CICIDS2017)) [31]. This data set is available Kaggle website.

Steps of Training by ***Polak Ribiere Polyak (PRP)***

```
Start
Select Data set,
Load training data% train data tr label
Apply Labeling on data,
for i = 1:length(reduce)
    val = tr_label2(i);
    t(i,val) = 1;
end
%Apply Training Using Polak Ribiere Polyak (PRP)
net1 = cascade forward net(size(x,2),'traincgp')
net1.train Param epochs = 30; % Number of iterations
net1.train Param goal = 1e-5; %
net1.train Param.min_grad = 1e-6;%
net1 = train(net1,x',t'); % Apply Cascaded feed forward
Perform(net,t,y) % Performance Calculation
Accuracy = match * 100 / length(reduce)
Modified Data Unique id Selected Features
Testnetunique idselected feature % Training Data
End
```

D. Testing of D-DoS Attack Detection

Now discuss the testing for proposed CFFNN[21] based polak ribiere polyak (PRP). The DDoS attack classify benign attack, DoS Hulk, and DoS slow loris.

Testing of Cyber DDoS Attack

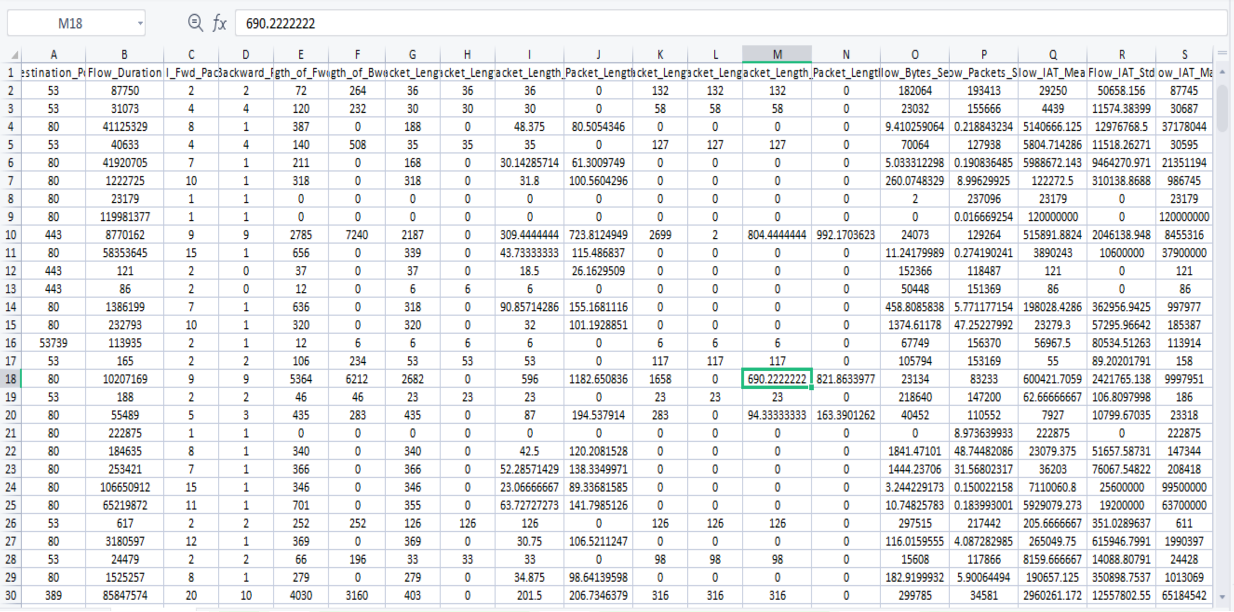
```
Start
[a,b,exc data]= xlsread('data-set');% Read data set
Load Final Test% Unique id elected feature
for j = 1:30 % Unique features cal.
    rw = t_data(j);
    Lookup =unique_id{selected_feat(j)};% Save features
    s = find(rw==lookup); % find unique
    t_data2(j) = s;
End
for y = net1(t_data2'); % Apply CFFNN
    detected = unique_id{69}(loc); % Unique 69 feature
end
if (strcmp(req_cat,'BENIGN')) % Detection attack
    actual(pos) = 1 ;
elseif(strcmp(req_cat,'DoS Hulk')) % Detection attack
    actual(pos) = 2 ;
Else,
    actual(pos) = 3; % DoS slow loris,
End,
detect(pos) = loc;,
End,
% Performance Parameters Calculation,
mat,selectivity, sensitivity, specificity, accuracy,
End
```

