National College of Ireland

DDOS attack detection and mitigation using statistical and machine learning methods in SDN

MSc Research Project
Cloud Computing

Vishal Kumar Singh
Student ID: x18201687

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Vishal Kumar Singh
Student ID:	x18201687
Programme:	Cloud Computing
Year:	2020
Module:	MSc Research Project
Supervisor:	Vikas Sahni
Submission Due Date:	17/08/2020
Project Title:	DDOS attack detection and mitigation using statistical and machine learning methods in SDN
Word Count:	5894
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

I agree to an electronic copy of my thesis being made publicly available on TRAP the National College of Ireland's Institutional Repository for consultation.

Signature:	
Date:	16th August 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

DDOS attack detection and mitigation using statistical and machine learning methods in SDN

Vishal Kumar Singh
x18201687

Abstract

Software defined networks is the future of networking as it decouples the data plane and control plane of the network devices to provide a centralized control over the network. SDN has abilities to provide superior management and security of a network and allows us to program the network for better ease of use and performance. However, SDN is vulnerable to attacks, DDOS attacks are the most dangerous and threatening attacks in a network, as it can flood the network and block access to the server network with large counts of packets and make use of network resources to deny response for further requests incoming. In a cloud environment DDOS attacks are known only to increase. The method presented is to combine statistical and machine learning methods to efficiently detect and mitigate DDOS attacks in SDN. Implementation of this method is done using ryu controller and mininet network simulator with openflow SDN protocol, the machine learning algorithm implemented has achieved accuracy of 99.26% and a detection rate of 100% in detecting and mitigating DDOS attacks in a software defined network.

1 Introduction

Cloud computing trend has seen a rapid growth in the last couple of years in fields of industries and academic, due to the essential characteristics it can offer over traditional network. Software define network (SDN) has seen significant growth in the field of networking and trending cloud networks. SDN is a networking technology which improves the network performance and management, it enables us to have a centralized management of the network and allows us to program the network devices(Xia et al. (2015)).

Software defined networking decouples the data and control plane of the network device and enables the control plane to be programmed by a SDN controller. SDN architecture is comprised of three-layer namely infrastructure layer where are the networking devices like switches and hosts are present, control layer where the controller is implemented and the application layer for networking applications. SDN is used to monitor and control the network from in single place, which helps in changing and configuring network devices easily in the network. It interns improves the saclability, performance, controllability and provides flexibility and cloud management. Software defined network-based cloud environments are deployed in cloud computing networks to have better secirity and control over the network and provide networking as a service (NAAS) (Yan et al. (2016)).

1.1 Motivation

Cloud computing technologies is booming with more and more demand for cloud services, many companies and organizations are moving their services to cloud in hope for a better performance and security. Humans now have an everlasting dependence on the internet with their personal data on it, due to this people always have security concerns. Security measures needs to be in place to keep those data intact and secure from security network attacks. DDOS attacks are on such network attacks which block access to the server and deny services for the cloud customers. Many research and measures have been taken to prevent this from happening, one such measure is to use software defined networks to achieve this goal.

1.2 Research Question

The research question presented during the initial stage of research in computing:-

"Can Software defined networking improve the detection and mitigation of distributed denial of services (DDOS) attacks in a cloud network environment"

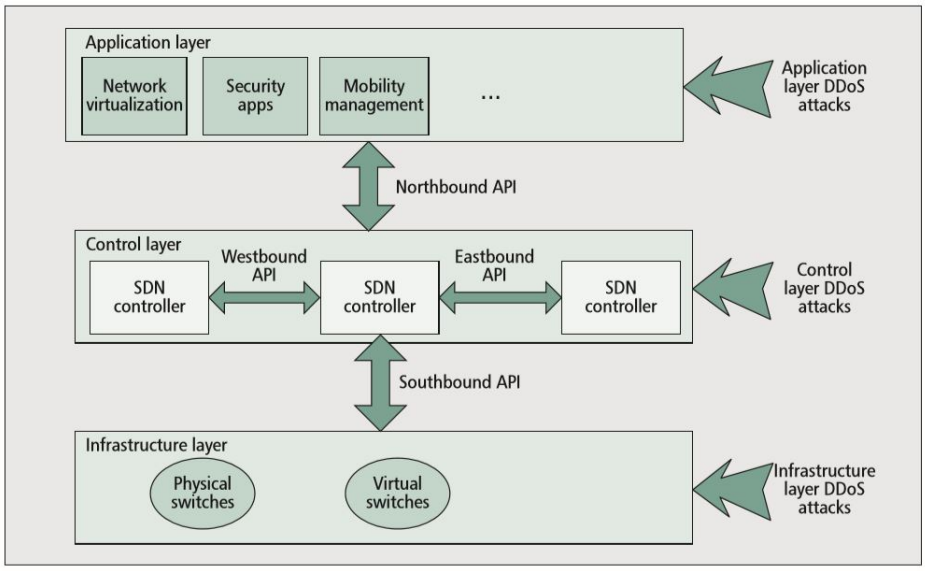


Figure 1: SDN architecture with vulnerabilities of DDOS attacks.(Yan and Yu (2015))

The above image shows the different layers of the SDN vulnerable to DDOS attacks. SDN decouples the network layers, hence has chances security issues on different layers of SDN architecture. Distributed denial of service(DDOS) attacks are common, hazardous and dangerous attacks for the cloud network. DOS attacks is when any attacker tries to block the network with traffic packets and force the network deny any further services for other incoming requests also the network becomes unavailable and utilize all the network resources (Yan and Yu (2015)).
Dos attack is when a single attacker is attacking the network and DDOS attack is when many attackers or botnets are sending traffic packets/requests at the same time. According to research article published by Akamai on network security stated that DDOS attacks are increasing over the years by upto 125%. However, SDN has capabilities to

have better security and centralized management to detect and prevent DDOS attacks on the network.

These security issues in software defined networks has to be solved, many research and work have been done to come up with new techniques and strategies to prevent any kind of attacks. DDOS attacks in cloud network has been an open research challenge for a decade now (Ahmad et al. (2015)) with more and more new methods to prevent in from happening. In this research project i have presented a method overcome this challenge and detect and mitigate DDOS attacks using software defined networks.

1.3 Contribution

The method presented in this work is to combine statistical analysis and machine learning methods. In statistical method, 4 features are extracted from the network traffic and collected in a dataset. This dataset is used to train the support vector machine classifier machine learning algorithm to predict DDOS attacks in the network and detect malicoius traffic early and mitigate it by blocking the source and port in the network. The implemented method is tested and achieved an accuracy of 99.26% in predicting the attack and 100% of the malicious traffic was detected with 0 false alarm of the traffic in the network.

