IJRAR.ORG E-ISSN: 2348-1269, P-ISSN: 2349-5138

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

Detection of DDoS attack in SDN environment using KNN algorithm

Madathi M
Department of Information Technology
National Engineering College, Kovilpatti, Tuticorin District, TamilNadu, India

Harini R
Department of Information Technology
National Engineering College, Kovilpatti, Tuticorin District, TamilNadu, India

Monikaa R
Department of Information Technology
National Engineering College, Kovilpatti, Tuticorin District, TamilNadu, India

Gowthami N
Assistant Professor, Department of Information Technology
National Engineering College, Kovilpatti, Tuticorin District, TamilNadu, India

Abstract - Software-Defined Networking (SDN) is a network architecture which is an optimization of traditional networks. SDN is used for programmatic management which is ideal for high-bandwidth applications. The centralized control being a primary benefit can also be a risk . If the unauthorized user succeeds in gaining access to the controller, he will have complete control of the system. The controller is extremely vulnerable to the Distributed Denial of Service (DDoS) assaults. SDN differs from traditional networks by decoupling the network control and forwarding functions. The control plane and data plane deals with network control and forwarding function respectively. SDN identifies malicious traffic and link failure present in the network. Apart from this flexibility, it is vulnerable to Distributed Denial of Service (DDoS) attacks which halts the whole network. To mitigate this problem, we propose to classify the traffic containing DDoS attacks by using machine learning techniques such as the KNN algorithm. In this project, we enter the features of DDoS attacks into a CSV file and train the algorithm. We use KNN for the classification process and hence we can able to detect the DDoS attack

I - INTRODUCTION

The exponential enhance the use of diverse packages over the world wide web led to a rise in security threats, such as Distributed Denial of Service (DDoS) attacks [1]. The main goal of the DDoS attack is to make an internet carrier inaccessible by consuming resources like bandwidth, memory or CPU of the target system.The DDoS detection problem is a classic problem in the field of intrusion detection systems, therefore, there is exhaustive prior art on the subject. However, DDoS attacks continue to be one of the biggest cyber threats affecting the financial, health, retail, gaming, and political sectors and resulting in financial loss [2], [3]. In 2019 DDoS attack size increased 273%. In addition, 91% of the victims reported that the attack saturated their internet bandwidth. In April2019, the most comprehensive network and application layer attacks were seen with 580 million packets per second (PPS) [2].Another assault lasted for thirteen days and generated 292,000 Requests according to Second (RPS). Additionally, DDoS attack metrics increased by 84% in the last quarter of 2019 [3].

In general, DDoS assaults are divided into corporations of bandwidth depletion assaults and resource depletion assaults [4]. Bandwidth depletion attacks deny service from the target system by flooding the target network with too many packets. Resource exhaustion attacks aim to consume the target system's processing resources by using malformed packets that exploit network protocols. This article examines resource depletion flood type DDoS attacks.

II–RELATED WORKS

Nisha Ahuja A et al.[1] "Automated DDOS attack detection in software defined networking". Proposed to classify normal traffic from DDOS attack traffic using the machine learning technique. The important contribution of this paper is to identify the novel features for DDoS attack detections. New features are saved in the CSV file to create the dataset and machine learning algorithms are trained on the created SDN dataset. Several previous works on DDoS detection have either used a non-SDN dataset or the research data has not been made public. Software Defined Networking (SDN) is the networking architecture defined by the software program. In SDN, the network traffic is managed by software, which leads the traffic between hosts.

Jing Wu, et al. [2] "Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models". In this study, The SDN-based detection systems developed for DDoS attacks were analyzed by using machine learning systems. In the first proposed approach, by analyzing flow data, algorithms with 98.3% accuracy ensure the detection of attacks without discriminating the type of traffic. With 97.7% sensitivity, KNN algorithms can perform this control by facilitating the charge of the controller. With the feature selection methods utilized in the study, initially 12 features were selected and therefore the selected subset of features was trained using classifiers. The number of features selected was determined by the algorithm itself or by the threshold value assigned to the algorithm. By changing this threshold value, different numbers of features are often selected and trained by the classifier. Multiple numbers of functions can change the precision.

