

PREVENTION OF DDoS ATTACK IN PRIVATE NETWORK

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

SUBMITTED BY

Name	Univ. Roll No.
Mousam Maji	10800120066
Adarsh Bharti	10800120112
Tabbassum Parween	10800120089
Riya Gorai	10800120117

UNDER THE GUIDANCE OF

Samanta Hazra

Assistant Professor



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
ASANSOL ENGINEERING COLLAGE
AFFILIATED TO
MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY

June, 2024

Contents

Certificate of Recommendation	iii
Certificate of Approval.....	iv
Acknowledgement.....	v
Abstract.....	vi
List of Figures.....	vii
List of Tables.....	viii
1. Preface.....	
1.1 Introduction.....	1
1.2 Motivation of the project.....	2
1.3 Basic description of the project.....	3
2. Literature Review.....	
2.1 General.....	5
2.2 Review of related works	7
3. Related Theories and Algorithms.....	
3.1 Fundamental theories underlying the work.....	10
3.2 Fundamental algorithms.....	12
4. Proposed model/algorithm.....	
4.1 Proposed model.....	14
4.2 Proposed algorithms.....	16
5. Simulation Results.....	
4.1 Experimental set up	18
4.2 Experimental results.....	24
6. Discussion and Conclusion	
6.1 Discussion.....	30
6.2 Future work.....	30
6.3 Conclusion.....	30
References.....	



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**

ASANSOL ENGINEERING COLLEGE

Vivekananda Sarani, Kanyapur, Asansol, West Bengal – 713305

Certificate of Recommendation

I hereby recommend that the thesis entitled, “**Prevention of DDoS attack in Private Network**” carried out under my supervision by the group of students listed below may be accepted in partial fulfilment of the requirement for the degree of “Bachelor of Technology in Computer Science and Engineering” of Asansol Engineering College under MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY.

Name	Univ. Roll No.
Mousam Maji	10800120066
Adarsh Bharti	10800120112
Tabbassum Parween	10800120089
Riya Gorai	10800120117

.....
(Samanta Hazra)
Thesis Supervisor
Dept. of Computer Science and
Engineering,
Asansol Engineering College,
Asansol-713305

Countersigned:

.....
(Dr. Monish Chatterjee)
Head of the Department
Dept. of Computer Science and
Engineering,
Asansol Engineering College,
Asansol-713305



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING**
ASANSOL ENGINEERING COLLEGE
Vivekananda Sarani, Kanyapur, Asansol, West Bengal – 713305

Certificate of Approval

The thesis is hereby approved as creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance in the partial fulfilment of the degree for which it has been submitted. It is understood that by this approval the undersigned does not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it is submitted.

.....
(Samanta Hazra)
Thesis Supervisor
Dept. of Computer Science and
Engineering,
Asansol Engineering College,
Asansol-713305

Acknowledgement

It is our great privilege to express our profound and sincere gratitude to our Thesis Supervisor, Samanta Hazra for providing us very cooperative and valuable guidance at every stage of the project work being carried out under his supervision. His valuable advice and instructions in carrying out the present study has been a very rewarding and pleasurable experience that has greatly benefited us throughout the period of work.

We would like to convey our sincere gratitude towards Dr. Monish Chatterjee, Head of the Department, Asansol Engineering College for providing us the requisite support for timely completion of our work. We would also like to pay our heartiest thanks and gratitude to all the teachers of the Department of Computer Science and Engineering, Asansol Engineering College for various suggestions being provided in attaining success in our work.

We would like to express our earnest thanks to Mr. Suman Mallick, of CSE Project Lab for his technical assistance provided during our project work.

Finally, I would like to express my deep sense of gratitude to my parents for their constant motivation and support throughout my work.

.....
Mousam Maji

.....
Adarsh Bharti

.....
Tabbassum Parween

.....
Riya Gorai

Abstract

This project explores the effectiveness of Software Defined Networking (SDN) in enhancing the detection and mitigation of Distributed Denial of Service (DDOS) attacks within a cloud environment. The study leverages both statistical analysis and machine learning techniques to identify and address malicious traffic.

Methodology: The project employs a combination of statistical analysis and machine learning to classify network traffic. Key features such as the speed of IP sources, flow count, speed of flow entries, and the ratio of pair-flow entries are analyzed. Machine learning models, specifically Support Vector Machines (SVM) and Decision Trees, are trained using datasets containing both normal and attack traffic to predict and categorize incoming traffic as either benign or malicious.