This report has been divided into sections as follows: Section 2- Related Work narrates the work done by other researchers in this field. Section 3- Methodology details the method presented in this study/work with the tools and technologies used. Section 4- Design specification describes the SDN framework and the network topology design created for simulation. Section 5- Presents the implementation ways and how the tools and software were put to use in this project. Section 6- Presents the evaluations configuration and experiments conducted to present the results and working of the project. Section 7- Is the conclusion and future work of the project.

2 Related Work

Software-Defined Networking often is described as the 'Future of Networking' these days. This technology is rapidly growing and has been a huge success reason being the decouplement of the control plane from the data plane of the networking devices. The control plane allows centralized management of the entire network and switches, thereby the need to configure each networking device is eliminated. The data plane, consists of the switching devices allows forwarding the network traffic based on policies programmed by the applications running on top of the programmed controller. This significant advancement in field of networking has help out accelerating service delivery and provide more flexibility, agility in provisioning both physical and virtual network devices from a centralised location.
The literature review has been divided into three sections namely Traffic analysis in SDN, DDOS attack detection and mitigation using SDN and Machine learning approaches for anomaly detection. These sections contains detailed review of the work done many researches in the fields and methods in preventing DDOS attack using SDN.

2.1 Traffic Analysis In SDN

This sections describes the methods and work presented and implemented by researches in analysing the incoming traffic in a software defined network. DDOS attacks can be detected and traffic analysis are done using these two techniques:

- Analysing the network data
- SNMP analysis of data

1) Analysing the network data This method involves finding out malicious traffic in the network to identify the likelihood of DDOS attack. Every network traffic has a set of features, the features of the traffic has to be extracted by defining some characteristics of the attack traffic. The behaviour of and features of the normal traffic also has to be captured to distinguish the traffic from attack traffic. Several techniques and features extraction parameters are proposed by many researchers. The authors Xu and Liu (2016) have presented a concurrent algorithmic method which changes the monitoring capabilities of the flow on the switches in the network to quickly identify potential victims, malicious traffic and suspicious attackers. The authors have experimented the designed method and the claims made by the author are well supported by the graphs and results in the article. However the article does not give enough details about the tools and simulation methods used to achieve the results obtained, which achieved high accuracy by capturing asymmetric flow.

Wang et al. (2019) have proposed four feature extraction methods which include collecting traffic data and counting the byte rate, symmetric and asymmetric flow variations and counting small amounts of packets incoming in the network. The proposed algorithm is implemented using ryu controller and simulated using mininet, the results shown claims to have reduced controller response time under DDOS attack.
A similar approach but with different feature extraction parameters with a density peak clustering algorithm have been implemented by He et al. (2017). At first the anomaly detection takes place by collecting the traffic features and then finds the correlation with the malicious traffic characteristics for strong and weak correlation factors and then the density peak clustering algorithm is applied to the correlated data for the DDOS attack detection in the network. The authors have implemented the algorithms using python 2.7.9 with MINE package and some open source Python machine learning libraries for comparison. The authors claims that this method out performs the other ML approaches but does not comply with the real time data of the incoming traffic in the network.

2) Simple network Management Protocol uses (MIB) Management information base wherein the traffic data is collected and stored. Analysing the MIB information improves the detection of DDOS attacks by integrating (IDS) intrusion detection system. A feasible approach to combact DDOS attack in SDN is suggested by Nhu-Ngoc Dao et al. (2015). This method focuses on flooding attack on the network, the method is to count the number of packets incoming, the counter of IPs in the network and average connections of users, these parameters are compared with a threshold values to detect the attack. The authors have experimented this method and simulated using mininet, but the controller details are not detailed and this methods can easily be bypassed by multiple IP sources and delayed attacks. The article does not produce enough details for the experiment to be reproduced.

In this article Jin et al. (2003) have proposed a defense mechanism against DDOS spoofed traffic by using a hop-count filtering method, every IP packet has to take certain number of hops to reach its destination. The authors method is to monitor the hops count and TTL-Time to live value in IP header, the authors claims that hop count filtering method can identify 90% of the spoofed IP packets. The experiments were conducted using linux kernal and generating TCP and ICMP traffic. The results are well demonstrated using graphs, however the experimental setup is much complicated and time consuming to perform.

2.2 DDOS Attack Detection and Mitigation Using SDN

This section includes the related work done in detecting and mitigating DDOS attacks in SDN using traditional methods and novel detection strategies used by other researchers.

Software defined-anti DDOS defence mechanism approach has been presented by Cui et al. (2016) in which the strategy is to have attack detection trigger mechanism with attack trackback system. The detection approach is based on the related work done by other researchers, model used has four stages namely initial stage, detection stage, traceback stage and mitigation stage. The novel work presented tracebacks the source IP from which the DDOS attack is incoming, it also determines the number of records from the traffic and velocity of the packets is also calculated to detect malicious traffic in the network. The method is implemented using ryu controller which is a python based controller and the simulation environment is setup using mininet, the authors claims are supported with well documented results and has evidences to reproduce the experiments conducted.

Bhushan and Gupta (2019) approach is to have a mitigation technique for DDOS attacks by arranging the size of the flow table in the software defined network, as most DDOS attacks top up the flow table entries and thus blocking new entries in the switch. The architecture has two databases, blacklisted sources and flow table status. The flow table status holds the flow entries statues in the network from all the switches, the blacklist status stores the source IPs of the malicious attack traffic incoming in the network. Once the flow table is full it check for the nearest switch to redirect the traffic to avoid flow table size blockage. This method infact lets the malicious traffic in the network, which can be dangerous of the network and does not have an early detection scheme. The authors have experiment the method and presented the results and the implementation is done using pox controller which is a python based controller and simulation environment is created using mininet. The evaluations are documented and detailed to reproduce and duplicate the experiment, limitation being having no early detection strategy.

In this study Alshamrani et al. (2017) have presented a defense system using SDN to defeat DDOS attacks. The novel work presented in this work is different from other techniques, usually when the network is under attack usually the packets are detected and mitigated by dropping the packets instead the packets are sent to a honeypot model. Honeypot model presented learns the behaviour of the attacker and monitors the IP to extract features and use them to further enhance the detection system. The proposed algorithm is experimented and implemented on pox controller and simulated using mininet,

the outcomes are shown using graphs and the claims and arguments made are justified. However, the implemented work is well documented and does not have evidences to reproduce the experiment and monitoring the IP won't be of much use as the attacker may not use the IP again and could have been botnets.

Another work is to combine open-flow and sflow to have an anomaly detection mechanism on SDN environments by Giotis et al. (2014), the work has detailed explanation and organized approach for anomaly traffic detection. The architecture has two modules, anomaly detection module and traffic collector. The traffic flow is analysed after collection, the anomaly detection uses an entropy-based algorithm to detect malicious traffic in the network. The architecture uses open-flow and sflow protocols, implementation is done using nox controller to program using python language and network simulation is done using mininet. The method presented can run with bigger network topologies and can handle real-time stats, the work is well detailed and can be reproduced. The limitation of the method implemented is using an outdated controller application and mitigation strategy is not efficient.

