# NETWORK APPLICATION FOR DETECTION AND MITIGATION OF DDoS ATTACKS IN SOFTWARE DEFINED NETWORKS

**K BhavaniShankar[1], Kshama Shivakavi[2], Rajith Kumar B K[3]**

*[1]K Bhavanishankar, UG Student, Department of ECE,RV College of Engineering,Bengaluru-59*
*[2]Kshama Shivakavi, UG Student, Department of ECE, RV College of Engineering, Bengaluru-59*
*[3]Rajith Kumar B K, Assistant Professor, Dept. of ECE, Engineering, RV college of Engineering, Karnataka, India*

---***---

**Abstract -** *Software Defined Networks (SDN) is an emerging technology in the era of modern technology which is manageable, dynamic, cost-effective and adaptable. The network control is directly programmable. Network intelligence is centrally managed by the SDN controller using which the flows can be generated. Hacker takes control of many ip address and overwhelm the website with more number of requests than it actually can accommodate. Then the website is under Distributed Denial of Service(DDoS) attack. Our project is to develop a network application for detection and mitigation of DDoS in SDN. Support Vector Machine(SVM) is a machine learning classification algorithm, it is used to classify the data. Our objectives are to detect the DDoS attack using SVM and reroute the flows in the case of a DDoS to a deep inspection box and mitigate the requests if not from trusted sources.*

*The classification algorithm SVM is trained with five features that are sufficient to classify whether the traffic is normal or anomalous. The five features are Speed of Source IP, Standard deviation of flow Packets, Standard deviation of byte Packets, Ratio of pair flow entries, Speed of flow entries. We need to extract the above features from the network and send them to trained SVM for classification, If classified as attack traffic add mitigative flows to the switch. The software that are used in our project are miniedit for creating the topology and python programming in Ubuntu.*

*Network topology is created using miniedit. When normal flows are introduced into the network SVM classifies it as normal and when the website is overwhelmed with more number of requests then SVM classifies it as anomalous and mitigative flows are added. Thus detection and mitigation of DDoS attack is done in SDN.*

*Detection of DDoS attack is achieved in SDN with 94% accuracy. When attack is detected it is successfully mitigated using deep inspection box. To improve the accuracy improve the feature correlation to include wider range of attacks where the difference in these parameters is subtle.*

***Key Words*: Software Defined Networking, DDoS attacks, Machine Learning, Support Vector Machine, Deep Inspection Box.**

## 1. INTRODUCTION

### 1.1 Software Defined Networks

SDN is an architecture that makes network sharp and flexible. The goal of SDN is to improve the network control by informing the company and service providers to react quickly to the changes going on in various work requirements. The main function of a SDN network is that a network engineer can easily modify the traffic without checking with each and every network device with the help of a centralized control console .The SDN controller directs the switches to provide services when they are needed irrespective of the so many connections between server and devices. It consists of 3 layers. They are application layer, control layer and infrastructure layer. The application layer consists of the network functions and applications. SDN architecture connects the different layers via API .The control layer has the SDN controller software which acts as the brain of SDN. The infrastructure layer consists of the physical switches in the network.

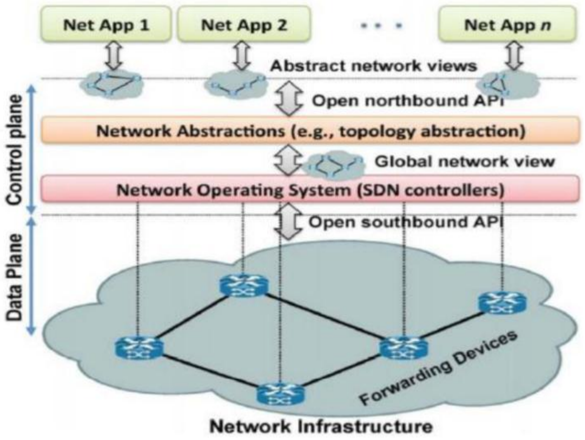


**Fig-1:** Architecture of the Software Defined Networks

### 1.2 Ddos Attacks

A DDoS is a cyberattack on a server, website or network by flooding it with internet traffic. Aim is to send more no of requests than the server can accommodate.

