Research Article **ISSN: 2393 – 9532**

International Journal of Arts and Science Research

Journal home page: www.ijasrjournal.com

https://doi.org/10.36673/IJASR.2020.v07.i01.A02

DETECTING DDOS ATTACK USING ADVANCED SUPPORT VECTOR MACHINES (ASVM) IN IOT NETWORK

V. V. R. Raman*[1] and Om Prakash Yadav[2]

[1*]Department of Computer Science, Aurora's Degree and P.G. College, Hyderabad, Telangana, India.
[2]Department Humanities and Sciences, Research Scholar Enrolment No: SSSCSE1515, SSSUTMS- Sehore, Madhya Pradesh, India.

ABSTRACT

The Distributed Denials of Service (DDOS) attacks have strong influence on the cyber world. The DDoS attacks are a serious threat security to network world. The normal functioning of the organization have been terminations by cyber-attack like; Internet protocol (IP) spoofing, bandwidth overflow, consuming memory resources etc. and causes a huge loss of industry. The Advance Support Vector Machine (ASVM) is focused to analyzing the pattern of the DDOS attacks and protects users from attacks. This paper presents a of research work on Detection from DDOS attack recognize by ASVM techniques with the use of identifying DDOS attack patterns and analyze patterns by using machine learning algorithms. The packet instrument Wireshark and ASVM is employed to implement the projected system. The results show that the proposed detection of DDoS attack using ASVM prototype has high detection accuracy (99%) decrease of the false positives and false negatives rates compared to conventional detection models.

KEYWORDS

DDOS, Machine Learning, KNN, ASVM and Wireshark.

Author for Correspondence:

Raman V V R,
Department of Computer Science,
Aurora's Degree and P.G. College,
Hyderabad, Telangana, India.

Email: ramanvvr@adc.edu.in

INTRODUCTION

Now a day's distributed denial-of-service (DDoS) attacks are becoming big challenges for industry which is more powerful and more sophisticated. While the growing of industry availability of attack tools and global botnets[1], the pool of possible attacks is also increase. Trusting on humans to prevent and block attacks is simply not enough, and organizations reliant on manual-based DDoS protection and mitigation services are not fully protected from today's threats.

"According to[2], the term Distributed Denial of Service (DDoS) was originally coined by Gligor in an operating system context[3], but since became widely adopted". If the DoS attack involving more than one computer to target a victim in a coordinated manner is called a Distributed Denial of Service (DDoS) attack. The detection of attack focuses on using machine-learning techniques methods[4]. DDoS attacks are a critical issue for companies that have been integrating their technology to public networks, allowing multiple attackers to access data or render services to large companies or countries[5].

A DDoS attack consists[6] into throw tens or hundreds of thousands of requests per second to a server from different locations or IPs; the concept of "Distributor" is concerning that these requests are made from hundreds of thousands of infected machines (commonly called "zombies") which are governed by "botnets" in a coordinated way at the same time, i.e. Smurf attacks, SYN Flood[7], which area sum of bandwidth, memory usage and target's processing, usually no servers could handle ending in a collapse of service because it cannot answer every request[8]; therefore it's necessary the development of new techniques and prototypes to detect fraudulent attacks of concurrent requests in an effective and efficient way also it's necessary in order to avoid the unavailability of service and economic losses.

In this paper, we proposed existing Advance Support Vector Machine (ASVM) algorithm an improvement that is the Advanced Support Vector Machine (ASVM) technique to detect DDoS attacks. We have explored three research problems with our proposed technique[9]. The first problem is the extension of the multiclass problem in the Advance Support Vector Machine (ASVM) algorithm. If the ASVM algorithm is applied in a DDoS attack detection problem on an IoT network, some of the network traffic attributes are multi value attributes[10].

However, the ASVM is originally designed for a binary arrangement. Therefore, multiclass classification is a big problem for applying ASVM. The second problem is that ASVM algorithm will take more time for training and testing methods. The ASVM classifier gives a high classification accuracy and low false-positive rate. The third problem is the efficiency of the centralized network. Therefore, the most important issue for our proposed network infrastructure in the use multiple controllers.

We create test cases of the proposed model by using Miniedit and Open Daylight controllers[11]. In the traffic generation process, we generate normal traffics, UDP flooding DDoS attack traffics[12], and SYN flooding DDoS attack traffics[3]. We can collect the traffic packet from each switch in the traffic collection process. In the feature generation process, average number of packets in a flow, generate the volumetric features, average number of flow bytes and the asymmetric features, amount of packet variations in a flow, the variation of byte flow and the average duration of traffics in the sampling interval.

We propose the Advance Support Vector Machine (ASVM) method for classification evaluation. In this evaluation process is evaluate the classification result by measuring false alarm rate, detection rate, and accuracy.

Internet of Thinks (IoT)

IoT gathers and analyzes real data from the physical world and translates it into workable uses based on industry; market; consumer; or even AI, algorithmic and programmatic needs. In the architecture combines and types of layers like connective layering, core services layers, cloud-based data centers, such as Ethernet, 4G and 5G, embedded and sensory-based learning's[13].

The IoT network architecture includes a few simple layers: Collection, Operational and Distribution. IoT network architecture components: The major component in IoT are six i.e. sensor, communication, security, gateway, platform and application[14].

From home appliances to office devices, our "anytime, anywhere" needs require every peripheral to connect to the internet and our smartphones. But concurrently, the new IT landscape has created a massive attack vector. The Sonic Wall's Annual Threat Report discovered a 217% increase in IoT attacks, while their Q3 Threat Data Report discovered 25 million attacks in the third quarter alone, a 33% increase that shows the continued relevance of IoT attacks in 2020[15].

IoT devices collect our private data for seemingly genuine purposes, but when a hacker gains access to those devices, they offer the perfect means for spying and tracking. IoT devices were usually exploited for creating botnet armies to launch distributed denial-of-service attacks, but in April 2019, Microsoft announced that[16]. Russian state-sponsored hackers used IoT devices to breach corporate networks. The attackers initially broke into a voice over IP phone, an office printer and a video decoder and then used that foothold to scan for other vulnerabilities within their target's internal networks[17].

Security technologies are necessary to protect IoT devices and platforms from breaches. Connected devices that have been in use for many years must communicate safely and securely with newer connected devices[14].

The Detection of DDoS Attack on IoT by Using Advanced Support Vector Machine (ASVM): the DDoS attack will be detected on the IoT network by using the Advanced Support Vector Machine (ASVM) method. The proposed research presents a customizable DDoS defense framework which generates DDoS attack alerts by considering the application’s security requirements[18]. Our proposed framework has been motivated by the concept that different applications required different security. From our future framework, a DDoS attack detection solution must include a customizable reaction mechanism for generating DDoS attack alerts.

In our proposed system leverages the programming and dynamic nature in IoT and implements an adaptive DDoS protection mechanism. Figure No.3 illustrates the architecture of the proposed framework[16].

Attackers or normal users have been sent the packets to the Open Flow Switches. When the packet arrives at the Open Flow switch, the packet information will be checked such as the information on the packet header fields including source port, destination port, source IP address, and destination IP address[19]. The information of the incoming packets will be checked against the flow entries, if a match is found then a specified action can be executed. Otherwise, the packet will be sent to the Open Daylight controller via the southbound API using a packet in control message. Controllers are connected as a cluster[17]. The Open Daylight controller the traffics when it is arrived at cluster, they will be forwarded via the northbound API to the Detection of DDoS attack by ASVM of application layer.

The packet will be categorized as a DDoS attack traffic or a normal traffic[20]. The components of our proposed framework consist of four modules including the traffic data collection, the feature extraction, and the classification of attack or normal and the traffic generation by ASVM method. The flooding-based DDoS attacks and normal traffics are two kinds of are generated. We have collected the traffic data from each Open Flowswitch[21].

The ASVM method have been five features which is extracted and classified as DDoS attacks or normal traffics by. The graphical representation of these modules can be seen in Figure No.4.