IV. SIMULATION AND RESULTS

4.1 Introduction

In this section we are describing out the implementation detail and designing issues for our proposed research work. By searching we have observed that for our proposed work the MATLAB 2020 is well known platform to perform suggested approach

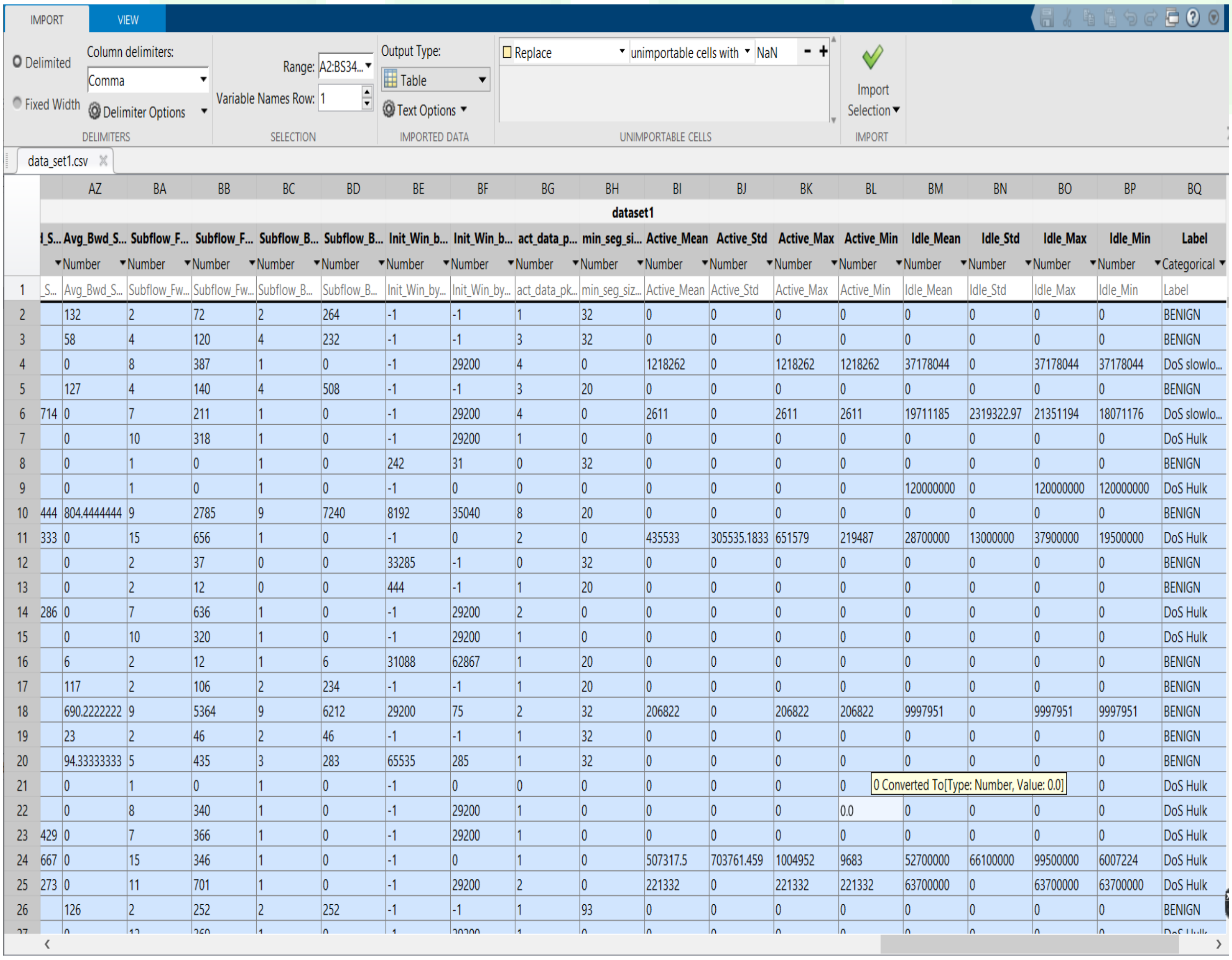
Data set



M18 fx 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	Backward_P	gth_of_Fw	gth_of_Bw	acket_Leng	cket_Leng	acket_Length_	Packet_Length	cket_Leng	acket_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