Nguyen Ngoc Tuan, et al [3] "A Robust TCP-SYN Flood Mitigation Scheme Using Machine Learning Based on SDN proposed a new TCP-SYN flood attack mitigation in SDN networks using machine learning". They propose a new TCP-SYN flood attack mitigation in SDN networks using machine learning By the usage of a test bed, we put into effect the proposed algorithms, compare their accuracy and cope with the trade-off among the accuracy and capability of the safety device. The effects display that the algorithms can mitigate TCP-SYN Flood assault over 96. A light-weight and fast machine-learning algorithm based on KNN is used to detect and mitigate by tracing back IP sources of attack. Normal traffic is almost not affected. In order for the system to operate in real time, the window size as the measuring duration, is an important factor. They also propose a method to optimize this value based on system performance and traffic inputs. Experimental results from the test bed show that 97% of attack flows are detected and dropped. As for future research, refined methods for DDoS mitigation are being developed based on new data sets collected from ISPs to improve accuracy and the performance of the system.

Wenwen Sun et al [4] "An Improved Method of DDoS Attack Detection for Controller of SDN design a DDoS attack detection system and method for SDN controller". They propose a DDoS attack detection method for SDN controller. The entropy value is used to divide the detection result into normal status and abnormal status. Using the characteristics of centralized control in the SDN network, the OpenFlow switch information is obtained in real time and the 8-tuple elements closely related to DDoS is extracted. The BiLSTM algorithm processes key features of traffic to detect whether a DDoS attack has occurred on the network of abnormal state. The method has the advantages of comprehensively extracting and analyzing the key attributes of the traffic under the SDN architecture, and reducing the overhead of the entire network by setting a threshold. In the SDN environment, the effectiveness of the method is verified.

Sanjeetha R, et al [5] "Mitigation of DDoS attack instigated by compromised switches on SDN controller by analyzing the flow rule request traffic". Proposed to demonstrate how a DDoS attack can be initiated on an SDN controller by the compromised switches whose idle and hard timeout values are manipulated to send repeated flow table entry requests to the controller. Moreover, a solution is also proposed to detect such an attack within the second repeated request and to mitigate it immediately. This solution is very effective because the attack is detected instantly instead of calculating a threshold based on the number of 11 flow entry requests on the SDN controller by compromising the switches to send many flow table requests so that its resources are running out. We additionally find a solution for mitigating the identical via way of means of detecting the DDoS assault and mitigating it immediately. The paper can be extended to use other mathematical models to identify the deviation of switch flow requirements and a comparison can be performed with each other to check its accuracy.

Jing Wu, et al. [6] "A new framework for DDOS attack detection and defense in SDN environment". They deploy a DDoS detection trigger mechanism which is apply on the data plane. This mechanism uses the CPU resources of switches to count the sending rate of packeting messages on switches. Once it detects the possible presence of a DDoS attack, it alerts the controller to detect the anomaly so that the controller can react quickly to the detection trigger mechanism. In the future, they will try to exploit the technology of streaming computing to reduce the burden of a single controller to ensure the efficiency of DDoS detection and network quality under large-scale network traffic.The burden of the controller increases and the efficiency of DDoS detection decreases when the network is under larger-scale network traffic will flow by changing its several features. In the SDN simulation environment we established, the host h1 sends DDoS attack packets to the network at the 10th second. At the same time, the detection trigger mechanism on the data plane finds that there is a suspicious DDoS attack flow on the switch at the beginning of the DDoS attack, that is, at the 10th second, and alerts the controller. After receiving the warning, the controller extracts the traffic features passed through the switch and detects it. After detecting whether it is a DDoS attack flow within a few seconds, the controller adopts corresponding attack defense measures and issues corresponding commands to the switches Therefore, the indicators gradually returned to the normal level after the DDO attack. Although SDN has many advantages, it also faces the threat of DDoS attacks, the most common security threat in the network.