6 Features to classify a Ddos Attack:-

1. The speed of source IP (SSIP)

$$SSIP = \frac{Sum\_IP_{src}}{T}$$

2. The speed of source port (SSP)

$$SSP = \frac{Sum\_port_{src}}{T}$$

3. The Standard Deviation of Flow Packets(SDFP)

$$SDFP = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(packets_i - Mean\_packets)^2}$$

4. The Deviation of Flow Bytes (SDFB)

$$SDFB = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(bytes_i - Mean\_bytes)^2}$$

5. The speed of flow entries(SFE)

$$SFE = \frac{N}{T}$$

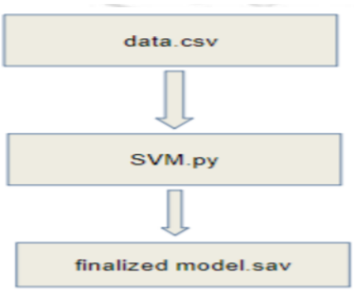
6. The Ratio of Pair-Flow (RPF)

$$RPF = \frac{2 * Pair\_Sum}{N}$$

## 2. Methodology

Various files and scripts that have been used to train the SVM for a given dataset. The files to create the network topology and the files which are used for detecting and mitigating the DDoS attack, they are:-

1. data.csv 2 .SVM.py 3. finalisedmodel.sav 4. Project1.mn 5. test.sh 6. collectE.sh 7. collectE.py 8. live.csv 9. Inspect.py



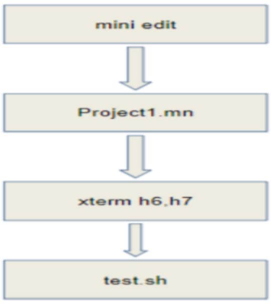


**Fig 2:** Training SVM **Fig 3**:Create network topology

### 2.1 Simulation of DDos Attack

In this section the simulation of DDoS attack and how the flows are sent to the deep inspection box for mitigating the attack are discussed.

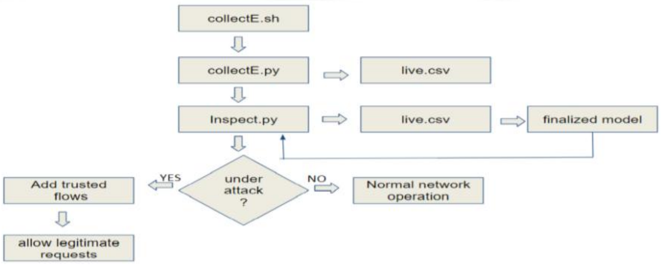


**Fig 4**: Flow chart for simulation of DDoS attack

CollectE.sh gives access to flows in the switch. The four variables(packet No, bytes no, Source IP, Destination IP) are collected from flows and sent to collectE.py. It calls Inspect.py and if it returns as there is an attack, mitigating flows are added in the switch.

### 2.1 Results

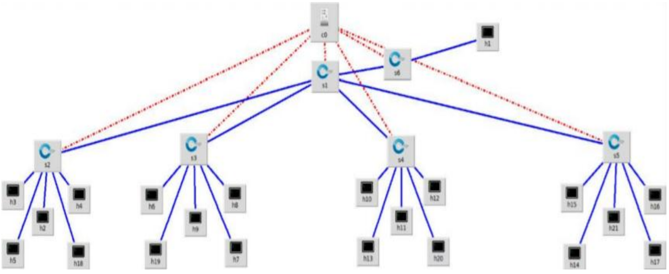
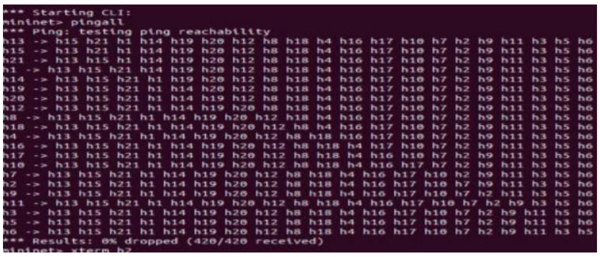
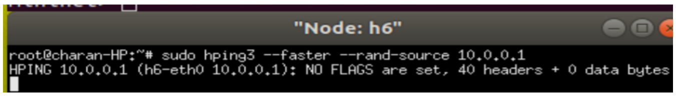


