DDOS ATTACK : DETECTION & MITIGATION

A Project Report submitted in partial fulfillment of the requirements for the award of the degree of

## Bachelor of Technology in

**Computer Science and Engineering**

By

**Pratham Dhadhiya (112115114)**

## Under the Supervision of: Dr. Anupama Arun Semester:VII

**Department of Computer Science and Engineering Indian Institute of Information Technology, Pune (An Institute of National Importance by an Act of Parliament)**

**November 2024**

# BONAFIDE CERTIFICATE

This is to certify that the project report entitled **“DDOS Attack: Detection and Mitigation”** submitted by **Pratham Dhadhiya** bearing the **MIS No: 112115114** in completion of his project work under the guidance of **Dr. Anupama Arun** is accepted for the project report submission in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in the **Department of Computer Science and Engineering**, Indian Institute of Information Technology, Pune (IIIT Pune), during the academic year **2024-25**.

### Supervisor’s Name HoD Name

**Dr. Anupama Arun Dr.Bhupendra Singh**

Project Supervisor Head of the

Department Department

Designation of the Supervisor Assistant Professor

Department of the Supervisor Department of

CSE CSE

IIIT Pune IIIT Pune

Project Viva-voce held on

# Undertaking for Plagiarism

I **Pratham Dhadhiya** solemnly declare that research work presented in the **report/dissertation** titled **“DDOS Attack: Detection and Mitigation "** is solely **my** research work with no significant contribution from any other person. Small contribution/help wherever taken has been duly acknowledged and that complete report/dissertation has been written by **me**. I understand the zero tolerance policy of **I** declare that no portion of my **report/dissertation** has been plagiarized and any material used as reference is properly referred/cited. I undertake that if I am found guilty of any formal plagiarism in the above titled thesis even after award of the degree, the Institute reserves the rights to withdraw/revoke my **B. Tech** degree.

### Student Name and Signature with Date

**Pratham Dhadhiya (112115114)**

# Conflict of Interest

### Manuscript title : DDOS Attack: Detection and Mitigation

The author whose name is listed immediately below certify that he have NO affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers’ bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

### Student Name and Signature with Date

**Pratham Dhadhiya (112115114)**

# ACKNOWLEDGEMENT

This project would not have been possible without the help and cooperation of many. We would like to thank the people who helped us directly and indirectly in the completion of this project work.

First and foremost, I would like to express my gratitude to our honorable Director, **Dr. Prem Lal Patel**, for providing his kind support in various aspects. I would like to express my gratitude to project guide **Dr. Anupama Arun**, **Department of CSE**, for providing excellent guidance, encouragement, inspiration, constant and timely support throughout this **B.Tech Project**.I would like to express my gratitude to the **Head of Department Dr. Bhupendra Singh**, **Department of CSE**, for providing his kind support in various aspects. I would also like to thank all the faculty members in the **Department of CSE** and our classmates for their steadfast and strong support and engagement with this project.

**DDOS ATTACK : DETECTION & MITIGATION**

# Abstract

Software-defined networking (SDN) is the future of networking in which all network devices have to completely separate their data plane and control plane. This may lead to an opportunity to centralize the control system so that more powerful management, security, and scalability can be made for the network. This helps program the network configurations for smooth operations, improvement, and by and large simplification of the network architecture. However, with these benefits, SDN is still exposed to one of the most serious security threats: Distributed Denial-of-Service attacks. DDoS attacks overwhelm a network with so much traffic that it overloads the resources and thus blocks access to the server from legitimate users. This is a consequence of particular concern in cloud environments since the associated risk with eventual DDoS attacks will be higher.

Since DDoS attacks have gained much significance in SDN environments, several researchers are investigating novel defense methodologies, which aim at integrating statistical and machine learning-based efficient detection and mitigation approaches. By using machine learning techniques, it will enable SDN to detect patterns for DDoS attacks to take quick actions, thus providing adaptive security. This proactive approach ensures that any development of resilience within the SDN infrastructure is resilient against threats that the traditional methods of security may not handle.

A special implementation to show the effectiveness of this approach involves using the Ryu controller and Mininet network simulator, which are both compatible with the widely used OpenFlow protocol in SDN. The machine learning model implemented in this setup achieved an accuracy rate of 99.26%, with a detection rate of 100% in identifying and mitigating DDoS attacks within a software-defined network. These results emphasize the potentiality of a combination of machine learning and SDN to provide robust, scalable, and intelligent network security solutions to address growing threats due to DDoS attacks in modern network environments.