Traffic Generation

The generation of two normal traffics and DDoS attack traffics are implemented in this work. The two DDoS attacks are SYN flooding attacks and UDP flooding attacks[14]. The UDP flooding attack is a one type of DDoS attack in which the random ports on the target’s host will be flooded with IP packets using User Datagram Protocol (UDP). Under a UDP flooding attack, first, the victim’s IP addresses are determined[22]; then the source port and the destination port are initialized to 1 and 80. Each time, 2000 packets are generated. The packets interarrival time for UDP attack traffics is 0.03 seconds[23]. The Scapy, a packet generation tool for computer networks written in python language, is used for generating the packets in this work.

The packet is created with the source IP for each random source IP address and the destination IP using scapy. The Scapy can forge or decode packets; The Scapy can send the packets on the wire; The Scapy can capture the packets; and Scapy can match the requests and the replies. The Scapy can also handle tasks like probing, unit tests, attacks, scanning, trace routing and network discovery[24]. After the packet is created, it must be sent to the destination IP address within the time interval. The step by step process of the UDP flooding attack on the IoT network can be seen in Figure No.5.

The type of DDoS attack in the SYN flooding attack is that exploits the normal three-way handshake procedure to consume the resources on the targeted server and render it unresponsive by using the TCP connection[18]. Under a SYN flooding attack, the number of packets, the victim IP addresses and the victim Port must be determined. Then, the victim IP and IP packet with a random source IP will be generated. We also need to create the TCP packet with a random source port, packet sequence, and time window, the victim port and 's' flag.

At last, both TCP packets and the IP will be sent to the victim host. The step by step process of a SYN flooding attack on the IoT network can be seen in Figure No.6.

The normal traffics are also produced as shown in Figure No.7.

The last number of host's destination IP address for a normal traffic generation must be determined. Every time, 1000 packets are generated because the average number of packets at a normal condition is approximately 1000 packets. The packets interarrival time for normal traffic generation is 0.1 second. The each time the random source IP address is used. The Scapy is also used for creating the normal traffic packets to be sent to the destination host.

Traffic Data Collection

For the detection of a DDoS attack on aIoT network, the traffic data collection is the main part of the system. We can use the Open Flow protocol to collect the traffic data information from the Open Flow switches[25]. In IoT, the traffic data are stored in the flow table within the Open Flow switches. When we want to extract the traffic data, the Open Flow switch responds to the on p_flow_stats_requst message and periodically sends this request message to the controller[21].

The Open Daylight controller is used in our research to manage and control the data-obtaining period and Flow-deleting period within the time interval[26]. In order to collect the traffic data, we can send the Flow request command, "sudoovs-ofctl dump-.flows s1" to each switch. Flow information of the Flow table. An example of the extracted traffic. Flow information from a switch is shown in Figure No.8.

ASVM Classification of Attack or Normal Traffic

The Advanced Support Vector Machine (ASVM) method is utilized to classify each packet to be attack or normal traffic. ASVM is a supervised machine learning algorithm that can be used on both classification and regression problems[27]. ASVM is widely used in many application areas because of its high accuracy, ability to deal with high-dimensional data, and flexibility in Modelling diverse data[28]. ASVM is originally used for liner two-class classification problems. In a sample linear two-class classification problem, the assumption is that there are two classes -1 (negative class) and +1 (positive class), Small letter '*x*' denotes a vector with components x_i.

The dataset of n points can be shown as.

$$D = \{(x_i, y_i)\}_{i=1}^{n} \qquad (1)$$

Where x_i denotes the i^{th} characteristic vector in a dataset and y_i is the label associated with x_i. The value of y_i is +1 or -1. The example of linear classification by ASVM is shown in Figure No.8.

According to Figure No.6, there is a straight line separating the vector of class +1 from the vector of class -1. Is straight line is denoted as $w.x+b=0$, where the vector w is called the weight vector and the scalar b is called the bias. The hyperplane of the class label 1 above the straight line is denoted as $w.x+b=1$ and another hyperplane of the class label -1 below the straight line is denoted as $w.x+b=-1$. When the dataset is linearly separable, this two hyperplanes can be seen as parallel and the distance between them must be as large as possible. The distance between them is calculated as follows:

$$distance\ between\ two\ hyperplane = \frac{2}{||w||} \qquad (2)$$

Therefore, the distance between the planes must be maximized. As a result, $w^2/2$ must be minimized. We also need to consider the avoidance of the data points from falling into the margin. We need to add the constraint for each "i" either $w \cdot X_i - b \geq 1$if y_i= - 1 or $w \cdot X_i - b \leq -1$, if y_i=-1.

The constraint for each data points need to be lied at the correct side of the margin which is $y_i\,(w \cdot X_i - b) \geq$

1, for all $1 \leq i \leq n$. Therefore, the optimization problem here is minimize $||w||^2/2$ subject to $y_i (w \cdot X_i - b) \geq 1$, for i., n. In practice, the data are not linearly separable. There are multiclass. Now and again, the maximization of margin can cause an error because of a misclassification of the data. In this work, we improve the ASVM with Advanced Support Vector Machine (ASVM). We need to deliberate the slack variables (ξ_i) and the classification error (*C*). Slack variable is the variable that measures the distance of the point to its marginal hyperplane[29].

The optimal problem is shown in the following equation 3:

$$\text{minimize } \frac{||w||}{2} + C \sum_{i=1}^{n} \xi i \qquad (3)$$

Subject to $y_i(w.x_i - b) \geq 1 - \xi\_i, \xi\_i \geq 0$.

The classification error, C > 0, gives the relative rank of maximizing the margin and minimizing the amount of slack. In a multiclass classification problem, we need to consider the classifier sentence including one-versus-one and one-versus-some. In one-versus-one, the classification pattern is constructed as $n(n - 1)/2$. There are two classes. The first class sample is trained as a positive sample and the second class sample is trained as a negative one. All of these classifiers are needed to classify the data in the testing phases. In one-against-some, the classification pattern is constructed such that each class is trained with the remaining n - 1 class. One class of the sample is denoted as positive, and all other samples are denoted as negatives.

When we make a decision, it is needed to produce a real-valued confidence score. When we use the ASVM algorithm in the classification problem, the most important thing is choosing the kernel function. The Kernel function K(x_n, x_i) is takings the dataset into a higher dimension space in order to make it possible to separate the data[30]. This work in kernel function of the form

$$(x_n.x_i) \leftarrow K\,(x_n, x_i) = (\,\phi(x_n).\,\phi(x_i)), \qquad (4)$$

Where x_n is the support vector data with n = 1, 2, 3, 4, n. The most useful kernel functions of ASVM algorithm are a linear kernel function, Radial Basis Function (RBF), sigmoid, and polynomial. Kernel functions are listed in Table No.1.

In this system, we have detected SYN flooding attacks and UDP. Nature of both attacks is normal distribution. In this work, OVS (one-versus-some) decision function and linear kernel are used for classifying the DDoS attack and the normal traffics.

Experimental Result and Analysis

The experiments in this work are conducted on the Mininet (version 2.3.0d1) emulator in order to create the IoT network topology on an Ubuntu 16.04 VMware. Our VMware is implemented with 1processors, 1MB of RAM, and 20 GB of hard disk. There are the varieties of different controllers: Floodlight, NOX, Ryu, ONOS, POX, and Open Daylight. Among them, the network topology is used for controlling through Open Daylight (version Beryllium) controller. Open Daylight is an open source Java based IoT controller that is supported by VMware, managed by the Linux Foundation[31].

The Open Daylight controller has a very large platform with a lot of plugins and features. Mininet is a network emulator that runs the collection of routers, end-hosts, switches, and links on a single Linux kernel, and its results are as same as a real network[32]. Most DDoS attacks use at least three hosts, at least one switch is used, and the number of hosts can be up to approximately one hundred hosts; and the number of controllers can range from one to as much as possible used.