III - SYSTEM IMPLEMENTATION

PROPOSED SYSTEM

In the proposed system we use a Machine learning model which is called as K-Nearest Neighbour which is suitable for Accurate Prediction.

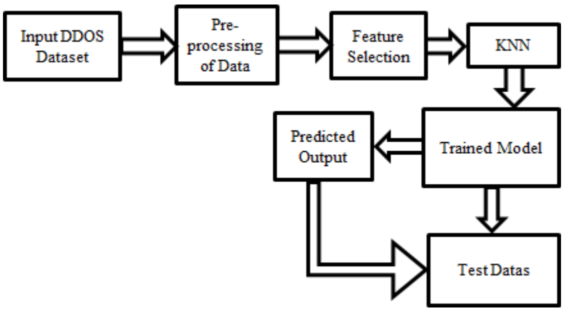


Fig .1 Proposed System Architecture

MODULE DESCRIPTION

At first, the data are collected from the input dataset by using the panda’s library. Then we pre-process the data by dropping null values, then we make feature selection by selecting input features for feeding it in the KNN module, we design a KNN model which can able to give high accuracy, the extracted features are inserted into the KNN model and the machine gets trained. After training, we predict DDoS attack data’s by feeding test data’s into the model.

DATA PRE-PROCESSING

At first, the dataset is fetched by using the panda’s library and then we save the data inside a pandas data frame, At first, this dataset consists of lots of null values, then we drop all the null values because our Machine learning model cannot able process null values.

MACHINE LEARNING

Machine learning is a part of man-made reasoning (AI) and software engineering which centers around the utilization of information and calculations to impersonate the way that people learn, continuously further developing its exactness. Machine learning is an important part of the developing field of information science. Using measurable techniques, calculations are prepared to make orders or forecasts, revealing key bits of knowledge inside information mining projects. These bits of knowledge in this way drive decision-production inside applications and organizations, in a perfect world affecting key development measurements.

As large information proceeds to expand and develop, the market demand for information researchers will increment, expecting them to aid the distinguishing proof of the most applicable business questions and consequently the information to respond to them.

K-NEAREST NEIGHBOR

- K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on the Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories.
- The KNN algorithm stores all available data and ranks a new data point based on similarity.
- This means that when new data is displayed, it can easily be classified into a good suite category using the KNN algorithm.
- K-NN is a non-parametric algorithm, which means it does not make any assumptions on underlying data.
- It is also called lazy learning algorithm because it does not immediately learn from the training set but stores the dataset and performs an action on the dataset while classifying.
- The KNN algorithm in the training phase simply stores the data set and when it receives new data, it categorizes it into a category very similar to the new data**.**
- **Example:** Suppose, we have an image of a creature that looks similar to a cat and dog, but we want to know whether it is a cat or dog, So, for this identification, we can use the KNN algorithm, since it works on a measure of similarity. Our KNN model will find similar features of the new dataset to images of dogs and cats and, based on the most similar features, place it in the cat or dog category.

Why do we need a K-NN Algorithm?

Let us consider two categories i.e.Category A and Category B and we have a new data point x1, so this data point will be in which of these categories. To solve this kind of problem, we need a KNN algorithm. With the help of KNN algorithm, we can easily able to identify the category or class of a particular dataset. Consider the below diagram:

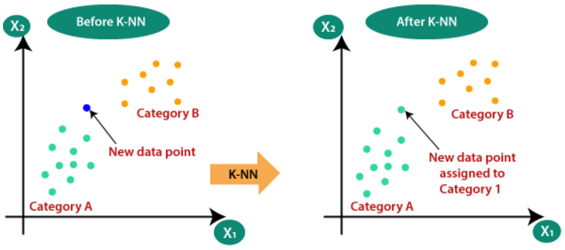


Fig .2 KNN Algorithm

How does K-NN work?

