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 SOFTWARE-DEFINED NETWORKING USING SVM ALGORITHM

Lakshmi Prabha S
Department of Information Technology
National Engineering College, Kovilpatti, Tuticorin District, TamilNadu, India

Bala Subhashini SP
Department of Information Technology
National Engineering College, Kovilpatti, Tuticorin District, TamilNadu, India

Maria Velantina A
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 networking paradigm that has redefined the term network by allowing network administrators to programmatically initialize, control, change, and manage network behavior. The centralized control that is a primary benefit of SDN can also be a huge security risk. If the intruder succeeds in gaining access to the central controller, he will have complete control of the system. The controller is extremely vulnerable to the Distributed Denial of Service (DDoS) assaults, which cause the system's resources to be depleted, resulting in the controller's services becoming unavailable. This has led to the development of numerous algorithms and strategies. However, on the topic of SDN networks, less research has been done. One such method is to use machine learning algorithms to categorize connections as valid or illegal. To discover the suspicious and hazardous traffic patterns, we employ a machine learning approach called the Support Vector Machine (SVM) classifier.

Key Words: Software-Defined Networking, Distributed Denial of Service, Support Vector Machine.

– INTRODUCTION

Many DDoS attack detection approaches have been presented by researchers focusing on traditional network design. Lin and Wang suggested a DDoS detection and defense system but the method used three Open flow management tools with slow standards to perform anomaly detection, so the deployment and operation are complex. Yang et al presented a method in which flow information and IP entropy characteristic information are combined which is detected by a single flow information and IP entropy characteristic information that has a higher and more accurate detection effect. Although information entropy is flexible and convenient, it must be paired with other technologies to determine the threshold and multi-element weight distribution. Saied et al advanced that, to identify DDoS assaults, the approach must analyze the features of each protocol of TCP/UDP/ICMP using the training ANN algorithm, which is difficult and inefficient.

Many SOM technique is used to identify DDoS assaults by extracting DDoS-related flow statistics. This approach has a low consumption rate and a high detection rate. The crucial issue is the extraction of time intervals. The downside of this approach is that the detection has some hysteresis and the attack behavior is not detected in a timely and precise manner. The authors provided a framework for detecting and mitigating DDoS assaults in a large-scale network, it is not appropriate for small-scale implementation.

It is proposed to implement a DDoS attack detection technique based on a genuine source and destination IP address database. Based on the nonparametric cumulative algorithm CUSUM, it analyses the anomalous features of the source and destination IP addresses when a DDoS assault happens and effectively checks the DDoS attack, but the approach must be adjusted and determined.

– RELATED WORKS

Nisha Ahuja, et al., (2021) article proposed using machine learning techniques such as logistic regression, SVC, KNN, Random Forest, Ensemble classifier, ANN, SVC-RF to identify benign traffic from DDoS assault traffic. The hybrid model of Support Vector classifier with Random Forest (SVC-RF) classifies traffic with the greatest testing accuracy of 98.8% and a very low false alarm rate.

Kshira sagar sahoo, et al., (2020) article proposed that SVM is used in conjunction with a kernel principal component analysis (KPCA) and a genetic approach accuracy (GA). An improved kernel function (N-RBF) is presented to decrease the noise generated by feature discrepancies. The experimental findings suggested that the proposed model achieves more accurate classification and better generalization than single-SVM and the proposed model can be implemented within the controller to build security rules that will restrict attackers from launching assaults.

Huseyin Polat et al., (2020) article proposed that DDoS attacks in SDN were detected using machine learning-based models. They trained and tested datasets with and without feature selection methods using classification models such as Support Vector Machine (SVM), Naive Bayes (NB), Artificial Neural Network (ANN), and K-Nearest Neighbors (KNN). According to the test findings the usage of the wrapper feature selection with a KNN classifier achieved the greatest accuracy rate (98.3%) in DDoS attack detection.

