IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

DETECTING DDOS ATTACKS IN SOFTWARE-DEFINED NETWORKS THROUGH FEATURE SELECTION METHODS AND MACHINE LEARNING MODELS

R.Evangelin Gladys , Mrs.P.Jasmine Lois Ebenazar MCA..,M.Phil.,
Department of Computer Applications, Sarah Tucker College, Thirunelveli-7.

Abstract: For the easy and flexible management of large scale networks, Software-Defined Networking (SDN) is a strong candidate technology that offers centralisation and programmable interfaces for making complex decisions in a dynamic and seamless manner. On the one hand, there are opportunities for individuals and businesses to build and improve services and applications based on their requirements in the SDN. On the other hand, SDN poses a new array of privacy and security threats, such as Distributed Denial of Service (DDoS) attacks. For detecting and mitigating potential threats, Machine Learning (ML) is an effective approach that has a quick response to anomalies. In this article, we analyse and compare the performance, using different ML techniques, to detect DDoS attacks in SDN, where both experimental datasets and self-generated traffic data are evaluated. Moreover, we propose a simple supervised learning (SL) model to detect flooding DDoS attacks against the SDN controller via the fluctuation of flows. We verify the outcome through simulations and measurements over a real testbed. Our main goal is to find a lightweight SL model to detect DDoS attacks with data and features that can be easily obtained. In this study, DDoS attacks in SDN were detected using machine learning-based models. First, specific features were obtained from SDN for the dataset in normal conditions and under DDoS attack traffic. Then, a new dataset was created using feature selection methods on the existing dataset. Feature selection methods were preferred to simplify the models, facilitate their interpretation, and provide a shorter training time. Both datasets, created with and without feature selection methods, were trained and tested with K-Nearest Neighbors (KNN) classification models. The test results showed that the use of the wrapper feature selection with a KNN classifier achieved the highest accuracy rate Our results show that SL is able to detect DDoS attacks with a single feature. The performance of the analysed SL algorithms is influenced by the size of training set and parameters used. The accuracy of prediction using the same SL model could be entirely different depending on the training set. This project is developed using PYTHON, PyCharm as IDE, Kaggle dataset**.**

I. INTRODUCTION

Software-Defined Networking(SDN) is the networking architecture defined by the software program. In SDN, network traffic is controlled by the software which directs the traffic between the hosts. This is unlike the current network architecture where the traffic control is hardware-based which is defined by the switches, routers and, other network infrastructure. The centralized SDN architecture is designed in such a way that SDN switch is comprised of only the data plane and the control plane is moved to another entity known as the controller. The controller acts as the brain of the network and it is a network wide centralized entity where all the switches in the network abide by the decision made by the controller.SDN Controller — Translates the requirements from the SDN application layer to the SDN data paths. It also provides the SDN applications with a central repository of network policies, a view of the networks and network traffic.SDN Data path — Implements switches that move data packets on a network.

Control functions are taken from the switch and given to the controller, which is the brain of the network in SDN architecture. Parent level rules are easily applied to the network with the help of the controller. The controller can add new rules to the transmission devices and change the existing rules. It can carry out these changes by communicating with transmission devices via a secure channel through the OpenFlow protocol. Continuity and the unity of data traffic are ensured through this channel. If this secure channel breaks, the connection between the controller and transmission devices breaks.

SDN architecture is the target for DDoS attacks. While the attacker is attacking the SDN network, it has three main targets, as shown in **Figure 1**: to consume the sources of the controller, to occupy the bandwidth of the channel between the controller and the switch, and to fill the flow tables in the switch with unnecessary flows. In DDoS attacks against the controller, the attacker sends a large number of packets to the OpenFlow switch via zombie users. It is difficult for the controller to differentiate between traffic sent by the attacker and legal traffic.

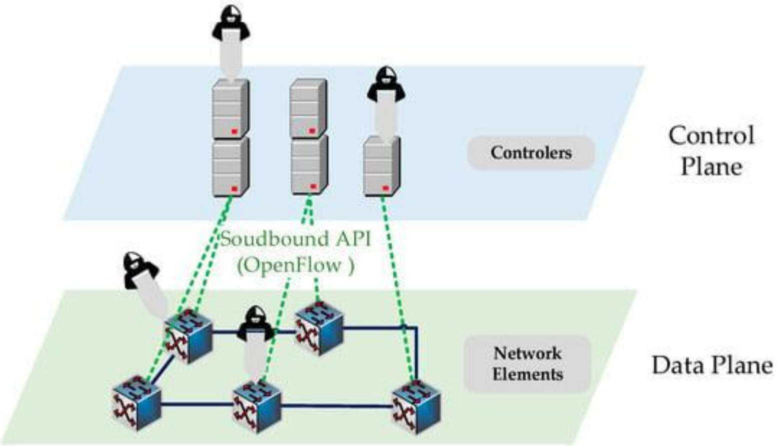


Figure 1 Main targets of the Distributed Denial of Service (DDoS) attacks on Software Defined Networking (SDN).