The working of K-NN algorithm can be explained based on the below algorithm

- **Step-1:**First we want to select the number K of the neighbors
- **Step-2:** Calculate the Euclidean distance of the number K of neighbors
- **Step-3:** Take the K nearest neighbors according to the calculated Euclidean distance
- **Step-4**: Among these k neighbors, the number of data points in each category matters
- **Step-5**: Then Assign the new data points to it category that the quantity of the neighbor ismaximum.
- **Step-6:** Our model is ready.

Suppose we have a new data point and we need to enter it into the required category. Consider the below image:

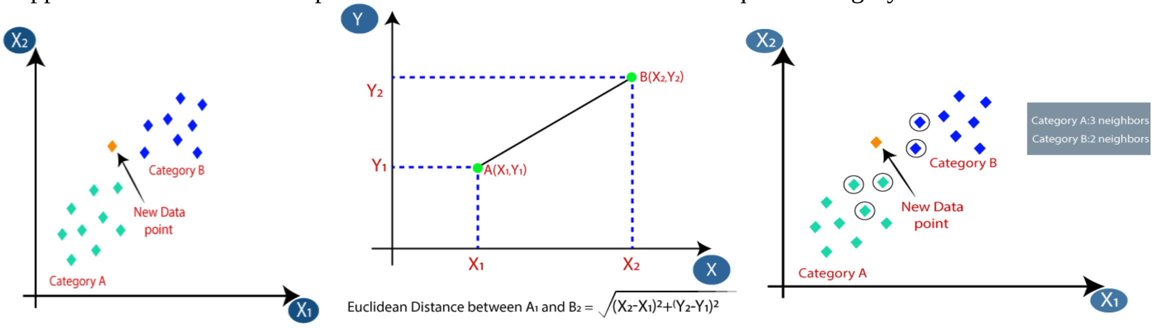
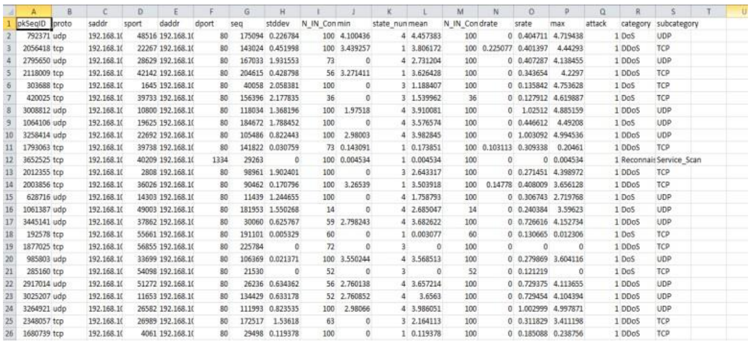


Fig.3 Working of KNN Algorithm

First we will choose the number of neighbors, Here we will choose k=5.Then we will calculate the Euclidean distance between the data points. Euclidean distance is defined as the distance between two points By calculating the Euclidean distance, we got the nearest neighbors, such as three nearest neighbors in category A and two nearest neighbors in category B. As we can see the 3 nearest neighbors are from category A, hence this new data point must belong to category A.