Parvinder Singh Saini, et al., (2020) used a machine learning-based approach in this paper to detect and classify various types of network traffic flows. The proposed method is validated using a new dataset containing a mix of modern type's attacks such as HTTP flood, SID DoS, normal traffic. WEKA, a machine learning tool, is used to classify various types of attacks.

Dong Li, et al., (2020) proposed based on SVM, this research suggested a new model for detecting DDoS attacks in SDN. Firstly the model extracts many essential features from packet-in data and uses entropy to measure the distribution of each feature before utilizing a trained Support Vector Machine (SVM) method to detect the DDoS attack.

Myo Myint Oo, et al., (2019) proposed a DDoS attack detection method based on SDN that cause the least amount of disruption to legitimate user activities and to propose ASVM technique as an enhancement of the existing SVM algorithm to detect DDoS attacks. They measured a false alarm rate, a detection rate, and accuracy to assess the outcomes with the shortest training and testing times, and detection approach had a detection accuracy of around 97%.

Puming Wanga et al., (2019) proposed that the article proceeded at the state-of-the-art of SDN security from a data standpoint and some common network attack detection (NAD) methods, such as machine learning-based methods and statistical methods, are examined. To detect attacks tensor principal component analysis (TPCA) and a review of the most recent data-driven SDN frameworks, a tensor-based big data-driven SDN threat detection framework for SDN security are presented. Finally, a case study is presented to demonstrate the efficacy of the suggested framework.

Jin Ye, et al., (2018) have proposed a method that in this research, the SDN environment is established using the mini net and food light simulation platforms, the switch low table's 6-tuple characteristic values are extracted, and a DDoS attack model is built using the SVM classification methods. With a small amount of low gathering, the method's average accuracy rate is 95.24 percent, according to the studies. Their work has a lot of potential for detecting DDoS attacks in SDN.

Nisharani Meti et al., (2017) proposed the utilizing SVM and Neural Network classifiers to detect intrusions and DDoS attacks. They learned with a training dataset to create a classifier model. The predictor model is built used two classification techniques: SVM and NN. When the server receives a new request from a client, the request is passed to the model, which predicts whether the connection is normal or abnormal. Nature determined a new connection (normal or abnormality) using knowledge of prior connections and the predictive model developed on top of them.

– METHODOLOGY

A. DETECTION OF DDoS ATTACK USING SVM

In our work, DDoS detection is the ultimate aim. The main goal of the project is to identify whether the incoming request is normal traffic or a DDOS attack. For detecting this, we use an SVM algorithm which creates a hyperplane to classify the request. SVM or Support Vector Machine is a linear model that may be used to solve classification and regression issues. It can handle linear and nonlinear problems and is useful for a wide range of practical applications such as Face Detection, Classification of Images, text and Hypertext recognition, and so on. A hyperplane is a decision border that separates the two classes and a data point on either side of the hyperplane belongs to different classes. The size of the hyperplane is determined by the number of input characteristics in the dataset. The hyperplane will be a line if we have two input characteristics. Similarly, if the number of features is three, it will be a two-dimensional plane. The decision to apply SVM to our problem is because it works well for classification problems with a large number of features as in our case and some of the advantages of this model are that it uses a large amount of data for training and a small amount of data for testing, which can improve the accuracy of the classification and it can also be memory efficient. The fig.1 below explains the flow diagram of detecting DDoS attacks using SVM.

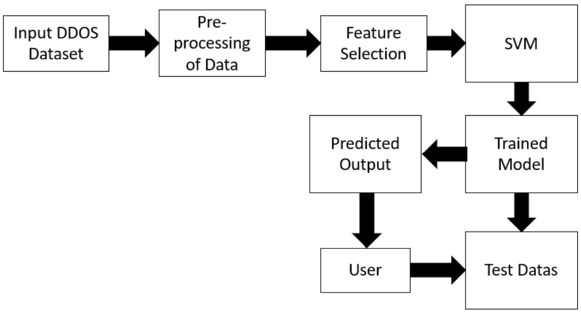


Fig.1 Flow diagram of detecting DDoS attack using SVM

The data is initially extracted from the annotated traffic dataset using the panda's package.The data is then preprocessed by removing null values, and input features are selected for training the model.The trained model is then fed with test data which uses an SVM algorithm to detect whether traffic is DDOS attack or normal traffic.