The OpenFlow switch seeks a match in the flow input by checking the packet header (source port, target port, source IP address, target IP address, etc.). If there is no match, the packet is forwarded to the controller by encapsulating the packet header in the flow request with the OpenFlow protocol (OFPT) PACKET_IN message. Then, the controller responds with the OFPT FLOW_MOD message. This message involves the process to be carried out on the packet and the timeout of the flow in the flow table assigned for the packet [8]. As the number of packets forwarded to the controller increases, the sources of the controller are consumed (bandwidth, memory, and CPU), new flow input for the new legal packets arriving at the network cannot be processed, and the SDN architecture collapses.

II. LITERATURE SURVEY

In[1], Software-Defined Networking (SDN) is a networking paradigm that has redefined the term network by making the network devices programmable. SDN helps network engineers to monitor the network expeditely, control the network from a central point, identify malicious traffic and link failure in easy and efficient manner. Besides such flexibility provided by SDN, it is also vulnerable to attacks such as DDoS which can halt the complete network. To mitigate this attack, the paper proposes to classify the benign traffic from DDoS attack traffic by using machine learning technique. The major contribution of this paper is identification of novel features for DDoS attack detections. Novel features are logged into CSV file to create the dataset and machine learning algorithms are trained on the created SDN dataset. Various work which has already been done for DDoS attack detection either used a non-SDN dataset or the research data is not made public. A novel hybrid machine learning model is utilized to perform the classification. Results show that the hybrid model of Support Vector classifier with Random Forest (SVC-RF) classifies the traffic with the highest testing accuracy of 98.8% with a very low false alarm rate.

In[2], In this paper, we propose a new framework of cooperative detection methods of control plane and data plane, which effectively improve the detection accuracy and efficiency, and prevent DDoS attacks on SDN.

In[3], The Distributed Denial of Service (DDoS) attack has seriously impaired network availability for decades and still there is no effective defense mechanism against it. However, the emerging Software Defined Networking (SDN) provides a new way to reconsider the defense against DDoS attacks. In this paper, we propose two methods to detect the DDoS attack in SDN. One method adopts the degree of DDoS attack to identify the DDoS attack. The other method uses the improved K-Nearest Neighbors (KNN) algorithm based on Machine Learning (ML) to discover the DDoS attack. The results of the theoretical analysis and the experimental results on datasets show that our proposed methods can better detect the DDoS attack compared with other methods.

In[4], Software Defined Networking (SDN) offers several advantages such as manageability, scaling, and improved performance. However, SDN involves specific security problems, especially if its controller is defenseless against Distributed Denial of Service (DDoS) attacks. The process and communication capacity of the controller is overloaded when DDoS attacks occur against the SDN controller. Consequently, as a result of the unnecessary flow produced by the controller for the attack packets, the capacity of the switch flow table becomes full, leading the network performance to decline to a critical threshold. In this study, DDoS attacks in SDN were detected using machine learning-based models. First, specific features were obtained from SDN for the dataset in normal conditions and under DDoS attack traffic. Then, a new dataset was created using feature selection methods on the existing dataset. Feature selection methods were preferred to simplify the models, facilitate their interpretation, and provide a shorter training time. Both datasets, created with and without feature selection methods, were trained and tested with Support Vector Machine (SVM), Naive Bayes (NB), Artificial Neural Network (ANN), and K-Nearest Neighbors (KNN) classification models. The test results showed that the use of the wrapper feature selection with a KNN classifier achieved the highest accuracy rate (98.3%) in DDoS attack detection. The results suggest that machine learning and feature selection algorithms can achieve better results in the detection of DDoS attacks in SDN with promising reductions in processing loads and times.