2.3 Machine Learning Approaches for Attack Detection

This section narrates the different machine learning algorithms and methodologies used by several researchers to predict and detect a DDOS traffic in a SDN.
In this study the Santos et al. (2019) have discussed about the different machine learning approaches which can be used with software defined network to detect and mitigate DDOS traffic in a network. The authors shows the implementation of four ML algorithms namely, MLP, Decision tree, Support vector machine(SVM) and random forest, these ML algorithms were simulated using mininet. The results shows that random forest algorithms achieves the best accuracy and decision tree algorithms gives the best processing time for the DDOS attack detection. However, there were few drawbacks in the implementation to classify flow table attack and bandwidth attack. Overall the article is well constructed and detailed.
A similar study, in which the auhtors Sahoo et al. (2018) have compared many machine learning algorithms namely k-nearest neighbour(KNN), Naive bayes(NB) , Support vector machine(SVM), Random forest, Linear regression(LR). Experiment results show that Linear regression(LR) and random forest ML algorithms produced the best prediction accuracy of 98% with random forest having less execution time than LR. However the results the shown in the article, but the tools used for the implementation and simulation are not detailed and hence this work cannot be reproduced.

Machine learning algorithms are always evolving and getting better in prediction day by day, we have seen support vector machine ML method. But here Myint Oo et al. (2019) proposes an advanced support vector machine based ML algorithm. In this study ASVM algorithm collects data from feature extraction stage and classifies parameters to predict DDOS attacks in SDN, this technique claims to reduce the testing and training time for the machine learning algorithm to perform its tasks. The implementation is done on opendaylight controller and simulation environment in implemented using mininet and the authors claims ASVM method has a detection accuarcy of 97% with the fastest testing and training time. The claims made by the author are well supported by the graphical results in the article.

Dehkordi et al. (2020) have implemented a combination of entropy based method and machine learning method, authors approach has three stages namely traffic data collector, entropy thresholds and ML classifiers. The data is collected and entropy based static thresholds are applied to better simply the malicious traffic and these datasets are then applied using machine learning algorithms. The experimental setup is done using floodlight controller and simulation and network topology is done using mininet, the results shows that this approach achieves better accuracy and prediction results in detecting DDOS attacks.

This work presented by Mohammed et al. (2018) have collaborated the traditional methods with machine learning by training the model with NSL-KDD dataset to provide better accuracy in DDOS attack prediction. The experimental setup is done using four SDN controllers, POX, ONOS, Ryu and opendaylight and the network simulation is done using mininet. A similar approach but using deep learning in openflow based SDN is proposed by Li et al. (2018), In which classical neural networks models such as CNN, RNN, and LSTM are applied with deep learning to achieve better detection of malicious DDOS traffic incoming in a software defined network.

Comparison Table			
References	Technique	Implementation	Comments/ limitations
Wang et al. (2019)	Safeguard scheme with feature extraction	Ryu controller and Mininet	Reduced controller response time, Future implement in real network.
Bhushan and Gupta (2019)	Flow table size recovery and blacklisted sources	Pox controller and Miniet	Late detection of DDOS traffic and lets traffic in the network.
Myint Oo et al. (2019)	Advanced SVM with feature extraction	Opendaylight Controller and Mininet	Accuracy of 97% with fast test and training time.
Dehkordi et al. (2020)	Entropy ML Classifier with static thresholds	Floodlight Controller with Mininet	Better accuracy but with limited network Design.
The Presented Work	Statistical analysis with SVM ML Algorithm	Ryu Controller with openFlow Switch and Mininet	Accuracy of 99.26% and 0% false alarm limitation being single network topology.

2.4 Conclusion

The work presented by the researchers in ways to analyse the incoming traffic in software defined network has shown that every network flow has certain parameters and features which can be monitored and collected to extract the exact features required to detect malicious DDOS traffic in a network. Different methods presented by researchers in a way ensure better security from DDOS attacks using their novel methods and implementation. Using machine learning algorithms to detect anomalies in the network is still a major research challenge as there are many ML methods, the related work done using ML methods shows that it can achieve better accuracy in detecting anomalies in the network traffic. Implementations methods seen in related work does differ in some cases. However, ryu controller is one of the popular python based programmable controller and mininet being the most used and best choice for creating the simulation environment for software defined networks.

3 Methodology

The traditional network systems are highly prone to attacks and can lead to privacy issues and data leak of packet information from the network. To avoid suck kind of network attacks on the public network this work presents a software defined network based DDOS attack detection and mitigation method using statistical network analysis and machine learning methods. SDN based network enables the separation of the control plane and data plane from the network devices. A centralised management mechanism is established in order to prevent the network from unauthorized access. Every network traffic incoming has few characteristics and parameters defined for every network packet flow, these characterizations are collected as training and test features for our method to prevent DDOS attacks on the network using software defined networking. The following features and parameters are monitored and collected for detecting DDoS attacks:

1. Speed of IP Sources: This feature gives the total number of TP sources incoming in the network within a particular time interval. Abbreviated as SSIP, it is defined as;

$$SSIP = \frac{SumIP_{src}}{T}$$

where $SumIP_{src}$ is the total number of IP sources incoming in every flow and T is the sampling time intervals. The time interval T is set to three seconds such that the detection system monitors and collects data of flows every three seconds and stores the number of source IPs during this duration. The controller needs to have sufficient data of both normal and attack traffic for the machine learning algorithm to predict the attacks. For normal attacks the SSIP is usually low and for attack the count is usually higher.

2. FlowCount of the Traffic: Every network traffic incoming in the network has a particular number of flow counts. Normal traffic has fewer flow counts than DDOS attack traffic.

3. Speed of Flow Entries: This is the total number of flow entries to the switch in the network within a particular time interval. Abbreviated as SFE, it is defined as;

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

This is a very relevant character of attack traffic detection because the number of flows entries increases significantly in a fixed interval of time in case of DDOS attacks as compared to the speed of flow entries value of the normal traffic flows.

4. Ratio of Pair-Flow Entries: This is the total number of flow entries incoming in the switch which are the interactive IPs divided by the total number of flows in the T time period. Abbreviated as RPF, it is defined as;

$$RPF = \frac{SrcIPs}{N}$$

where SrcIPs is the total number of collaborative IPs in the network flow and N is the total number of IPs. Under normal traffic conditions, the i^{th} flow IP source will be the same as the IP of the destination of the j^{th} flow and the j^{th} flow will have the same IP source as the destination IP of the i^{th} flow. This accounts for an interactive flow which won't be the case when it is a DDoS attack traffic. Under attack, flow entries to the host destination at time T increases rapidly and the destination host is unable to respond to them.
Therefore the attack traffic will have an abrupt decrease in the total number of collaborative flows as the DDOS attack starts. The total number of collaborative flows is divided by the total number of flows to make this detection parameter expandable to the network under different operating conditions.