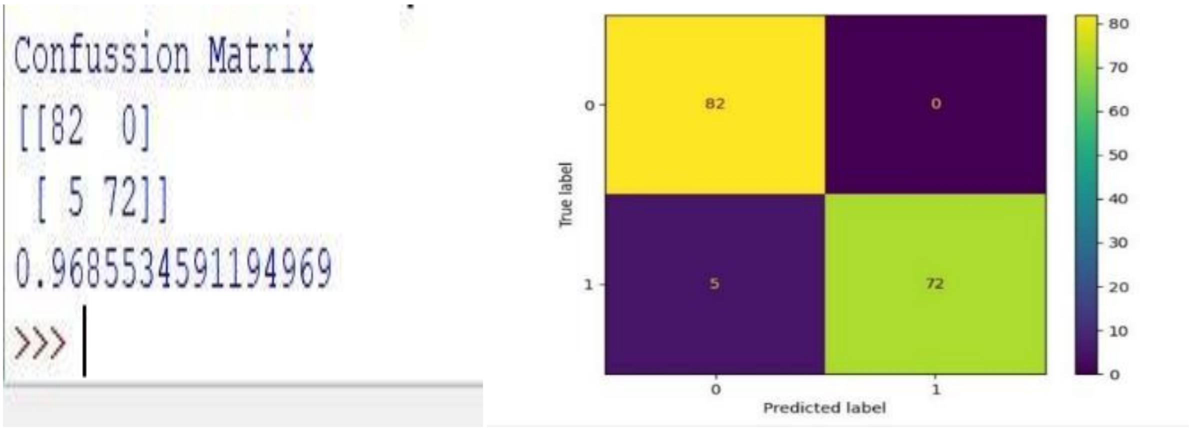
IV – RESULTS& DISCUSSION

The input dataset consists of 17 columns and more than 1000 rows. It is a DDoS detection dataset that is collected from the Kaggle website.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	pkSeqID	proto	saddr	sport	daddr	dport	seq	stddev	N_IN_Con	min	state_nun	mean	N_IN_Con	drate	srate	max	attack	category	subcategory	
2	792371	udp	192.168.1(	48516	192.168.1(	80	175094	0.226784	100	4.100436	4	4.457383	100	0	0.404711	4.719438	1	DoS	UDP	
3	2056418	tcp	192.168.1(	22267	192.168.1(	80	143024	0.451998	100	3.439257	1	3.806172	100	0.225077	0.401397	4.44293	1	DDoS	TCP	
4	2795650	udp	192.168.1(	28629	192.168.1(	80	167033	1.931553	73	0	4	2.731204	100	0	0.407287	4.138455	1	DDoS	UDP	
5	2118009	tcp	192.168.1(	42142	192.168.1(	80	204615	0.428798	56	3.271411	1	3.626428	100	0	0.343654	4.2297	1	DDoS	TCP	
6	303688	tcp	192.168.1(	1645	192.168.1(	80	40058	2.058381	100	0	3	1.188407	100	0	0.135842	4.753628	1	DoS	TCP	
7	420025	tcp	192.168.1(	39733	192.168.1(	80	156396	2.177835	36	0	3	1.539962	36	0	0.127912	4.619887	1	DoS	TCP	
8	3008812	udp	192.168.1(	10800	192.168.1(	80	118034	1.368196	100	1.97518	4	3.910081	100	0	1.02512	4.885159	1	DDoS	UDP	
9	1064106	udp	192.168.1(	19625	192.168.1(	80	184672	1.788452	100	0	4	3.576574	100	0	0.446612	4.49208	1	DoS	UDP	
10	3258414	udp	192.168.1(	22692	192.168.1(	80	105486	0.822443	100	2.98003	4	3.982845	100	0	1.003092	4.994536	1	DDoS	UDP	
11	1793063	tcp	192.168.1(	39738	192.168.1(	80	141822	0.030759	73	0.143091	1	0.173851	100	0.103113	0.309338	0.20461	1	DDoS	TCP	
12	3652525	tcp	192.168.1(	40209	192.168.1(	1334	29263	0	100	0.004534	1	0.004534	100	0	0	0.004534	1	Reconnais	Service_Scan	
13	2012355	tcp	192.168.1(	2808	192.168.1(	80	98961	1.902401	100	0	3	2.643317	100	0	0.271451	4.398972	1	DDoS	TCP	
14	2003856	tcp	192.168.1(	36026	192.168.1(	80	90462	0.170796	100	3.26539	1	3.503918	100	0.14778	0.408009	3.656128	1	DDoS	TCP	
15	628716	udp	192.168.1(	14303	192.168.1(	80	11439	1.244655	100	0	4	1.758793	100	0	0.306743	2.719768	1	DoS	UDP	
16	1061387	udp	192.168.1(	49003	192.168.1(	80	181953	1.550268	14	0	4	2.685047	14	0	0.240384	3.59623	1	DoS	UDP	
17	3445141	udp	192.168.1(	37862	192.168.1(	80	30060	0.625767	59	2.798243	4	3.682622	100	0	0.726616	4.152734	1	DDoS	UDP	
18	192578	tcp	192.168.1(	55661	192.168.1(	80	191101	0.005329	60	0	1	0.003077	60	0	0.130665	0.012306	1	DoS	TCP	
19	1877025	tcp	192.168.1(	56855	192.168.1(	80	225784	0	72	0	3	0	100	0	0	0	1	DDoS	TCP	
20	985803	udp	192.168.1(	33699	192.168.1(	80	106369	0.021371	100	3.550244	4	3.568513	100	0	0.279869	3.604116	1	DoS	UDP	
21	285160	tcp	192.168.1(	54098	192.168.1(	80	21530	0	52	0	3	0	52	0	0.121219	0	1	DoS	TCP	
22	2917014	udp	192.168.1(	51272	192.168.1(	80	26236	0.634362	56	2.760138	4	3.657214	100	0	0.729375	4.113655	1	DDoS	UDP	
23	3025207	udp	192.168.1(	11653	192.168.1(	80	134429	0.633178	52	2.760852	4	3.6563	100	0	0.729454	4.104394	1	DDoS	UDP	
24	3264921	udp	192.168.1(	26582	192.168.1(	80	111993	0.823535	100	2.98066	4	3.986051	100	0	1.002999	4.997871	1	DDoS	UDP	
25	2348057	tcp	192.168.1(	26989	192.168.1(	80	172517	1.53618	63	0	3	2.164113	100	0	0.311829	3.411198	1	DDoS	TCP	
26	1680739	tcp	192.168.1(	4061	192.168.1(	80	29498	0.119378	100	0	1	0.119378	100	0	0.185088	0.238756	1	DDoS	TCP	