Figure 2. Data Set In Excel



DELIMITERS: ● Delimited ○ Fixed Width; Column delimiters: Comma; Delimiter Options

SELECTION: Range: A2:BS34...; Variable Names Row: 1

IMPORTED DATA: Output Type: Table; Text Options

UNIMPORTABLE CELLS: Replace unimportable cells with NaN

data_set1.csv

		AZ	BA	BB	BC	BD	BE	BF	BG	BH	BI	BJ	BK	BL	BM	BN	BO	BP	BQ
										dataset1									
	I_S...	Avg_Bwd_S...	Subflow_F...	Subflow_F...	Subflow_B...	Subflow_B...	Init_Win_b...	Init_Win_b...	act_data_p...	min_seg_si...	Active_Mean	Active_Std	Active_Max	Active_Min	Idle_Mean	Idle_Std	Idle_Max	Idle_Min	Label
		Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Number	Categorical
1	_S...	Avg_Bwd_S...	Subflow_Fw...	Subflow_Fw...	Subflow_B...	Subflow_B...	Init_Win_by...	Init_Win_by...	act_data_pk...	min_seg_siz...	Active_Mean	Active_Std	Active_Max	Active_Min	Idle_Mean	Idle_Std	Idle_Max	Idle_Min	Label
2		132	2	72	2	264	-1	-1	1	32	0	0	0	0	0	0	0	0	BENIGN
3		58	4	120	4	232	-1	-1	3	32	0	0	0	0	0	0	0	0	BENIGN
4		0	8	387	1	0	-1	29200	4	0	1218262	0	1218262	1218262	37178044	0	37178044	37178044	DoS slowlo...
5		127	4	140	4	508	-1	-1	3	20	0	0	0	0	0	0	0	0	BENIGN
6	714	0	7	211	1	0	-1	29200	4	0	2611	0	2611	2611	19711185	2319322.97	21351194	18071176	DoS slowlo...
7		0	10	318	1	0	-1	29200	1	0	0	0	0	0	0	0	0	0	DoS Hulk
8		0	1	0	1	0	242	31	0	32	0	0	0	0	0	0	0	0	BENIGN
9		0	1	0	1	0	-1	0	0	0	0	0	0	0	120000000	0	120000000	120000000	DoS Hulk
10	444	804.4444444	9	2785	9	7240	8192	35040	8	20	0	0	0	0	0	0	0	0	BENIGN
11	333	0	15	656	1	0	-1	0	2	0	435533	305535.1833	651579	219487	28700000	13000000	37900000	19500000	DoS Hulk
12		0	2	37	0	0	33285	-1	0	32	0	0	0	0	0	0	0	0	BENIGN
13		0	2	12	0	0	444	-1	1	20	0	0	0	0	0	0	0	0	BENIGN
14	286	0	7	636	1	0	-1	29200	2	0	0	0	0	0	0	0	0	0	DoS Hulk
15		0	10	320	1	0	-1	29200	1	0	0	0	0	0	0	0	0	0	DoS Hulk
16		6	2	12	1	6	31088	62867	1	20	0	0	0	0	0	0	0	0	BENIGN
17		117	2	106	2	234	-1	-1	1	20	0	0	0	0	0	0	0	0	BENIGN
18		690.2222222	9	5364	9	6212	29200	75	2	32	206822	0	206822	206822	9997951	0	9997951	9997951	BENIGN
19		23	2	46	2	46	-1	-1	1	32	0	0	0	0	0	0	0	0	BENIGN
20		94.33333333	5	435	3	283	65535	285	1	32	0	0	0	0	0	0	0	0	BENIGN
21		0	1	0	1	0	-1	0	0	0	0	0	0	0	0 Converted To[Type: Number, Value: 0.0]			0	DoS Hulk
22		0	8	340	1	0	-1	29200	1	0	0	0	0	0.0	0	0	0	0	DoS Hulk
23	429	0	7	366	1	0	-1	29200	1	0	0	0	0	0	0	0	0	0	DoS Hulk
24	667	0	15	346	1	0	-1	0	1	0	507317.5	703761.459	1004952	9683	52700000	66100000	99500000	6007224	DoS Hulk
25	273	0	11	701	1	0	-1	29200	2	0	221332	0	221332	221332	63700000	0	63700000	63700000	DoS Hulk
26		126	2	252	2	252	-1	-1	1	93	0	0	0	0	0	0	0	0	BENIGN