These are the four parameters and characteristic features extracted from every incoming traffic flow which are programmed in the SDN ryu controller. Using these extracted features data, the SVM/Decision tree machine learning algorithm is trained to spot the malicious traffic incoming in the network and classify it as normal or DDOS traffic.

3.1 Machine learning Algorithm

Support vector machine (SVM) classifier a supervised machine learning algorithm which differentiates hyperplane, SVM learns from the data which is provided to it and trained with, it compares the data assigned with the data used for training the algorithm. It creates a pattern map which differentiates the normal traffic features with the features of the attack traffic.

Decision tree classifier is also a machine learning algorithm works similar to SVM, with the difference being in the way it interprets data and classifies it. Decision tree algorithm breaks the data into smaller bits and final result is obtained in a tree structure.

Python has inbuilt libraries for SVM and decision tree classifier algorithm. The controller is programmed to work with both the machine learning algorithms. However, the presented method is tested and experimented with support vector machine (SVM) ML algorithm.

3.2 Development and simulation Platform Tools

In this workt, the algorithm is implemented in the virtual environment that is created by the VMware Workstation. In Virtual Machine, Ubuntu 20.04 is installed for creating the operating environment for the simulation. The following tools and technologies were used to implement the presented methodology.

OpenFlow is a communications protocol for SDN that gives access to the forwarding plane of a network switch or router over the network in software defined network.

Mininet is a network emulator which creates a network of virtual hosts, switches, controllers, and links. Mininet hosts run standard Linux network software, and its switches support OpenFlow for highly flexible custom routing and Software-Defined Networking.In a virtual environment to simulate a large network, Mininet is the open-source network simulator for Software Defined Network. The primary reason to use the Mininet is that it supports OpenFlow Protocol, which is essential for the network configuration and computation for Software Defined Network. It also provides an inexpensive platform for developing, testing, and creating custom topologies in the network.

Ryu Controller is an open, software-defined networking (SDN) Controller designed to increase the agility of the network by making it easy to manage and adapt how traffic is handled. Which is a Python-based programmable controller tool.

Iperf is a network performance tool that is used to measure the bandwidth and datagram loss in a network. This project measures the Transport Control Protocol (TCP) and User Datagram Protocol (UDP) network throughput and data streams. The iperf tool helps to measure the network performance by creating a client and server functionality for both source and destination node

4 Design Specification

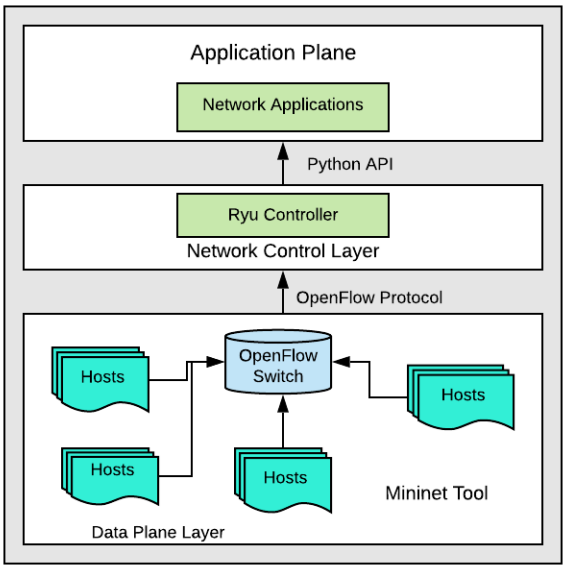


Figure 2: SDN Framework.

The presented SDN framework, data plane has multiple node/hosts virtually created using mininet and all these are connected to the openflow switch which the defines the SDN protocols and the openflow protocol communicates with the control plane of the framework. Control plane controls the data plane and the switches and define rules and also monitors the network traffic flow, here Ryu controller is used as the controller which provides the programming capabilities and allows us to control the routing operations in

the network. The control plane is programmed using pyhton as ryu is a python based controller and uses a python based API to communicate with the application layer, which in our case is network traffic applications.

4.1 Network Design

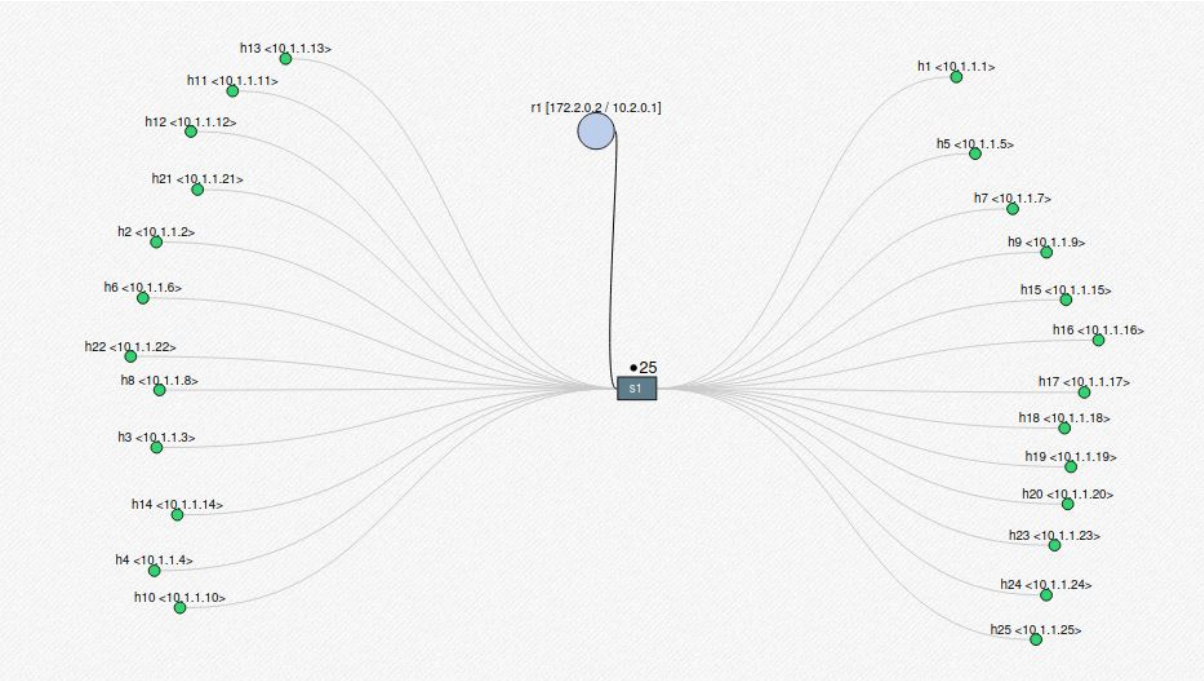


Figure 3: Mininet Network Design.