Fig.4 Dataset



```
Confussion Matrix
[[82  0]
 [ 5 72]]
0.9685534591194969
>>>
```

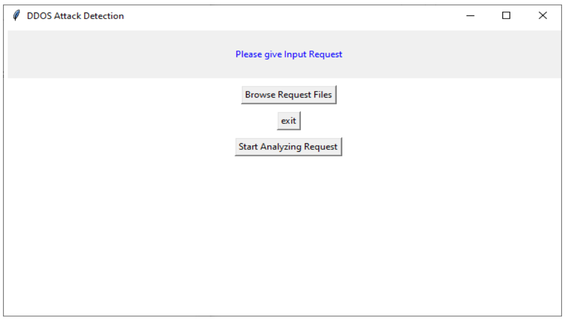
Fig.5 Results of KNN Algorithm

The above fig.5 shows the performance metrics of the KNN Algorithm. It shows accuracy score, precision score. In the KNN algorithm, we got 96% accuracy

A DDoS detection dataset was processed for this study, outliers were identified and eliminated, and a variety of classification techniques KNN . Table 1 shows that when accuracy and precision are taken into account, KNN perform good accuracy.

TABLE 1: Accuracy and Precision

Algorithm	**Accuracy**	**Precision**
KNN	96.9	1.00

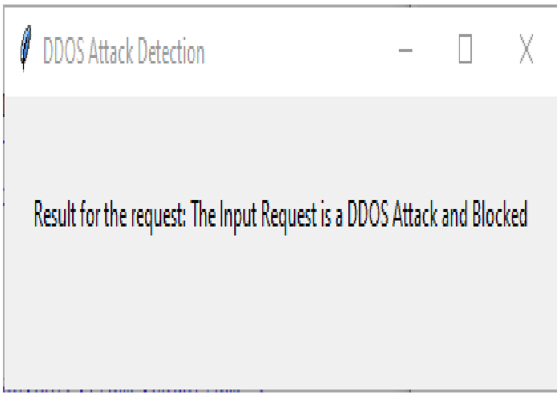


DDOS Attack Detection

Please give Input Request

Fig.6 Test Application

We designed a separate application to test our trained model. Here we give the requested dataset to the machine to predict whether the request is a DDoS Attack or not