**Keywords:** Software-Defined Networking (SDN), DDoS Attacks, Machine Learning,

OpenFlow Protocol

i

# TABLE OF CONTENTS

### Abstract i

[List of Figures/Symbols/Nomenclature iv](#_bookmark0)

[List of Tables v](#_bookmark1)

1. Introduction 1
   1. [Overview of work 1](#_bookmark2)
   2. [Literature Review 1](#_bookmark3)
   3. Motivation of work 4
   4. [Research Gap. 5](#_bookmark4)
2. Problem Statement 6
   1. [Research Objectives 6](#_bookmark5)
   2. [Analysis And Design 7](#_bookmark6)
3. Proposed Work 9
   1. [Methodology of work. 9](#_bookmark7)
   2. [Hardware & Software specifications. 11](#_bookmark8)
   3. [Dataset Description 11](#_bookmark9)
4. Results and Discussion 14
5. Conclusion and Future Scope 20

**References**

# List of Figures [/ Symbols/ Nomenclature](#_bookmark0)

**Figure 1.** SDN architecture with vulnerabilities of DDOS attacks.(Yan and Yu (2015)) [3]

**Figure 2.** SDN Framework

**Figure 3.** Mininet Network Design

**Figure 4.** Normal Traffic Prediction for Case Study 1 **Figure 5.** Attack Traffic Prediction for Case Study 1 **Figure 6.** SVM Decision Boundary for Case Study 1 **Figure 7.** Accuracy score for Case Study 1

**Figure 8**. Detection Rate for Case Study 1

**Figure 9.** Attack Traffic Prediction for Case Study 2 **Figure 10.** SVM Decision Boundary for Case Study 2 **Figure 11.** Accuracy score for Case Study 2

**Figure 12.** Detection Rate for Case Study 2

iv

# List of Tables

**Table 1.** Comparison Table for Literature Review

v

# Chapter 1 Introduction

## Overview of Work

Cloud computing, with several advantages over traditional networks, has grown very fast across industries and academia. Software-Defined Networking (SDN) plays an important role in improving cloud networks for better performance, scalability, and manageability. Through decoupling the data and control planes of network devices, SDN can allow their centralized management and programming, having the ability of the control plane to be managed by an SDN controller. It consists of three layers: an infrastructure layer made up of network devices; a control layer, the controller; and an application layer that hosts the network applications. SDN-based cloud environments reinforce network flexibility, security, and manageability to enable "Networking as a Service," sometimes referred to as NaaS.

## Literature Review

Software-Defined Networking (SDN) is increasingly being recognized as the "future of networking" due to its rapid growth and widespread success. One of the key reasons behind its success is the separation of the control plane from the data plane in network devices. This decoupling allows for centralized management of the entire network, eliminating the need to configure each individual device. The data plane, which consists of the switches, forwards network traffic based on policies set by applications running on the centralized controller. This advancement in networking technology has significantly accelerated service delivery and offers greater flexibility and agility, enabling the provisioning of both physical and virtual network devices from a central location.

The literature review is divided into three main sections: Traffic Analysis in SDN, DDoS Attack Detection and Mitigation using SDN, and Machine Learning Approaches for Anomaly Detection. These sections provide a comprehensive overview of research in these areas, particularly focusing on methods for preventing DDoS attacks through SDN.

### Traffic Analysis in SDN

This section discusses methods proposed by researchers for analyzing incoming traffic in Software-Defined Networks (SDN) to detect DDoS attacks. Two primary techniques are commonly used for traffic analysis:

* + - 1. **Network Data Analysis**: This method focuses on identifying malicious traffic by extracting features of both attack and normal traffic. Researchers have proposed various techniques for feature extraction. For example, Xu and Liu (2016) **[16]** introduced an algorithm to enhance flow monitoring on switches, enabling quick identification of potential victims and attackers. While the results showed high accuracy, the article lacked details on the tools and simulation methods used. Similarly, Wang et al. (2019) **[9]** proposed four feature extraction methods, including byte rate counting and flow variations, implemented using the Ryu controller and Mininet to reduce controller response time during DDoS attacks. He et al. (2017) **[6]** used a density peak clustering algorithm to correlate traffic features and detect DDoS

attacks, claiming their method outperformed other machine learning approaches, though it didn’t support real-time traffic analysis.

* + - 1. **SNMP Analysis**: The Simple Network Management Protocol (SNMP) uses the Management Information Base (MIB) to collect and store traffic data, which can aid in DDoS detection by integrating an Intrusion Detection System (IDS). Nhu-Ngoc Dao et al. (2015) **[14]** proposed a method for detecting DDoS flooding attacks by monitoring packet count, IP sources, and average user connections. Although the method was simulated using Mininet, it could be bypassed by multiple IP sources or delayed attacks. The article did not provide sufficient details for reproducing the experiment. Another defense mechanism proposed by Jin et al. (2003) **[13]** used hop- count filtering to detect spoofed IP traffic by monitoring the number of hops and TTL (Time to Live) values in IP headers. While the method showed promising results, it was complex and time-consuming to implement.

These methods offer various approaches for detecting and mitigating DDoS attacks, though some have limitations in terms of real-time implementation and practicality.

### DDOS Attack Detection and Mitigation Using SDN

This section discusses related work in detecting and mitigating DDoS attacks in Software- Defined Networking (SDN) using both traditional methods and innovative detection strategies proposed by researchers.

