DDoS Detection and Analysis in SDN-based Environment Using Support Vector Machine Classifier

Kokila RT[1], S. Thamarai Selvi[1], Kannan Govindarajan[2]
[1]Department of Computer Technology, Anna University (MIT Campus), Chennai
[2]MIMOS, Malaysia.
{kokilart@gmail.com, stselvi@annauniv.edu, kannan.gridlab@gmail.com}

Abstract*—Software Defined Networking (SDN) provides separation of data plane and control plane. The controller has centralized control of the entire network. SDN offers the ability to program the network and allows dynamic creation of flow policies. The controller is vulnerable to Distributed Denial of Service (DDoS) attacks that leads to resource exhaustion which causes non-reachability of services given by the controller. The detection of DDoS requires adaptive and accurate classifier that does decision making from uncertain information. It is critical to detect the attack in the controller at earlier stage. SVM is widely used classifier with high accuracy and less false positive rate. We analyze the SVM classifier and compare it with other classifiers for DDoS detection. The experiments show that SVM performs accurate classification than others.

Keywords—SDN, DDoS, SVM, OpenFlow, DARPA dataset

I.INTRODUCTION

Computer network consists of a group of devices like switches and routers that are controlled via proprietary interfaces implemented on it. The network administrator is responsible for configuring the network policies in those devices using simple command line interface. This task has to be accomplished with limited tools. The existing network device interfaces are closed and collaboration among multiple vendor software is a challenging issue. It creates a barrier for creating innovations in the networking. Due to the emerging trend of Internet, the network conditions are changing tremendously. It is difficult to perform real world experiments (deployment of new protocol) in a large production environment.

Open Networking Foundation (ONF) defines Software Defined Networking (SDN) as "physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices" [1]. In SDN, the data plane acts simply as packet forwarding hardware and control plane acts as "brain" of the device. The network control plane is easily programmable and it provides an abstraction of the underlying network infrastructure. It simplifies the networking devices and they accept the instructions from the centralized controller (control plane) [2].The operator does not need to configure the network devices individually; the routing and forwarding decisions has been implemented in the centralized SDN controller [3]

978-1-4799-8159-5/14/$31.00©2014 IEEE.

SDN controller interacts with networking devices using southbound API's like OpenFlow and interaction with applications is performed using northbound API's. OpenFlow [22] is a standard that allows researchers to run experimental protocols in the network. It is based on Ethernet switch, which has an internal flow table. It provides a standard interface called OpenFlow protocol to add or modify the entries in the flow table.

The switches in SDN environment can be OpenFlow switches (only forwards the packets) or hybrid OpenFlow-Ethernet switches (bridging, routing along with forwarding). In the traditional network switches or routers, the fast packet forwarding (data plane) and routing decisions (control plane) occurs in the same device. However, OpenFlow switch in SDN environment separates these two functions. The components of OpenFlow switch are shown in Fig 1.

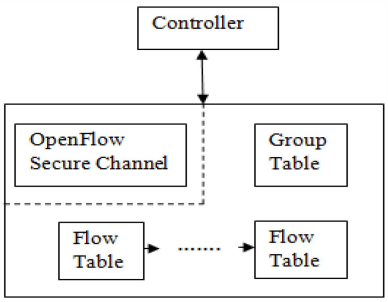

Fig 1 Components of OpenFlow Switch [5]

OpenFlow switch consists of flow tables and group table that is used to perform packet lookup and forwarding. The OpenFlow secure channel connects the switch to external controller. The communication between switch and controller occurs through OpenFlow protocol. The controller uses OpenFlow protocol to create, modify and remove flow entries in the flow table. The secure channel is a TLS (Transport Layer Security) or TCP (Transmission Control Protocol) connection established between the controller and the switch. The Group table consists of group entries and contains actions applicable for packets sent to specific groups. The sample flow table entry is shown in Fig 2.

Packet Header Match Fields	Priority	Counters	Cookies	Timeout	Instructions

Fig 2 Components of flow entry in flow table

There may be many rules for the same flow. Priority specifies matching precedence for an entry. The combination of packet header match fields and priority uniquely identifies an entry in the flow table. Counters will be set for each flow that indicates received packets per flow, received bytes per flow, received packets per port, etc. Cookies are used by the controller to filter the flow statistics. The timeout can be hard timeout or soft timeout. Hard timeout mentions the time needed for a flow to expire since it was installed initially. Soft timeout value mentions the time needed for a flow to expire since the last packet match. The instructions specify the action set for an entry. The packet header fields used to match the flow table entry with the incoming packet to the switch are shown in Fig 3.