B. SUPPORT VECTOR MACHINE

The Support Vector Machine is a well-known Supervised Learning technique that can address classification and regression problems. It is, however, mostly employed in Machine Learning for Classification challenges. The purpose of the SVM algorithm is to discover the optimal line or decision boundary for categorizing n-dimensional space into classes so that subsequent data points can be conveniently placed in the relevant category. The optimal choice boundary is referred to as a hyperplane. SVM selects the extreme points/vectors that aid in the creation of the hyperplane. These extreme situations are known as support vectors, and the algorithm is known as the Support Vector Machine. Consider the Fig.2 support vector machine below, which shows two distinct categories that are separated by a decision boundary or hyperplane.

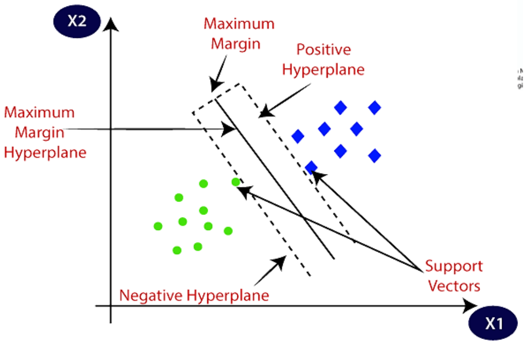


Fig.2 Support Vector Machine

C. WORKING OF SVM

1. Identify the right hyper-plane (Scenario-1)

There are three hyper-planes in this area (A, B, and C). Select the appropriate hyper-plane to classify stars and circles. Fig.3 below explains how the right hyper-plane classifies the stars and circles.

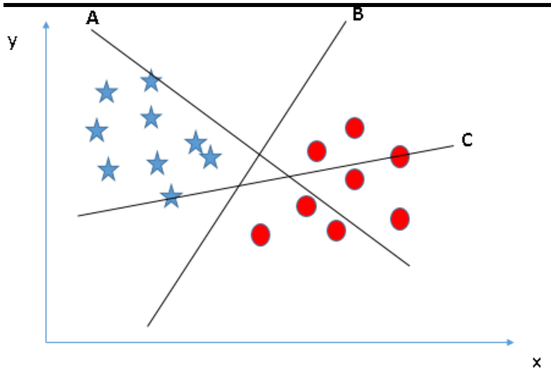


Fig.3 Right hyper-plane (Scenario-1)

2. Identify the right hyper-plane (Scenario-2)

Here, we have three hyper-planes (A, B, and C), each of which effectively separates the classes. Now, how do we find the correct hyper-plane? Fig.4 below explains how the classes are being segregated.

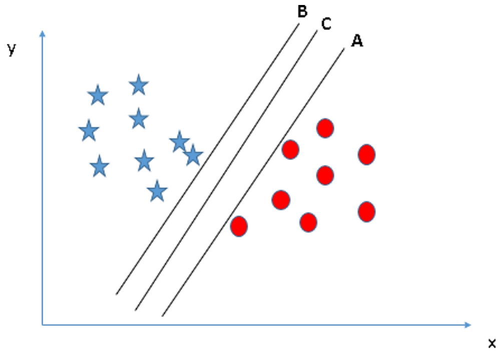


Fig.4 Right hyper-plane (Scenario-2)

In this case, maximising the distances between the nearest data point (of either class) and the hyper-plane will assist us in selecting the appropriate hyper-plane. A margin is the term for this distance.

3. Identify the right hyper-plane (Scenario-3)

Use the rules as discussed in the previous section to identify the right hyperplane. Fig.5 below explains how to identify the right hyperplane.

