International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org
Volume 11, Issue 8, August 2022 **ISSN 2319 – 4847**

Machine Learning Methods for Distributed DoS Attacks: Traffic Generation, Collection and Classification in an SDN Environment

Arvind T[1], Dr. K Radhika[2]

[1]Department of CSE, UCE, OU, Hyderabad, India.

[2]Professor, Department of IT, CBIT, Hyderabad, India

ABSTRACT

In an SDN environment, one of the most serious risks that might emerge is a distributed denial of service (DDoS) attack. It is a form of attack in which a large number of bogus packets are delivered into the network from various sources in order to drain network resources. In this research, we employed an SDN environment to generate and collect DDoS and normal traffic.SYN flooding, UDP flooding, and ICMP flooding attacks were produced using the hping3 tool, while normal traffic was produced using the ping and iperf. The produced traffic was assembled into a dataset and classified using machine learning models, namely Naive Bayes, Logistic Regression, SVM, KNN and Gradient Boost models. The evaluation's findings demonstrate that KNN and Gradient Boost classifiers outperform other models in terms of accuracy, Precision, Recall, F1-score, AUC-ROC, training time and testing time.

Keywords: SDN, DDoS attacks, Mininet, Iperf, hping3, Machine Learning Models.

1. INTRODUCTION

SDN is a framework that separates the control plane from the data plane, making network administration easier [28]. This concentrated potential of the SDN turns out to be a main reason for its failure [1-2,22]. DDoS is among the most prevalent attacks against the centralized controller in which a significant number of bogus packets are fed into the network by a group of hacked hosts using techniques such as spoofing, etc., resulting in the loss of the SDN architecture. These assaults are categorized into three types, namely volume-based, protocol-based and application-based.

Volume-based: The main goal of these attack types is to deprive the network bandwidth in order to prevent the provision of proper services [3].The main attacks included in this category are ICMP floods, UDP floods, and reflection and amplification assaults.

UDP Flooding attack: Attacks using UDP flooding send large numbers of User Datagram Protocol (UDP) packets to a server's arbitrary ports in an effort to overwhelm and deplete its resources [4].

ICMP Flooding attack: It shatters the targeted device with an ICMP echo-request packet, making it unavailable to regular traffic.

Amplification attacks: An example of an amplification assault is DNS Amplification, which is launched from DNS systems that are open to the public. In order to search the DNS server for "any" resource record, the attacker replaces the victim's source IP with his or her target IP. The server then notifies the receivers of the whole zone information in a response.

NTP Amplification: In this case, an attacker repeatedly requests the NTP server's "get monlist" information while posing as a trusted source IP address. The NTP server then acknowledges and provides a list of the latest 600 hosts to connect to it.

Protocol-based: these attacks' main goal is to use up all available server resources. SYN flooding, the Ping of Death, Smurf assaults, etc. are some of its features.

Attacks like the SYN Flood, which take advantage of the 3-way handshake in the TCP connection sequence, may be considered DDoS attacks. The attacker often sends numerous SYN packets to each port on the targeted server using a bogus IP address [3]. The oblivious server reacts by delivering a SYN-ACK message to each open port.

Ping of Death: occurs when an attacker sends too many ICMP datagram's to the target node, causing it to hang, restart, or even crash.
Smurf attack: To make sure that every host in the network responds to an ICMP request, it sends ICMP packets with the victim's source IP spoofing. As a result, the victim's node sees high traffic.
Application Level-based: Application Level-based Layer 7-based assaults, like HTTP-based ones, are low volumetric attacks. These are the most challenging assaults because they are hard to identify and counter.
Slowloris attack: is a denial-of-service tactic that enables the attacker to establish and maintain several HTTP connections with the target server at once, ultimately shutting it down.
The remainder of the document is divided into the following segments: The second segment comprises the literature survey, while the third segment contains the implementation environments and methodology. Segment four covers machine learning models and assessment metrics. The fifth segment contains the results and discussion, while the sixth segment discusses the conclusions and future scope.