Ingress Port	VLAN ID	VLAN Priority	Ethernet			IP			TCP	
			SA	DA	Type	SA	DA	Protocol	Src Port	Dst Port

Fig 3 Packet Header matching fields

The paper is organized as follows. An overview of various DDoS attacks is presented in section II, security issues in SDN controller are discussed in section III. The existing methods for intrusion detection in SDN and network intrusion detection using the concepts of SDN are described in section IV. The Section V discusses about some of the existing multiclass SVM classification methods. The DDoS detection method is presented in section VI. The experimental results and discussions are available in section VII with concluding remarks in section VIII.

I. DISTRIBUTED DENIAL OF SERVICE ATTACK

Brief description of DDoS attack and its major classification are presented here. DDoS attack is launched by multiple compromised computers called as bots or zombies targeting a single system. The four major components of DDoS attack are the real attacker, compromised hosts called as handlers or masters capable of controlling multiple agents using software programs, the agent hosts who generate a large number of packets towards the victim host, and the target host to which the attack is launched. Taxonomies of DDoS, tools used to launch the attack and possible countermeasures are discussed in detail [6].

UPD flood: The victim system attacked by sending UDP (User Datagram Protocol) packets continuously to specific or random port.

ICMP flood attack: Large number of ICMP (Internet Control Message Protocol) echo request (ping flood) packets with spoofed source IP address is sent to the victim.

Smurf attack: Reflection or amplification attack is targeted against routers and servers where the ICMP packets are redirected to these amplifiers with a spoofed source IP address. The spoofed address will be victim host IP address. UDP and ICMP flood attack sources can be easily tracked, but it is difficult to track the source of Smurf attack.

Fraggle attack: Similar to Smurf attack, but uses UDP packet instead of ICMP packet. Here also the victim's IP address is used as a spoofed source IP address in the attack packets.

SNMP amplification attack: SNMP (Simple Network Management Protocol) is used to monitor the devices such as router, printers and firewalls attached to the network. SNMP uses default communication string which allows programs to get the configuration information of the devices. The GetBulk request can be sent to retrieve the configuration details. Attackers send this request using a default communication string with a spoofed source IP address of the target system. Thus the victim system is overwhelmed with responses.

Coremelt Attack: The zombies will be divided into two groups. The attacker instructs the zombies to communicate with the zombie in other group leading to sending and receiving huge data. It is difficult to track this attack as the communication happens via legitimate packets. Instead of targeting the single host, zombies communicate with each other to create network flood [7].

HTTP flood: The web server is flooded with HTTP (Hyper Text Transfer Protocol) requests. It is a volumetric attack and does not belong to reflection or spoofing techniques.

SIP flood attack: Voice over IP (VoIP) communications uses Session Initiation protocol (SIP) for call signaling. SIP phone can be easily flooded with messages so that it cannot serve legitimate requests.

Land Attack: Large numbers of packets are sent with same host and destination IP address and port number that crash the system.

TCP SYN attack: The weakness of the Transmission Control protocol (TCP) is used for launching this attack. The attacker sends large number of SYN (Synchronize) requests to the server. The server replies to the request by sending SYN + ACK (Acknowledge) packet and waits for the ACK packet from the client. Now the attacker doesn't send ACK packet, and the server waits for nonexistent ACK. The limited buffer queue of the server becomes full and incoming valid requests are rejected.

CGI Request attack: The attacker sends large number of Common Gateway Interface (CGI) request that consumes CPU cycles of the victim.

Authentication Server attack: The authentication server verifies the bogus signature sent by the attacker which consumes more resources than generating the signature.

