**University of Miskolc**

**Analysis And Implementation Of ML And DL Techniques In The Detection Of DDoS Attacks**

**MARADI BALACHANDRA VINAY KARTHIK**

**AZF2YA**

Abstract:

In the technological era, the use of infrastructure gadgets is increasing dramatically, which resulted in an increase in cyber intrusions. Among them, Distributed Denial of Service (DDoS) attacks become more difficult to detect. DDoS attacks create a significant threat to the security and integrity of today’s essential infrastructure, computer networks, and information systems. Detecting DDoS attacks can be challenging before you can take action to mitigate them. In this paper, we have implemented various Machine Learning algorithms for the detection of DDoS attacks. Several classifiers such as Random Forest (RF), K- nearest neighbor (KNN), Support Vector Machine (SVM), Multi Level Perceptron and Extra Trees Classifier were implemented and their performance is measured on NSL KDD benchmark dataset. The experimental results show that, the Random Forest model performed well compared to state-of-art methods with an accuracy of 99% on train data and 86% on full test dataset.

I. INTRODUCTION and PROBLEM STATEMENT

Technology has become an inherent component of human life in today’s society. Indeed, everything from the corporate sector to educational institutions has migrated from offline to online during the Covid-19 pandemic. As a result, the number of incursions and attacks using Internet-based technology has increased exponentially. The Distributed Denial of Service (DDoS) attack, which can quickly devastate Web services and applications, is one of the most dangerous threats that has emerged. The goal of such attacks is to slow down the target server by flooding it with bogus traffic created by infected devices (botnets) in the network. A botnet is a collection of devices that have been compromised with computer viruses and have been taken over by a malicious actor. Botnets can be used to carry harmful operations such as spamming, data theft, spyware, false ad clicks, and distributed denial-of-service (DDoS) assaults.