Our IoT test bed consists of one hundred hosts (h1 to h100), nine switches (s1 to s9), and three controllers (c0, c1, c2). Four subnets are arranged in our test bed. The experiments are set up on Miniedit. Miniedit is a simple GUI editor for Mininet. Figure No.9 shows our implemented test bed.

After running the test bed, the network flows have been added to the nine switches. Open Flow protocol (version Open Flow13) and Open Virtual Switch (OVS) are used in our test bed. OVS is a multilayer virtual switch, production quality which has licensed under the open source Apache 2.0 license[33]. We have been the command, for example, in switch s1 as "shovs-ofctl add-flow s1 in-port=1, action=flood" at our test bed terminal. 126 flows are added for nine switches[34].

In our test bed, each traffic type is generated from 100 scenarios. There are three types of traffics including UDP flooding attacks, SYN flooding attacks and normal traffics. Under a UDP flooding attack scenario, we use at least nine hosts and five hosts as the attacker hosts and four hosts as the victims.

The four hosts are assigned as attacker hosts and only one victim host under a SYN flooding attack. The traffic generation is started first then the traffic flow information in each scenario and from each switch will be manually collected from each switch[35]. The collection of traffic data and after processing the generation for each scenario, five different traffic features are extracted in order for the ASVM to detect the DDoS attack. In this experiment normal traffic is 200 seconds and the sampling traffic collection time for attack traffics. The result of the first feature and ANPI for normal traffics is shown in Figure No.10.

In the within sampling time the trend of the curve has gradually fluctuated. The attack traffics in ANPI feature are shown in Figure No.11.

During the numbers of packets, the attack period is growing rapidly. The trend of the curve is fluctuated at first, and sometimes, the value reached the highest point depending on the randomly generated attack traffic packets. The result of the second feature of ANBI in the sampling interval for normal traffics is shown in Figure No.12.

The trend of the curve is fluctuated depending on the number of flow bytes for the normal traffics. The value of ANBI for attack traffics within the sampling time is expressed in Figure No.13.

The attackers send a large number of packets as fast as possible, but they do not consider the data value. Therefore, the ANBI value of attack traffic is commonly from up to down and sometimes speciously reaches the highest point. The result of the third feature, VPI for normal traffics is shown in Figure No.14.

Normally, the variation relatively unchanged of the flow packets. For the attack case, however, the VPI changes quickly. The attack traffics of VPI curve trend is shown in Figure No.15.

When the attacks occur within the sampling time, the variation of traffics has fluctuated, and sometimes, it reaches the highest point. The normal traffics of VBI are the result of the fourth feature, shown in Figure No.16.

The trend of the curve is gradually fluctuated, and sometimes, it reaches the lowest

Points at sampling time 65 and 169 seconds. When the attack arises in the sampling time, the attackers did not reflect the flow byte values of the sending packets. Therefore, the curve trend progressively grows up and down as shown in Figure No.17.

The result of the last feature, ADTI for normal traffics and attack traffics, is shown in Figures No.18 and No.19, respectively.

The curve of both types is the same, but the ADTI value of the attack traffics is apparently greater than that of the normal traffics. The extracted features from the traffic data have been stored as the feature dataset, namely, IoT traffic DS. The next step is the classification of these dataset by the ASVM method[36,37]. The classification process is shown in Figure No.20.

First, IoT traffic DS is read and the Type field and the last fields are separated. The data is then split into Training DS and Testing DS using a cross-validation method in order to reduce an over threating problem[38]. Next, the ASVM model is produced by using the Training DS. The Linear kernel, classification error “C” ($C > 0$), OVS decision function and the auto Gamma value are used in our ASVM. After the training process is done, the resulting model is used for categorizing the Testing DS[26]. The confusion matrix is used for the performance evaluation of the classification results. The classification report for three classes is generated. Finally, the accuracy of our proposed organization result from the Training DS and the Testing DS is also produced.

Evaluation of Prediction Result

Training DS and testing DS are multidimensional data. We have solved our research’s first problem of multiclass training time and testing time of ASVM algorithm has been solved by using the linear kernel with penalty parameter of the classification error term, ‘C,’ allowing for the value of “gamma” and “OVS” decision function shape. Detection rate, accuracy and False alarm rate are used for evaluating our detection result. False alarm rate is the error rate of our detection system that is the incorrect result on a normal behaviour[39].

Thus, less false alarm rate is preferred. Detection rate is the correct rate for detecting the malicious traffics. The higher detection rate is the better system performance. Accuracy is the measurement of the system that correctly categorizes in two normal traffics and malicious traffics. All three measures equation is shown in the following equations:

$$False\ alarm\ Rate = \frac{FP}{TP + FP} * 100\ \%\ FP$$

$$Detection\ Rate = \frac{TP}{TP + FN} * 100\%$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} * 100\%$$

True positive (TP) is the amount of network traffics that are correctly detected attack or normal traffics and forwarded. The amount of network traffics is True Negative (TN) is that are correctly detected and dropped. The amount of network traffics is False Positive (FP) that are incorrectly detected and forwarded.

The amount of network traffics is False Negative (FN) that are incorrectly detected and dropped. In this experiment, we have been trained and tested with the cross validation method of splitting rate from 10% to 90% of IoT Traffics DS. The experimental result can be seen in Table No.2.

According to the experimental results shown in Table No.2, the average accuracy of the detection is 0.97, the average false alarm rate is 0.02, and the average detection rate is 0.97. The testing time and training time for each rate are approximately 55 seconds and 50 seconds, respectively.

Table No.1: Different kernel function

S.No	Kernel functions	Formula
1	linear	$K(x_n, x_i) = (x_n, x_i)$
2	RBF	$K(x_n, x_i) = \exp(-\gamma\|x_n - x_i\|^2 + C)$
3	Sigmoid	$K(x_n, x_i) = \tanh(\gamma(x_n, x_i) + r)$
4	Polynomial	$K(x_n, x_i) = (\gamma(x_n, x_i) + r)^d$

Table No.2: The experimental result

Spit Rate	Training Data (%)	Testing Data (%)	False alarm rate	Detection rate	Accuracy
0.1	90	10	0	1	1
0.2	80	20	0.06	0.93	0.93
0.3	70	30	0.02	0.98	0.97
0.4	60	40	0.03	0.97	0.96
0.5	50	50	0.01	0.99	0.98
0.6	40	60	0.01	0.99	0.98
0.7	30	70	0.01	0.99	0.98
0.8	20	80	0.03	0.96	0.96
0.9	10	90	0.03	0.97	0.97