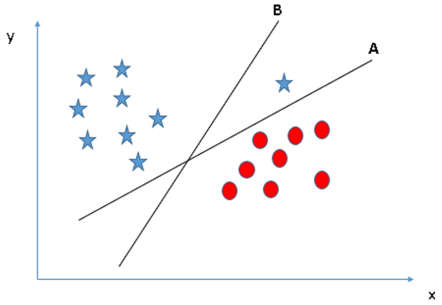


Fig.5 Right hyper-plane (Scenario-3)

4. Find the hyper-plane to segregate to classes (Scenario-4)

In the scenario below, we can't have a linear hyper-plane between the two classes, so how does SVM classify these two classes? We've only looked at the linear hyper-plane so far.Fig.6 below explains how the SVM classifies two classes.

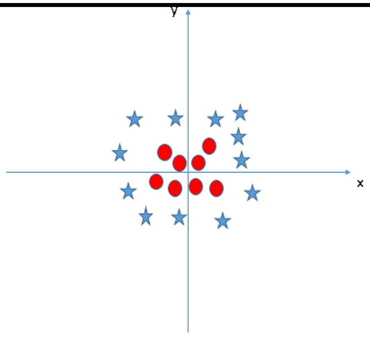
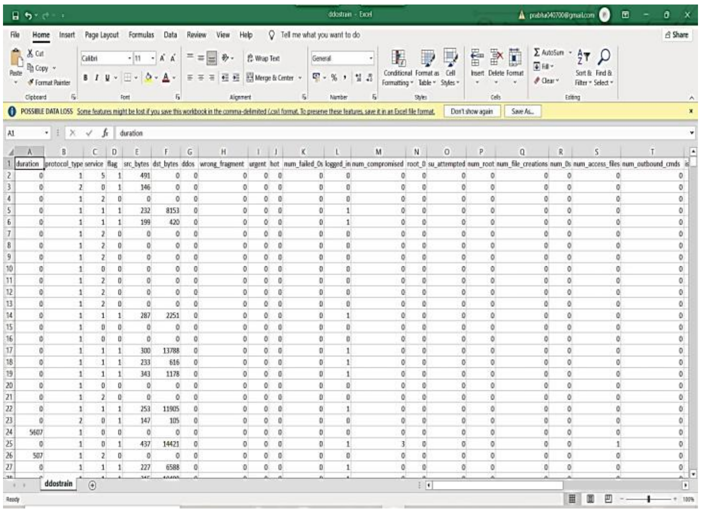


Fig.6 Right hyper-plane (Scenario-4)

- DATASET AND DATA PROCESSING

A. DATASET FOR DDoS ATTACK

We collected the DDoS dataset from the Kaggle website. This dataset contains 1000 traffic patterns with 42 attributes, of which we have considered 22 features for detection. We have used 80% of the data for training and 20% of the data for testing. It describes all kind of data that are considered as DDoS attacks and all are the data's which are considered normal. At first, the dataset is fetched by using the panda's library and then we save the data's 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 to process null values. The Dataset consists of a few DDoS Attacks. The most common types are incorporated in this dataset are: - TCP Flood Attack, UDP Flood Attack, and ICMP (Ping) Flood Attack. These attacks constitute the most common threats that are advancing at an alarming rate, in their sophistication and frequency.



POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.

A1: duration

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	duration	protocol_type	service	flag	src_bytes	dst_bytes	ddos	wrong_fragment	urgent	hot	num_failed_0s	logged_in	num_compromised	root_0	su_attempted	num_root	num_file_creations	num_0s	num_access_files	num_outbound_cmds
2	0	1	5	1	491	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	2	0	1	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	1	1	1	232	8153	0	0	0	0	0	1	0	0	0	0	0	0	0	0
6	0	1	1	1	199	420	0	0	0	0	0	1	0	0	0	0	0	0	0	0
7	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	1	1	1	287	2251	0	0	0	0	0	1	0	0	0	0	0	0	0	0
15	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	1	1	1	300	13788	0	0	0	0	0	1	0	0	0	0	0	0	0	0
18	0	1	1	1	233	616	0	0	0	0	0	1	0	0	0	0	0	0	0	0
19	0	1	1	1	343	1178	0	0	0	0	0	1	0	0	0	0	0	0	0	0
20	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	0	1	1	1	253	11905	0	0	0	0	0	1	0	0	0	0	0	0	0	0
23	0	2	0	1	147	105	0	0	0	0	0	0	0	0	0	0	0	0	0	0
24	5607	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	1	0	1	437	14421	0	0	0	0	0	1	3	0	0	0	0	0	1	0
26	507	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	1	1	1	227	6588	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Fig.7 Dataset for DDoS Attack