Development and Simulation Tools:

- **Platform:** Virtual machine setup with Ubuntu 20.04 OS
- **SDN Protocol:** OpenFlow
- **Controller:** Ryu (Python-based)
- **Simulation:** Mininet, Hping3, Iperf

Key Findings: The implementation demonstrated that SDN could effectively manage and mitigate DDOS attacks by utilizing advanced traffic analysis and machine learning prediction. However, challenges such as the accuracy of the dataset and the potential for trusted IPs to be used in attacks highlight the need for further refinement.

Future Work: Future research will focus on deploying this system in real-time network traffic and expanding the network topology to include multiple controllers and switches. Additionally, more parameters and features will be defined to enhance detection accuracy.

This project contributes to the field by providing a framework for integrating SDN with machine learning to create a robust defense mechanism against DDOS attacks in cloud environments.

List of Figures

Fig. 1	SDN Framework.	20
Fig. 2	Mininet Network Design	21
Fig. 3	Flowchart of the Presented Method	22
Fig. 4(a)	Case Study 1 - SFE	24
Fig. 4(b)	Case Study 1 - SSP	24
Fig. 5(a)	Case Study 1 - RFIP	24
Fig. 5(b)	Case Study 1 - Flowcount	24
Fig. 6	Case Study 1 - Normal Traffic Prediction	25
Fig. 7	Case Study 1 - Attack Traffic Prediction	25
Fig. 8	Case Study 1 - SVM Decision Boundary	26
Fig. 9	Case Study 1 - Accuracy Score	26
Fig. 10	Case Study 1 - Detection Rate	26
Fig. 11(a)	Case Study 2 - SFE	27
Fig. 11(b)	Case Study 2 - SSP	27
Fig. 12(a)	Case Study 2 - RFIP	27
Fig. 12(b)	Case Study 2 - Flowcount	27
Fig. 13	Case Study 2 - Attack Traffic Prediction	28
Fig. 14	Case Study 2 - SVM Decision Boundary	28
Fig. 15	Case Study 2 - Accuracy Score	28
Fig. 16	Case Study 2 - Detection Rate	29

List of Tables

Table 1	Comparison Table
---------	------------------

9

1. Preface

1.1 Introduction

In the modern digital era, Distributed Denial of Service (DDOS) attacks have emerged as a significant threat to the stability and security of networked systems, particularly within cloud environments. These attacks aim to overwhelm a network or service with an excessive amount of traffic, rendering it unavailable to legitimate users. The increasing sophistication and frequency of DDOS attacks necessitate the development of advanced detection and mitigation strategies to safeguard network integrity.

Software Defined Networking (SDN) offers a promising solution to this challenge. By decoupling the control plane from the data plane, SDN provides a flexible and programmable network architecture that can dynamically adapt to various security threats. This project investigates the potential of SDN to enhance the detection and mitigation of DDOS attacks using a combination of statistical analysis and machine learning techniques.

The core objective of this research is to determine whether SDN can improve the accuracy and efficiency of DDOS attack detection and mitigation compared to traditional network architectures. To achieve this, the project involves analysing key traffic features, such as the speed of IP sources, flow count, speed of flow entries, and the ratio of pair-flow entries. These features are then used to train machine learning models, specifically Support Vector Machines (SVM) and Decision Trees, to classify network traffic as normal or malicious.

The project setup includes a virtual machine running Ubuntu 20.04 OS, utilizing the OpenFlow protocol for SDN and the Ryu controller. Simulation tools like Mininet, Hping3, and Iperf are employed to generate and evaluate network traffic under various conditions.

Through this research, we aim to demonstrate the viability of integrating SDN with machine learning to create a robust defence mechanism against DDOS attacks. The findings from this study will contribute to the development of more secure and resilient network infrastructures, capable of effectively countering the evolving threat landscape posed by DDOS attacks.

1.2 Motivation of the project

Increasing Importance of SDNs:

Software Defined Networks (SDNs) have become crucial in modern networking due to their centralized control and management capabilities. They enable flexible, efficient, and scalable network management, making them integral to contemporary network infrastructures.

Vulnerability to DDoS Attacks:

Despite their advantages, SDNs are particularly susceptible to Distributed Denial of Service (DDoS) attacks. The centralized nature of SDNs can be exploited by attackers to overwhelm network resources, leading to significant service disruptions.

Inadequacy of Traditional Security Mechanisms:

Description: Traditional network security solutions often struggle to effectively detect and mitigate DDoS attacks in SDNs. The dynamic and complex traffic patterns in SDNs require more sophisticated detection and mitigation strategies than those provided by conventional methods.

Growing Frequency and Sophistication of DDoS Attacks:

DDoS attacks are becoming more frequent and sophisticated, posing a severe threat to network stability and security. This escalation necessitates the development of advanced techniques to keep pace with evolving attack methods.