II. SECURITY IN SDN

A. Problem Discussion

OpenFlow switch checks the incoming packet (packet header fields such as source port, destination port, source IP address, destination IP address etc.) against the flow entries, if a match is found then the specified action can be executed. Otherwise, the packet will be sent to the controller using PacketIn control message. When a large number of spoofed IP addresses packets are sent together, there will not be a match found in flow table and packet will be sent to the controller. Using this processing delay the malicious attacker can modify the flow entries and make the legitimate packet to be dropped, clone the flow table entries which leads to overflow in the flow table. There will not be enough memory space to accept the new flow instructions given by the controller. The controller tries to process the legitimate and spoofed packets continuously and its resources are exhausted. This can be described as DDoS attack against the controller. Under this attack, the

controller becomes unreachable and it will not be able to process the new legitimate packets. Our approach treats this scenario as launching DDoS attack after establishing the connection between switch and controller. Fig 2 illustrates this scenario.

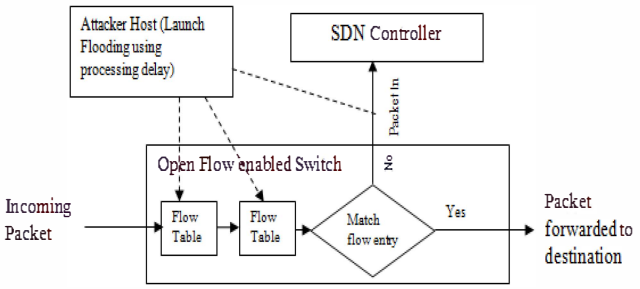

Fig 2 Modification to flow table by malicious host

The controller has the centralized network intelligence. Anyone who has access to the server in which the controller software is running can get access to the network. It introduces the possibility of controlling entire network. The applications like firewall, load balancing, routing, traffic engineering will be running on top of the SDN control plane. Once the access to controller applications is obtained for e.g. firewall application, then new Access Control List (ACL) can be created. Use of TLS/SSL connection between the switch and the controller doesn't guarantee secure communication. When the TLS connection is lost, the switch will try to connect to a backup controller if it is available. This is called "fail secure mode" or "fail standalone mode". In this mode the switch can use flow tables in the way it wishes, the switch may add, modify or delete any entry in the flow table [5].

The communication between switch and controller can happen in two ways. The operator can configure the switches with the IP address of the controller or the controller can initiate the Hello request. During connection breakup, an attacker can send Hello request to the switch acting as a legitimate controller and get access to flow tables. This scenario is treated as launching DDoS attack before establishing the communication and it is illustrated in Fig 3.

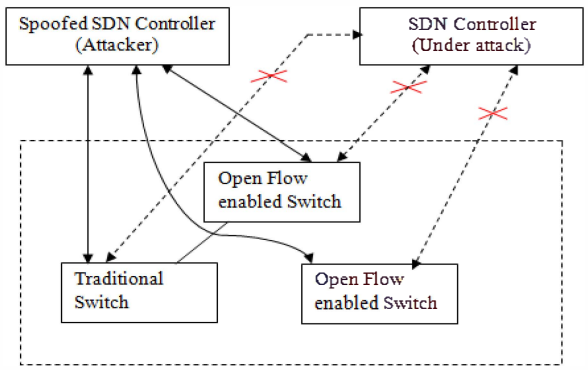

Fig 3 Malicious controller getting access to the entire network

B. *Effects of DDoS in SDN*

The brain of the network in SDN environment is the controller. It acts as operating systems to the switches. If the TLS connection is broken, the hybrid switches can operate in normal mode that does both routing and decision making. Incorrect flows can be installed and it can affect the performance of the controller.

The threat vectors with respect SDN environment and possible solutions for that are discussed in [4]. In this authors described various possibilities of attack with respect to switch and the controller. The forged flows can be injected into switch flow table resulting in launching of DoS (Denial of Service) or DDoS attack against switches and controllers. A possible solution could be deploying an intrusion detection system.

C. *Objective*

DDoS attack against the controller can be detected using the machine learning algorithm that was trained with attack and normal patterns. Hence, this paper explores the possibility of launching DDoS attacks and detection of DDoS using the SVM Classifier. The experiments are carried out using DARPA dataset [20].

III. RELATED WORKS

DDoS attacks and its detection methods is a long term research topic. However, very limited research is done in the area of security issues with respect to SDN environments. We describe some of intrusion detection techniques related to SDN environment. K. Giotis et al. [8] proposed anomaly detection mechanism on SDN using Openflow and sFlow. The packet sampling capability of sFlow is utilized for traffic gathering and comparison of this approach with sFlow is presented. The statistical entropy method is used for anomaly detection. The OpenFlow protocol is used to mitigate the attack by modifying the priority values of existing flows and installing new flows with drop action and high priority. But entropy method has a strong assumption that traffic data follow a certain normal distribution. The detection rate decreases if the assumption is incorrect.