The above Fig.7 shows the list of the following attributes used in the dataset for DDoS attacks. The following is the list of attributes that are particularly used for detecting the request.

– RESULTS AND DISCUSSION

The below Fig.8 shows the histogram of outcome in which 0 depicts the normal attack whereas 1 depicts the DDoS attack.

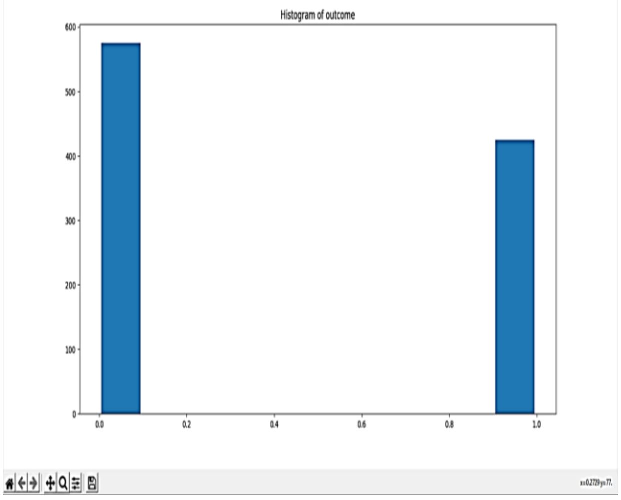
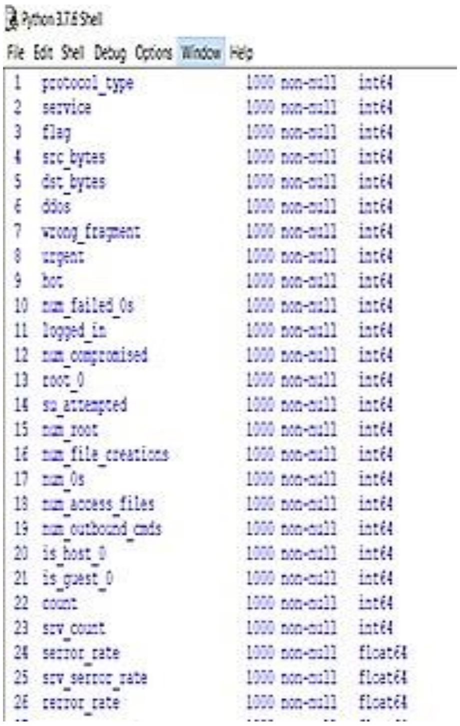