Need for Advanced Detection and Mitigation Techniques:

There is a critical need for innovative approaches to enhance the detection and mitigation of DDoS attacks. Leveraging statistical analysis and machine learning offers the potential to significantly improve the accuracy and efficiency of identifying malicious traffic.

Ensuring Reliable and Continuous Service Availability:

The primary goal of this project is to develop solutions that ensure reliable and continuous service availability in SDNs. By effectively detecting and mitigating DDoS attacks, the project aims to maintain network performance and prevent service disruptions.

Contributing to Network Security Research:

This project aspires to contribute to the broader field of network security research. By developing and validating new methods for DDoS detection and mitigation, it aims to provide a foundation for future advancements in SDN and cloud computing security.

Enhancing Resilience of Network Infrastructures:

The project aims to strengthen the resilience of network infrastructures against DDoS attacks. By integrating advanced detection and mitigation strategies, it seeks to build more robust and secure networks capable of withstanding sophisticated cyber threats.

These motivations underscore the critical need for the project and highlight its potential impact on improving network security in SDN environments.

1.3 Basic description of the project

Objective:

The primary objective of this project is to enhance the detection and mitigation of Distributed Denial of Service (DDOS) attacks within Software Defined Networking (SDN) environments by leveraging statistical analysis and machine learning techniques.

Key Components:

Software Defined Networking (SDN):

SDN is a networking paradigm that separates the control plane from the data plane, enabling centralized and programmable network management. This flexibility allows for dynamic adaptation to various network conditions and threats.

DDOS Attacks:

DDOS attacks aim to disrupt services by overwhelming network resources with excessive traffic. These attacks can cause significant downtime and financial loss, making their detection and mitigation crucial for maintaining network stability.

Statistical Analysis:

The project involves the analysis of network traffic using statistical methods to identify abnormal patterns indicative of DDOS attacks. Key features such as the speed of IP sources, flow count, and the ratio of pair-flow entries are examined.

Machine Learning Techniques:

Machine learning models, specifically Support Vector Machines (SVM) and Decision Trees, are trained on datasets containing both normal and attack traffic. These models are used to classify incoming traffic and predict potential DDOS attacks.

Implementation Tools:

Platform: Virtual machine with Ubuntu 20.04 OS.

SDN Protocol: OpenFlow

Controller: Ryu (a Python-based SDN controller)

Simulation Tools: Mininet for network simulation, Hping3 and Iperf for generating and testing network traffic.

Methodology:

Data Collection: Network traffic data is collected and labelled as normal or attack traffic.

Feature Extraction: Statistical features relevant to DDOS detection are extracted from the collected data.

Model Training: Machine learning models are trained using the extracted features.

Traffic Classification: The trained models are used to classify and predict traffic in real-time, identifying and mitigating DDOS attacks.

Outcomes:

Improved Detection Accuracy: The integration of statistical analysis and machine learning enhances the accuracy of DDOS attack detection.

Effective Mitigation: The system can dynamically adapt to and mitigate detected attacks, ensuring minimal disruption to network services.

Scalable and Flexible Solution: The use of SDN allows for a scalable and flexible approach to network security, adaptable to various network environments and attack scenarios.

Future Work:

Real-Time Deployment: Further research will focus on deploying the system in real-time network environments.

Extended Network Topology: Expanding the network topology to include multiple controllers and switches for enhanced security.

Additional Features: Defining more parameters and features to improve detection accuracy.

This project aims to create a robust defence mechanism against DDOS attacks, contributing to the development of more secure and resilient SDN infrastructures.

2. Literature Review

2.1 General

The literature on DDOS attack detection and mitigation in Software Defined Networking (SDN) environments spans various methodologies, including statistical analysis, machine learning, and hybrid approaches. This review provides an overview of key research contributions and findings in this domain.

1. Software Defined Networking (SDN):

- **Overview:** SDN is a paradigm that decouples the control plane from the data plane, enabling centralized and programmable network management. It allows for dynamic configuration, efficient resource utilization, and improved network management.
- **Significance:** SDN's centralized control makes it easier to monitor and manage traffic, providing an advantageous framework for implementing security measures against DDOS attacks.

2. DDOS Attacks:

- **Nature of DDOS Attacks:** DDOS attacks aim to make network services unavailable by overwhelming them with a flood of malicious traffic. These attacks can target various network layers and exploit different vulnerabilities.
- **Challenges in Detection:** Detecting DDOS attacks in SDNs involves distinguishing between legitimate high-volume traffic and malicious traffic, which can be challenging due to the dynamic nature of network traffic.

3. Statistical Analysis for DDOS Detection:

- **Methods:** Statistical methods involve analysing traffic patterns and identifying anomalies that indicate potential DDOS attacks. Common metrics include packet arrival rates, flow counts, and traffic distribution.
- **Advantages:** Statistical analysis provides a straightforward approach to detecting deviations from normal traffic behaviour, which can be indicative of attacks.
- **Limitations:** Purely statistical methods may struggle to adapt to evolving attack patterns and might generate false positives or negatives.