The network topology is designed using mininet network simulator, the network has 25 hosts/nodes and one single openflow switch and one ryu controller. All the hosts are connected to the switch and the switch is connected to the controller. All of these hosts and switch are controlled by the ryu controller, any port under attack will be blocked immediately.

5 Implementation

The presented method uses both statistical and machine learning methods to detect and mitigate DDOS attacks in a software defined network. The implemented method requires to train the SVM ML algorithm for detecting the attack in a network.
Data Collection module has to collect data of both normal traffic and attack traffic and stores the data in a CSV file for the ML algorithm to use. At first normal traffic data has to collected and then the attack traffic data, it is recommended to collect normal traffic data again after attack traffic data for better accuracy. The data is collected considering all three statistical parameters of feature extraction defined in the methodology namely speed of source IP, speed of flow entries and ratio of flowpair entries in different columns.

Detection and Mitigation takes place after the data has been collected and the controller is set to detection state, then when the normal traffic is generated the SVM algorithm

predicts it as normal traffic and when the attack traffic is generated it instantly detects the traffic as DDOS attack traffic and blocks the port from which the traffic is incoming. Once the port has been blocked the controller is set to a 120sec hardtime, after which it unblocks the port. But, if the attack is still active it again detects and blocks the port for another 120 seconds. After blocking the normal traffic flow is allowed in the network from other ports. This process keeps on going as long as the attack lasts.

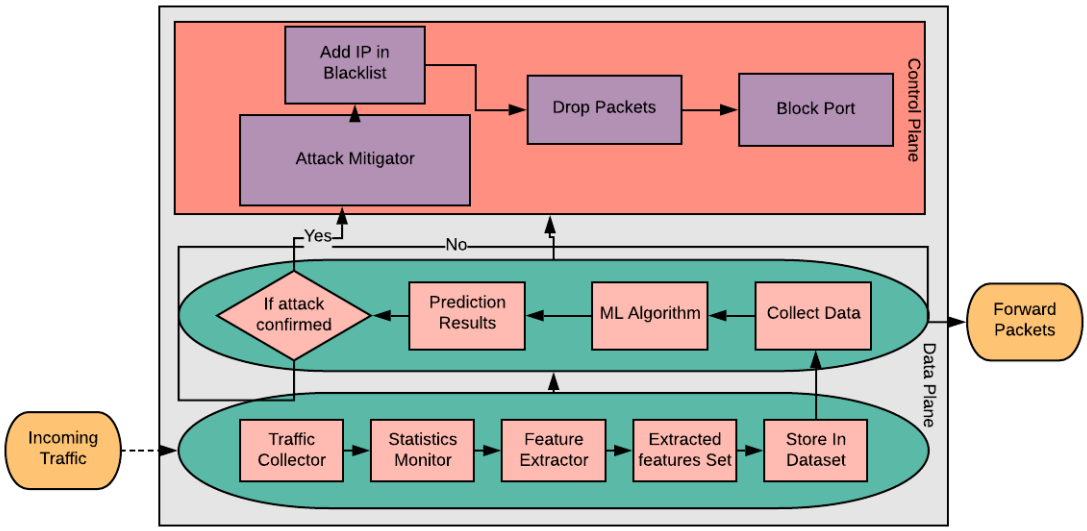


Figure 4: Flowchart of the Presented Method.

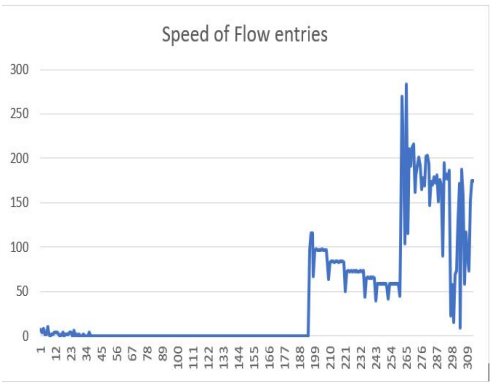
Software defined networks has various protocols and controllers, each of those are designed to perform in a certain way and provide efficiency and flexibility in a particular aspect. The presented method is implemented using the most popular and out performing tools for the detection and mitigation of DDOS attacks in a software defined network.
Openflow Protocol is the most popular and standard protocol for software defined networks, hence openVswitch is used for this project.As the presented method is a combination of statistical and machine learning methods, the logic and techniques are programmed using python. Statistical method include parameters such as speed of source IP, speed of flow entires and ratio of flowpair entries, all of these logic is programmed in the controller.
Ryu Controller is an open-source python based programmable controller, which is used to define the rules and logic for the switches to follow in the methodology.
Mininet is a network simulator and creates a virtual network topology with controller, switches and hosts, in this work a single openVswitch with 10 and 25 hosts are created for multiple tests.
Hping3 is a packet generator which generates TCP/IP traffic in the network, it is mostly used to test network security. Normal and attack traffic scripts are written to generate traffic automatically using this tool.
Iperf is also a network traffic generator and network performance tester, which in this work is used to generate traffic manually.
All of these tools are installed in ubuntu 20.04.1 LTS operating system which is installed VMware Workstation.

6 Evaluation

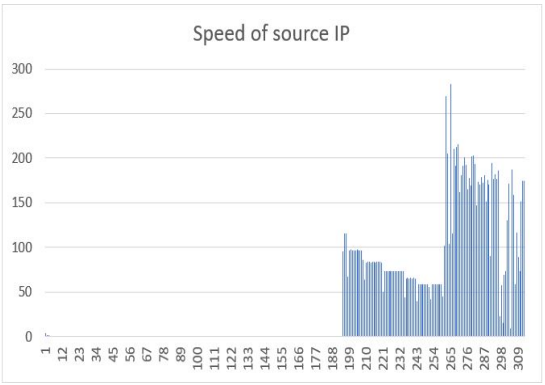
This section presents the tests conducted on the SDN with normal and attack traffic being sent to the network from different ports and the overall detection and mitigation process with the accuracy and detection rate of the implemented method. **The datasets** created first has the 600+ samples of normal traffic data and 300+ samples of attack traffic data stored for the SVM algorithm to train and analyses to predict the attack. The tests are conducted for 300 seconds with 2 seconds interval for traffic collection, SVM predicts the traffic every 2 seconds.

6.1 Experiment / Case Study 1

In this experiment the normal traffic is sent from all the ports and attack is being sent from port/host 1 in the network with incoming traffic being captured every 3 seconds. The network topology is created using mininet which has 1 openflow switch with 10 hosts in the network.