**Fig 5:** SDN Topology created in miniedit



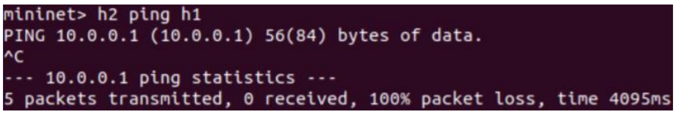
```
*** Starting CLI:
mininet> pingall
*** Ping: testing ping reachability
h13 -> h15 h21 h1 h14 h19 h20 h12 h8 h18 h4 h16 h17 h10 h7 h2 h9 h11 h3 h5 h6
h15 -> h13 h21 h1 h14 h19 h20 h12 h8 h18 h4 h16 h17 h10 h7 h2 h9 h11 h3 h5 h6
h21 -> h13 h15 h1 h14 h19 h20 h12 h8 h18 h4 h16 h17 h10 h7 h2 h9 h11 h3 h5 h6
h1 -> h13 h15 h21 h14 h19 h20 h12 h8 h18 h4 h16 h17 h10 h7 h2 h9 h11 h3 h5 h6
h14 -> h13 h15 h21 h1 h19 h20 h12 h8 h18 h4 h16 h17 h10 h7 h2 h9 h11 h3 h5 h6
h19 -> h13 h15 h21 h1 h14 h20 h12 h8 h18 h4 h16 h17 h10 h7 h2 h9 h11 h3 h5 h6
h20 -> h13 h15 h21 h1 h14 h19 h12 h8 h18 h4 h16 h17 h10 h7 h2 h9 h11 h3 h5 h6
h12 -> h13 h15 h21 h1 h14 h19 h20 h8 h18 h4 h16 h17 h10 h7 h2 h9 h11 h3 h5 h6
h8 -> h13 h15 h21 h1 h14 h19 h20 h12 h18 h4 h16 h17 h10 h7 h2 h9 h11 h3 h5 h6
h18 -> h13 h15 h21 h1 h14 h19 h20 h12 h8 h4 h16 h17 h10 h7 h2 h9 h11 h3 h5 h6
h4 -> h13 h15 h21 h1 h14 h19 h20 h12 h8 h18 h16 h17 h10 h7 h2 h9 h11 h3 h5 h6
h16 -> h13 h15 h21 h1 h14 h19 h20 h12 h8 h18 h4 h17 h10 h7 h2 h9 h11 h3 h5 h6
h17 -> h13 h15 h21 h1 h14 h19 h20 h12 h8 h18 h4 h16 h10 h7 h2 h9 h11 h3 h5 h6
h10 -> h13 h15 h21 h1 h14 h19 h20 h12 h8 h18 h4 h16 h17 h7 h2 h9 h11 h3 h5 h6
h7 -> h13 h15 h21 h1 h14 h19 h20 h12 h8 h18 h4 h16 h17 h10 h2 h9 h11 h3 h5 h6
h2 -> h13 h15 h21 h1 h14 h19 h20 h12 h8 h18 h4 h16 h17 h10 h7 h9 h11 h3 h5 h6
h9 -> h13 h15 h21 h1 h14 h19 h20 h12 h8 h18 h4 h16 h17 h10 h7 h2 h11 h3 h5 h6
h11 -> h13 h15 h21 h1 h14 h19 h20 h12 h8 h18 h4 h16 h17 h10 h7 h2 h9 h3 h5 h6
h3 -> h13 h15 h21 h1 h14 h19 h20 h12 h8 h18 h4 h16 h17 h10 h7 h2 h9 h11 h5 h6
h5 -> h13 h15 h21 h1 h14 h19 h20 h12 h8 h18 h4 h16 h17 h10 h7 h2 h9 h11 h3 h6
h6 -> h13 h15 h21 h1 h14 h19 h20 h12 h8 h18 h4 h16 h17 h10 h7 h2 h9 h11 h3 h5
*** Results: 0% dropped (420/420 received)
mininet> xterm h2
```