In[5], Keeping Internet users protected from cyberattacks and other threats is one of the most prominent security challenges for network operators nowadays. Among other critical threats, distributed denial-of-service (DDoS) becomes one of the most widespread attacks in the Internet, which is very challenging to mitigate appropriately as DDoS attacks cause the system to stop working by resource exhaustion. Software-defined networking (SDN) has recently emerged as a new networking technology offering unprecedented programmability that allows network operators to configure and manage their infrastructures dynamically. The flexible processing and centralized management of the SDN controller allow flexibly deploying complex security algorithms and mitigation methods. In this paper, we propose a novel DDoS attack mitigation in SDN-based Internet Service Provider (ISP) networks for TCP-SYN and ICMP flood attacks utilizing machine learning approach, i.e., *K*-Nearest-Neighbor (KNN) and XGBoost. By deploying a testbed, we implement the proposed algorithms, evaluate their accuracy, and address the trade-off between the accuracy and mitigation efficiency. Through extensive experiments, the results show that the algorithms can efficiently mitigate the attack by over 98.0% while benign traffic is not affected.

In[6], The Software-Defined Network (SDN) is a new network paradigm that promises more dynamic and efficiently manageable network architecture for new-generation networks. With its programmable central controller approach, network operators can easily manage and control the whole network. However, at the same time, due to its centralized structure, it is the target of many attack vectors. Distributed Denial of Service (DDoS) attacks are the most effective attack vector to the SDN. The purpose of this study is to classify the SDN traffic as normal or attack traffic using machine learning algorithms equipped with Neighbourhood Component Analysis (NCA). We handle a public "DDoS attack SDN Dataset" including a total of 23 features. The dataset consists of Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Internet Control Message Protocol (ICMP) normal and attack traffics. The dataset, including more than 100 thousand recordings, has statistical features such as byte_count, duration_sec, packet rate, and packet per flow, except for features that define source and target machines. We use the NCA algorithm to reveal the most relevant features by feature selection and perform an effective classification. After preprocessing and feature selection stages, the obtained dataset was classified by k-Nearest Neighbor (kNN), Decision Tree (DT), Artificial Neural Network (ANN), and Support Vector Machine (SVM) algorithms. The experimental results show that DT has a better accuracy rate than the other algorithms with 100% classification achievement.

In[7], Software Defined Network (SDN) is a new network architecture which separates the data plane from the control plane. The SDN controller implements the control plane and switches implement the data plane. Many papers discuss about DDoS attacks on primary servers present in SDN and how they can be mitigated with the help of controller. In our paper we show how DDoS attack can be instigated on the SDN controller by manipulating the flow table entries of switches, such that they send continuous requests to the controller and exhaust its resources. This is a new, but one of the possible way in which a DDoS attack can be performed on controller. We show the vulnerability of SDN for this kind of attack. We further propose a solution for mitigating it, by running a DDoS Detection module which uses variation of flow entry request traffic from all switches in the network to identify compromised switches and blocks them completely.

In[8], Software Defined Networking (SDN) is one of the most commonly used network architectures in recent years. With the substantial increase in the number of Internet users, network security threats appear more frequently, which brings more concerns to SDN. Distributed denial of Service (DDoS) attacks are one of the most dangerous and frequent attacks in software defined networks. The traditional attack detection method using entropy has some defects such as slow attack detection and poor detection effect. In order to solve this problem, this paper proposed a method of fusion entropy, which detects attacks by measuring the randomness of network events. This method has the advantages of fast attack detection speed and obvious decrease in entropy value. The complementarity of information entropy and log energy entropy is effectively utilized. The experimental results show that the entropy value of the attack scenarios 91.25% lower than normal scenarios, which has greater advantages and significance compared with other attack detection methods.

In[9], Software Defined Networks (SDN) provides separation of data plane and control plane, which can be used for implementing various network solutions like traffic engineering, intrusion detection load balancing, etc. However, there are few issues relating to SDN that needs to be addressed, one of them being Distributed Denial of Service (DDoS) attack on the centralized controller. Many researchers have contributed various solutions for identifying and mitigating such attacks. However, the intruders often find new ways of performing such DDoS attacks and hence the detection of such attacks takes more time and resources. In this paper, the aim is 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. Furthermore, a solution is also proposed to detect such an attack within the second repeated request and mitigate it immediately. This solution is highly efficient as the attack is detected instantly instead of calculating a threshold based on the number of flow entry requests to identify whether the traffic is attack traffic or a genuine one.