2. LITERATURE SURVEY

DDOs attacks may be detected using a variety of approaches, including statistical, machine learning, deep learning techniques, and others.

An entropy-based DDoS detection approach was put forward by Mousavi et al. [9] to recognise a reduction in randomness in the flow of packets arriving at controllers to identify an early DDoS assault. When an incoming packet exceeds the window size, entropy is computed to detect a DDoS attack. For the window size, the frequencies for the target IP addresses are recorded. A similar technique was presented by Bhavani K et al. [12]. The authors calculated the mean entropy and the rate of percentage decline, which aids in early detection of DDoS assaults before the controller gets overwhelmed. In a single controller context, the proposed approach is successful in detecting DDoS assaults. Cong Fan et al. [13] presented a fusion entropy approach to evaluate network traffic randomness in the SDN network environment to identify DDoS attacks. The traffic was generated using the Scapy programme. The suggested method efficiently identifies the attacks by having an entropy value that is 91.25 percent lower than the entropy of the normal traffic flow. The suggested strategy, on the other hand, employs a fixed threshold, which decreases detection rates while increasing false positive rates. Furthermore, there is a little information regarding the dataset and the attributes utilized to identify DDoS attacks.

In their SVM model, Kshira et al. [27] suggested using KPCA to reduce the size of the feature vectors and GA to optimise different SVM parameters. An enhanced kernel function (N-RBF) is created in order to reduce the noise brought on by feature discrepancies. According to the experimental findings, the recommended model offers greater generality and more reliable categorization than single-SVM. Additionally, the controller might utilize the suggested model to develop security rules that would forbid further attacker attempts. Polat et al. [14] investigated the efficacy of four ML models with and without feature selection to detect DDoS attacks. The authors used four feature selection methods: Filter, wrapper and embedded-based to extract 12 most relevant features on the generated dataset. Following feature selection, each machine learning algorithm chooses 6 to 10 essential features, and training the model with these key characteristics rather than all 12 features often increases detection accuracy.KNN employing wrapper-based selection is the most accurate model, with an accuracy of 98.3 percent when trained with six key features and 95.67 percent when trained with twelve. According to Dong et al.[15,18], a flow is believed to be a vector containing values for length, duration, size, and rate. The square root of the difference between each flow characteristic is used to compute the distance between two flows. A weight value is added to the KNN model to further highlight the significance of a neighbour flow; a closer neighbour has a greater weight value, which strengthens its impact on the prediction. Nisharani and colleagues [21] evaluated three machine learning models: SVM, Naive Bayes, and Neural Network. In the experimental configuration, Mininet and the Ryu controller were employed. When compared to other approaches, the experimental findings revealed that the SVM model had the greatest accuracy, recall, and precision.

Obaid et al. [7] detected and mitigated DDoS attacks in an SDN network using a variety of machine learning models, including J48, RF, SVM, and KNN.The experiment was carried out on Ubuntu with the help of mininet and Ryu controllers, and Weka was utilised to train and evaluate the machine learning models. Normal traffic was generated using Tshark, and attack traffic was generated using hgping3.According to the experimental data, J48 outperformed the other models. For better traffic classification and an enhanced History-based IP Filtering system (eHIPF) for faster and more accurate attack detection, Trung et al. [28] created a hybrid machine learning model based on SVM and SOM algorithms.

Service function chaining was used during the testing on an SDN-based cloud. The results of the experiment revealed that the proposed strategy outperformed the alternative methods.