**Fig 6** :Network is up and running(pingall)



```
root@charan-HP:~# sudo hping3 --faster --rand-source 10.0.0.1
HPING 10.0.0.1 (h6-eth0 10.0.0.1): NO FLAGS are set, 40 headers + 0 data bytes
```

**Fig 7:** Simulating DDoS from host h6.



```
mininet> h2 ping h1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
^C
--- 10.0.0.1 ping statistics ---
5 packets transmitted, 0 received, 100% packet loss, time 4095ms
```

**Fig 8**: host h2 cannot ping h1

When host h6 starts attack on server h1. h1 is overwhelmed with the requests from h6 and h1 stops responding to h2. So server cannot serve to legitimate users. Attacks results in slow response.

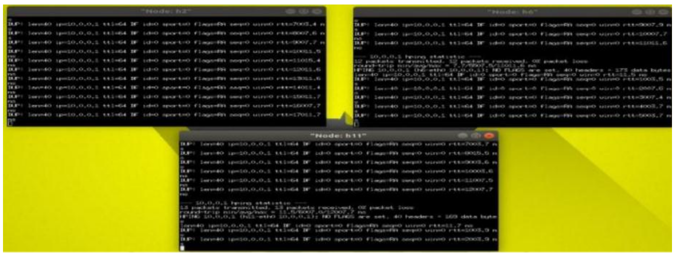
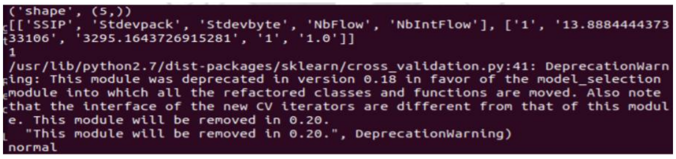


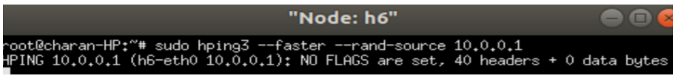
**Fig 9** : test.sh in hosts h6, h11, h14

Test.sh will create regulated normal traffic flows in hosts when it is run to 10.0.0.1 (our server h1)



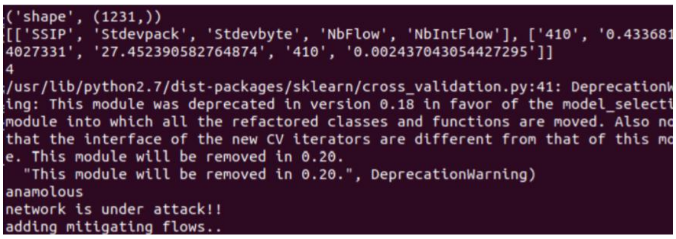
```
('shape', (5,))
[['SSIP', 'Stdevpack', 'Stdevbyte', 'NbFlow', 'NbIntFlow'], ['1', '13.8884444373
33106', '3295.1643726915281', '1', '1.0']]
1
/usr/lib/python2.7/dist-packages/sklearn/cross_validation.py:41: DeprecationWarn
ing: This module was deprecated in version 0.18 in favor of the model_selection
module into which all the refactored classes and functions are moved. Also note
that the interface of the new CV iterators are different from that of this modul
e. This module will be removed in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)
normal
```

**Fig 10**: collectE.sh result for normal Network operation



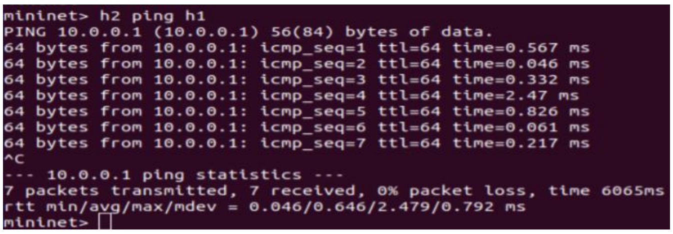
```
oot@charan-HP:~# sudo hping3 --faster --rand-source 10.0.0.1
HPING 10.0.0.1 (h6-eth0 10.0.0.1): NO FLAGS are set, 40 headers + 0 data bytes
```