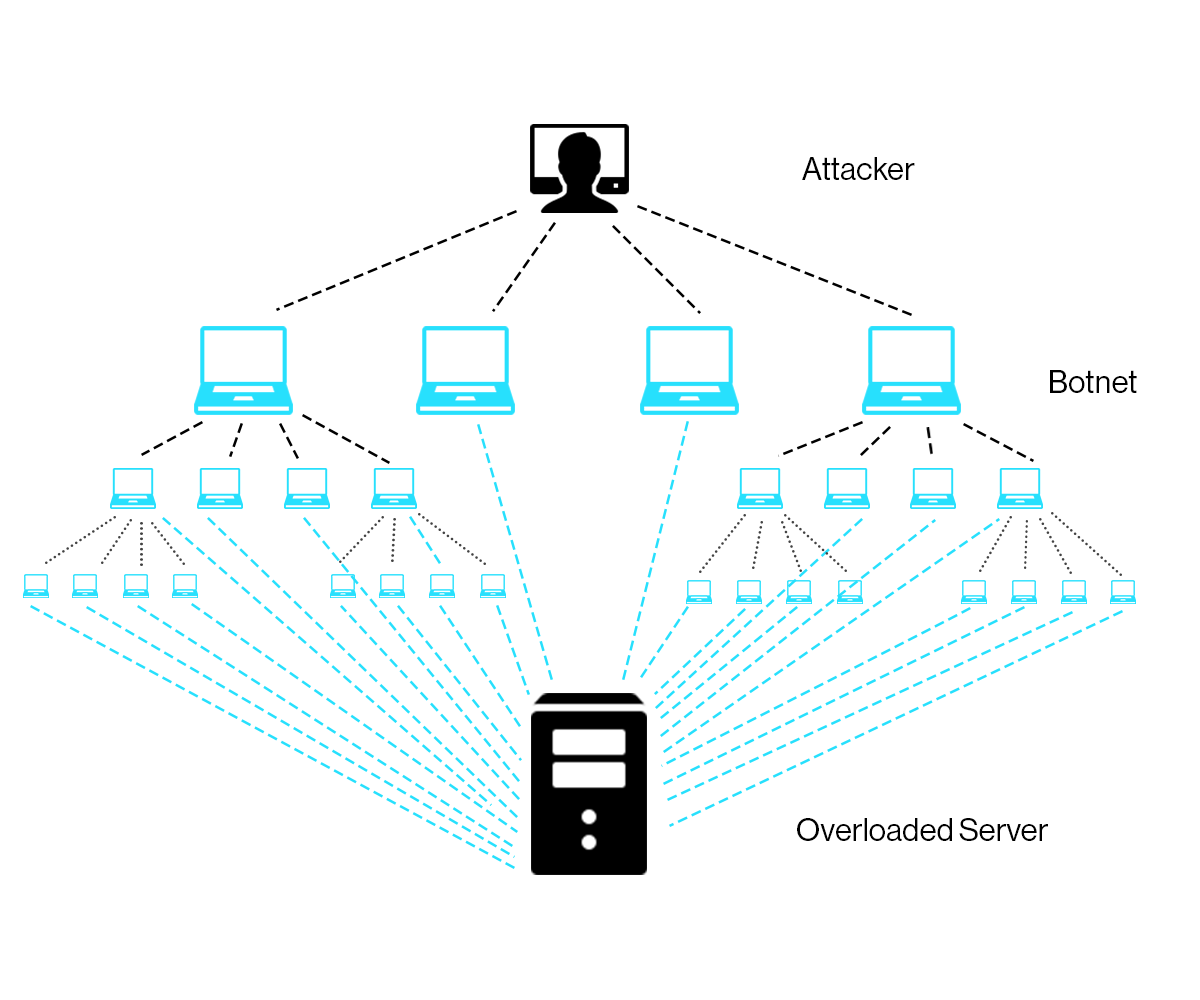


Figure 1: DDoS Attacks

DDoS attacks are a subset of DoS attacks that arise when an offender exploits several computing devices in order to disrupt a targeted victim’s normal traffic. According to Daily Swig report, the Cryptocurrency exchange EXMO was flooded with 30 GB of traffic per second in February 2021, and the services was unavailable for 2 hours.

Moreover, this attack not only disrupts the services but it impacted the entire network infrastructure including Application programming interface (API), webpages, web server API and exchange charts. According to Microsoft, the incidence of DDoS assaults jumped about 25 percent during the first quarter of 2021 over to the 4th quarter of the previous year. Furthermore, the biggest attack throughput recorded in 2021 was 625 GBPS, decreasing from 1 TBPS in the third quarter of 2020.

Intrusion Detection Systems (IDSs) are one of the many methods capable of protecting target machine from such assaults. IDS provide protection depending on the needs of the customers and can be applied on both the client / server sides. Furthermore, IDS is combined with machine learning techniques to automatically make clever and intelligent decisions. Machine learning algorithms can make decisions depending on the type of attacks that the system is experiencing, while still preserving the network’s integrity, privacy, and security. DDoS assaults can be classified into three kinds depending on the target and behavior.

In Volume based attacks, the purpose of the assault is to saturate the bandwidth of the targeted site, and the size of the attack is measured in bits per second (Bps). These include UDP floods, ICMP floods, and other spoofed-packet floods. Protocol based attacks include SYN floods, fragmented packet attacks, Ping of Death, Smurf DDoS, etc. This form of assault uses network bandwidth or intermediate communication infrastructure like firewalls and load balancers, and is calculated in packets per second (PPS). The Application layer attacks include Low-and-slow attacks, GET/POST floods, attacks on Apache, Windows, or OpenBSD vulnerabilities, and more. The aim of these assaults, which are made up of seemingly genuine and harmless requests, is to overwhelm the web service, and the size is measured in Requests per second (RPS).

The above mentioned incidents highlight the need for a reliable way of detecting DDoS attacks. There are numerous approaches available. Statistical approaches could be used to detect abnormal variations in resource utilization caused by a DDoS attack. The drawback with statistics-based detection is that the usual network packet distribution cannot be determined. Rather, only a similar behavior may be replicated.

Some approaches that use data mining tools can detect the assault with a high accuracy rate. However, in most cases, these technologies cannot be applied in real-time computation. Therefore, In this paper, We have employed various machine learning algorithms for detection of DDoS attacks and their performance is measured on NSL KDD benchmark dataset.