Fig.8 – Histogram of outcome

0 - Normal Attack 1 - DDoS Attack

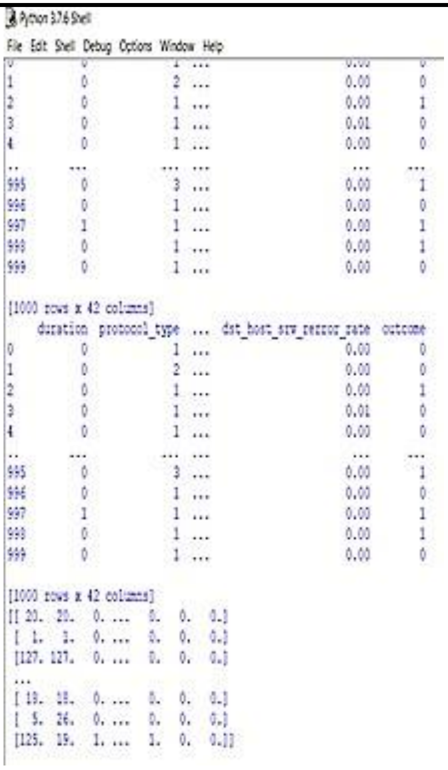
The below Fig.9 shows the list of attributes that are used in the dataset for detecting the behavior of a request which is displayed by using the line of **print(dataset.info())** in the python code.



```
1   protocol_type           1000 non-null   int64
2   service                 1000 non-null   int64
3   flag                    1000 non-null   int64
4   src_bytes               1000 non-null   int64
5   dst_bytes               1000 non-null   int64
6   ddos                    1000 non-null   int64
7   wrong_fragment          1000 non-null   int64
8   urgent                  1000 non-null   int64
9   hot                     1000 non-null   int64
10  num_failed_0s           1000 non-null   int64
11  logged_in               1000 non-null   int64
12  num_compromised         1000 non-null   int64
13  root_0                  1000 non-null   int64
14  su_attempted            1000 non-null   int64
15  num_root                1000 non-null   int64
16  num_file_creations      1000 non-null   int64
17  num_0s                  1000 non-null   int64
18  num_access_files        1000 non-null   int64
19  num_outbound_cmds       1000 non-null   int64
20  is_host_0               1000 non-null   int64
21  is_guest_0              1000 non-null   int64
22  count                   1000 non-null   int64
23  srv_count               1000 non-null   int64
24  serror_rate             1000 non-null   float64
25  srv_serror_rate         1000 non-null   float64
26  rerror_rate             1000 non-null   float64
```

Fig.9 – List of attributes used in the dataset

The below Fig.10 shows the confusion matrix of test data which is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. Confusion matrices are useful because they give direct comparisons of values like True positive, False Positives, True Negatives, and False Negatives.

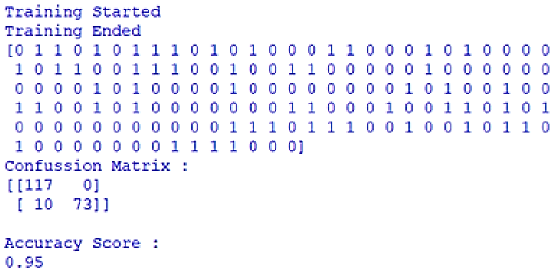


```
1        0              2  ...                        0.00       0
2        0              1  ...                        0.00       1
3        0              1  ...                        0.01       0
4        0              1  ...                        0.00       0
..     ...            ...  ...                         ...     ...
995      0              3  ...                        0.00       1
996      0              1  ...                        0.00       0
997      1              1  ...                        0.00       1
998      0              1  ...                        0.00       1
999      0              1  ...                        0.00       0

[1000 rows x 42 columns]
     duration  protocol_type  ...  dst_host_srv_rerror_rate  outcome
0           0              1  ...                      0.00        0
1           0              2  ...                      0.00        0
2           0              1  ...                      0.00        1
3           0              1  ...                      0.01        0
4           0              1  ...                      0.00        0
..        ...            ...  ...                       ...      ...
995         0              3  ...                      0.00        1
996         0              1  ...                      0.00        0
997         1              1  ...                      0.00        1
998         0              1  ...                      0.00        1
999         0              1  ...                      0.00        0

[1000 rows x 42 columns]
[[ 20.  20.   0. ...   0.   0.   0.]
 [  1.   1.   0. ...   0.   0.   0.]
 [127. 127.   0. ...   0.   0.   0.]
 ...
 [ 19.  18.   0. ...   0.   0.   0.]
 [  5.  26.   0. ...   0.   0.   0.]
 [125.  19.   1. ...   1.   0.   0.]]
```

Fig.10 – Confusion matrix of test data



```
Training Started
Training Ended
[0 1 1 0 1 0 1 1 1 0 1 0 1 0 0 0 1 1 0 0 0 1 0 1 0 0 0 0
 1 0 1 1 0 0 1 1 1 0 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 0 0 0
 0 0 0 0 1 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0
 1 1 0 0 1 0 1 0 0 0 0 0 0 0 1 1 0 0 0 1 0 0 1 1 0 1 0 1
 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 1 1 1 0 0 1 0 0 1 0 1 1 0
 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0]
Confussion Matrix :
[[117   0]
 [ 10  73]]

Accuracy Score :
0.95
```

Fig.11– Accuracy Score of test data

The above Fig.11 shows the accuracy score of test data which shows a 95% rate for the DDoS detection using a support vector machine.