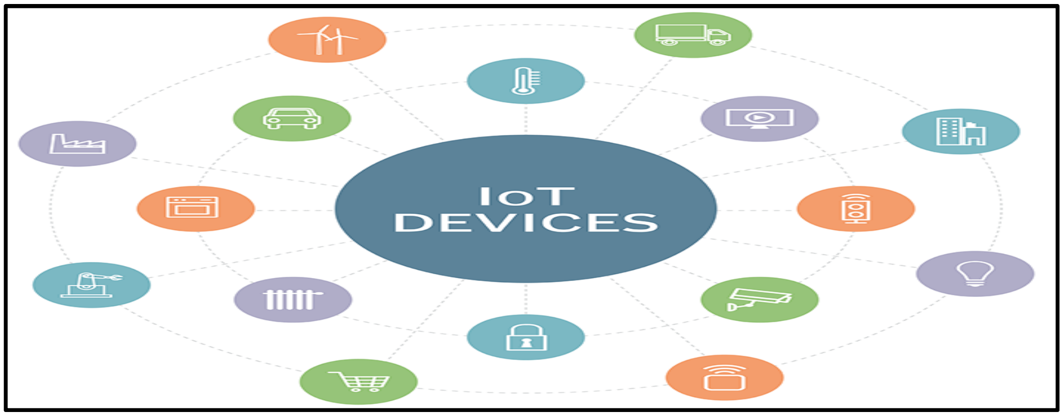


Figure No.1: Use of IoT Device

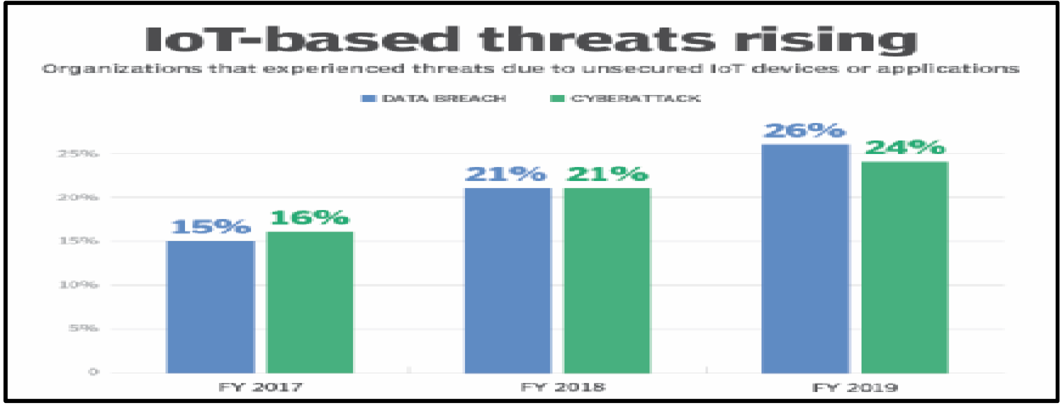


Figure No.2: Last 3 Year Attack report

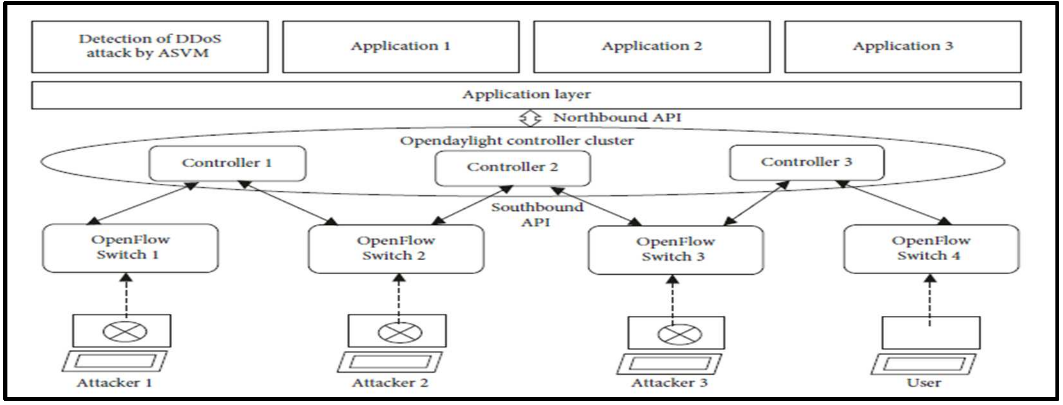


Figure No.3: The proposed IoT Network based DDoS attack detection framework

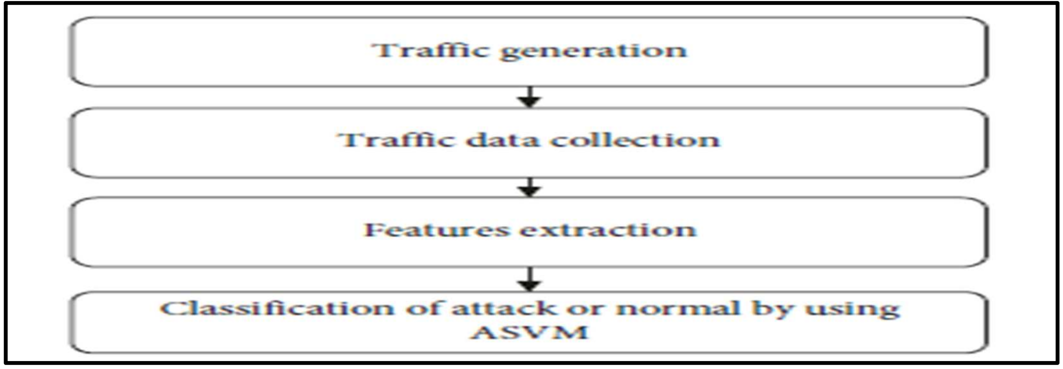


Figure No.4: Proposed system framework with four modules

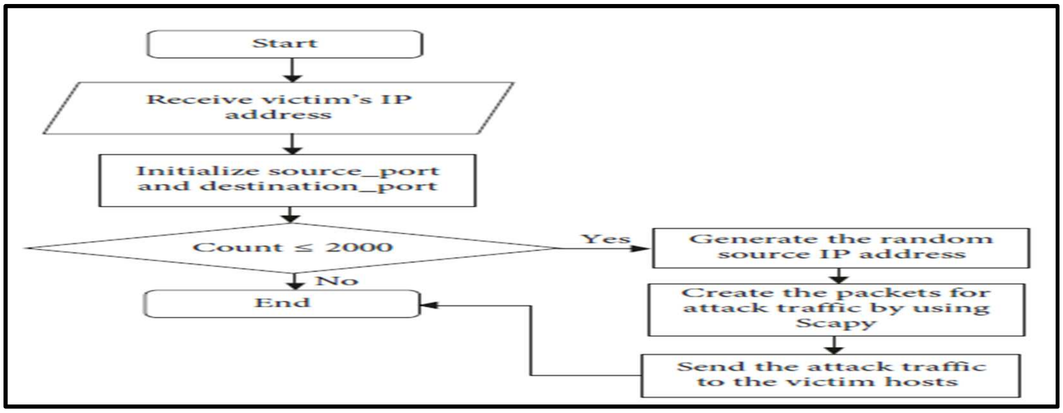

Start
Receive victim's IP address
Initialize source_port and destination_port
Count ≤ 2000
Yes
No
End
Generate the random source IP address
Create the packets for attack traffic by using Scapy
Send the attack traffic to the victim hosts

Figure No.5: UDP attack Step by step process

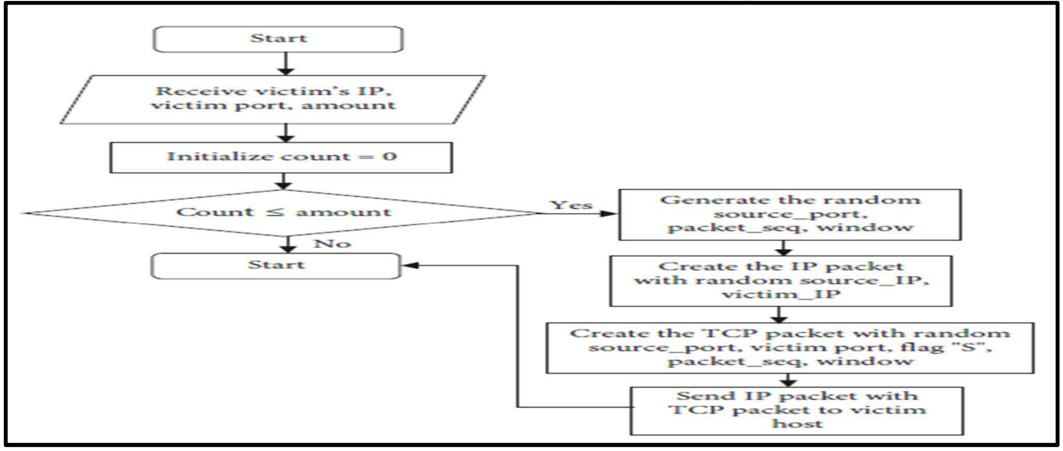

Start
Receive victim's IP, victim port, amount
Initialize count = 0
Count ≤ amount
Yes
No
Start
Generate the random source_port, packet_seq, window
Create the IP packet with random source_IP, victim_IP
Create the TCP packet with random source_port, victim port, flag "S", packet_seq, window
Send IP packet with TCP packet to victim host