DDOS Attack Detection

Result for the request: The Input Request is a DDOS Attack and Blocked

Fig.7 Result

V - CONCLUSION

Eventhough SDN has many advantages, also addresses the threat of DDO, the most common security threat in the network. As an advantage of SDN, the central control also causes the SDN controller more susceptible to the security threats of DDOS attacks. In response to this problem, in this paper, we analyze the detection and defense mechanism of DDoS attacks using KNN machine learning algorithm. Experiments are constructed to prove that the detection methods proposed in this paper can achieve good results. SDN has a controller which controls the networking functions. We will place the trained python code in controller. By sending the DDoS traffic pattern from any hosts, the controller can detect the DDoS attack and preventing the network from halting and thus saving the network. Therefore in future, we will try to create a SDN network using mininet software. Mininet is a network emulator.Mininet creates a network of virtual hosts, switches, controllers, and links. Mininet hosts run standard Linux Network software, and their switches are compatible with the OpenFlow for highly flexible adjustments routing and Software-Defined Networking. By using KNN, we can predict the DDoS attack 96% of accuracy.

REFERENCE

[1] Nisha Ahuja,GauravSingal,Debajyoti Mukhopadhyay, Neeraj Kumar “Automated DDOS attack detection in software defined networking”, Volume 187 ,1 August2021.

[2] Huseyin Polat, Onur Polat and Aydin Cetin “Detecting DDoS Attacks in SoftwareDefined Networks Through Feature Selection Methods and Machine LearningModels”,Volume 12, 29 January 2020.

[3] N. N. Tuan, P. H. Hung, N. D. Nghia, N. Van Tho, T. V. Phan and N. H. Thanh, “A Robust TCP-SYN Flood Mitigation Scheme Using Machine Learning Based on SDN,” International Conference on Information and Communication Technology Convergence,2019,volume 19, pp. 363-368.

[4] W. Sun, Y. Li and S. Guan, “An Improved Method of DDoS Attack Detection for Controller of SDN,” 2019 IEEE 2nd International Conference on Computer and Communication Engineering Technology (CCET), 2019, volume 19, pp. 249-253.

[5] Sanjeetha R, Shikhar Srivastava, Rishab Pokharna,Syed Shafiq,Dr. Anita Kanavalli “Mitigation of DDoS attack instigated bycompromised switches on SDN controller byanalyzing the flow rule request traffic”, volume 7, May 2019, pp 46 – 47.

[6] L.Tan, Y. Pan, J. Wu, J. Zhou, H. Jiang and Y. Deng, “A New Framework for DDoS Attack Detection and Defense in SDN Environment,” in IEEE Access, volume 8, pp. 161908-161919, 2018.

[7] J.A. Pérez-Díaz, I. A. Valdovinos, K. -K. R. Choo and D. Zhu, "A Flexible SDN-Based Architecture for Identifying and Mitigating Low-Rate DDoS Attacks Using Machine Learning," in IEEE Access, volume 8, pp. 155859-155872, 2018.

[8] Y.Xu, H. Sun, F. Xiang and Z. Sun, "Efficient DDoS Detection Based on K-FKNN in Software Defined Networks," in IEEE Access, vol. 7, pp. 160536-160545, 2017.

[9] S. Dong and M. Sarem, "DDoS Attack Detection Method Based on Improved KNNWith the Degree of DDoS Attack in Software-Defined Networks," in IEEE Access, volume 8, pp. 5039-5048, 2017.

[10] Y. Xu, H. Sun, F. Xiang and Z. Sun, "Efficient DDoS Detection Based on K-FKNN in Software Defined Networks," in IEEE Access, vol. 7, pp. 160536-160545, 2016.