Fauzi D et al. [16] established a solution based on ML models, comprising SVM, employing linear and radial basis function kernels, KNN, DT, RF, MLP, and Gaussian NB. The dataset was developed using port statistics. The results of the evaluation revealed that the SVM model outperforms the other models. Deepa v et al. [19] devised an ensemble-based DDoS detection method that comprised NB, KNN, SVM, and Self-organizing maps (SOM), KNN-SOM, NB-SOM, and SVM-SOM. The authors employed the CAIDA 2016 dataset, and the models were assessed for accuracy, detection rate, and false alarm rate. The proposed models outperform the other models. Wenchao et al. [26] use the recursive feature removal approach and suggested a detection method based on LeNet-5 CNN to choose the most important 49 characteristics. To cut computational costs, they eliminate the first pooling layer and the last completely linked layer from their architecture. The testing findings suggest that this strategy identifies assaults more accurately. Abhiroop et al. [10] detected flow-table overflow threats in the SDN data plane using three distinct machine learning methods: SVM, NB, and NN. To extract features from open flow switches and produce training data, the open flow protocol was used. Scapy generates three forms of flood traffic: TCP, UDP, and ICMP. When employing the five features used in machine learning techniques, the findings demonstrate that the SVM has a lower accuracy rate than the other classifiers.

A novel technique for visualizing network data using Convolution Neural Networks (CNN) and graphical heat maps was proposed by McCullough and colleagues [23].The obtained results are compared to the models of Long Short-Term Memory (LSTM) and SVM.The results show that botnet-based DDoS assaults may be accurately identified when network traffic is investigated using graphical heat maps by CNN. DDoS attacks are identified via packet analysis by Hu et al. [20].SNORT notifies the network administrator of an attack when it finds it. The SNORT rule could be customised to suit the user's requirements. The SNORT has a drawback that genuine transmissions may result in false alaram s. SNORT was used by Zohaib H et al. [17] to develop a DDoS intrusion detection system. They have built Snort rules to detect DDoS attacks; however the model establishes more false alarms since they restrict certain valid requests.

3. IMPLEMENTATION ENVIRONMENT AND METHODOLOGY

Mininet[11], Ryu controller, Python, and Jupyter notebook were installed on Ubuntu 20.04 LTS for the experiments. The iperf and ping programmes were used to generate regular traffic, whereas the hping3 utility was used to generate attack traffic.