Cui et al. (2016) presented a software-defined anti-DDoS defense mechanism, featuring a four-stage process: initial stage, detection stage, traceback stage, and mitigation stage. The approach uses a trigger mechanism for attack detection and a traceback system to identify the source IP of incoming DDoS attacks. It also calculates packet velocity and traffic records to detect malicious behavior. The method, implemented using the Ryu controller and simulated with Mininet, was supported by well-documented results, allowing for reproducibility of the experiments. **[3]**

Bhushan and Gupta (2019) proposed a mitigation technique that focuses on managing the flow table size in SDNs. DDoS attacks often overwhelm the flow table, preventing new entries from being added to the switch. Their solution uses two databases: one for blacklisting malicious IPs and another for monitoring flow table status across switches. If the flow table is full, traffic is redirected to the nearest switch to avoid blockage. However, this approach does not have an early detection strategy and may allow malicious traffic to remain in the network. The method, implemented with the POX controller and Mininet, is well-documented, but lacks an early detection mechanism. **[2]**

Alshamrani et al. (2017) presented an SDN-based defense system where packets under attack are redirected to a honeypot instead of being dropped. The honeypot learns attacker behavior and monitors IPs to enhance detection. This novel approach, implemented with the POX controller and Mininet, is well-documented, but it may not be effective if the attacker switches IPs or uses botnets. Despite this, the method is documented with evidence, though it lacks complete reproducibility. **[5]**

Giotis et al. (2014) combined OpenFlow and sFlow to create an anomaly detection system for SDNs. The architecture consists of two modules: an anomaly detection module and a traffic collector. Traffic is analyzed after collection, and an entropy-based algorithm is used to identify malicious traffic. The method, implemented with the NOX controller and Mininet, can handle larger network topologies and real-time statistics. While it is well-documented and reproducible, the method uses an outdated controller and lacks an efficient mitigation strategy. **[12]**

These approaches offer valuable insights into DDoS attack detection and mitigation in SDNs, but each has its own limitations, such as the absence of early detection strategies or inefficient mitigation mechanisms.

### Machine Learning Approaches for Attack Detection

This section explores various machine learning (ML) algorithms and methodologies used by researchers to predict and detect DDoS traffic in Software-Defined Networks (SDN).

Santos et al. (2019) discussed different ML approaches for DDoS detection and mitigation in SDNs. They implemented four ML algorithms: MLP, Decision Tree, Support Vector Machine (SVM), and Random Forest, simulated using Mininet. The results indicated that the Random Forest algorithm achieved the highest accuracy, while the Decision Tree algorithm provided the best processing time for DDoS detection. However, the study had some limitations in classifying flow table and bandwidth attacks. Despite these, the paper is comprehensive and well-documented. **[8]**

Similarly, Sahoo et al. (2018) compared multiple ML algorithms, including K-Nearest Neighbor (KNN), Naive Bayes (NB), SVM, Random Forest, and Linear Regression (LR). Their experiments showed that LR and Random Forest achieved the best prediction accuracy of 98%, with Random Forest having a lower execution time than LR. However, the tools and simulation details were not provided, making the study difficult to reproduce. **[15]**

Myint Oo et al. (2019) proposed an advanced SVM-based algorithm (ASVM) for DDoS detection in SDNs. The ASVM algorithm classifies parameters collected from the feature extraction stage to predict DDoS attacks. This approach claims to reduce training and testing time while achieving 97% detection accuracy. The study used the OpenDaylight controller and Mininet for simulation, and the results were well-supported by graphical evidence. **[17]**

Dehkordi et al. (2020) combined entropy-based methods with machine learning for DDoS detection. Their approach had three stages: traffic data collection, entropy thresholds, and ML classifiers. Traffic was first filtered using static entropy thresholds, then analyzed with ML algorithms. Their experiments, conducted with the Floodlight controller and Mininet, showed improved accuracy and prediction for detecting DDoS attacks. **[4]**

These studies show the evolution of machine learning techniques for DDoS detection, with each offering unique advantages and challenges in SDN environments.

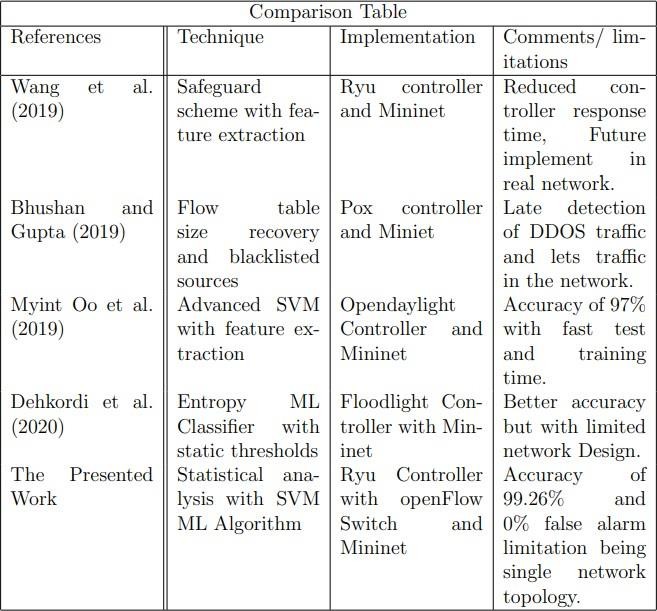


Table 1. Comparison Table for Literature Review

## Motivation of the Work

The demand for cloud services is surging, with more companies and organizations moving to the cloud to achieve better performance and security. As our reliance on the internet grows, along with the amount of personal data stored online, concerns about data security continue to rise. Robust security measures are essential to protect this data from various network threats. One major threat is the Distributed Denial-of-Service (DDoS) attack, which disrupts server access and denies cloud services to users. Extensive research and methods have been developed to counter these attacks, and one effective solution is leveraging Software-Defined Networking (SDN) to strengthen network security.