Figure 3Data Set In MATLAB

4.2 Result Parameters

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

4.4.1 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.

4.4.2 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.

4.4.3 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.

4.4.4 True Negative (T.N.)

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

4.4.5 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. The FN is also a key consideration in several industrial applications, such as the identification of weeds or diseases and the overall accuracy of the detection scheme. If the weeds or diseased plants are not removed, they can quickly spread or expand, threatening net production even after application of a specific treatment. Any approach with more accuracy, but a great amount of FN may represent a larger risk.

$$\text{Accuracy} = (TP + TN) / S \qquad (4.1)$$

When S There were a total of FP false positives (deaths mistaken for plants) as well as FNR false negatives in the test set, with FP being the amount of FP false positives. It's the likelihood of a positive test if the plant in question is showing signs of disease. The accuracy is the ratio of addition of number of correct production (TP+TN) and total number of production (TP + TN + FP + FN) .

$$\text{Acc} = \frac{TP+TN}{TP+TN+FP+FN} \qquad (4.2)$$

where: TP = True positive; FP = False positive; TN = True negative; FN = False negative

4.4.6 Precision

Predictability is one measure of a machine learning model's efficiency, and accuracy measures the model's ability to make accurate predictions. According to this definition, reliability is a ratio of genuine positives shared by all predicted positives (i.e., the number of true positives plus the number of false positives).

Precision (P):

$$P = \sum\left(\frac{tp}{tp + fp}\right) \times 100 \qquad (4.3)$$

4.4.7 Selectivity

The challenge of estimating the number of records, known as "selectivity estimate," is a common one in database systems. Query selectivity prediction with numerous associated characteristics is particularly difficult to get right.

Sensitivity (Se)

$$Se = \sum\left(\frac{tp}{tp + fn}\right) \times 100 \qquad (4.4)$$

4.4.8 Sensitivity

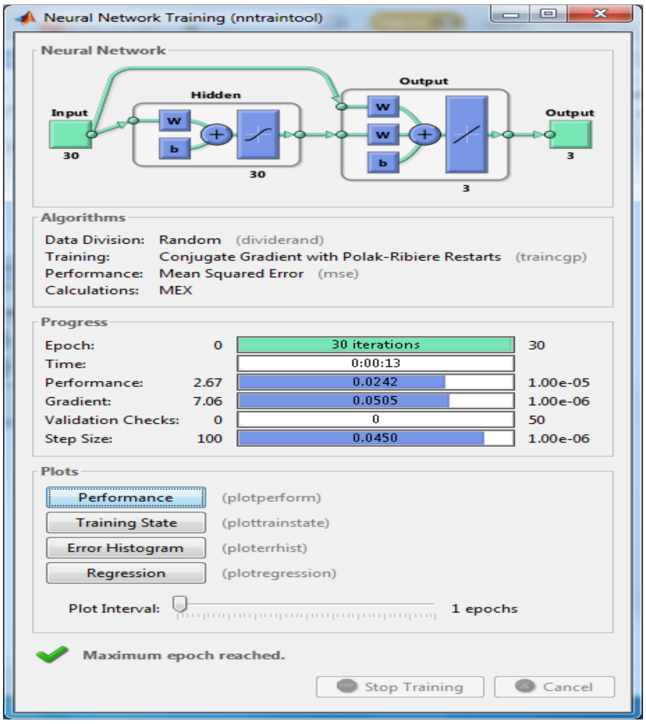
To determine a model's sensitivity, we look at how well it can estimate true positives in each of the categories that are accessible to us. An evaluation of a model's ability to predict the true negatives of each available category is called specificity.