- CONCLUSION

Although SDN offers numerous advantages, it is vulnerable to DDoS assaults, the most prevalent network security concern. As a benefit of SDN, centralized control makes the SDN controller more exposed to security risks such as DDoS assaults. In response to this issue, we examine the detection and protection mechanism of DDoS assaults using the SVM machine learning method in this study which shows an accuracy of 95%. Experiments are designed to demonstrate the effectiveness of the detection methods suggested in this research.

SDN provides a virtual network that runs actual software on the network's components, allowing it to be used to test networking software interactively. It has a controller which controls the networking functions. The trained Python code will be placed in the controller. The controller can identify a DDoS assault and prevent the network from pausing by sending the DDoS traffic pattern from any server, so saving the network.

- FUTURE WORK

In the future, we will examine the detection mechanism of DDoS attacks using the Random forest and ID3 classification machine learning methods. Therefore in the future, we can create an SDN network using mini net software. Mininet, an open-source network simulator, is intended to aid in research and education in the field of Software Defined Networking systems, and also we will examine the detection mechanism of DDoS attacks using the Random forest and ID3 classification machine learning methods.

REFERENCES

[1] Nisha Ahuja, Gaurav Singal, Debajyoti Mukhopadhyay, Neeraj Kumar, “Automated DDOS attack detection in software-defined networking”, ELSEVIER, Journal of Network and Computer Applications Volume 187 (2021), 103108.

[2] R Kshira sagar sahoo, Bata krishna tripathy, Kshirasagar naik, Somula ramasubbareddy, Balamurugan balusamy, Manju khari, DanielBurgos, “An Evolutionary SVM Model for DDOS Attack Detection in Software Defined Networks”, IEEE Access – Volume 8, 2020.

[3] Huseyin Polat, Onur Polat and Aydin Cetin, “Detecting DDoS Attacks in Software-Defined Networks Through Feature Selection Methods and Machine Learning Models”, MDPI Sustainability - Volume 12 - Issue 3, 2020.

[4] Parvinder Singh Saini, Sunny Behal, Sajal Bhatia, “ Detection of DDoS Attacks using Machine Learning Algorithms”, IEEE- 2020 7th International Conference on Computing for Sustainable Global Development.

[5] Dong Li, ChangYu, Qizha Zhou and Junqing Yu, “Using SVM to Detect DDoS Attack in SDN Network”, IOP Conference Series: Materials Science and Engineering, Volume 466 – 2018.

[6] Myo MyintOo, Sinchai Kamolphiwong, Thossaporn Kamolphiwong, and Sangsuree Vasupongayy, “Advanced Support Vector Machine- (ASVM-) Based Detection for Distributed Denial of Service (DDoS) Attack on Software Defined Networking”, Hindawi - Journal of Computer Networks and Communications – Volume 2019.

[7] Puming Wanga, Laurence T. Yanga, Xin Niea, Zhian Rena, Jintao Li , Liwei Kuange, “Data-driven software defined network attack detection : State-of-the-art and perspectives” , ELSEVIER - Information Sciences Volume 513, 2019.

[8] Jin Ye, Xiangyang Cheng, Jian Zhu, Luting Feng, and Ling Song, “A DDoS Attack Detection Method Based on SVM in Software Defined Network”, Hindawi Security and Communication Networks – Volume 2018.

[9] Nisharani Meti, Narayan D G and V. P. Baligar, “Detection of Distributed Denial of Service Attacks using Machine Learning Algorithms in Software Defined Network”, IEEE - 2017 International Conference on Advances in Computing, Communications and Informatics.