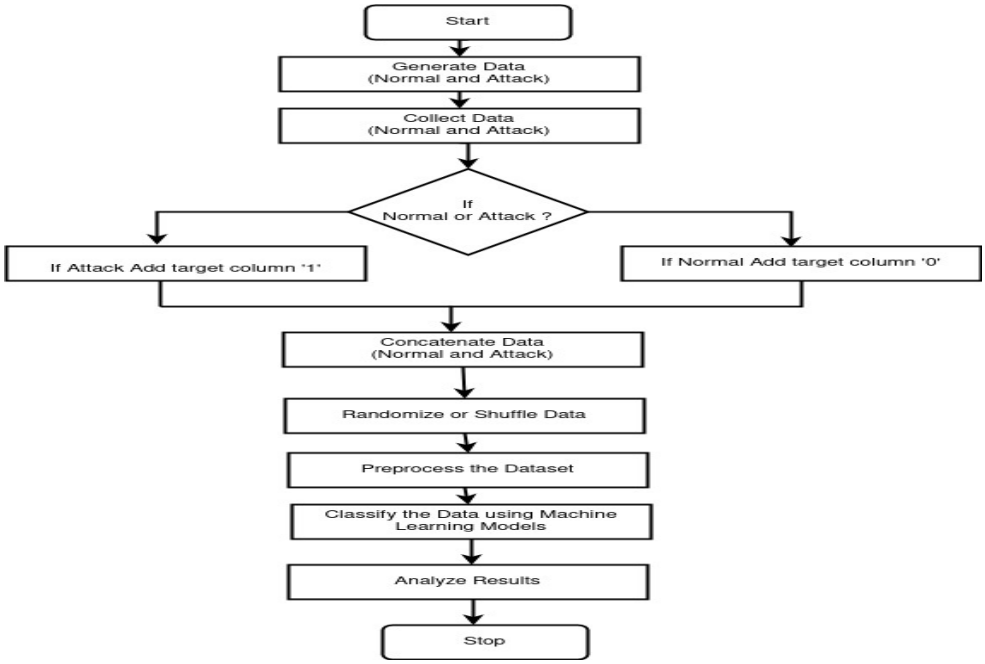

Figure 1 Methodology

3.1 Traffic Generation and Collection

The process involves the generation of the normal and DDoS traffic (ICMP, UDP and TCP floods) using iperf, ping and hping3 tools in a python script executed under mininet emulator. Flow based features were extracted for both Normal and attack traffic and collected into separate datasets in CSV format. Attack traffic was captured for about 20 minutes and normal traffic was captured for about 16 hours in order to balance the normal and attack traffic samples [1, 5-8]. The collected dataset contains 24 features such as timestamp, datapath_id, flow_id, src_ip, tp_src, dst_ip, tp_dst, ip_proto, icmp_code, icmp_type, flow_dur_sec, flow_dur_nsec, idle_tout, hard_tout, flags, byt_count, pac_count, in_port, pac_count_per_sec, byt_count_per_sec, pac_count_per_nsec, byt_count_per_nsec. The total number of samples in the final dataset is 201660, which comprises 69388 normal samples and 132272 attack samples.

Table 1: Dataset Traffic Distribution

Traffic	ICMP	UDP	TCP	Total
Normal	31464	9948	27976	69388
Attack	44072	36604	51596	132272
Total	75536	46552	79572	201660

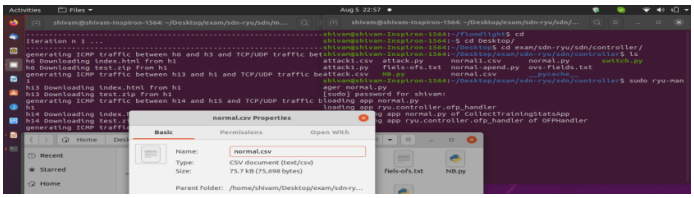

Figure 2 Normal traffic generation process

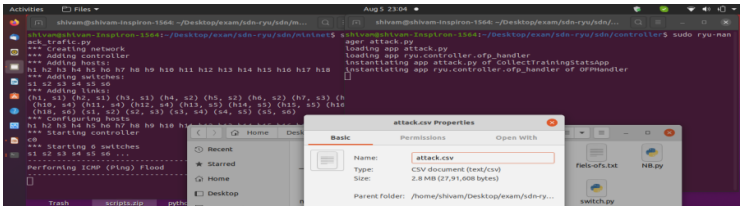

Figure 3 Attack traffic generation process

3.2 Adding the target column for normal and attack datasets

The target column was added to the normal and attack datasets using shell scripts. The target column was added to each row of the normal and attack datasets using the Linux awk utility. In the normal dataset, the target column was set to '0,' but in the attack dataset, the target column was set to '1'.

3.3 Concatenation of datasets

To concatenate datasets, a Linux shell script was utilized. The script concatenates the normal and attack datasets using the Linux cat function.

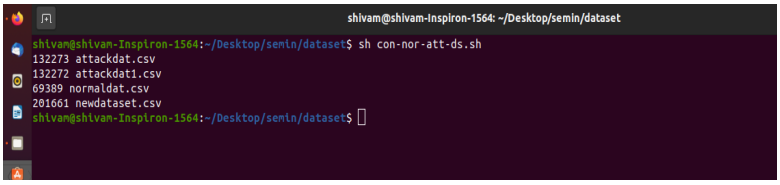

Figure 4 Datasets concatenation

3.4 Randomizing dataset rows

The shell script was used to randomize the rows of the dataset. To shuffle the dataset rows, the Linux shuf tool was utilized.