## Research Gap

* **Real-time Detection and Scalability**: Existing methods lack real-time capabilities and scalability for large SDN environments. **[6]**
* **Holistic Detection Approaches**: There is a need for combining multiple traffic features to improve detection robustness. **[9]**
* **Early Detection and Efficient Mitigation**: Early detection strategies are missing, and mitigation occurs only after flow table congestion. **[2]**
* **Improved Traceback Mechanisms**: Traceback methods are ineffective against advanced attack methods like IP spoofing and botnets. **[3]**
* **Hybrid Machine Learning Models**: There is a need for combining different machine learning models to improve detection accuracy and efficiency. **[15] & [7]**
* **Real-World Validation**: Most studies rely on idealized datasets, and there’s a lack of real-world SDN traffic for training. **[7]**
* **Unaddressed Challenges with Botnets and IP Spoofing**: Many methods fail to address more advanced DDoS techniques like botnets and IP spoofing. **[8] & [15]**

# Chapter 2 Problem Statement

The project aims to develop a DDoS attack detection and mitigation system for Software Defined Networks (SDN) using machine learning algorithms. Traditional network systems are vulnerable to DDoS attacks, which can lead to severe privacy issues and data loss. This project utilizes statistical network analysis and machine learning, specifically Support Vector Machine (SVM), to detect and mitigate such attacks. By analyzing features such as IP source speed, flow counts, flow entry speed, and pair-flow ratios, the system identifies abnormal traffic patterns in real-time, distinguishing between normal and attack traffic, thereby enhancing network security and performance.

## Research Objectives

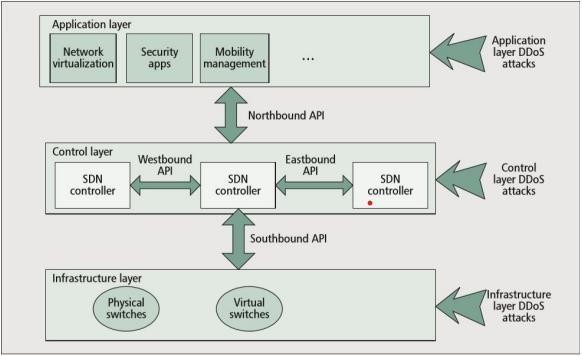


Figure 1: SDN architecture with vulnerabilities of DDOS attacks.(Yan and Yu (2015)) **[3]**

The above image illustrates the different layers of the SDN that are vulnerable to DDOS attacks. Based on this, the research objectives of this project can be outlined as follows:

* + - **Identify Vulnerabilities in SDN**: To examine and highlight the vulnerabilities in the different layers of the SDN architecture that are susceptible to Distributed Denial of Service (DDOS) attacks.
    - **Understand DDOS Attacks**: To explore the nature of DDOS attacks, including the difference between DOS and DDOS attacks, and how they affect the cloud network.
    - **Assess the Increasing Threat of DDOS Attacks**: To understand the growing frequency and severity of DDOS attacks, as reported by industry studies like Akamai, and their impact on cloud network security.
    - **Evaluate SDN Capabilities**: To investigate how SDN's centralized management and security features can be leveraged to detect and mitigate DDOS attacks effectively.
    - **Develop a Detection and Mitigation Method**: To propose and implement a method that uses SDN to detect and mitigate DDOS attacks in a cloud network environment, improving network security.
    - **Contribute to Ongoing Research**: To contribute to the ongoing research on DDOS attack prevention in cloud networks by developing a novel method for detection and mitigation within SDN-based environments.

## Analysis and Design

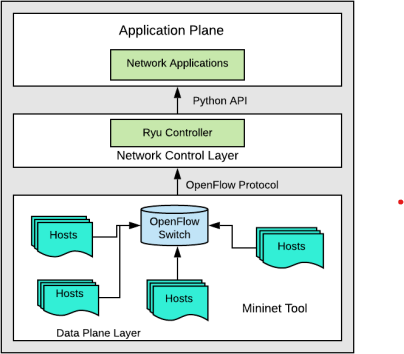


Figure 2: SDN Framework

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.

### 2.2.1 Network Design

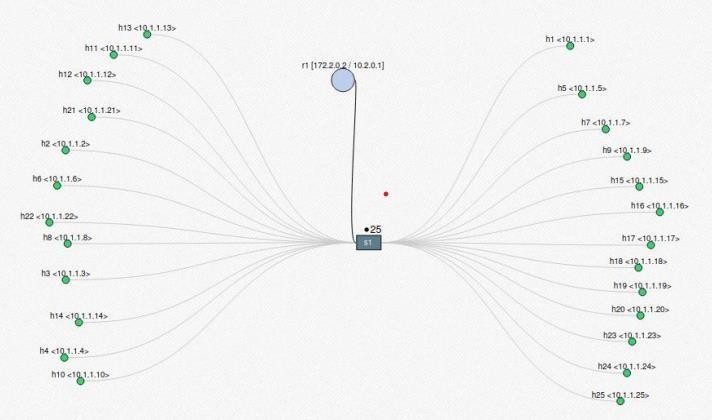


Figure 3: Mininet Network Design.