II. BACKGROUND AND RELATED WORK

A. Distributed Denial of Service attack (DDoS)

DoS and DDoS attacks have similar intentions and are frequently carried out in the same manner. All these sorts of threats aim to cause a denial of service to their target by overwhelming either the victim’s network capacity or framework such as CPU or memory. In broad sense, causing a denial of service attack implies flooding a victim, such as a web application, with huge number of traffic. When a user gets more data packets than its network capacity or system resources can handle, it will face difficulties in accessing or transmit portions of the intended data traffic. The main difference between DoS and DDoS attacks is that, DoS attacks originate from a single source whereas, DDoS attacks originate from dispersed range of networks with the help of Zombies/Botnets. Both sort of incidents have found to be effective and lethal, but DDoS attacks seem to be the most challenging to manage. The main motivation for this is the potential size of a DDoS attack. Figure 2, shows a typical example of DDoS attack.

B. DDoS Attack Detection in Machine Learning

Artificial intelligence, machine learning, statistics, information theory, and pattern recognition are just a few of the data mining techniques that have refined IDSs (Intrusion Detection System). ML methods have gained attention in the field of DDoS detection and prevention. DDoS detection approaches based on Machine Learning can be classified as supervised, unsupervised, and semi-supervised. In Supervised ML techniques, machines are trained using well-labelled training data and on the basis of data machine predicts output. The basic goal is to identify a mapping function that will connect the input variable (X) and the output variable (Y). On the other hand, Unsupervised ML techniques use unlabeled dataset, and all allowed to act on that data without any supervision. Unsupervised learning’s major purpose is to discover a dataset’s underlying structure, categorize data based on similarities, then represent that dataset in a compressed fashion. Furthermore, Semi-supervised methods also take advantage of the capabilities of both labelled and unlabeled datasets.

C. Related Work

In [3], Tahir Alyas discussed Detection and mitigation of DDoS attacks in cloud computing using a machine learning algorithm. The two most effective algorithms Naïve Bayes and Random forest, concluding with that Naïve Bayes produce more positives. In [4], focused on A study for the DDOS attack classification method. Here, three receptive classification algorithms can be compared to Naive Bayes, Decision tree, Artificial neural network from that Artificial neural network produces the best accuracy rate of 84.3% compared with the other two techniques. In [6], the paper proposed Research on multiple Machine learning for anomaly detection. In this approach NSL-KDD dataset is divided into two dataset its performance can be compared with the confusion matrix.

Fadil and team used the NB method to predict the existence of DDoS attacks based on the mean and standard deviation of network packets and achieved precise results. An RF is a collection of decision trees. The majority of the outcomes of individual decision trees determine the classification [5]. In [7], Wang and team proved that with well-computed key features in DDoS data, experimental results show an RF algorithm can attain valid classification performance and an optimal feature subset.

Li and Lu in [8] combined Long Short-Term Memory (LSTM) and Bayesian methods to detect DDoS attacks. LSTM is suitable for events with long intervals and delays in the time domain. In other words, LSTM can control the value of the indefinite length of time and decide whether the information should be retained or removed. The author used LSTM to identify the confidence index of DDoS attacks and further used the Bayesian method to make a second judgment to improve detection accuracy.

In [9], Yong and team applied machine learning models to detect web shell to build secure solutions for IoT networks. Ensemble methods, including random forest (RF), extremely randomized trees (ET), and Voting, are used to improve the performances of these machine learning models. Their findings show that RF and ET are suitable for lightweight IoT scenarios, and the Voting method is effective for heavyweight IoT scenarios.

Hemalatha and team in [10] propose an efficient malware detection system based on deep learning. The system uses a reweighted class-balanced loss function in the final classification layer of the DenseNet model to achieve significant performance improvements in classifying malware by handling imbalanced data issues. Comprehensive experiments performed on four benchmark malware datasets showed that the proposed scheme has superior performance with a higher detection rate and lower computational cost.

Idhammad et al. , presented a technique for detecting DDoS attacks using a feed-forward neural network in[11]. They gathered inbound traffic first, then used a correlation-based feature selection method to extract relevant information. The UNSW-NB15 and NSLKDD datasets were used to validate their method.