In[10], The rapid usage of the Internet for the last few decades has lead to the deployment of high-speed networks in commercial and educational institutions. As network traffic is increasing, security challenges are also increasing in the high-speed network. Although the Intrusion Detection System (IDS) has a significant role in spotting potential attacks, the heavy traffic flow causes severe technical challenges relating to monitoring and detecting the network activities. Moreover, the devastating nature of the Distributed Denial-of-Service (DDoS) attack draws out as a significant cyber-attack regardless of the emergence of Software Defined Network (SDN) architecture. This paper proposes a novel framework to address the performance issues of IDS and the design issues of SDN about DDoS attacks by incorporating intelligence in the data layer using Data Plane Development Kit (DPDK) in the SDN architecture. This novel framework is named as DPDK based DDoS Detection (D3) framework, since DPDK provides fast packet processing and monitoring in the data plane. Moreover, the statistical anomaly detection algorithm implemented in the data plane as Virtual Network Function (VNF) using DPDK offers fast detection of DDoS attacks. The experimental results of the D3 framework guarantee both efficiency and effect of the novel IDS framework. The publicly available CIC DoS datasets also ensure the detection effect of a single statistical anomaly detection algorithm against the DDoS attack.

III. METHODOLOGY

At first the datas are read from the input dataset by using pandas library. Then the data will be pre-processed 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 in to the KNN model and the machine gets trained. After training we predict DDos attack datas by feeding test datas into the model.

- In our work, the detection of DDoS is one of the most important processes. In this project, the main goal is to classify traffic as DDOs attack or normal traffic.

- For a wide range of problems, a large number of samples will be selected, but the sample size will not be large. Algorithms work best when it comes to speed, but data sharing is not a problem.

- The main advantage of this model is the use of a large amount of data for training and a small amount of data for testing, which can improve the **accuracy of the classification**.

- Proposed flow diagram consists of Data Pre-Processing, Feature Selection, Trained Model, Test Data.