Datacenter Overloading problems due to DDoS and other internal factors such as workload changes, operator errors were discussed by Ye Wang et al. in [10]. The capability of OpenFlow switch was utilized to monitor the network traffic and multidimensional flow aggregation mechanism is used to identify the overloading flows. Adaptive rate control using toxin-antitoxin mechanism is applied to suspicious flows to reduce false alarm rate. The packet counters of OpenFlow switch have the accumulated value from the time when flow rule was installed. But anomaly detection technique requires data only during last time period. DDoS flooding attack detection using OpenFlow with trained SOM classifier was proposed by Braga et al. [15]. Only DDoS detection method has been discussed and mitigation mechanism was not considered. Jeffrey et al. [16] discussed about ALARMS flow specification language to limit the amount of traffic to be forwarded to the controller. The network traffic was copied into other systems using span ports available in switch which creates overhead. Detailed security analysis of OpenFlow was presented in [17].

Lisa Schehlmann and Harald Baier proposed COFFEE [11] which utilizes OpenFlow protocol to identify the botnet activities and erase it. The network flow is monitored using Cisco technology NetFlow. The suspected flows are further validated by sending those packets to controller to extract more features. The detection was done using machine learning algorithms and reaction to the attack was done using an OpenFlow protocol by installing higher priority rules. This method doesn't delay the network traffic until the inspection is completed.

Defending of Scanning, Worm propagation attacks using SDN controller was discussed by Jafar et al. [13]. The

end hosts are assigned with random virtual IP by the controller and the translation is done with real IP during communication. Security enhancements that can be made to the network using SDN and the security challenges in SDN were discussed in [14].

Wenying Feng et al. [9] proposed an intrusion detection method by combing SVM with ant colony networks. They concluded that CSVAC (Combining support vectors with Ant colony) shows better results than SVM and Clustering based on Self-Organized Ant Colony Network (CSOACN). Multiclass SVM classification was done using One-against-all method which trains N classifiers and consults all N classifiers for testing unknown sample. This increases the testing time, which is critical for detecting intrusion at an earlier stage. DDoS detection using an ensemble of adaptive and hybrid neuro-fuzzy was proposed by Arun Raj Kumar and Selvakumar [12]. KDD 99 dataset is taken for evaluation purpose and NFBoost algorithm gives high accuracy with less false positive rate.

The existing methods are based on traditional network and SDN was used as a mechanism to detect and mitigate the DDoS attack. But initiating an attack against the controller causes switches losing its operating system. So far, very limited papers address the security of SDN controller.

IV. MULTICLASS SVM CLASSIFICATION

SVM is a supervised learning algorithm that recognizes patterns by analyzing the data and use the pattern for classification. Though it was initially designed as a binary SVM classifier, it has been extended to support multiclass classification. Generally multiclass problem is decomposed into binary problems and these classifiers are trained. This section describes some widely used SVM methods.

A. One-against-One (OVO)

This algorithm constructs N(N-1)/2 two classifiers and samples of the first class are trained as positive and samples of the second class as negative. Majority voting is applied to combine the classifiers while testing new unknown sample. All the classifiers are consulted to classify the data in testing phases.

B. One-against-all (OVA)

N binary classifiers are constructed for the N class problem and each class is trained against remaining N-1 classes. But the disadvantage is all N-1 classifiers have to be tested to predict the sample point.

C. Binary Tree of SVM

Binary Tree of SVM (BTS) proposed by Fei. B and Liu J [18] provide high classification efficiency for multiclass problems. It decreases the number of binary classifiers to the greatest extent without increasing the complexity of the original problem. Testing time is better than both OVO and OVA. But the disadvantage is training time is high as it tests all samples with trained SVM to build the sub nodes of the tree.

D. Binary Decision Tree SVM

SVM classifier utilizing Binary Decision Tree (SVM–BDT) was proposed in [19]. The classes are recursively divided into two groups by calculating the gravity centers of each group. The classes with biggest Euclidean distance are assigned to two different groups. Then classes with smallest distance are assigned to the same group. They showed that only $\log_2 N$ classifiers need to be consulted to classify the test sample. But the BDT may not always be height balanced and tree may be skewed on left or right side.