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

# Chapter 3 Proposed Work

This work proposes a method for detecting and mitigating DDOS attacks in Software-Defined Networks (SDN) using statistical network analysis and machine learning. By separating the control and data planes, SDN enables centralized management, enhancing security. The method monitors key traffic features like IP source speed, flow counts, and flow entry speeds to detect abnormalities. Using machine learning algorithms, such as Support Vector Machine (SVM) and Decision Trees, the system classifies traffic as normal or malicious. Developed in a virtual environment with tools like Mininet, OpenFlow, and the Ryu controller, this approach aims to efficiently prevent DDOS attacks while minimizing false alarms.

## Methodology of work

Traditional network systems are vulnerable to various attacks, which can lead to privacy issues and data leaks from network packet information. To safeguard against such threats in public networks, this work proposes a method for detecting and mitigating DDOS attacks using Software-Defined Networking (SDN) combined with statistical network analysis and machine learning techniques. SDN separates the control and data planes, allowing for centralized management and better protection against unauthorized network access.

For detecting DDOS attacks, incoming network traffic is analyzed through specific features and parameters, which are collected for both training and testing. These features are essential for distinguishing normal traffic from attack traffic. The following key parameters are monitored:

* + 1. **Speed of IP Sources (SSIP):** This feature tracks the number of incoming IP sources within a specific time interval. SSIP is defined as the total number of incoming IP sources (SumIPsrc) divided by the sampling time interval (T). The system monitors and collects data on IP sources every three seconds. For normal traffic, SSIP is typically low, while during an attack, the count of incoming sources tends to be higher.

SSIP = (Sum IPsrc) /T

* + 1. **Flow Count of the Traffic:** This refers to the number of traffic flows entering the network. Normal traffic usually has fewer flows compared to DDOS attack traffic, which results in a higher flow count.
    2. **Speed of Flow Entries (SFE):** This parameter measures the number of flow entries into the network switch within a given time interval. Defined as the total number of flow entries (N) divided by the time interval (T), SFE increases significantly during a DDOS attack, unlike in normal traffic where it remains stable.

SFE = N /T

* + 1. **Ratio of Pair-Flow Entries (RPF):** This feature calculates the ratio of interactive flow entries (SrcIPs) to the total number of flows within a time period (N). In normal

conditions, flow entries from source IPs correspond to the destination IPs of other flows, but during a DDOS attack, the flow pattern becomes erratic, and the number of interactive flows drops drastically.

RPF = (Src IPs) /N

These four features are collected from each incoming traffic flow and programmed into the SDN Ryu controller. The data collected is then used to train machine learning algorithms, such as Support Vector Machine (SVM) and Decision Trees, to classify the network traffic as either normal or malicious (DDOS). This approach aims to provide an effective mechanism for detecting and mitigating DDOS attacks within SDN environments.

### [7] .

* + 1. **Machine Learning Algorithms**

The **Support Vector Machine (SVM)** is a supervised machine learning algorithm that classifies data by finding a hyperplane that best separates the data into different classes. SVM learns from the provided training data and compares it to incoming traffic data, creating a pattern map that distinguishes between normal traffic and attack traffic. This enables the system to classify network flows as either normal or malicious based on their characteristics.

The **Decision Tree Classifier** operates in a similar manner to SVM, but it uses a different approach to data interpretation. It breaks down the data into smaller subsets, and the final classification result is represented in the form of a tree structure. Python’s built-in libraries support both SVM and Decision Tree classifiers. The controller in this study is programmed to work with both algorithms, but the primary method tested and used in this research is the **SVM** machine learning algorithm.

### Development and Simulation Platform Tools

The algorithm is implemented in a virtual environment created using **VMware Workstation**, with **Ubuntu 20.04** installed in a virtual machine to establish the operating environment for the simulation. The following tools and technologies are used for the implementation of the methodology:

* + - * **OpenFlow** is a communications protocol for SDN that facilitates access to the forwarding plane of network switches or routers. This enables fine-grained control over network traffic and is integral to the Software-Defined Networking architecture.
      * **Mininet** is a network emulator that allows the creation of a network with virtual hosts, switches, controllers, and links. It supports OpenFlow, which is critical for SDN-based network configurations. Mininet is especially useful because it provides an inexpensive platform for developing, testing, and simulating custom network topologies in a virtual environment.
      * **Ryu Controller** is an open-source, Python-based SDN controller that enhances the agility of network management by making it easier to control how traffic is handled. It provides a programmable interface that allows for dynamic network management.
      * **Iperf** is a network performance tool used to measure network throughput and

datagram loss. In this project, Iperf is used to measure the Transport Control Protocol (TCP) and User Datagram Protocol (UDP) throughput, as well as to monitor data streams. It facilitates the evaluation of network performance by simulating client- server functionality between source and destination nodes.