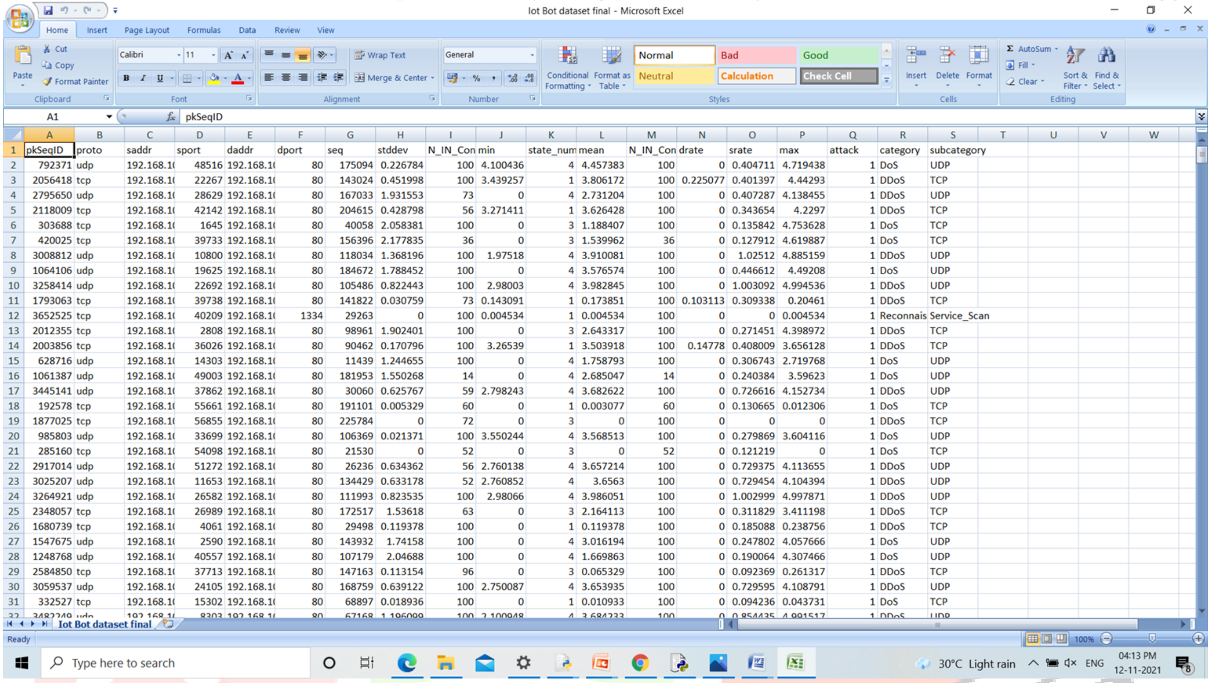
MODULES

- **INPUT DATASET**
- **PREPROCESSING**
- **DEATURE SELECTION**
- **TRAINING MODEL**
- **PREDICTION MODEL**

The modules explanation is below

- **INPUT DATASET**

At first the datas are read from the input dataset by using pandas library. The dataset is downloaded from kaggle website.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	pkSeqID	proto	saddr	sport	daddr	dport	seq	stddev	N_IN_Con	min	state_num	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
27	1547675	udp	192.168.1(	2590	192.168.1(	80	143932	1.74158	100	0	4	3.016194	100	0	0.247802	4.057666	1	DoS	UDP
28	1248768	udp	192.168.1(	40557	192.168.1(	80	107179	2.04688	100	0	4	1.669863	100	0	0.190064	4.307466	1	DoS	UDP
29	2584850	tcp	192.168.1(	37713	192.168.1(	80	147163	0.113154	96	0	3	0.065329	100	0	0.092369	0.261317	1	DDoS	TCP
30	3059537	udp	192.168.1(	24105	192.168.1(	80	168759	0.639122	100	2.750087	4	3.653935	100	0	0.729595	4.108791	1	DDoS	UDP
31	332527	tcp	192.168.1(	15302	192.168.1(	80	68897	0.018936	100	0	1	0.010933	100	0	0.094236	0.043731	1	DoS	TCP
32	3482249	udp	192.168.1(	8303	192.168.1(	80	67168	1.196099	100	2.100948	4	3.684233	100	0	0.854435	4.991517	1	DDoS	UDP

- **Dataset description**

1.pkSeqID-row identifier
2.proto-Textual representation of transaction protocols present in network flow
3.saddr-source address
4.sport-source port
5.daddr-destination address
6.dport-destination port
7.seq-Argus sequence number
8.stddev-Standard deviation of aggregated records
9.N_IN_Conn_P_SrcIP-Number of inbound connections per destination IP.
10.min-Minimum duration of aggregated records
11.state_number-Numerical representation of feature state
12.mean-Average duration of aggregated records
13.N_IN_Conn_P_DstIP-Number of inbound connections per source IP
14.drate-Destination-to-source packets per second
15.srate-Source-to-destination packets per second
16.max-Maximum duration of aggregated records
17.attack-Class label: 0 for Normal traffic, 1 for Attack Traffic
18.category-Traffic category
19.subcategory-Traffic subcategory

PREPROCESSING

Data preprocessing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for a building and training Machine Learning models.Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Preprocessing is a technique that is used to convert the raw data into a clean data set.

Missing data: Take a look for missing data fields, blank spaces in text, or unanswered survey questions. This could be due to human error or incomplete data. To take care of missing data, you’ll have to perform data cleaning.

Data cleaning is the process of adding missing data and correcting, repairing, or removing incorrect or irrelevant data from a data set. Dating cleaning is the most important step of preprocessing because it will ensure that your data is ready to go for your downstream needs.

Data cleaning will correct all of the inconsistent data you uncovered in your data quality assessment. Depending on the kind of data you're working with, there are a number of possible cleaners you'll need to run your data through.

FEATURE SELECTION

Feature Selection is the method of reducing the input variable to your model by using only relevant data and getting rid of noise in data. It is the process of automatically choosing relevant features for your machine learning model based on the type of problem you are trying to solve.Relief feature scoring is based on the identification of feature value differences between nearest neighbor instance pairs. If a feature value difference is observed in a neighboring instance pair with the same class (a 'hit'), the feature score decreases.

TRAINING MODEL

The process of training an ML model involves providing an ML algorithm (that is, the learning algorithm) with training data to learn from. The term ML model refers to the model artifact that is created by the training process.

PREDICTION MODEL

Predictive modeling is a mathematical process used to predict future events or outcomes by analyzing patterns in a given set of input data. It is a crucial component of predictive analytics, a type of data analytics which uses current and historical data to forecast activity, behavior and trends. This work is used KNN here.

ALGORITHM

- we consider in our implementation of the algorithm. Therefore, distance metric and K value are two important considerations while using the KNN algorithm. Euclidean distance is the most popular distance metric.
- Firstly, we will choose the number of neighbors, so we will choose the k=3.
- Next, we will calculate the Euclidean distance between the data points. The Euclidean distance is the distance between two points.