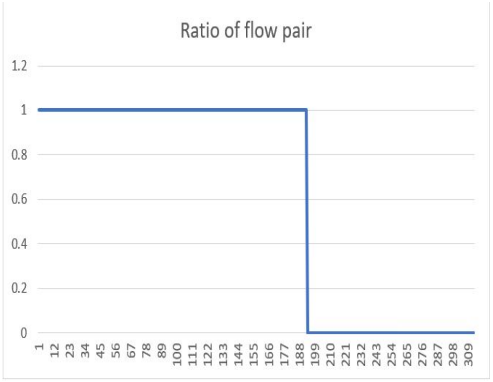


(a) SFE

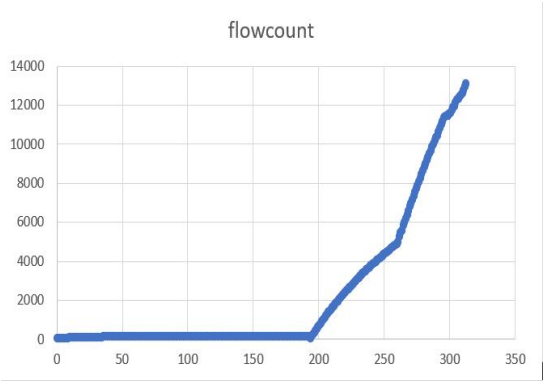


(b) SSP

In the graphs A and B, X axis is the Data counts and Y axis is the speed count of the flow entries and source IP. From graph it is shown how the speed of flow entries and speed of IP sources increases when attack traffic is sent in the network, whereas the straight line being the normal traffic flow in the network.



(a) RFIP



(b) Flowcount

The graphs A shows the ratio of flow pairs reduced when the network has attack traffic incoming and graph B shows the flowcount of the normal traffic and attack traffic.

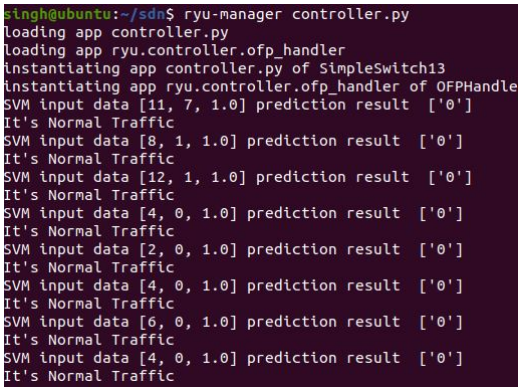


Figure 7: Normal Traffic Prediction

SVM machine learning algorithm predicting the traffic as normal traffic.

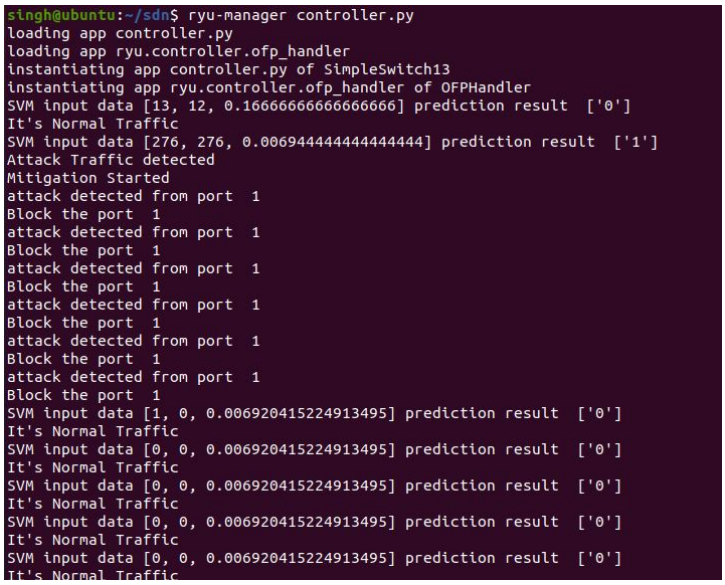


Figure 8: Attack Traffic Prediction

SVM machine learning algorithm predicting the traffic as DDOS attack traffic and mitigation process being started instantly and blocking the port 1 from which attack traffic is incoming.

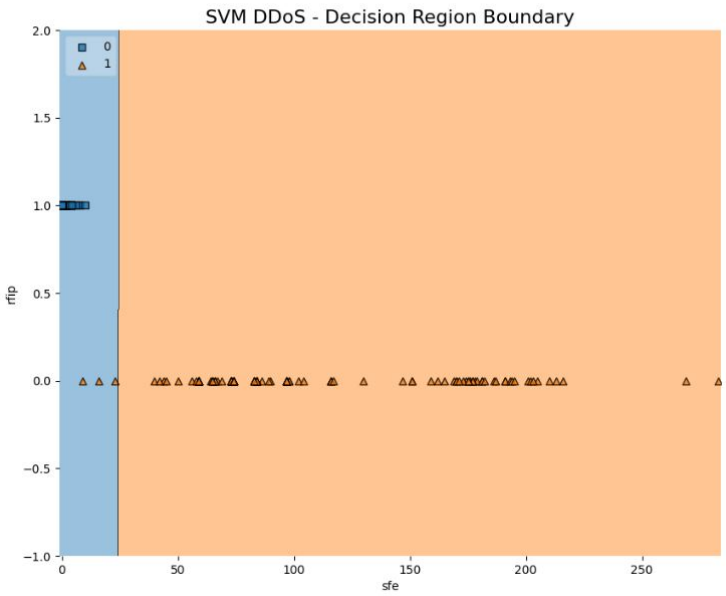


Figure 9: SVM Decision Boundary

The above image shows the decision taken by SVM ML algorithm, blue area being the normal traffic and orange area being the attack traffic in the network.

```
singh@ubuntu:~/sdn/analysis$ python accuracy_score.py
Accuracy is  98.71794871794873
cross-validation score 0.9957446808510639
```

Figure 10: Accuracy Score

The accuracy score achieved by the presented method is 98.71% and cross validation score with the training data and test data achieved is 99.57%.