V. SYSTEM OVERVIEW

In order to detect the DDoS attack, the intrusion detection system should be fed with traffic information. The system utilizes widely used SVM classifier to detect the attack. SVM can learn the pattern with few training samples and produce an accurate classification by reducing the false positive data. This is achieved using the generalization capability, which refers to the ability of trained machine to classify unknown samples with the model learned from training dataset. SVM always find a global optimum solution rather than stopping with local optimum. SVM performs linear separation by finding an optimum hyperplane (largest margin) that separates two classes. The training examples that are closer to the hyperplane are called support vectors. SVM is linear classifier and kernel functions are used to support the nonlinear classification. Commonly used kernel functions are linear, polynomial, radial basis function and sigmoidal. A kernel function takes a dataset and transforms into higher dimension through the use of some of the functions described above. The transformed data become linearly separable in higher dimension, though it is not linearly separable in the original dimension. The Radial Basis Function (RBF) kernel supports nonlinear classification. It can be defined as

$$K(x_i, x_j) = \exp(\gamma \| x_i - x_j \|^2), \gamma > 0 \quad (1)$$

where x_i and x_j denotes the training data points and gamma is the adjustable width parameter of resulting classifier.

The of DDoS detection using SVM classifier is shown in Fig 4. The traffic data can have attributes like Source IP address, Destination IP address, Source port, Destination port, Protocol used for communication and the length of the packet. Some of these attributes will be multi-valued attributes. These attributes have to be converted as binary attributes, which has only two states or values. This conversion will be useful to perform the normalization process that helps to prevent higher values in the attribute dominating the lower range values.

Let the number of values in multi-valued attribute is n, after the conversion n binary attributes will be created to represent it. The value of the binary attribute has the value 1 when the nominal attributes take that particular value otherwise it is 0. During the normalization process, the attribute value is scaled to fit specific range (e.g. [0, 1]). The SVM classifier is trained with training data set and model is built upon it. Using the pattern recognized, an SVM classifier performs prediction of the category for new unknown traffic sample. The outcome of the classifier for the test data point would be either normal or attack.

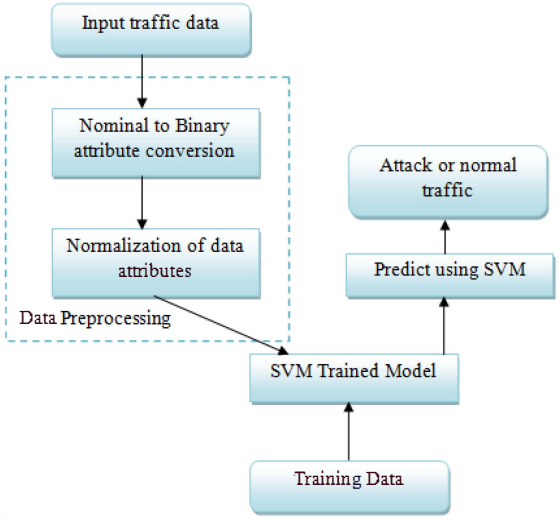

Fig 4 System architecture

VI. EXPERIMENTAL RESULTS

In this section, we provide the results of multiclass classification using SVM classifier. The 2000 DARPA intrusion detection scenario specific dataset provided by MIT Lincoln lab is taken for evaluation [20]. The dataset contains DDoS attack launched by a novice attacker. This attack scenario is carried out over multiple network and audit sessions. These sessions have been grouped into 5 attack phases over the course of which the adversary probes, breaks-in, installs Trojan mstream DDoS software, and launches a DDoS attack against an off-site server. The brief description of attack scenarios are given below.

1. IPsweep of the AFB (Air Force base) from a remote site
2. Probe of live IP's to look for the sadmind daemon running on Solaris hosts
3. Breakins via the sadmind vulnerability, both successful and unsuccessful on those hosts
4. Installation of the Trojan mstream DDoS software on three hosts at the AFB
5. Launching the DDoS attack

This dataset includes only attack traffic. The normal traffic data are included from the 1998 DARPA dataset. The data instances are divided into two groups, training data, and testing data. Details of these datasets are given in Table 1.