4.4.9 Specificity

Percentage of anticipated negatives that were actually negatives is a measure of specificity (or true negative). False positives, in this case, may be defined as the percentage of genuine negatives that were incorrectly forecasted as positives. Also known as false positive rate, this percentage is quite high.

Specificity (Sp)

$$Sp = \sum\left(\frac{tp}{tn + tp}\right) \times 100 \qquad (18)$$



```
Data Division:   Random   (dividerand)
Training:        Conjugate Gradient with Polak-Ribiere Restarts   (traincgp)
Performance:     Mean Squared Error   (mse)
Calculations:    MEX

Epoch:                  0   30 iterations   30
Time:                       0:00:13
Performance:         2.67   0.0242          1.00e-05
Gradient:            7.06   0.0505          1.00e-06
Validation Checks:      0   0               50
Step Size:            100   0.0450          1.00e-06

Performance       (plotperform)
Training State    (plottrainstate)
Error Histogram   (ploterrhist)
Regression        (plotregression)

Plot Interval:    1 epochs

✓ Maximum epoch reached.
```

Fig. 4 NN simulation model

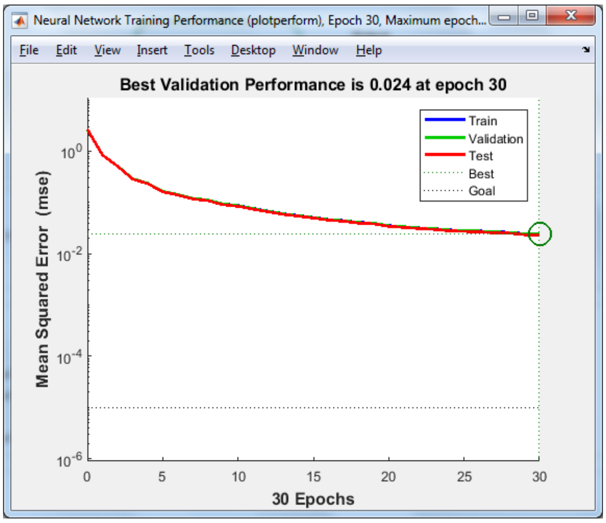


Fig. 5 Output of training validation performance

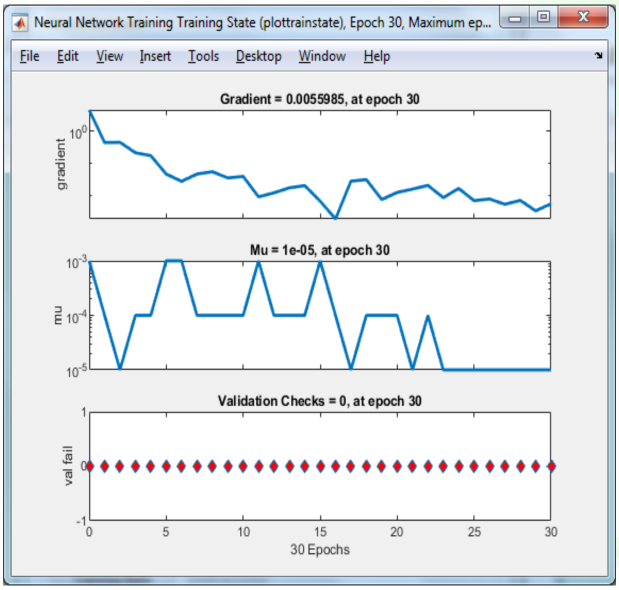


Figure 6 Shows the neural network training states

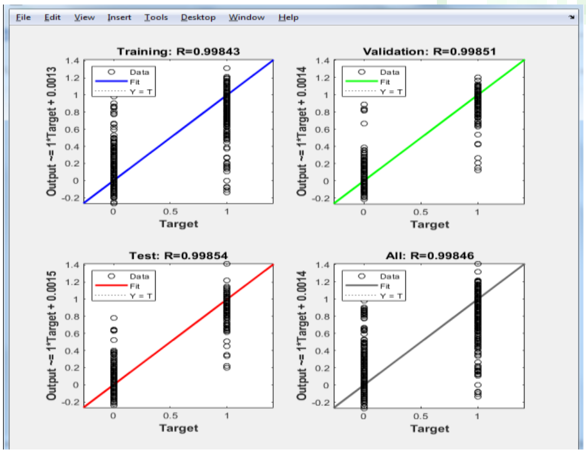


Figure 7 Shows the result outcomes of training, validation and test

Experimental Results 1

Proposed Accuracy (Acc)	99.12
Acc. hybrid	98.6070
True Positive	318 274 94
False Negative	0 4 3
False Positive	3 3 1
True Negative	372 412 595

V.CONCLUSIONS

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.As a result of a denial of service (DDoS) attack, a targeted system is unable to provide regular services to its legitimate customers. In this proposed work presented modified feature selected based neural network for efficient DDoS attack detection. For the implementation of proposed work MATLAB 2020 software. MATLAB is well known academic as well as industrial research software this work. The proposed method design and simulated in the R2020 MATLAB. There are different type of DDoS attack are present in internet.

References

[1] Akgun, Devrim, Selman Hizal, and Unal Cavusoglu. "A new DDoS attacks intrusion detection model based on deep learning for cybersecurity." Computers & Security 118 (2022): 102748.