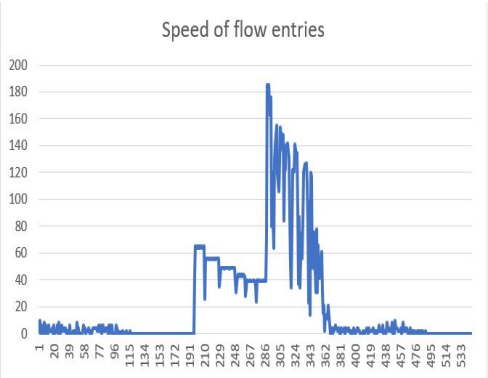
```
singh@ubuntu:~/sdn/analysis$ python detection_rate.py
Calculating Detection Ratio & False
Detection rate  0.9285714285714286
False Alarm rate  0.0
singh@ubuntu:~/sdn/analysis$ python graph.py
singh@ubuntu:~/sdn/analysis$
```

Figure 11: Detecion Rate

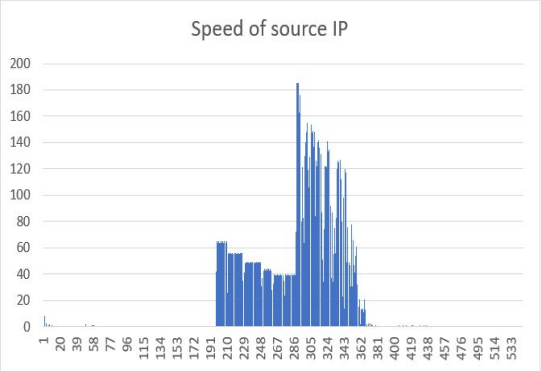
The DDOS attack traffic detection rate in the network achieved by the presented method is 92.8% and 0% false alarm, meaning no normal traffic was considered as attack traffic.

6.2 Experiment / Case Study 2

In this experiment the normal traffic is sent from all the ports and attack is being sent from port/host 8 in the network with incoming traffic being captured every 2 seconds. The network topology is created using mininet which has 1 openflow switch with 25 hosts in the network.

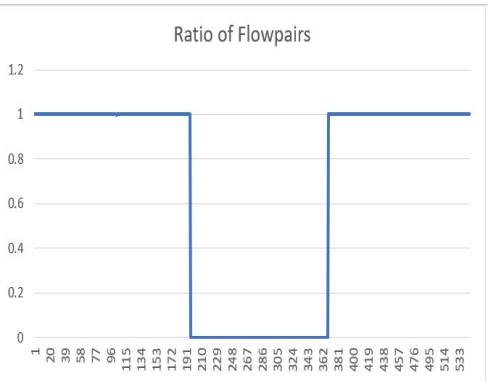


(a) SFE

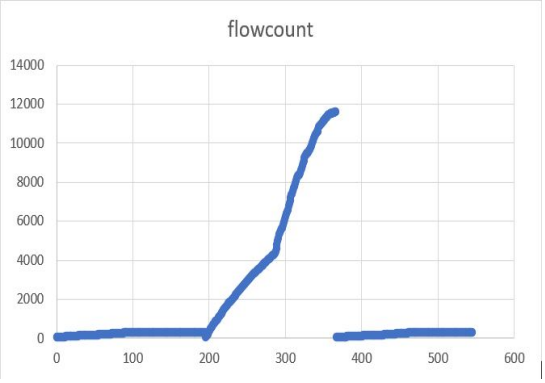


(b) SSP

In the graphs A and B, X axis is the Data counts and Y axis is the speed count of the flow entries and source IP. From graph it is shown how the speed of flow entries and speed of source IP increases when attack traffic is sent in the network, whereas the straight line being the normal traffic flow in the network. The graphs A shows the ratio of flow pairs



(a) RFIP



(b) Flowcount

reduced when the network has attack traffic incoming and graph B shows the flowcount of the normal traffic and attack traffic.

```
SVM input data [0, 0, 1.0] prediction result  ['0']
It's Normal Traffic
SVM input data [31, 257, 0.0] prediction result  ['1']
Attack Traffic detected
Mitigation Started
attack detected from port  8
Block the port  8
attack detected from port  8
Block the port  8
SVM input data [1, 0, 0.0] prediction result  ['1']
Attack Traffic detected
Mitigation Started
SVM input data [0, 0, 0.0] prediction result  ['1']
Attack Traffic detected
Mitigation Started
SVM input data [0, 0, 0.0] prediction result  ['1']
Attack Traffic detected
Mitigation Started
SVM input data [0, 0, 0.0] prediction result  ['1']
Attack Traffic detected
Mitigation Started
```

Figure 14: Attack Traffic Prediction

SVM machine learning algorithm predicting the traffic as DDOS attack traffic and mitigation process being started instantly and blocking the port 8 from which attack traffic is incoming.

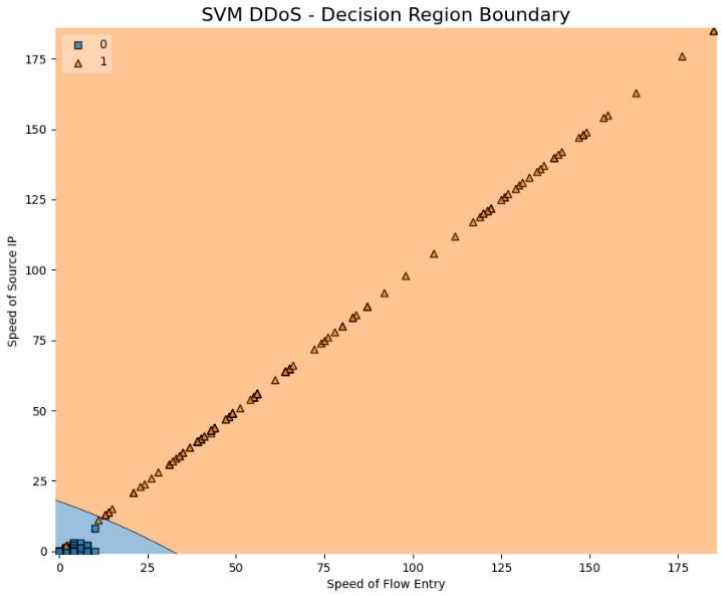


Figure 15: SVM Decision Boundary

The above image shows the decision taken by SVM ML algorithm, blue area being the normal traffic and orange area being the attack traffic in the network.

```
singh@ubuntu:~/Downloads/sdn2/analysis$ python accuracy_score.py
Accuracy is  99.26470588235294
cross-validation score 0.9975308641975309
```

Figure 16: Accuracy Score

The accuracy score achieved by the presented method is 99.26% and cross validation score with the training data and test data achieved is 99.75%.

```
singh@ubuntu:~/Downloads/sdn2/analysis$ python detection_rate.py
Calculating Detection Ratio & False
Detection rate  1.0
False Alarm rate  0.0
```

Figure 17: Detecion Rate

The DDOS attack traffic detection rate in the network achieved by the presented method is 100% and 0% false alarm, meaning no normal traffic was considered as attack traffic.

6.3 Discussion

The implemented method was experimented in 2 different attack cases. In experiment 1 the attack traffic is being sent from port 1 with only 10 hosts in the network, the results obtained showed that the SVM ML algorithm achieved accuracy of 98.71% and the cross validation score with the training data was 99.57% and the attack traffic detection rate was close to 100% with no false alarm meaning no normal traffic was considered as

malicious. In experiment 2 the attack traffic is being sent from port 8 with only 25 hosts in the network, the results obtained showed that the SVM ML algorithm achieved accuracy of 99.26% and the cross validation score with the training data was 99.75% and here as well the detection rate is 100% with no false alarm. These results show that the presented methods is very accurate in detecting malicious traffic in the network with zero false alarms, so no normal traffic is being refused access in the network.
However, if an IP address which is considered normal in trained data and is in trusted IP list is used to attack the network cannot be detected using this method and can dodge the security and get in the network, though chances to this happening is very low a comprehensive network traffic analyser must be designed to prevent these cases.