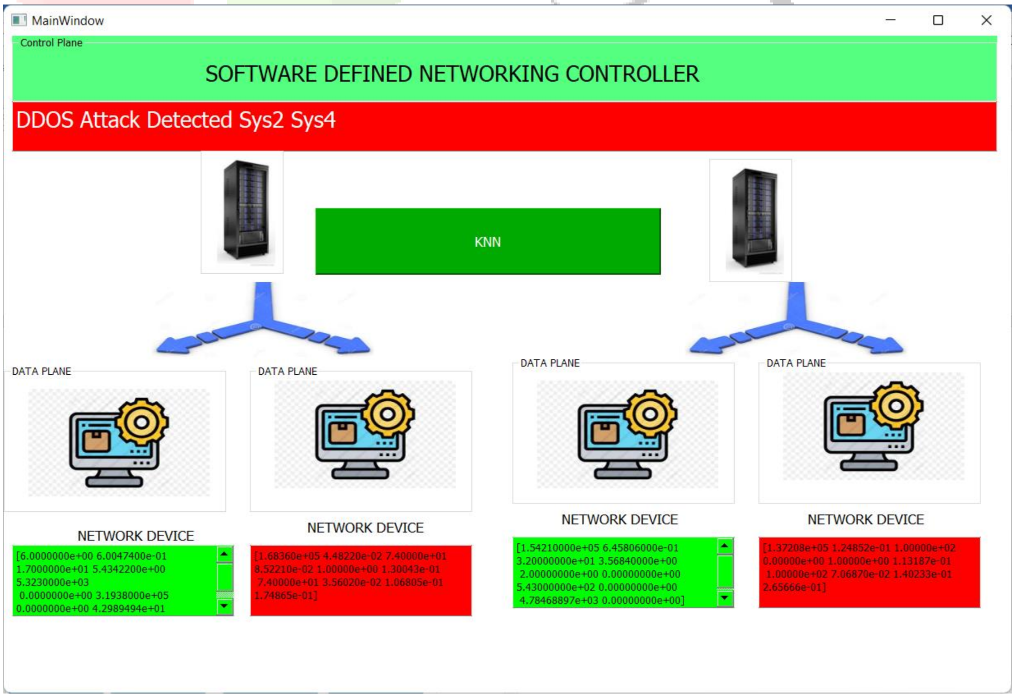
KNN

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

- ❖ Step-1: Select the number K of the neighbors
- ❖ Step-2: Calculate the Euclidean distance of K number of neighbors
- ❖ Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.
- ❖ Step-4: Among these k neighbors, count the number of the data points in each category.
- ❖ Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.
- ❖ Step-6: Our model is ready.

IV. EXPERIMENT AND ANALYSIS

The experimental result shows the output of the project.



DDOS Attack Detected Sys2 Sys4

```
[6.0000000e+00 6.0047400e-01
1.7000000e+01 5.4342200e+00
5.3230000e+03
 0.0000000e+00 3.1938000e+05
0.0000000e+00 4.2989494e+01
```

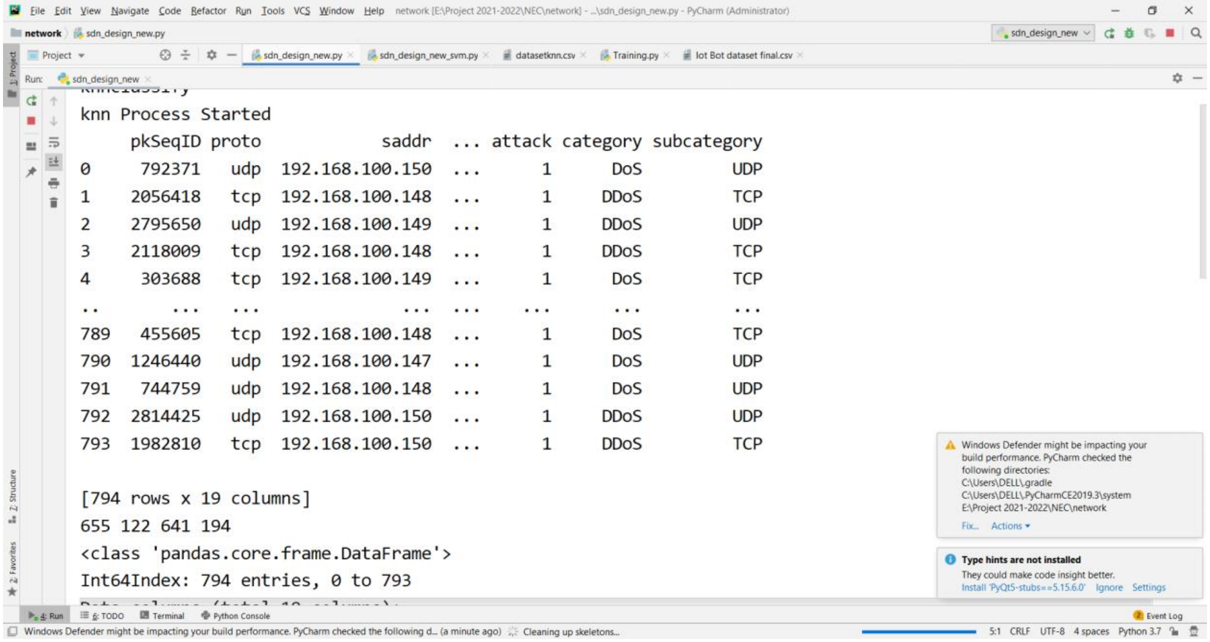
```
[1.68360e+05 4.48220e-02 7.40000e+01
8.52210e-02 1.00000e+00 1.30043e-01
 7.40000e+01 3.56020e-02 1.06805e-01
1.74865e-01]
```

```
[1.54210000e+05 6.45806000e-01
3.20000000e+01 3.56840000e+00
 2.00000000e+00 0.00000000e+00
5.43000000e+02 0.00000000e+00
 4.78468897e+03 0.00000000e+00]
```

```
[1.37208e+05 1.24852e-01 1.00000e+02
0.00000e+00 1.00000e+00 1.13187e-01
 1.00000e+02 7.06870e-02 1.40233e-01
2.65666e-01]
```