Figure No.6: SNY Flooding attack process step by step

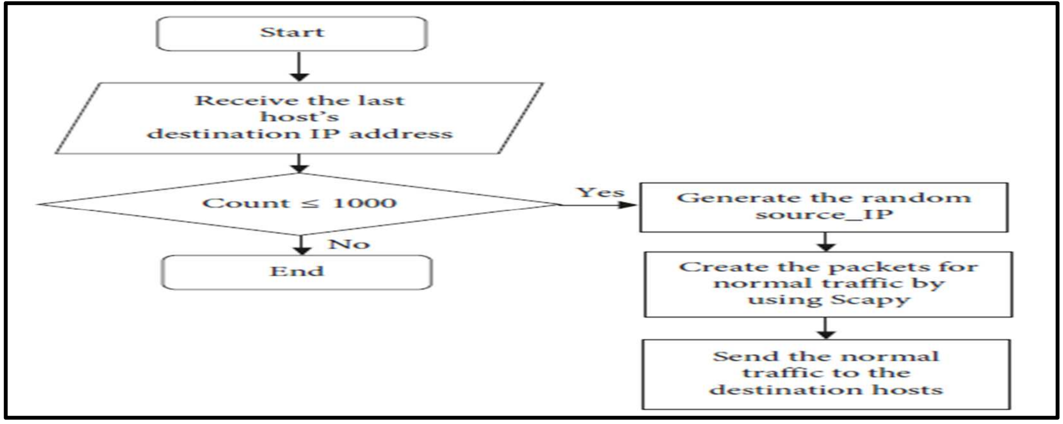
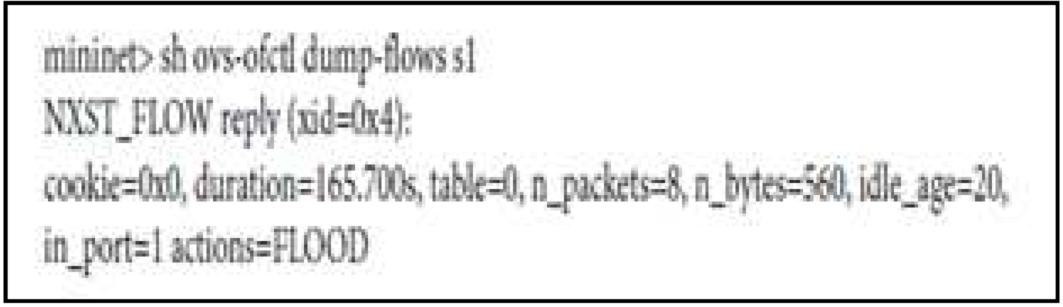

Start
Receive the last host's destination IP address
Count ≤ 1000
Yes
No
End
Generate the random source_IP
Create the packets for normal traffic by using Scapy
Send the normal traffic to the destination hosts

Figure No.7: Normal traffic generation process step by step



```
mininet> sh ovs-ofctl dump-flows s1
NXST_FLOW reply (xid=0x4):
cookie=0x0, duration=165.700s, table=0, n_packets=8, n_bytes=560, idle_age=20,
in_port=1 actions=FLOOD
```