7 Conclusion and Future Work

Software defined network provides us the capabilities to design and perform operations in the network by programming which is not case with traditional networks. Using SDN to detect and mitigate DDOS attacks in cloud environment was the main goal of this work. The implemented method is a combination of statistical features like source of IP, speed of flow entries , flowcount and ratio of flow-pair and SVM machine learning algorithm to detect and predict DDOS attacks in the network, experimented results shows the presented method can provide accuracy 0f 99.26% and malicious traffic detection rate of 100% with zero false predictions of the traffic. However, security is never full proof and can always be shatter same way implemented method has a drawback, an attack from trusted IP sources can be used to send malicious traffic in the network which the SVM won't be able to predict.
In future the implemented method can be designed to have multiple switches and controller in the network with a comprehensive network packet analyser. Currently 4 features are being used in the statistical analysis, furthermore features can be extracted and used with ML algorithm to have better and accurate prediction of malicious traffic.

References

Ahmad, I., Namal, S., Ylianttila, M. and Gurtov, A. (2015). Security in software defined networks: A survey, *IEEE Communications Surveys Tutorials* **17**(4): 2317–2346. JCR Impact Factor: 22.973 (2019).

Alshamrani, A., Chowdhary, A., Pisharody, S., Lu, D. and Huang, D. (2017). A defense system for defeating ddos attacks in sdn based networks, *Proceedings of the 15th ACM International Symposium on Mobility Management and Wireless Access*, MobiWac '17, Association for Computing Machinery, New York, NY, USA, p. 83–92. ERA Ranking: B.

Bhushan, K. and Gupta, B. B. (2019). Distributed denial of service (ddos) attack mitigation in software defined network (sdn)-based cloud computing environment, *Journal of Ambient Intelligence and Humanized Computing* **10**(5): 1985–1997. JCR Impact Factor: 1.910 (2019).

Cui, Y., Yan, L., Li, S., Xing, H., Pan, W., Zhu, J. and Zheng, X. (2016). Sd-anti-ddos: Fast and efficient ddos defense in software-defined networks, *Journal of Network and Computer Applications* **68**: 65 – 79. JCR Impact Factor: 5.273 (2019).

Dehkordi, A. B., Soltanaghaei, M. and Boroujeni, F. Z. (2020). The ddos attacks detection through machine learning and statistical methods in sdn, *The Journal of Supercomputing* pp. 1–33. JCR Impact Factor: 2.600 (2019).

Giotis, K., Argyropoulos, C., Androulidakis, G., Kalogeras, D. and Maglaris, V. (2014). Combining openflow and sflow for an effective and scalable anomaly detection and mitigation mechanism on sdn environments, *Computer Networks* **62**: 122 – 136. JCR Impact Factor: 3.030 (2019).

He, D., Chan, S., Ni, X. and Guizani, M. (2017). Software-defined-networking-enabled traffic anomaly detection and mitigation, *IEEE Internet of Things Journal* **4**(6): 1890–1898. JCR Impact Factor: 9.515 (2019).

Jin, C., Wang, H. and Shin, K. G. (2003). Hop-count filtering: An effective defense against spoofed ddos traffic, *Proceedings of the 10th ACM Conference on Computer and Communications Security*, CCS '03, Association for Computing Machinery, New York, NY, USA, p. 30–41. CORE Ranking: A*.

Li, C., Wu, Y., Yuan, X., Sun, Z., Wang, W., Li, X. and Gong, L. (2018). Detection and defense of ddos attack–based on deep learning in openflow-based sdn, *International Journal of Communication Systems* **31**(5): e3497. JCR Impact Factor: 1.278 (2019).

Mohammed, S. S., Hussain, R., Senko, O., Bimaganbetov, B., Lee, J., Hussain, F., Kerrache, C. A., Barka, E. and Bhuiyan, M. Z. A. (2018). A new machine learning-based collaborative ddos mitigation mechanism in software-defined network, *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, IEEE, pp. 1–8. CORE Ranking: B.

Myint Oo, M., Kamolphiwong, S., Kamolphiwong, T. and Vasupongayya, S. (2019). Advanced support vector machine-(asvm-) based detection for distributed denial of service

(ddos) attack on software defined networking (sdn), *Journal of Computer Networks and Communications* . JCR Impact Factor: 0.510 (2019).

Nhu-Ngoc Dao, Junho Park, Minho Park and Sungrae Cho (2015). A feasible method to combat against ddos attack in sdn network, *2015 International Conference on Information Networking (ICOIN)*, Cambodia, Cambodia, pp. 309–311. CORE Ranking: B.

Sahoo, K. S., Iqbal, A., Maiti, P. and Sahoo, B. (2018). A machine learning approach for predicting ddos traffic in software defined networks, *2018 International Conference on Information Technology (ICIT)*, Bhubaneswar, India, India, pp. 199–203. CORE Ranking: C.

Santos, R., Souza, D., Santo, W., Ribeiro, A. and Moreno, E. (2019). Machine learning algorithms to detect ddos attacks in sdn, *Concurrency and Computation: Practice and Experience* p. e5402. JCR Impact Factor: 1.167 (2019).

Wang, Y., Hu, T., Tang, G., Xie, J. and Lu, J. (2019). Sgs: Safe-guard scheme for protecting control plane against ddos attacks in software-defined networking, *IEEE Access* **7**: 34699–34710. JCR Impact Factor: 4.640 (2019).

Xia, W., Wen, Y., Foh, C. H., Niyato, D. and Xie, H. (2015). A survey on software-defined networking, *IEEE Communications Surveys Tutorials* **17**(1): 27–51. JCR Impact Factor: 22.973 (2019).

Xu, Y. and Liu, Y. (2016). Ddos attack detection under sdn context, *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, San Francisco, CA, USA, pp. 1–9. CORE Ranking: A*.

Yan, Q. and Yu, F. R. (2015). Distributed denial of service attacks in software-defined networking with cloud computing, *IEEE Communications Magazine* **53**(4): 52–59. JCR Impact Factor: 10.356 (2019).

Yan, Q., Yu, F. R., Gong, Q. and Li, J. (2016). Software-defined networking (sdn) and distributed denial of service (ddos) attacks in cloud computing environments: A survey, some research issues, and challenges, *IEEE Communications Surveys Tutorials* **18**(1): 602–622. JCR Impact Factor: 22.973 (2019).