Bhuvanesvari, Amma et al in [12], realized skill, deep intellect. The author extracted intelligence from radial basis functions composed of different levels of abstraction. Experiments were performed on the well-known Network Security Laboratory (NSL KDD) and UNSW-NB15 datasets. The authors claim that it is more accurate than other existing methods.

Intrusion detection using machine learning algorithms was proposed by Nuno Martins et al.in [13]. They used the KDD dataset. They used various supervised models to balance the classification method and improve its performance. In this paper, a comparative research was proposed using several classification methods, and the results were positive.

For evaluation, in [16] Larriva-Novo et al. Presented two bench- mark datasets, particularly UGR16 and UNSW-NB15, as well as the most widely used dataset KDD99. The scalar and standardization capabilities of the pre-processing technique are assessed. Various attribute arrangements are used to apply these pre-processing models. The purpose of this investigation is to assess this structure by utilizing various information A comparison work for network traffic classification was proposed by Xianwei Gao et al. in [15]. For intrusion detection, they used classification algorithm. The datasets CI- CIDS and KDD were obtained from the UCI repository. In comparison to other methods, they discovered support vector machine SVM to be one of the better.

For detecting several sorts of abnormalities in the network, Xiao et al. in[14] devised the most widely used kNN technique. The kNN method was used to identify as many bots as possible in the network. When it came to detecting unknown attacks, the accuracy of any method was increased.

III. PROPOSED METHODOLOGY

In this study, we have implemented various types of machine learning classifiers such as Random forest, KNN, SVM, Multi-Level Perceptron and Extra Trees Classifier. The experiment was conducted on well-known dataset NSL KDDdataset. The dataset contains normal and different kinds of attack types. Out of 4 classifications techniques, Random Forest performed well with an accuracy of 99% on train data and 86% on test data. Also, we have done feature preprocessing technique in which we have dropped some features which have low importance.

1. Classifiers Used

* Random forest
* KNN
* SVM
* Multi-Level Perceptron
* Extra Trees Classifier

IV. EXPERIMENTAL ANALYSIS AND RESULTS

A. Experimental Setup

The Codes were made using Google Colaboratory and are python notebooks of “.ipynb” format. The codes were run using the Google Colaboratory. The necessary packages were imported and can be found under the requirements notebook.

B. Dataset Description

The NSL KDD data set was created at the Australian Centre for Cyber Security’s Cyber Range Lab. It has about 175,000 training samples and 82,000+ test samples that have been classified as benign or harmful (including attack type).The distribution of normal and attack traffic is depicted in fig 4. As detailed in [31], the data comprises nine forms of more recent, low-footprint assaults.

The benign flow is also more representative of contemporary data traffic. Assaults compromise a far lesser amount of the data in this sample, accounting for less than 25% of the total. This data set was also created with the intention of more closely matching the content of the training and testing sets, as stated in [32].

|  |  |  |  |
| --- | --- | --- | --- |
|  | Test Data | Train Data | Total |
| Attacks | 12833 | 58630 | 71463 |
| Normal | 9711 | 67343 | 77054 |
| Total | 22544 | 125973 | 148517 |

Distribution of DATA in KDD train DataSet

C. The Analysis

The raw data in the NSL KDD has 43 features, which include integer and text data. We need to make the data numeric. Hence we employ preprocessing techniques to achieve this.

Now we have total 117 features in our dataset after the introduction of Dummies to replace the text data. The final shape of the NSL KDD data to be used is:

Train Data has shape: (125973, 117)

Test Data has shape: (22544, 117)

During pre-processing, we have used one-hot encoding and feature Engineering and we got 95.12% accuracy on train data and 87.86% accuracy on test data.

* Support Vector Machine: The preprocessing Notebook is run and the processed data is used to train the SVM and the Validation dataset is used to calculate the validation accuracies. The test data set is used to calculate the test accuracies and the evaluation function is called in both the cases. The results are recoded. The SVM is first configured with a Linear Kernel and then a polynomial kernel is used to compare the differneces and evaluate the best model.

SVM Parameters:class sklearn.svm.SVC(\*, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True,probability=False, tol=0.001, cache\_size=200, class\_weight=None, verbose=False,max\_iter=-1,decision\_function\_shape='ovr',break\_ties=False, random\_state=None).