**Fig 11:** Simulating DDoS from host h6



```
('shape', (1231,))
[['SSIP', 'Stdevpack', 'Stdevbyte', 'NbFlow', 'NbIntFlow'], ['410', '0.433681
4027331', '27.452390582764874', '410', '0.002437043054427295']]
4
/usr/lib/python2.7/dist-packages/sklearn/cross_validation.py:41: Deprecation
ing: This module was deprecated in version 0.18 in favor of the model_selecti
module into which all the refactored classes and functions are moved. Also no
that the interface of the new CV iterators are different from that of this mo
e. This module will be removed in 0.20.
  "This module will be removed in 0.20.", DeprecationWarning)
anamolous
network is under attack!!
adding mitigating flows..
```

**Fig 12**: Network is under attack, Mitigating flows are added



```
mininet> h2 ping h1
PING 10.0.0.1 (10.0.0.1) 56(84) bytes of data.
64 bytes from 10.0.0.1: icmp_seq=1 ttl=64 time=0.567 ms
64 bytes from 10.0.0.1: icmp_seq=2 ttl=64 time=0.046 ms
64 bytes from 10.0.0.1: icmp_seq=3 ttl=64 time=0.332 ms
64 bytes from 10.0.0.1: icmp_seq=4 ttl=64 time=2.47 ms
64 bytes from 10.0.0.1: icmp_seq=5 ttl=64 time=0.826 ms
64 bytes from 10.0.0.1: icmp_seq=6 ttl=64 time=0.061 ms
64 bytes from 10.0.0.1: icmp_seq=7 ttl=64 time=0.217 ms
^C
--- 10.0.0.1 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6065ms
rtt min/avg/max/mdev = 0.046/0.646/2.479/0.792 ms
mininet> 
```

**Fig 13**: host h2 being able to ping host h1 even under attack

Due to the addition of mitigative flows, host h2 is being able to communicate with host h1. Thus DDoS attack is detected and mitigated.

## 3. CONCLUSIONS

Growth of traffic is creating a huge challenge for network operators making it very hard to manage and monitor. In this paper, we discussed the six-tuple characteristic values related to DDoS attack and then use the support vector machine algorithm to judge the traffic and carry out DDoS attack detection. Detection is achieved via SVM and with 94% accuracy. When attack is detected it is successfully mitigated using deep packet inspection. DPI has the ability to identify types of traffic in the network at real-time, and to locate DPI at SDN layers, we split DPI into 2 parts. The first half is the DPI is located at network gateway which helps for scanning the traffic and forward results to the second part: the DPI controller with the task of managing the policy chain, helps the operators to have more control over networks that provide speed and flexibility.

## REFERENCES

[1] Z. Cai, Z. Wang, K. Zheng, and J. Cao, "A distributed TCAM coprocessor architecture for integrated longest prefix matching, policy filtering, and content filtering," IEEE Transactions on Computers, vol.62, no.3, pp.417–427, 2013.

[2] Y. Li, Z. Cai, and H. Xu, "LLMP: exploiting LLDP for latency measurement in softwaredefined data centre networks," Journal of Computer Science and Technology, vol.33, no.2, pp.277– 285, 2018

[3] J. G. Yang, X. T. Wang, and L. Q. Liu, "Based on traffic and IP entropy characteristics of DDoS attack detection method," Application Research of Computers, vol.33, no.4, pp.1145– 1149, 2016.

[4] A. Saied, R. E. Overill, and T. Radzik, "Detection of known and unknown DDoS attacks using artificial neural networks," Neurocomputing, vol.172, pp.385–393, 2016.

[5] R. Braga, E. Mota and A. Passito, "Light weight DDoS flooding attack detection using NOX/OpenFlow," in Proceedings of the 35th Annual IEEE Conference on Local Computer Networks (LCN'10), pp.408–415, Denver, Colo, USA, October, 2010

[6] N. Z. Bawany, J. A. Shamsi and K.Salah, "DDoS attack detection and mitigation using SDN: methods, practices, and solutions," Arabian Journal for Science and Engineering, vol. 42, no. 2, pp. 425–441,2017