## Hardware & Software specifications

### Hardware Specifications

* Processor: Intel Core i5 (or equivalent) or higher
* RAM: 8 GB or more
* Storage: 500 GB HDD/SSD
* Graphics Card: Optional, NVIDIA GeForce GTX 1050 (or equivalent) for machine learning tasks
* Network: Stable internet connection for SDN simulation

### Software Specifications

* Operating System:
  + Ubuntu 20.04 (used for creating the virtual machine environment and running the simulations).
  + VMware Workstation (used for setting up virtual environments to simulate network traffic and SDN management).
* Programming Languages:
  + Python 3.x (for developing the machine learning algorithms, SDN controller, and managing simulations).
* Libraries and Frameworks:
  + Scikit-learn: For implementing machine learning algorithms such as SVM and Decision Trees.
  + Ryu Controller: A Python-based SDN controller used for network management and simulation.
  + OpenFlow: For controlling the forwarding plane of network switches.
  + Mininet: A network emulator to simulate a network topology supporting OpenFlow.
  + Iperf: For measuring network performance, throughput, and datagram loss during traffic simulations.
* Virtualization/Containerization:
  + VMware Workstation (for creating virtual machines to run Ubuntu and simulate SDN environments).
* Development Environment:
  + Jupyter Notebook/VS Code: For writing and testing Python code.
  + Anaconda: For managing Python environments and dependencies.
  + Git: For version control, managing code, and collaborating if necessary.

## Dataset Description

The dataset used in this work comprises both normal and attack traffic samples, collected to train and evaluate the machine learning model. The dataset is specifically designed for the purpose of DDoS attack detection in a Software Defined Network (SDN) environment.

* **Normal Traffic**: The dataset includes over 600 samples of normal network traffic. These samples represent typical, everyday traffic flow in an SDN network under normal operating conditions. Each sample is characterized by various traffic features such as IP source speed, flow counts, flow entry speed, and the ratio of pair-flow entries. These normal traffic samples are used as the baseline for detecting deviations that indicate an attack.
* **Attack Traffic**: The dataset contains over 300 samples of attack traffic, specifically DDoS attacks. The attack traffic is generated to simulate real-world DDoS scenarios where the network experiences abnormal traffic spikes, including unusually high numbers of flow entries, IP sources, and non-interactive flow entries. These attack samples serve as the positive class for training the machine learning model to detect and classify abnormal traffic patterns.

### Data Collection

Traffic data is collected at regular intervals of 2 seconds over a total period of 300 seconds. The following features are monitored and captured for each sample:

* + 1. Speed of IP Sources (SSIP): Represents the total number of unique source IPs during each 2-second window.
    2. Flow Count: Tracks the number of flow entries in the network, with attack traffic showing a significantly higher flow count compared to normal traffic.
    3. Speed of Flow Entries (SFE): Measures the number of flow entries made to the switch within a specific time interval, with DDoS attacks showing a marked increase in the flow entry rate.
    4. Ratio of Pair-Flow Entries (RPF): Calculates the ratio of interactive flow entries, which typically decreases during a DDoS attack as the number of collaborative flows drops.

### Traffic Simulation

The dataset is generated through a combination of normal network behavior and simulated DDoS attacks. These attacks are designed to mimic common DDoS strategies, such as flooding the network with a large volume of traffic from multiple IP addresses. The network traffic is fed into the SDN controller for processing by the machine learning model (SVM in this case).

During the testing phase, the system continuously predicts whether the incoming traffic is normal or an attack every 2 seconds. The performance of the DDoS detection system is then measured based on the accuracy and detection rate of the model.

Testing Setup

* Test Duration: 300 seconds (5 minutes)
* Traffic Collection Interval: 2 seconds
* Traffic Sources: Multiple virtual hosts generate traffic using Mininet, which emulates the SDN network.
* Traffic Prediction: The SVM algorithm is tested by predicting the traffic class (normal or attack) at 2-second intervals.

This dataset serves as a critical resource for evaluating the effectiveness of the DDoS detection and mitigation method using SDN and machine learning algorithms. It allows for a real-time analysis of the system's ability to accurately differentiate between normal and attack traffic under various network conditions.

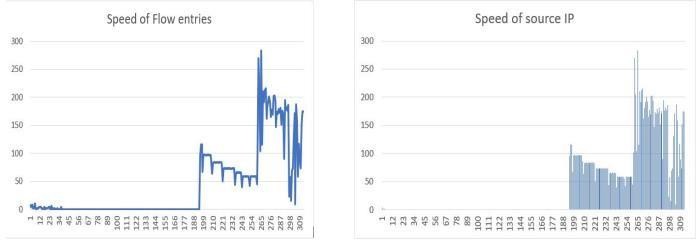
# Chapter 4 Results and Discussion

## Results-

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.

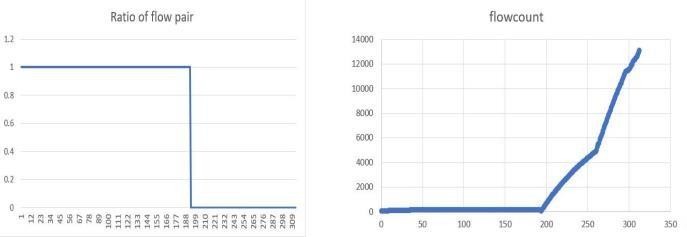
### Case Study 1-

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.