The parameter for the kernel used are “linear” and “poly”.

* K-Nearest Neighbors: The K-Nearest Neighbors algorithm is used with different number of Neighbors to evaluate the optimal K value which was found to be in the proximity of 500.

The KNN algorithm is also validated on the Validation data Set and on the Test DataSet. KNN is configured with 20 Neighbours and the validation and the test runs are performed and the evaluation results are obtained. The classification report is also generated for comparision.

class sklearn.neighbors.KNeighborsClassifier(n\_neighbors=5, \*, weights='uniform', algorithm='auto', leaf\_size=30, p=2, metric='minkowski', metric\_params=None, n\_jobs=None)

Here, n\_neighbors is specified to be ‘20’, ‘500’ and ‘2000’ and compared.

* Artificial Neural Networks: NN is configured with 3 layers and the validation and the test runs are performed and the evaluation results are obtained. The evaluation is calculated on the validation and test data set and the ReLU activation efficiency is calculated with 10 Epochs.

class sklearn.neural\_network.MLPClassifier ( hidden\_layer\_sizes = (100,), activation = 'relu', \* , solver='adam' , alpha=0.0001, batch\_size='auto' , learning\_rate = 'constant' , learning\_rate\_init = 0.001, power\_t = 0.5, max\_iter = 200, shuffle = True, random\_state = None, tol = 0.0001, verbose = False, warm\_start = False, momentum=0.9, nesterovs\_momentum = True, early\_stopping = False, validation\_fraction = 0.1, beta\_1 = 0.9, beta\_2 = 0.999, epsilon = 1e-08, n\_iter\_no\_change = 10, max\_fun=15000)

Here we specify the parameter activation is “relu”, the solver is “adam” and 3 layers specified and number of epochs = “10”

* Extra Trees Classifier: The preprocessing notebook is run to prepare the data, the test and train datasets are used to validate and test the model and the evaluation of efficiency is noted.

class sklearn.ensemble.ExtraTreesClassifier(n\_estimators=100, \*, criterion='gini', max\_depth=None, min\_samples\_split=2, min\_samples\_leaf=1, min\_weight\_fraction\_leaf=0.0, max\_features='auto', max\_leaf\_nodes=None, min\_impurity\_decrease=0.0, bootstrap=False, oob\_score=False, n\_jobs=None, random\_state=None, verbose=0, warm\_start=False, class\_weight=None, ccp\_alpha=0.0, max\_samples=None)

In this model, we specify the parameter n\_estimators=100 and criterion as “gini”.The dataset is then dimensionally reduced based on the best features using Principal Component Analysis where the dimensions are reduced to 10 best features of the dataset. The resulting PCA datasets are used for further analysis.Feature Selection is done to assess the impact of the features of the dataset. classsklearn.decomposition.PCA(n\_components=None, \*, copy=True, whiten=False, svd\_solver='auto', tol=0.0, iterated\_power='auto', random\_state=None)

Here for the Dimensionality Reduction we specify the parameter n\_components = 10.

* Random Forest Classifier: The Preprocessing Notebook is called to prepare the datasets, and the validation and test datasets are used to evaluate the prediction accuracies. And the evaluation results are noted for comparision.

class sklearn.ensemble.RandomForestClassifier(n\_estimators=100, \*, criterion='gini', max\_depth=None, min\_samples\_split=2, min\_samples\_leaf=1, min\_weight\_fraction\_leaf=0.0, max\_features='auto', max\_leaf\_nodes=None, min\_impurity\_decrease=0.0, bootstrap=True, oob\_score=False, n\_jobs=None, random\_state=None, verbose=0, warm\_start=False, class\_weight=None, ccp\_alpha=0.0, max\_samples=None)

Here we specify the parameters as ccp\_alpha=0, criterion='entropy', max\_depth=10, max\_features='log2',max\_leaf\_nodes=100,max\_samples=10000,min\_impurity\_decrease=0,min\_samples\_leaf=10,min\_weight\_fraction\_leaf=0,n\_estimators=50, n\_jobs=5, oob\_score=True, random\_state=10, warm\_start=True

The PCA.ipynb file is run to apply Dimensionaltiy reduction on the data sets with n\_components set as 10. The PCA datasets areused to analyze the performance of Random Forest Classifier and the performance evaluation results are recorded.