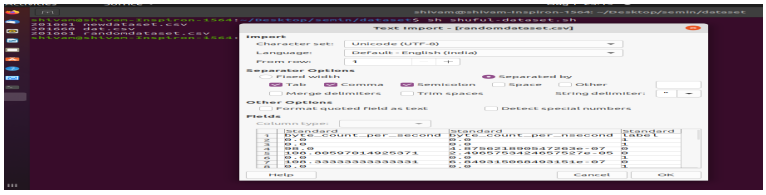

Figure 5 Shuffle dataset rows

3.5 Preprocess the dataset

The dataset was preprocessed before being put into ML models in order to turn it into a format that the ML models could understand. For example, src_ip and dst_ip are in dotted decimal notation and must be converted to integers before the model can comprehend and forecast them.

4. ML CLASSIFICATION AND EVALUATION METRICS

Finally, machine learning methods were used to classify the dataset's traffic. Among the machine learning models investigated for the paper are Logistic Regression, Naive Bayes, KNN, SVM, and Gradient Boost. The aforementioned models' accuracy, precision, recall, F1, AUC-ROC, training and testing durations were all investigated.

5. RESULTS AND DISCUSSION

Figure 8 depicts the accuracy, precision, recall, and f1 comparison of the models, while the table depicts the evaluation metric measure of the models. According to the results, the Gradient Boost and KNN models give approximately the same accuracy, precision, recall, and f1 score; however, the KNN model requires somewhat longer testing. The SVM model had the longest training and testing times, whilst the LR model had the lowest accuracy when compared to the other models.

Table 2: Evaluation-metric comparison

S.no	Model	Acc	Train Time	Test Time	Precision	Recall	F1	AUC
1	LR	0.3404	1.520	0.037	0.00	0.00	0.00	0.5
2	KNN	0.9993	1.738	0.999	0.99	0.99	0.99	0.9996
3	NB	0.7170	0.155	0.045	1.00	0.57	0.73	0.7834
4	GB	0.9999	143.540	0.299	1.00	0.99	0.99	0.9999
5	SVM	0.6595	21580.977	446.587	0.66	1.00	0.79	0.7784

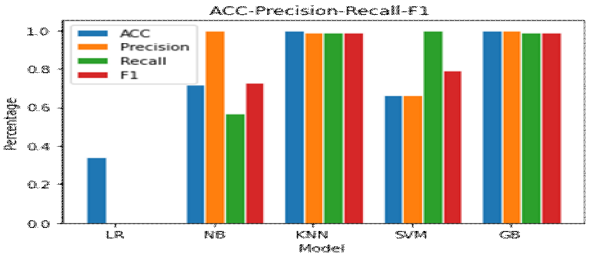

Figure 8 Acc-precison-recall-f1 comparisons

The graph in figure 9 demonstrates that the NB, LR, and KNN models have the shortest training times, whereas the SVM requires more time to train than the other models.

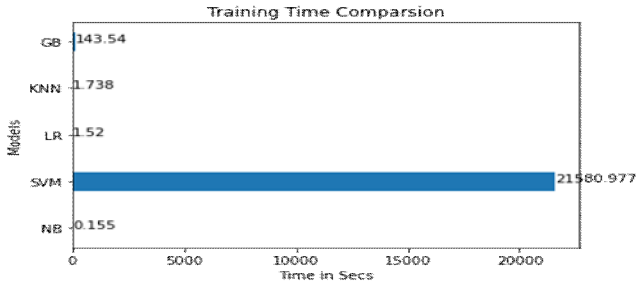

Figure 9 Training time comparison

The graph in fig. 10 demonstrates that, when compared to the other models, the testing time for the SVM model is the longest, whilst the testing times for the LR, NB, and Gradient Boost models are nearly identical.

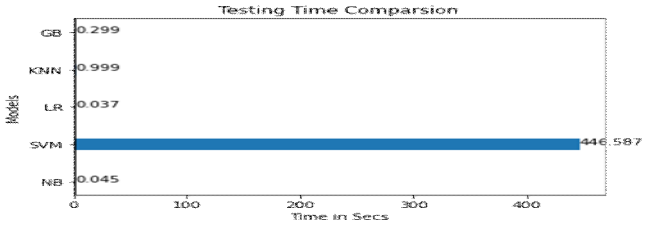

Figure 10 Testing time comparison

The graph in figure 11 demonstrates that the Gradient Boost and KNN models have AUCs that are practically equal to 1, while the LR model has the lowest AUC.