Figure No.8: The traffic flow information example from a switch

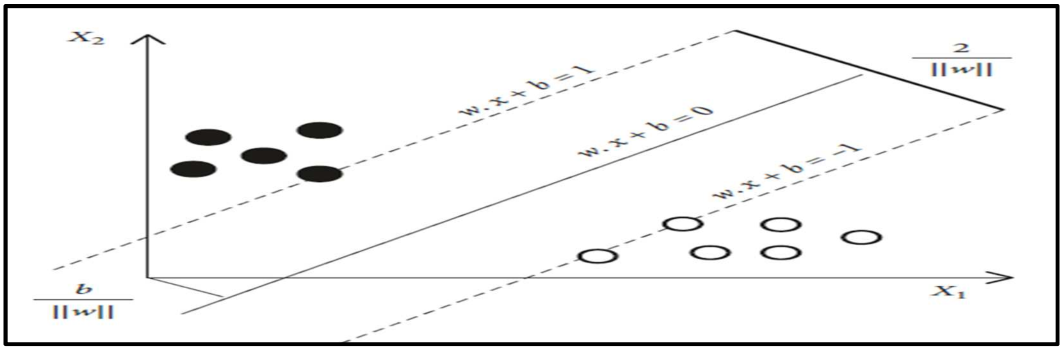


Figure No.9: The Linear ASVM

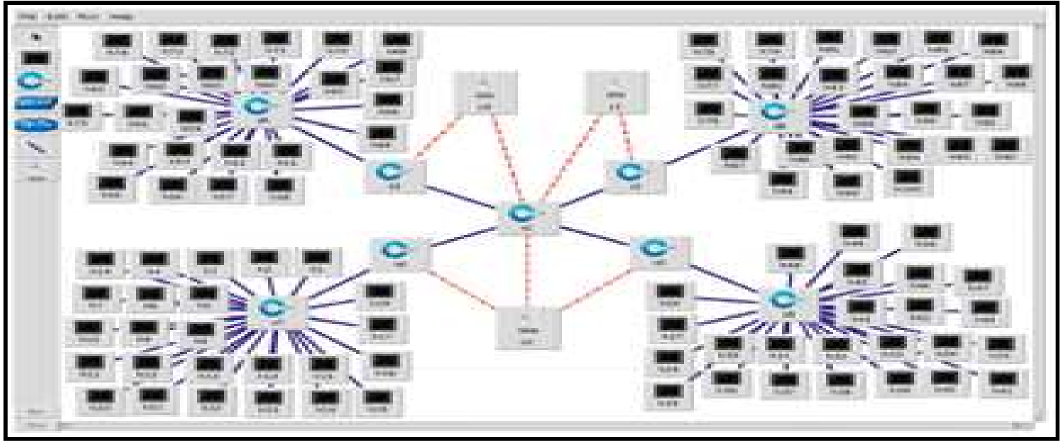


Figure No.9: DDoS attack by using ASVM in IoT test bed for detection

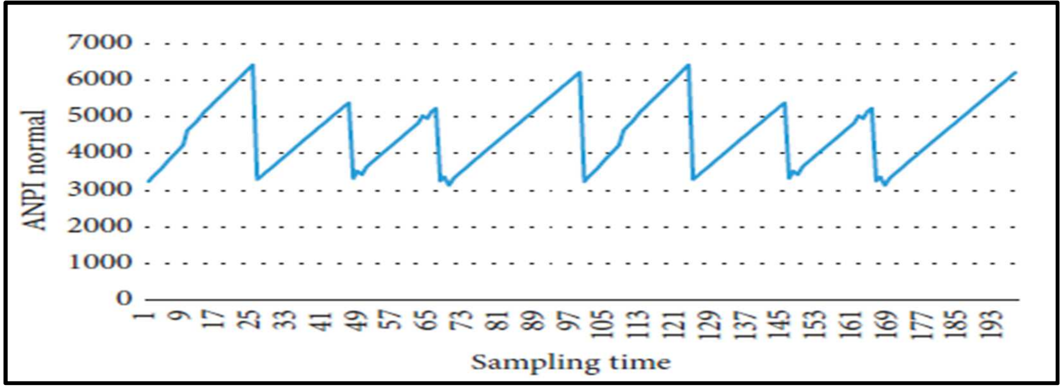


Figure No.10: Feature of ANPI for normal traffics

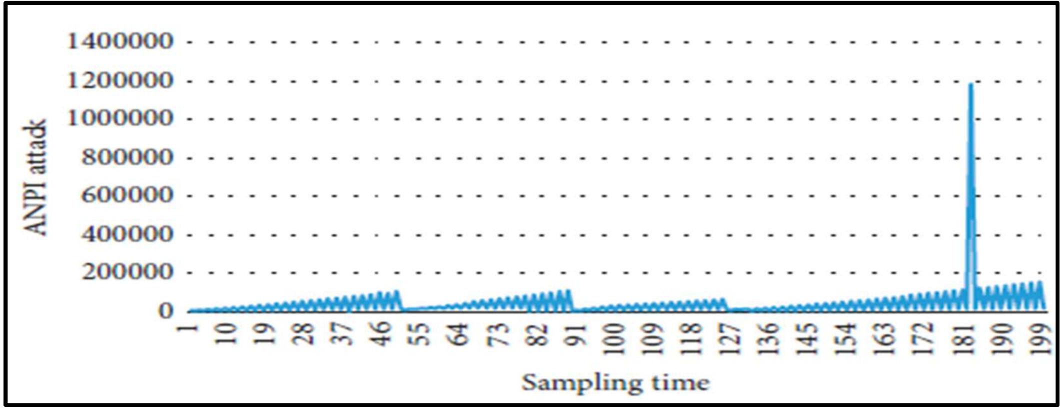


Figure No.11: Feature of ANPI for attack traffics

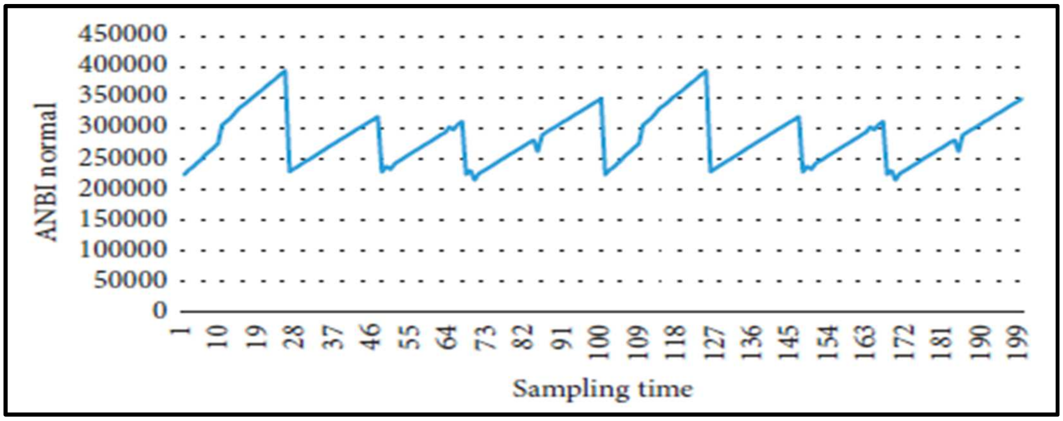


Figure No.12: Feature of ANBI for normal traffics

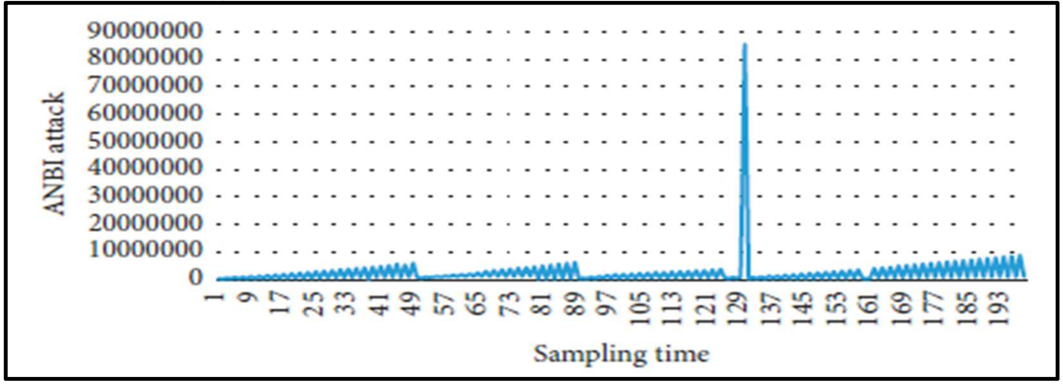


Figure No.13: Feature of ANBI for attack traffics

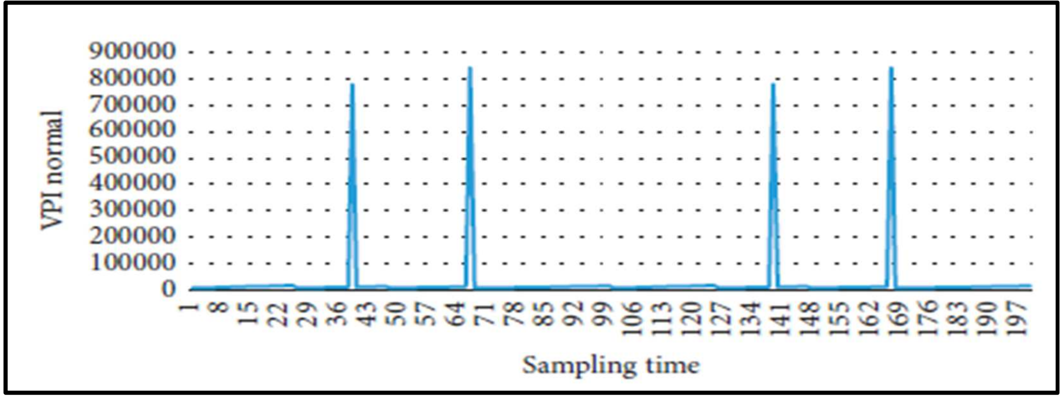


Figure No.14: Feature of VPI for normal traffics

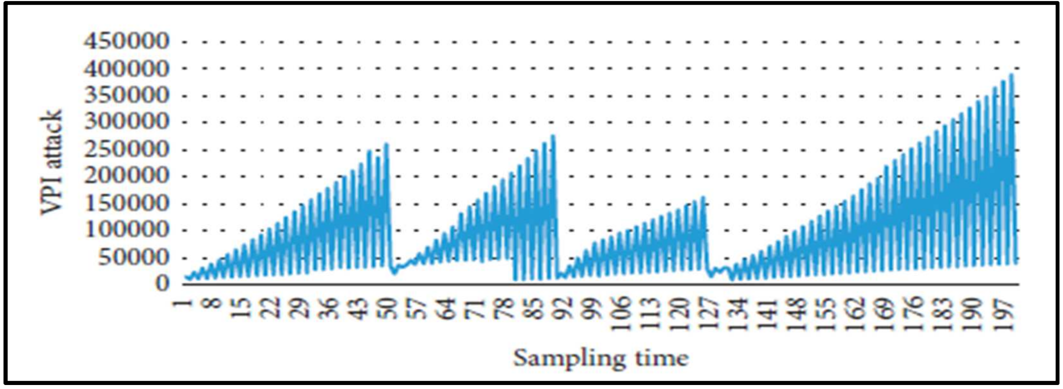


Figure No.15: Feature of VPI for attack traffics

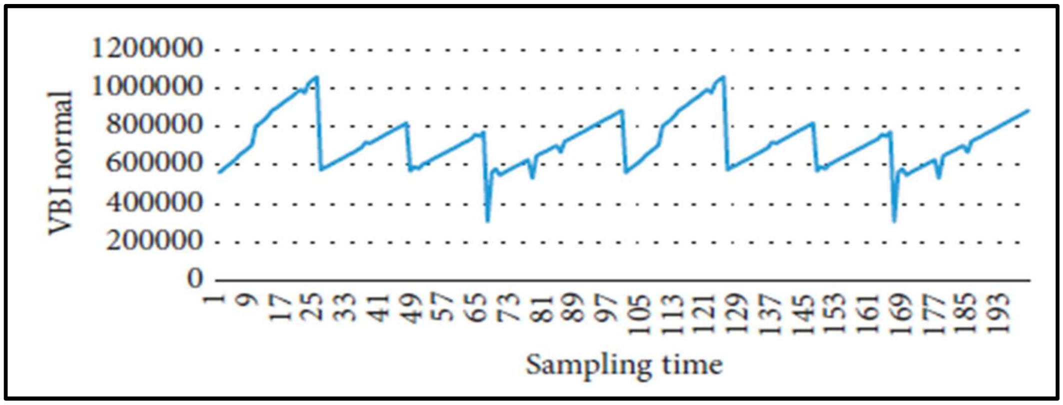


Figure No.16: Feature of VBI for normal traffics

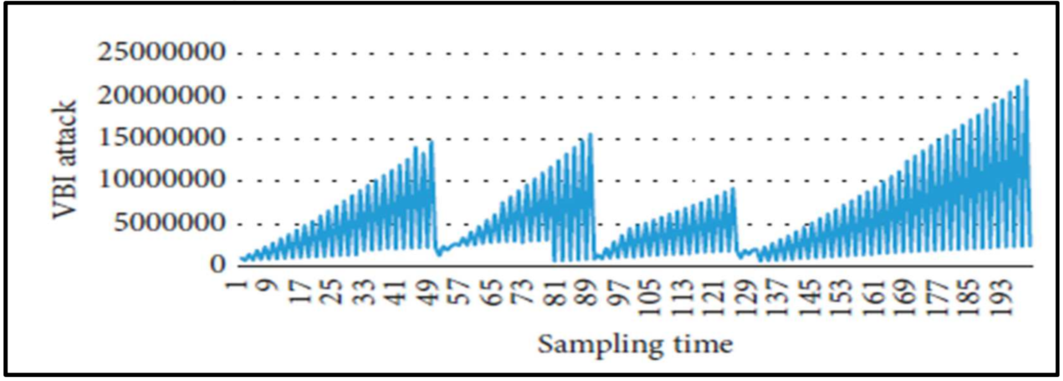


Figure No.17: Feature of VBI for attack traffics

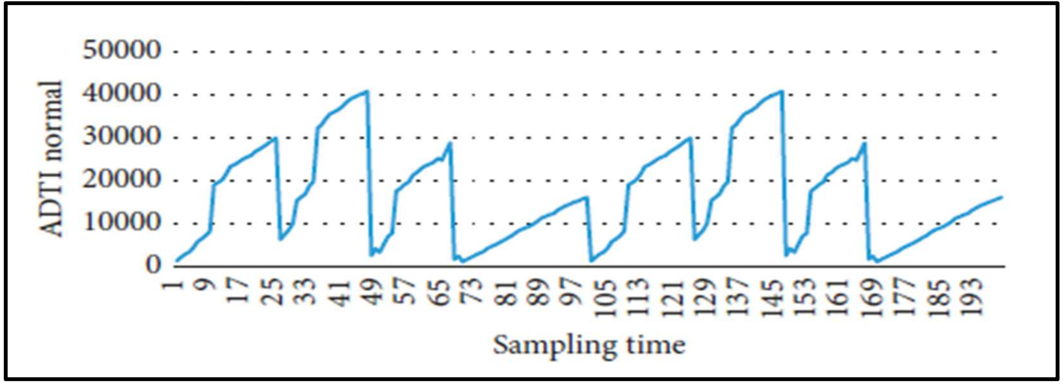


Figure No.18: Feature of ADTI for normal traffics

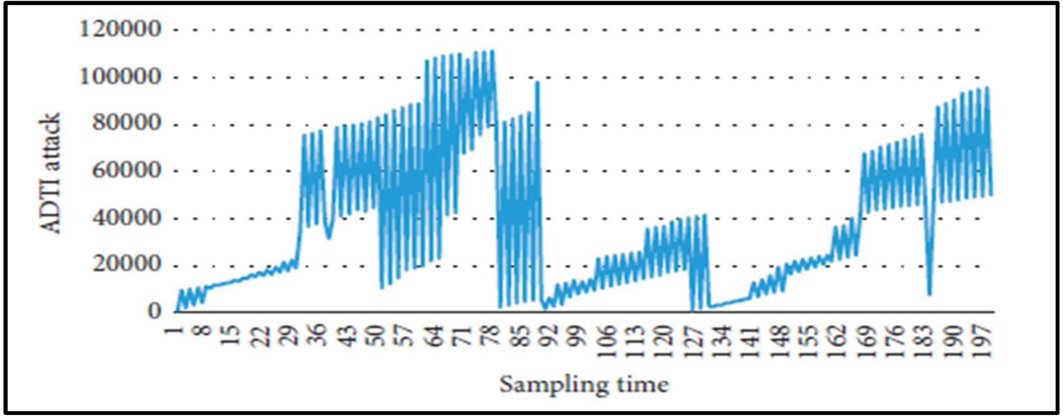


Figure No.19: Feature of ADTI for attack traffics

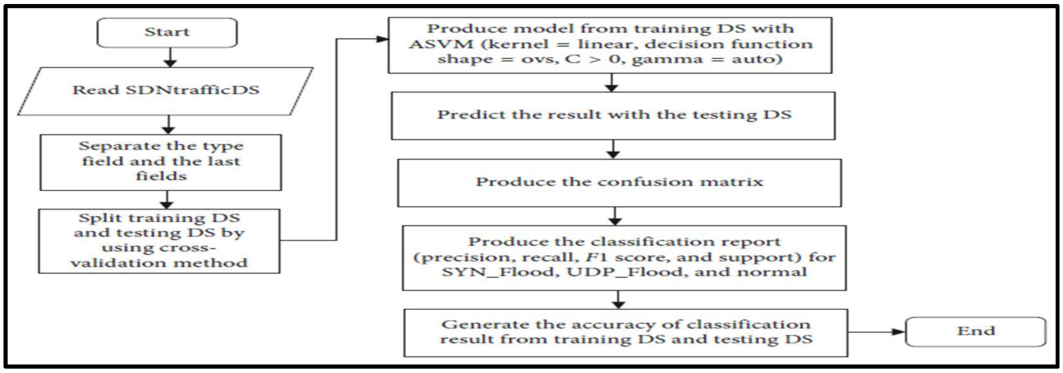

Start
Read SDNtrafficDS
Separate the type field and the last fields
Split training DS and testing DS by using cross-validation method
Produce model from training DS with ASVM (kernel = linear, decision function shape = ovs, C > 0, gamma = auto)
Predict the result with the testing DS
Produce the confusion matrix
Produce the classification report (precision, recall, F1 score, and support) for SYN_Flood, UDP_Flood, and normal
Generate the accuracy of classification result from training DS and testing DS
End

Figure No.20: The proposed classification method for System flow

CONCLUSION