* + 1. 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.



(c) RFIP (d) Flowcount

The graphs (c) shows the ratio of flow pairs reduced when the network has attack traffic incoming and graph (d) shows the flowcount of the normal traffic and attack traffic.

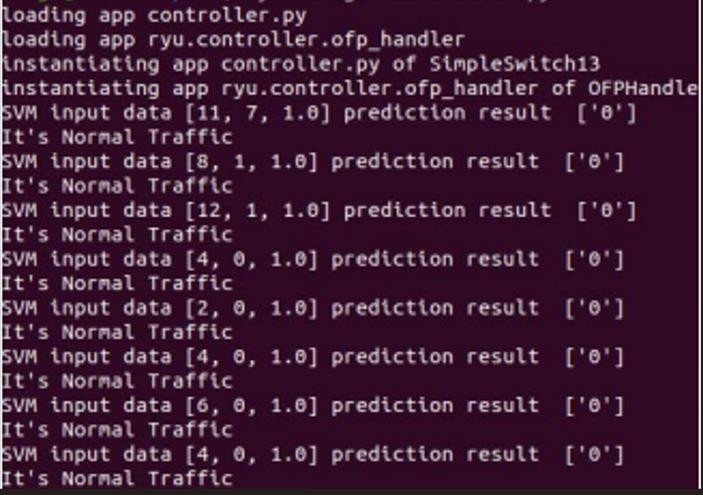


Figure 4. Normal Traffic Prediction

SVM machine learning algorithm predicting the traffic as normal traffic.

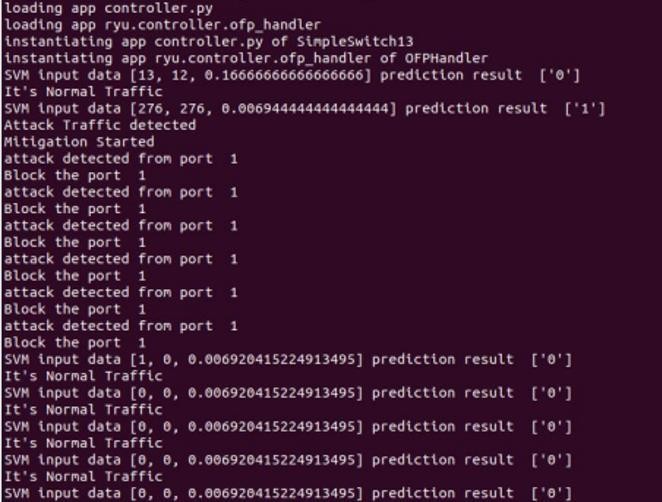


Figure 5. Attack Traffic Prediction

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.

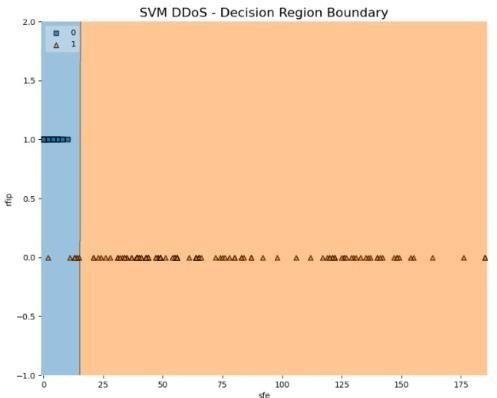


Figure 6: SVM Decision Boundary

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.

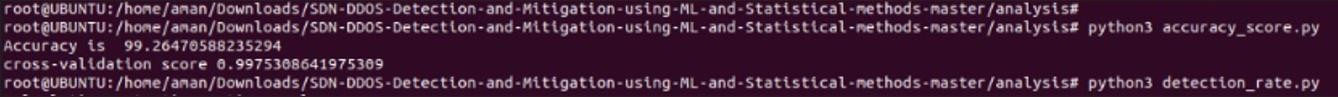


Figure 7: Accuracy Score



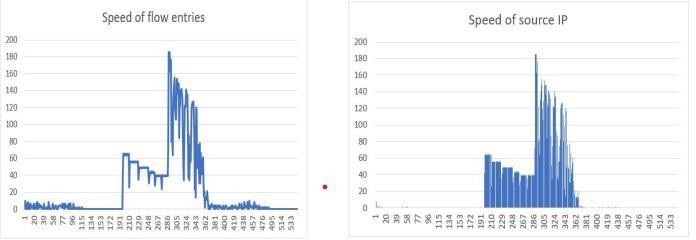
Figure 8: Detecion Rate

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%.

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

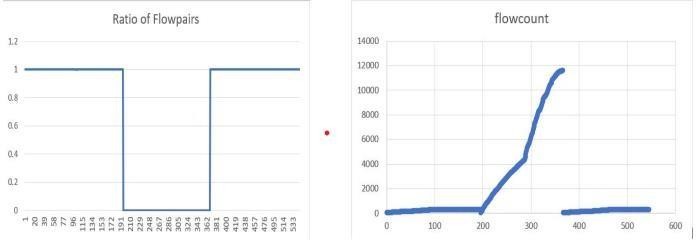
### Case study 2-

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.