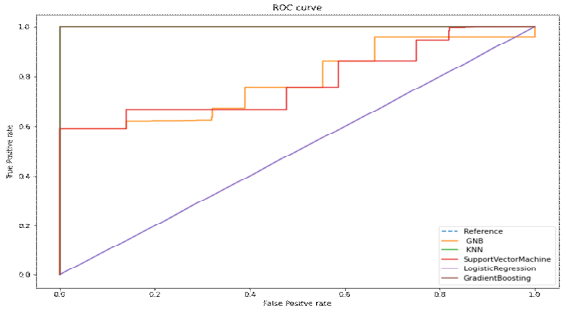

Figure 11 AUC-ROC comparisons

6. CONCLUSION AND FUTURE SCOPE

The experimental assessment revealed that Gradient Boost and KNN classifiers outperform other classifiers. The performance of the Gradient Boost and KNN models is nearly identical across most criteria, whereas the performance of the LR model is the lowest when compared to the other models. We will expand on this work in the future to create an efficient machine learning approach for detecting and mitigating DDoS assaults.

References

[1] Arvind T, Dr.K.Radhika, "An SDN Based DDoS Traffic Generation, Collection and Classification Using Machine Learning Techniques ", International Conference on Advanced Engineering Optimization Through Intelligent Techniques (AEOTIT),Sardar Vallabhbhai National Institute of Technology (SVNIT), 28-30 January 2022.

[2] Rochak S, Mayank D, Virender R,"Software-defined Networking based DDoS Defense Mechanisms", ACM Comput.Surv.52, 2, Article 28, https://doi.org/10.1145/3301614, 2019.

[3] Sarvan Ali, Maria Khalid A, Safi F, Muhammed AK,Abudullah A, Imdadullah K,"Detecting DDoS Attack on SDN Due to Vulnerabilities in OpenFlow", International Conference on Advances in the Emerging Computing Technologies (AECT),DOI:10.1109/aect47998.2020.9194211,2020.

[4] https://www.cloudflare.com/en-in/learning/ddos/udp-flood-ddos-attack [Accessed: Sept. 12, 2021].

[5] MS Elsayed,NAL Khac, AD Jurcut,"InSDN: A novel SDN intrusion dataset", IEEE Access 8, 165263-165284, 2020.

[6] Obaid R,Mohammad Ali G Q,Chang H L, "DDoS Attacks Detection and Mitigation in SDN using Machine Learning",IEEE World Congress on Services,DOI 10.1109/SERVICES.2019.00051,2019.

[7] B Zhang, T Zhang,Z Yu, "DDoS Detection and Prevention Based on Artificial Intelligence Techniques", 3rd IEEE International Conference on Computer and Communications, pp. 1276–1280, 2017.

[8] R Santos,D Souza,W Santo,A Ribeiro,Edward M ,"Machine learning algorithms to detect DDoS attacks in SDN", Concurrency and Computation: Practice and Experience, DOI:10.1002/cpe.5402,2020.

[9] Mousavi S M, Marc S H, "Early Detection of DDOS attack against SDN Controller", International Conference on Computing, Networking and Communications (ICNC),77-81,IEEE,2015.

[10] T Abhiroop, S Babu, B S Manoj," A machine learning approach for detecting DoS attacks in SDN switches",Proc. Twenty Fourth Nat. Conf. Commun., pp. 1–6, 2018.

[11] "Introduction to Mininet", GitHub, availableonline: https://github.com/mininet/mininet/wiki/Introduction-to-Mininet [Accessed: March. 05, 2020].

[12] Bavani K, Ramkumar M P, Emil Selvan G S R ," Statistical Approach Based Detetion of Distributed Denial Service Attack in a Software Defined Network",6th International Conference on Advanced computing & Communication Systems (ICACCS) ,pp.380-385,IEEE,2020.