Figure 2 Main window

The above figure illustrates the main windows of the project. There are four data plane and two servers. The servers are linked with SDN. When the attack is come to data plane, that node is colored as red. The intimation is going to SDN through server.



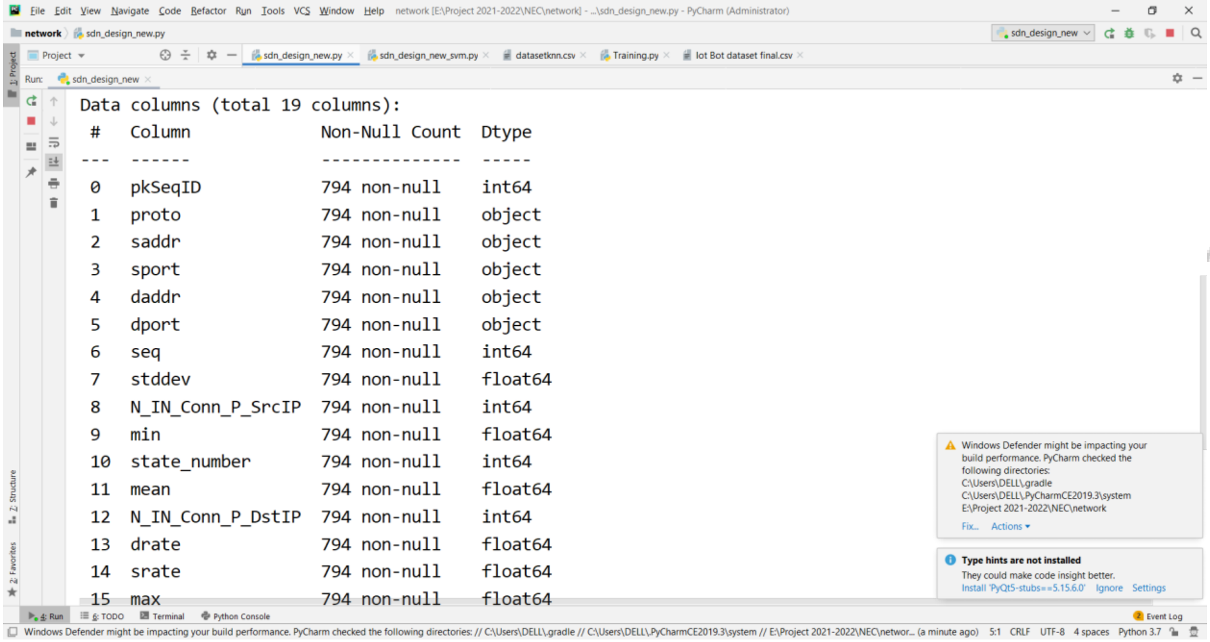
```
knn Process Started
    pkSeqID proto           saddr  ... attack category subcategory
0    792371   udp 192.168.100.150  ...      1      DoS         UDP
1   2056418   tcp 192.168.100.148  ...      1     DDoS         TCP
2   2795650   udp 192.168.100.149  ...      1     DDoS         UDP
3   2118009   tcp 192.168.100.148  ...      1     DDoS         TCP
4    303688   tcp 192.168.100.149  ...      1      DoS         TCP
..      ...   ...             ...  ...    ...      ...         ...
789  455605   tcp 192.168.100.148  ...      1      DoS         TCP
790 1246440   udp 192.168.100.147  ...      1      DoS         UDP
791  744759   udp 192.168.100.148  ...      1      DoS         UDP
792 2814425   udp 192.168.100.150  ...      1     DDoS         UDP
793 1982810   tcp 192.168.100.150  ...      1     DDoS         TCP

[794 rows x 19 columns]
655 122 641 194
<class 'pandas.core.frame.DataFrame'>
Int64Index: 794 entries, 0 to 793
```