In this paper, we proposed a way to detect two flooding based DDoS attacks using the proposed advanced ASVM method. Nowadays, most researches in the detection of the DDoS attack on IoT network used traditional networking dataset. In this work, a new dataset, IoT Traffics DS, is generated and used. Our emulated test bed is conducted using Mininet. In our test bed, nine switches, one hundred hosts, and three controllers are used. The existing researches in the security of SDN network used a single controller in their network setting.

In this work, on the other hand, three controllers are used. The one of controller has down because of the attack, another controller can still in 9-used. We used hundred scenarios for SYN flooding attacks and another one hundred scenarios for UDP flooding attack. Both normal traffic data and malicious traffic data are generated. The IoT traffic from the Open Flow switches is collected. The volumetric and asymmetric features from the IoT traffic are collected and extracted to create the dataset.

The Cross-validation method is testing the classification model and employed while training. Linear kernel is used in our ASVM algorithm. As a result, testing time and the training is reduced. The parameter of classification error (C), decision function shape (OVS) and gamma value, is considered. According to the experimental results 97% is and the overall accuracy of the proposed model. Our future works include an online detection system for DDoS attack on IoT network. In addition, other attack planes of IoT layer must also be consider.

ACKNOWLEDGEMENT

The authors which to express their sincere gratitude to Department of Computer Science, Aurora's Degree and P.G. College, Hyderabad, Telangana, India for providing necessary facilities to carry out this research work.