The PCA output data sets are in the shape of:

test PCA has shape: (22544, 10)

train PCA has shape: (125973, 10)

classsklearn.decomposition.PCA(n\_components=None, \*, copy=True, whiten=False, svd\_solver='auto', tol=0.0, iterated\_power='auto', random\_state=None)

Here for the Dimensionality Reduction we specify the parameter n\_components = 10.

RESULTS OF VARIOUS MODELS

|  |  |
| --- | --- |
| Model | Accuracy |
| LSVM Validation | 98.9971 |
| LSVM Test | 81.2056 |
| QSVM Validation | 99.4681 |
| QSVM Test | 81.0016 |
| KNN 20 Validation | 99.3332 |
| KNN 20 Test | 77.0981 |
| KNN 500 Validation | 96.2564 |
| KNN 500 Test | 79.1386 |
| KNN 2000 Validation | 93.9004 |
| KNN 2000 Test | 75.5145 |
| NN Validation | 98.4957 |
| NN Test | 77.3465 |
| NN ReLU | 56.9242 |
| RF Validation | 98.876 |
| RF Test | 78.2692 |
| RF PCA Validation | 98.6347 |
| RF PCA Test | 87.5222 |
| ETC Validation | 99.9206 |
| ETC Test | 79.5422 |
| ETC PCA Validation | 99.7523 |
| ETC PCA Test | 82.9356 |

D. Solution

The evaluation section showed that the Random Forest Classifier showed the most promising solution that can be applied to the Intrusion Detection System, Once the model is trained on the Train Dataset, the trained model can be used to predict the incoming packet traffic and provide real time protection against cyber-attack. The model can be further improved by training on much diverse datasets and improving the accuracy of the algorithm.

In this study, we have implemented 5 types of machine learning classifiers such as Random forest, KNN, SVM, MLP and ETC. The experiment was conducted on well-known dataset NSL KDD dataset. The dataset contains normal and different kinds of attack types. Out of 5 classifications techniques, Random Forest performed well with an accuracy of 99.34% on train data and 86.17% on test data. Also, we have done feature preprocessing technique in which we have dropped some features which have low importance.

The below flowchart depicts the proposed algorithm for the Intrusion Detection System

KDD Cup DataSet

Data PreProcessing

Random Forest Classifier

fit()

Trained Model

Prediction

model.predict()

Figure 8: Proposed Solution Algorithm

The Proposed Solution implements an Intrusion Detection System that has Random Forest Classifier as its core, the model is trained on an extensive and updated dataset. The trained model predicts the incoming packet as either attack or normal and provides real time protection of the network by allowing the normal packets to communicate and raise an alert when an intrusion is detected and withholding the connection with the sender. The model may also be configured to recursively train itself based on the predicted outcomes in a supervised manner.

V. PROGRAM CODE

The Program Codes can be found here:

https://github.com/vkvinay580/IDS-with-ML-DL

VI. CONCLUSION REFERENCES

[1]Gayathri Rajakumaran1 · Neelanarayanan Venkataraman1.” Raghava Rao Mukkamala2. “Denial of Service Attack Prediction Using Gradient Descent Algorithm”. Received: 10 June 2019 / Accepted: 4 October 2019 © Springer Nature Singapore Pte Ltd 2019.

[2] C.Gong,K.Sarac,A more practical approach for single-packet I traceback using packet logging and marking, IEEETrans. Parallel Distrib Syst 19(10)(2008)

[3]Aroosh Amjad, Tahir Alyas\*, Umer Farooq, Muhammad Arslan Tariq. “Detection and mitigation of DDoS attack in cloud computing using machine learning algorithm”.

[4] Ahmad Sanmorino. “A study for DDOS attack classification method” 1st International Conference on Advance and Scientific Innovation (ICASI).