Windows Defender might be impacting your build performance. PyCharm checked the following directories:
C:\Users\DELL\.gradle
C:\Users\DELL\.PyCharmCE2019.3\system
E:\Project 2021-2022\NEC\network

Type hints are not installed
They could make code insight better.
Install 'PyQt5-stubs==5.15.6.0' Ignore Settings

Figure 3 Dataset details

The above figure illustrates the details of the dataset. There are 794 rows and 19 attributes like source IP address, destination IP address, packet size and category etc.



```
Data columns (total 19 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   pkSeqID           794 non-null    int64
 1   proto             794 non-null    object
 2   saddr             794 non-null    object
 3   sport             794 non-null    object
 4   daddr             794 non-null    object
 5   dport             794 non-null    object
 6   seq               794 non-null    int64
 7   stddev            794 non-null    float64
 8   N_IN_Conn_P_SrcIP 794 non-null    int64
 9   min               794 non-null    float64
 10  state_number      794 non-null    int64
 11  mean              794 non-null    float64
 12  N_IN_Conn_P_DstIP 794 non-null    int64
 13  drate             794 non-null    float64
 14  srate             794 non-null    float64
 15  max               794 non-null    float64
```

Windows Defender might be impacting your build performance. PyCharm checked the following directories:
C:\Users\DELL\.gradle
C:\Users\DELL\.PyCharmCE2019.3\system
E:\Project 2021-2022\NEC\network

Type hints are not installed
They could make code insight better.
Install 'PyQt5-stubs==5.15.6.0' Ignore Settings

Figure 4 Attributes Information

The Above figure shows the attribute details with not null values and data type of the attributes.

V. CONLIUSION

In this study, SDN-based detection systems developed for DDoS attacks were analyzed with 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. Among the proposed systems, the second approach proceeds by labelling DDoS attacks as normal traffic and attack traffic. With 97.7% sensitivity, KNN algorithms can perform this control by lightening the burden of the controller.

With the feature selection methods used in the study, initially 12 features were selected and the selected subset of features was trained using classifiers. The number of features selected was determined either by the algorithm itself or by the threshold value given to the algorithm. By changing this threshold value, different numbers of features can be selected and trained by the classifier. Different numbers of features can change the accuracy. In general, we observed that the performance rate of all models is above 80% and the algorithms used for this dataset are successful. At the same time, network browsing, attacks between layers, and malicious software can be detected on SDN with this approach. For protecting and improving SDN structure, the second approach can be employed.

REFERENCES

[1] Nisha Ahuja , Gaurav Singal , Debajyoti Mukhopadhyay , Neeraj Kumar " Automated DDOS attack detection in software defined networking", Volume 187 ,1 August 2021

[2] 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.

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

[4] HuseyinPolat ,OnurPolat and Aydin Cetin "Detecting DDoS Attacks in Software- Defined Networks Through Feature Selection Methods and Machine Learning Models", Volume 12, 29 January 2020.

[5] 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, volume 19,2019.

[6] 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.

[7] Sanjeetha R , Shikhar Srivastava , RishabPokharna , SyedShafiq , Dr. Anita Kana valli "Mitigation of DDoS attack instigated by compromised switches on SDN controller by analyzing the flow rule request traffic", volume 7 , May 2019.

[8] 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), Volume 19,2019.\

[9] S. R, A. Pattanaik, A. Gupta and A. Kanavalli, "Early Detection and Diminution of DDoS attack instigated by compromised switches on the controller in Software Defined Networks," 2019 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), 2019, pp. 1-5, doi: 10.1109/DISCOVER47552.2019.9007925.

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