TABLE 1 DATASET DETAILS

Data Category	No. of training instances	No. of test instances
BreakIn	156	374
DDoS	963	1035
Installsw	318	204
IPSweep	101	684
Normal	2500	2501
Probe	54	94
Total	4092	4892

The classification accuracy and false positive rate of SVM depends on the parameters used. SVM accepts soft margin constant C as input parameter. A large value of C leads to high penalty values for misclassification errors. The optimizer tries to find smaller margin when the value of C is large, it will find larger margin for smaller values of C. The RBF kernel accepts gamma, adjustable width as input. The selection of parameters is done through Gridsearch method. The resultant optimum values of Gridsearch depend on the minimum and maximum values set for the grid size. The classification accuracy of varying gamma parameters is given in Table 2.

TABLE 2 ACCURACY WITH DIFFERENT PARAMETERS

Cost	Gamma	Classification Accuracy (%)	False Positive
10	0.1	94.23	.011
10	0.01	95.11	.008
10	0.001	93.86	.013

From the result, the classification accuracy was high when the gamma parameter is set to.01 and cost =10. The result of the SVM classifier in terms of confusion matrix is given in table 3.

TABLE 3 CONFUSION MATRIX OF TEST DATA SET

Actual Class	Classified Class					
	BreakIn	DDoS	Installsw	IPSweep	Normal	Probe
BreakIn	184	78	88	0	2	22
DDoS	0	1035	0	0	0	0
Installsw	13	30	160	0	0	0
IPSweep	0	0	0	683	1	0
Normal	1	0	0	0	2500	0
Probe	3	0	0	0	0	91

LIBSVM [21] package with RBF kernel is taken for experimental purpose. The results are compared with other DDoS detection methods listed below.

- Naïve Bayes
- Bagging
- Radial Basis Function Network
- J48 Decision Tree
- Random Forest

The result in Fig 5 shows SVM performs better in terms of accuracy and false positive rate. The details of false positive rate, training time and classification accuracy are given in table 4. Though the RBF network achieves similar results of SVM, the training time of the RBF is very high. SVM has high accuracy and less false positive rate compared to other methods. In terms of training time, Naïve Bayes and Random forest models perform better compared to other methods.

TABLE 4 COMPARISON OF SVM WITH OTHER METHODS

Method	Accuracy (%)	False Positive rate	Training Time (sec)
RBF	94.56	0.01	1320
SVM	95.11	0.008	120
Naïve Bayes	90.14	0.02	3
Bagging	91.49	0.024	60
J48	91.82	0.024	7
Random Forest	90.53	0.046	3

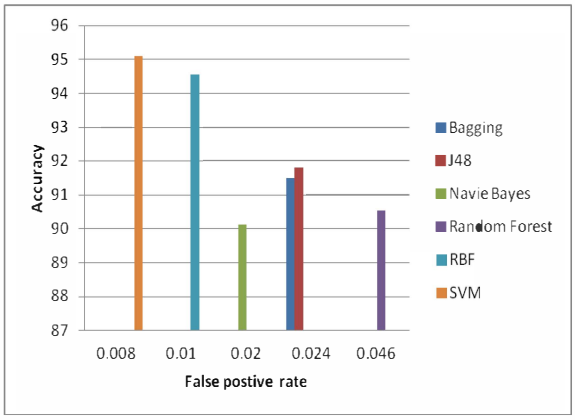

Fig 5 Comparison of classification methods

VII. CONCLUSION AND FUTURE WORK

SDN has emerged to improve the programmability within the network and also provides support for the dynamic nature of future functions. In order to achieve this goal, security challenges in SDN have to be addressed. This paper describes about one of the security issues in SDN controller. DDoS attack on the controller causes flow table flooding and dropping of legitimate packets. Hence, it is important to detect the DDoS attack in the earlier stage. The machine learning algorithms detects the DDoS attack with the pattern generated from the dataset. The experiments were carried out with existing DARPA dataset and results of SVM classifier is compared with other DDoS detection methods. Compared to other techniques, SVM classifier gives less false positive rate and high classification accuracy. However, SVM takes more time to train and generate the detection model, which is used to predict the traffic characteristics.

The future work aims to integrate the traffic pattern built in SVM with the SDN controller and detect the DDoS attack online. In further, the performance of the SVM classifier can be improved by combining AVL tree with SVM. The multiple binary SVM will be arranged in an AVL tree structure. The height balancing property of AVL tree helps to reduce the testing time.