[5] Fadil, A.; Riadi, I.; Aji, S. Review of detection DDoS attack detection using Naïve Bayes classifier for network forensics. Bull. Electr. Eng. Inform. 2017

[6] Shravan K, Shridhar Allagi and Rashmi Rachh, An Approach for Securing Big Data Environment using Machine Learning for Dynamic Processing of Portable Executable in Network, International Journal of Advanced Research in Engineering and Technology, 12(1), 2021

[7] Yuanyuan Sun 1, 2, 3a, Yongming Wang 1, 2b, Lili Guo 3c, Zhongsong Ma3, Shan Jin3 and Huiping Wang. “Researching on Multiple Machine Learning for Anomaly Detection”.

[8] J.Li;Y.Liu;L.Gu, DDoS attack detection based on neural network,in: Proceedings of the 2nd International Symposium on Aware Computing (ISAC),Tainan,1–4 Nov.2010

[9] Yong, B.; Wei, W.; Li, K.C.; Shen, J.; Zhou, Q.; Wozniak, M.; Połap, D.; Damaševičius, R. Ensemble machine learning approaches for webshell detection in Internet of things environments. Trans. Emerg. Telecommun. Technol. 2020

[10] Hemalatha, J.; Roseline, S.A.; Geetha, S.; Kadry, S.; Damaševiˇcius, R. An efficient DenseNet-based deep learning model for malware detection. Entropy 2021

[11] Idhammad M, Afdel K, Belouch M (2017) DoS detection methodbased on artificial neural networks. Int J Adv Comput Sci Appl, 2017

[12]Bhuvaneswari Amma N G, Selvakumar S (2020) A statistical class center based triangle area vector method for detection of denial of service attacks. Cluster Computing

[13] N. Martins, J. M. Cruz, T. Cruz and P. H. Abreu, "Adversarial machine learning applied to intrusion and malware scenarios: A systematic review", IEEE Access, vol. 8, 2020

[14] Zhang, J.; Li, Y.; Xiao, W.; Zhang, Z. Non-iterative and Fast Deep Learning: Multilayer Extreme Learning Machines. J. Franklin Inst. 2020, 357, 8925–8955.

[15] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, ‘‘An adaptive ensemble machine learning model for intrusion detection,’’ IEEE Access, vol. 7, 2019

[16] Larriva-Novo, X.A.; Vega-Barbas, M.; Villagra, V.A.; Sanz Rodrigo, M. Evaluation of Cybersecurity Data Set Characteristics for Their Applicability to Neural Networks Algorithms Detecting Cybersecurity Anomalies, 2020.

[17] G. Ajeetha and G. M. Priya, "Machine Learning Based DDoS Attack Detection," in 2019 Innovations in Power and Advanced Computing Technologies (i-PACT), 2019, vol. 1: IEEE

[18] Bendale, A.; Boult, T.E. Towards open set deep networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway Township, NJ, USA, 2016;.

[19] Sabeel, U.; Heydari, S.S.; Mohanka, H.; Bendhaou, Y.; Elgazzar, K.; El-Khatib, K. Evaluation of deep learning in detecting unknown network attacks. In Proceedings of the 2019 International Conference on Smart Applications, Communications and Networking, Sharm El Sheik, Egypt, 2019

[20] Jie-Hao, C.; Feng-Jiao, C.; Zhang. (2012) “DDoS defense system with test and neural network”. IEEE International Conference on Granular Computing (GrC), 11-13 Aug. 2012, Hangzhou, China,

[21] V.Akilandeswari; S.M.Shalinie, Probabilistic neural network based attack traffic classification, in:Proceedings of the Fourth International Conference on Advanced Computing(ICoAC),Chennai,13-15 Dec 2012.

[22]Leu F.; Pai C. (2010) “Detecting DoS and DDoS Attacks Using Chi -Square”, Fifth Inter-national Conference on Information Assurance and Security (IAS-09), 18-20 August 2009, Xian, pp.225-258 26.

[23]Xu, X. ;Wei, D. ; Zhang, Y. (2011) “Improved Detection Approach for Distributed Denial of Service Attack Based on SVM”. 2011 Third Pacific-Asia Conference on Circuits, Communications and Systems (PACCS) 17-18 July 2011, Wuhan.