* + 1. 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.



(c) RFIP (d) Flowcount

The graphs (c) shows the ratio of flow pairs reduced when the network has attack traffic incoming and graph (d) shows the flowcount of the normal traffic and attack traffic.

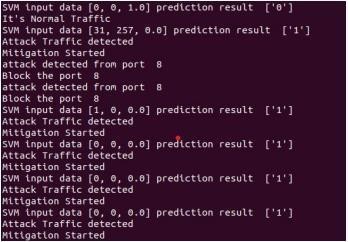


Figure 9: Attack Traffic Prediction

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.

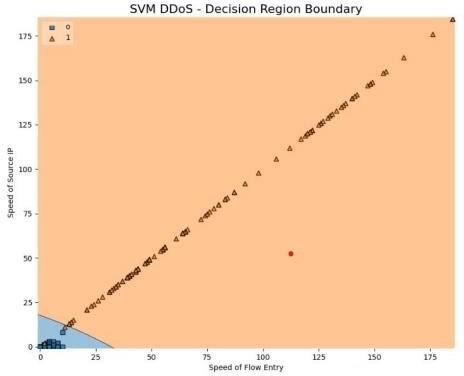


Figure 10: SVM Decision Boundary

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.

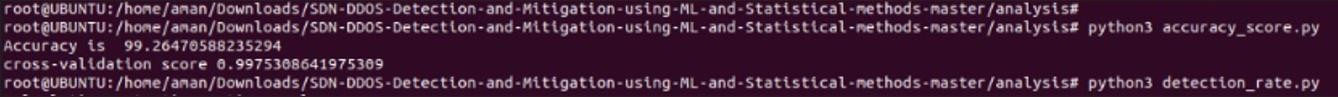


Figure 11: Accuracy Score



Figure 12: Detecion Rate

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%.

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.

## Discussion-

The implemented method was tested in two different attack scenarios. In Experiment 1, the attack traffic was sent from port 1 with only 10 hosts in the network. The results showed that the SVM machine learning algorithm achieved an accuracy of 98.71%, with a cross- validation score of 99.57% on the training data. The attack traffic detection rate was nearly 100%, and there were no false alarms, meaning no legitimate traffic was mistakenly flagged as malicious.

In Experiment 2, the attack traffic was sent from port 8, with 25 hosts in the network. The results indicated that the SVM algorithm achieved an accuracy of 99.26%, with a cross- validation score of 99.75% on the training data. Once again, the detection rate was 100%, and there were no false alarms.

These results demonstrate that the proposed method is highly effective at detecting malicious traffic without triggering any false positives, ensuring that legitimate traffic is not blocked from the network.

However, a potential limitation arises if an attack is launched from an IP address that is considered trusted or normal based on the training data. In such a case, the method may fail to detect the malicious activity, allowing the attacker to bypass the security. While the likelihood of this happening is low, it highlights the need for a more comprehensive network traffic analyzer to address these types of threats and further strengthen network security.

# Chapter 5 Conclusion and Future Scope

Software-defined networking (SDN) offers the ability to design and manage network operations through programming, a feature not available in traditional networks. The primary objective of this work was to leverage SDN for detecting and mitigating DDoS attacks in a cloud environment. The proposed method combines statistical features such as IP source, flow entry speed, flow count, and flow-pair ratio with an SVM (Support Vector Machine) machine learning algorithm to identify and predict DDoS attacks.

Experimental results demonstrate that this method achieves an accuracy of 99.26%, with a 100% detection rate for malicious traffic and zero false predictions.

However, no security system is completely foolproof. A limitation of the implemented method is that an attack originating from trusted IP sources could bypass detection by the SVM algorithm, since it may not recognize traffic from legitimate sources as malicious.

In the future, the approach could be enhanced by incorporating multiple switches and controllers within the network, along with a more comprehensive network packet analyzer. Currently, the method uses four statistical features, but additional features could be extracted and included in the machine learning algorithm for even more accurate predictions of malicious traffic.

# References

1. **Journal Articles**
   * **[1]** 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).
   * **[2]** 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).
   * **[3]** 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).
   * **[4]** 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).
   * **[5]** 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).

* + **[6]** 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).
  + **[7]** 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).
  + **[8]** 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).
  + **[9]** 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).
  + **[10**] 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).
  + **[11]** 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).

1. **Conference Papers**
   * **[12]** 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.

* + **[13]** 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\*.
  + **[14]** 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.
  + **[15]** 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, pp. 199–203. CORE Ranking: C.

1. **Surveys and Tutorials**
   * **[16]** Xia, W., Wen, Y., Foh, C. H., Niyato, D. and Xie, H. (2015). A survey on softwaredefined networking, *IEEE Communications Surveys Tutorials*, 17(1): 27–51. JCR Impact Factor: 22.973 (2019).
2. **Other Sources**
   * **[17]** 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).