REFERENCES

[1] "Software Defined Networking" https://www.opennetwor king.org/.

[2] "Software-Defined Networking: The New Norm for Networks," White Paper, Open Networking Foundation (ONF), Apr. 2012. [Online] Available: https://www. opennetworking.org/images/stories/downloads/white-papers/wp-sdn-newnorm.pdf.

[3] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Mag.*, vol. 51, no. 2, pp. 114-119, 2013.

[4] D. Kreutz, F.M.V. Ramos, and P. Verissimo, "Towards Secure and Dependable Software-Defined Networks", *ACM, HotSDN'13*, pp. 1-6, Aug, 2013.

[5] Open Networking Foundation, "OpenFlow Switch Specification, V1.3.2," Apr. 25, 2013, 131 pages, https://www.opennetworking.org/sdn-resources/onfspecifications/

[6] S. M. Specht and R.B. Lee "Distributed Denial Of Service: Taxonomies of Attacks, Tools and Countermeasures," *Proceedings of the International Workshop on Security in Parallel and Distributed Systems*, pp. 543-550, 2004.

[7] A. Studer and A. Perrig, "The Coremelt attack," *Proc of the 14th European conference on Research in computer security*, pp. 37-52, 2009.

[8] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and sFlow for an effective and Scalable anomaly detection and mitigation mechanism on SDN environments," *Journal on Computer networks,* Elsevier, vol. 62, pp. 122-136, Apr. 2014.

[9] W. Feng, Q. Zhang, G. Hu, and J. Huang, "Mining network data for intrusion detection through combining SVMs with ant colony networks," *Future Generation Computer Systems*, vol. 37, pp. 127-140, July 2014

[10] Y. Wang, Y. Zhang, V. Singh, C. Lumezanu, and G. Jiang, "NetFuse: Short-circuiting Traffic Surges in the Cloud," *IEEE International Conf on communications*, pp. 3514 – 3518, June 2013.

[11] L. Schehlmann and H. Baier, "COFFEE: a Concept based on OpenFlow to Filter and Erase Events of botnet activity at high-speed nodes," Proc. of Lecture Notes in Informatics, vol. p-220, pp. 2225-2239, Sep. 2013.

[12] P. Arun Raj Kumar and S. Selvakumar, "Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems," *Computer Communications*, vol. 36, pp. 303-319, Feb. 2013

[13] J. Jafarian, E. Al-Shaer, and Q. Duan, "OpenFlow Random Host Mutation : Transparent Moving Target Defense using Software Defined Networking," *HotSDN 12,* pp.127-132. Aug. 2012

[14] S. Scott-Hayward, G. O'Callaghan, and S. Sezer "SDN Security: A Survey," *IEEE SDN for Future Networks and Services*, pp. 1-7, Nov. 2013.

[15] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS Flooding Attack Detection Using NOX/OpenFlow," *IEEE 35th Annual Conference on Local Computer Networks*, pp. 408-415, 2010.

[16] J. R. Ballard, I. Rae, and A. Akella "Extensible and Scalable Network Monitoring using OpenSAFE", *Proc. Of the 2010 internet network management conference on Research on enterprise networking*, pp. 1-6, 2010.

[17] R. Kloti, V. Kotronics, and P. Smith, "OpenFlow: A Security Analysis," *IEEE International Conference on Network Protocols*, pp. 1-6, Oct. 2013.

[18] G. Madazarov, D. Gjorgevikj, and I. Chorbev, "A Multi-class SVM classifier utilizing Binary Decision Tree" *Informatica*, pp. 233-241, 2009.

[19] B. Fei and J. Liu, "Binary Tree of SVM: A new fast Multiclass Training and Classification Algorithm", *IEEE Transactions on Neural Networks*, vol. 17, no. 3, pp. 696-704, May 2006.

[20] DARPA 2000 Scenario Specific dataset available from : http://www.ll.mit.edu/mission/communications/cyber/CSTcorpora/ideval/data/2000/LLS_DDOS_1.0.html

[21] C. Chang, and C. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 27, pp. 1-39,Issue 3, Apr. 2011.

[22] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "OpenFlow: enabling innovation in campus networks," *SIGCOMM Comput. Commun.* Rev., vol. 38, no. 2, pp. 69–74, 2008.