[2] Ullah, Safi, Muazzam A. Khan, Jawad Ahmad, Sajjad Shaukat Jamal, Zil e Huma, Muhammad Tahir Hassan, Nikolaos Pitropakis, and William J. Buchanan. "HDL-IDS: a hybrid deep learning architecture for intrusion detection in the Internet of Vehicles." Sensors 22, no. 4 (2022): 1340.

[3] Saghezchi, Firooz B., Georgios Mantas, Manuel A. Violas, A. Manuel de Oliveira Duarte, and Jonathan Rodriguez. "Machine learning for DDoS attack detection in industry 4.0 CPPSs." Electronics 11, no. 4 (2022): 602.

[4] Mighan, Soosan Naderi, and Mohsen Kahani. "A novel scalable intrusion detection system based on deep learning." International Journal of Information Security 20, no. 3 (2021): 387-403.

[5] Snehi, Jyoti, Abhinav Bhandari, Manish Snehi, Urvashi Tandon, and Vidhu Baggan. "Global intrusion detection environments and platform for

anomaly-based intrusion detection systems." In Proceedings of Second International Conference on Computing, Communications, and Cyber-Security, pp. 817-831. Springer, Singapore, 2021.

[6] Liu, Lan, Pengcheng Wang, Jun Lin, and Langzhou Liu. "Intrusion detection of imbalanced network traffic based on machine learning and deep learning." IEEE Access 9 (2020): 7550-7563.

[7] Sarhan, Mohanad, Siamak Layeghy, Nour Moustafa, and Marius Portmann. "Netflow datasets for machine learning-based network intrusion detection systems." In Big Data Technologies and Applications, pp. 117-135. Springer, Cham, 2020.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] Virupakshar, Karan B., et al. "Distributed denial of service (DDoS) attacks detection system for OpenStack-based private cloud." Procedia Computer Science 167 (2020): 2297-2307.

[9] Singh, Maninder Pal, and Abhinav Bhandari. "New-flow based DDoS attacks in SDN: Taxonomy, rationales, and research challenges." Computer Communications 154 (2020): 509-527.

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] Dayal, Neelam, and Shashank Srivastava. "An RBF-PSO based approach for early detection of DDoS attacks in SDN." In 2018 10th International Conference on Communication Systems & Networks (COMSNETS), pp. 17-24. IEEE, 2018.

[16] 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.

[17] 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.

[18] 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.

[19] 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.