CONFLICT OF INTEREST

We declare that we have no conflict of interest.

BIBLIOGRAPHY

1. Akamai. State of the internet/security, *SOTI,* 4(5), 2018, 1-18.
2. Lakshminarayanan K, Adkins D, Perrig A, Stoica I. Taming ip packet flooding attacks, *ACM SIGCOMM Comp Comm. Review,* 34(1), 2004, 45-50.
3. Bani-Hani R and Al-Ali Z. SYN flooding attacks and countermeasures: a survey, *In Proceedings of ICICS, Beijing, China,* 2013, 1-7.
4. Akamai. Memcached Reflection Attacks: A NEW era for DDoS, *Akamai Technologies, Cambridge, MA, USA,* 2018, 1-2.
5. Lukaseder T, Shreya G, Frank K. Mitigation of flooding and slow DDoS attacks in a software-defined network, *In Proceedings of Cryptography and Security, Santa Barbara, CA, USA,* 2018, 1-3.
6. Badotra S and Singh J. Open daylight as a controller for software defined networking, *International Journal of Advanced Computer,* 8(5), 2017, 1105-1111.

7. Yihunie F, Eman A, Odeh A. Analysis of ping of death DoS and DDoS attacks, *In Proceedings of IEEE Long Island Systems, Applications and Technology Conference (LISAT), Farmingdale, NY, USA,* 2018, 1-4.
8. Bogdanoski M, Risteski A, Shuminoski T. TCP SYN flooding attack in wireless networks, *In proceedings of the conference: Innovations on communication seory, INCT, Istanbul, Turkey,* 2012, 1-8.
9. Gligor V D. A note on denial-of-service in operating systems, *IEEE Transactions on SE,* 10(3), 1984, 320-324.
10. Zakaria Bawany N and Shamsi J A. Application layer DDoS attack defense framework for Smart city using IoT," *In third International Conference on Computer Science, Computer Engineering, and Social Media (CSCESM), Thessaloniki, Greece,* 2016, 1-9.
11. Mujtiba S H and Beigh G R. Impact of DDoS attack (UDP flooding) on queuing models, *In Proceedings of the 2013 4th ICCCT, Allahabad, India,* 2013, 210-216.
12. Dang-Van T and Truong H. A multi-criteria based software defined networking system Architecture for DDoS attack mitigation, *REV Journal on Electronics and Communications,* 6(3-4), 2016, 50-60.
13. Gharvirian F and Bohlooli A. Neural network based protection of software defined network controller against distributed denial of service attacks, *IJE,* 30(11), 2017, 1714-1722.
14. https://internetofthingsagenda.techtarget.com /tip/IoT-network-architecture-shaped-by-business-requirements.
15. https://searchcio.techtarget.com/tip/How-IoT-5G-RPA-and-AI-are-opening-doors-to-cybersecurity-threats.
16. https://www.zdnet.com/article/microsoft-russian-state-hackers-are-using-iot-devices-to-breach-enterprise-networks/ https://www.zdnet.com/article/microsoft-russian-state-hackers-are-using-iot-devices-to-breach-enterprise-networks/
17. Anandshree Singh N, Johnson Singh K, De T. DDoS attack detection using naive bayes classifier through info gain feature selection, *In Proceedings of the International conference on informatics and analytics, Pondicherry, India,* 2016, 1-9.
18. MyintOo M, Sinchai K, Ossaporn K E. The design of IoT based detection for distributed denial of service (DDoS) attack, *In Proceedings of the 21st International Computer Science and Engineering Conference, Antalya, Turkey,* 2017.
19. Harshita H. Detection and prevention of ICMP flood DDOS attack, *International Journal of New Technology and Research (IJNTR),* 3(3), 2017, 63-69.
20. Pramana M I W, Purwanto Y, Yosef Suratman F. DDoS detection using modified K-means clustering with chain initialization over landmark window, *In Proceedings of the 2015 International Conference on Control, Electronics, Renewable Energy and Communications (ICCEREC), Bandung, Indonesia,* 2015, 7-11.
21. Kokila R T, Amarai Selvi S, Kannan G. DDoS detection and analysis in IoT-based environment using support vector machine classifier, *In Proceedings of the 2014 6thICoAC, Chennai, India,* 2014.
22. Verma and Kumar Xaxa D. A survey on HTTP flooding attack detection and mitigating methodologies, *International Journal of Innovations and Advancement in Computer Science,* 5(5), 2016, 18-21.
23. Kazuya S, Kentaro S, Nobuyuki T *et al.* A survey on Open Flow technologies, *IEICE Transactions on Communications,* E97.B(2), 2014, 375-386.
24. Chi Wu Y, Tseng H, Yang W, Hong Jan R. DDoS detection and trace back with decision tree and grey relational analysis, *In Proceedings of the 2009 3th ICMUEQ,* 2009, 306-314.
25. Acharya S and Tiwari N. Survey of DDoS attacks based on TCP/IP protocol

vulnerabilities, *IOSR Journal of Computer Engineering,* 18(3), 2016, 68-76.

26. Dayal N, Maity P, Srivastava S, Khondoker R. Research trends in security and DDoS in IoT, *Security and Communication Networks,* 9(18), 2016, 6368-6411.
27. Moore. Cross-validation for detecting and preventing over fitting, *School of Computer Science Carnegie Mellon University,* 2001, 1-63.
28. Evgeniou T and Pontil M. Support vector machines: theory and applications, *Machine Learning and Its Applications: Advanced Lectures,* 2049, 2001, 249-257.
29. Tang F, Tinno P, Gutierrez P A, Chen H. The benefits of modelling slack variables in SVMs, *Neural Computation,* 27(4), 2015, 954-981.
30. Balcan M F, Blum A, Vempala S. On kernels, margins, and low-dimensional mappings," Lecture Notes in Computer Science, *In Proceedings of the 15th International Conference on Algorithmic Learning Theory, Padova, Italy,* 3244, 2004, 194-205.
31. Asadollahi S, Goswami B, Gonsai A M. Implementation of IoT using Open DayLight controller, *An Proceedings of the International Conference on recent trends in IT Innovations, IJIRCCE, India,* 5(2), 2017, 218-227.
32. Keti F and Askar S. Emulation of software defined networks using mininet in different simulation environments, *In Proceedings of the 6th International Conference on Intelligent Systems, Modeling, and Simulation, Kuala Lumpur,* 2015, 1-6.
33. Pfaff. Open vSwitch, 2014. http://www.openvswitch.org//support/slides/brkt.pdf.
34. Kolahi S, Treseangrat K, Sarrafpour B. Analysis of UDP DDoS flood cyber-attack and defense mechanisms on Web Server with Linux Ubuntu 13, *In Proceedings of the 2015 ICCSP and their Applications (ICCSPA'15), London, UK,* 2015, 1-5.
35. Rajneet S. A study of DoS and DDoS-smurf attack and preventive measures, *IJCS Information Technology Research,* 2(4), 2014, 312-317.
36. Linxia L, Leung V C M, Chin-Feng L. Evolutionary algorithms in software defined networks: techniques, applications, and issues, *ZTE Communications,* 15(3), 2017, 20-35.
37. Tauber L. Introducing the normal distribution in a data analysis course: specific meaning contributed by the use of computers, *In Proceedings of the ICOTS 6: the Sixth International Conference on Teaching Statistics, Cape Town, South Africa,* 2002.
38. Benzekki K, El Fergougui A, Elbelrhiti Elalaoui A. Software-defined networking (IoT): A survey, *Security and Communication Networks,* 9(18), 2017, 5803-5833.
39. Ahuja Y and Yadav S K. Multiclass classification and support vector machine, *GJCSTI,* 12(11), 2012, 1-7.

Please cite this article in press as: Raman V V R and Om Prakash Yadav. Detecting DDOS attack using advanced support vector machines (ASVM) in IOT network, *International Journal of Arts and Science Research,* 7(1), 2020, 5-19.