[13] Cong F, Nitheesh M K, Chen S, Jiang H, Yiwen z, Carlene C, "Detection of DDoS Attacks in Software Defined Networking Using Entropy", Appl. Sci., 12(1), 370,2022.

[14] H. Polat, O. Polat, A. Cetin, "Detecting ddos attacks in software-defined networks through feature selection methods and machine learning models", Sustainability,12 (3), 1035,2020.

[15] Dong S, Sarem M, " DDoS Attack Detection Method Based on Improved KNN With the Degree of DDoS Attack in Software Defined Networks", IEEE Access,8:5039-48,2020.

[16] Fauzi D S S,Christian S K A, "Comparative Analysis of DDoS Detection Techniques Based on Machine Learning in OpenFlow Network",3rd ISRITI,pp.152-157,IEEE ,2020.

[17] Zohaib H, Shahzeb, Roman O, Sergiy G, Abnash Z, Masroor S, "Detection of Distributed Denial of Service Attacks Using Snort Rules in Cloud Computing & Remote Control Systems," IEEE 5th International Conference on Methods and Systems of Navigation and Motion Control(MSNMC), 2018.

[18] Dong S, Abbas K, R. Jain, ''A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments,'' IEEE Access,vol. 7, pp. 80813–80828, 2019.

[19] Deepa V, Sudar K, Deepalakshmi P,"Design of ensemble learning methods for DDoS detection in SDN environment",International Conference on Vision Towards Emerging Trends in Communication and Networking, ViTECoN, IEEE, 2019, pp. 1–6,2019.

[20] Hu, Zhengbing, Sergiy Gnatyuk, Oksana Koval, Viktor Gnatyuk, and Serhii Bondarovets, "Anomaly detection system in secure cloud computing environment," International Journal of Computer Network and Information Security 9, no. 4, p. 10, 2017.

[21] Nisharani M, Narayan D G,Baligar V P, "Detection of Distributed Denial of Service Attacks using Machine Learning Algorithms in Software Defined Networks",IEEE,pp.1366-1371,2017.

[22] Arvind T, Dr.K.Radhika, " Comparative Assessment of SDN Openflow Controllers under Minet Emulation Envronment",Vol.11,Issue 4,pp.081-085,IJETTCS,2022.
[23] McCullough, E.; Iqbal, R.; Katangur, A. "Analysis of Machine Learning Techniques for Lightweight DDoS Attack Detection on IoT Networks", In Forthcoming Networks and Sustainability in the IoT Era. FoNeS-IoT,Springer: Berlin/Heidelberg,Germany, Volume353, pp.96–110, 2021.
[24] Ryu Documentation, availableonline: https://ryu.readthedocs.io/en/latest/getting_started.html.
[25] Jupyter notebook, availableonline: https://jupyter.org/install
[26] Wenchao C,Qiong L, Asif M Q, Wie L, Kehe W, "An adaptive LeNet-5 model for anomaly detection", Inf. Secur. J. Glob.Perspect, Vol.30 (7),19–29, 2021.
[27] Kshira S, Bata K T, Kshira S N, Somula R, Balamurugan B, Manju K, Daniel B," An Evlolutioary SVM model for DDoS Attack Detection in Software Defined Networks', Vol.8,pp.132502-132513, IEEE Access,2020.
[28] Trung V P, Minho P, "Efficient Distributed Denial-of-service Attack Defense in SDN-Based Cloud ",Vol.7,pp.18701-18714, IEEE Access,2019.
[29] RYU SDN Framework Ryubook 1.0 documentation, .Retrieved from https://osrg.github.io/ryu-book/en/html.
[30] Song W, Juan F B, Karina G C, Akram Al-Hourani, Sithamparanathan K, Muhammad R, Giovanni R, "Detecting flooding DDoS attacks in software defined networks using supervised learning techniques" , Engineering Science and Technology, an International Journal, 2022