4. Machine Learning Techniques:

- **Overview:** Machine learning models can learn from historical traffic data to classify and predict malicious traffic. Techniques such as Support Vector Machines (SVM), Decision Trees, and Neural Networks are commonly used.
- **Training and Validation:** These models are trained on labelled datasets containing both normal and attack traffic. The models learn to recognize patterns associated with DDOS attacks.
- **Effectiveness:** Machine learning models can achieve high accuracy in detecting attacks and adapt to new attack patterns over time. However, their performance depends on the quality and representativeness of the training data.

5. Hybrid Approaches:

- **Combination of Methods:** Hybrid approaches combine statistical analysis and machine learning to leverage the strengths of both methods. This can enhance detection accuracy and robustness.
- **Examples:** Some studies use statistical features as inputs for machine learning models, improving the models' ability to detect complex attack patterns.
- **Benefits:** Hybrid methods can reduce false positives and negatives and provide a more comprehensive detection mechanism.

6. Implementation in SDN:

- **SDN Controllers:** SDN controllers, such as Ryu and OpenFlow, play a critical role in managing traffic and implementing security measures. They can be programmed to use detection algorithms and respond to detected attacks.
- **OpenFlow Protocol:** OpenFlow is a widely used SDN protocol that allows the controller to interact with the data plane. It enables fine-grained control over traffic flows, facilitating effective attack mitigation.
- **Simulation Tools:** Tools like Mininet are used to simulate SDN environments and evaluate the performance of detection and mitigation strategies. They provide a controlled environment for testing and validation.

7. Notable Research Contributions:

- **Early Detection Systems:** Research has proposed various early detection systems using real-time traffic analysis and machine learning. These systems aim to detect attacks before they can cause significant damage.
- **Adaptive Security Mechanisms:** Adaptive mechanisms adjust their detection parameters based on current network conditions, improving their responsiveness to changing attack patterns.
- **Resource Management:** Effective resource management strategies are critical for mitigating DDOS attacks. Some studies focus on optimizing resource allocation in SDNs to ensure service availability during attacks.

8. Future Directions:

- **Real-Time Implementation:** There is ongoing research to implement these detection and mitigation strategies in real-time, operational networks.
- **Scalability and Robustness:** Future work aims to enhance the scalability and robustness of detection systems, ensuring they can handle large-scale networks and sophisticated attacks.
- **Integration with Other Technologies:** Integrating SDN with other technologies, such as Network Function Virtualization (NFV) and blockchain, is a promising area for improving network security.

2.2 Review of related works

Statistical Analysis Approaches:

1. He and Xu (2012):

- a. **Contribution:** Proposed a statistical method for anomaly detection in SDNs based on traffic entropy.
- b. **Findings:** Their approach effectively identified abnormal traffic patterns, but it required a well-defined baseline of normal traffic behavior for accurate detection.

2. Wang et al. (2015):

- a. **Contribution:** Developed a traffic anomaly detection system using statistical features such as packet arrival rates and flow counts.
- b. **Findings:** The system demonstrated good performance in identifying DDOS attacks but faced challenges in distinguishing between legitimate high-volume traffic and attack traffic.

Machine Learning Techniques:

1. Kang et al. (2016):

- a. **Contribution:** Applied Support Vector Machines (SVM) to classify network traffic in SDNs.
- b. **Findings:** The SVM model achieved high accuracy in detecting DDOS attacks, particularly when combined with feature selection techniques to enhance model performance.

2. Javaid et al. (2016):

- a. **Contribution:** Proposed a machine learning-based intrusion detection system using Decision Trees.
- b. **Findings:** The system effectively detected DDOS attacks with low false positive rates, demonstrating the potential of decision trees for real-time traffic classification.

3. Zhou et al. (2018):

- a. **Contribution:** Implemented a deep learning approach using Neural Networks for DDOS detection in SDNs.
- b. **Findings:** Deep learning models outperformed traditional machine learning models in terms of detection accuracy, though they required more computational resources and training data.

Hybrid Approaches:

1. Doriguzzi-Corin et al. (2019):

- a. **Contribution:** Developed a hybrid detection system combining statistical analysis and machine learning.
- b. **Findings:** The hybrid approach significantly improved detection accuracy and reduced false positives by leveraging the strengths of both methods.

2. Ahmed and Kim (2020):

- a. **Contribution:** Proposed an adaptive hybrid system that adjusts its detection parameters based on real-time network conditions.
- b. **Findings:** The adaptive system demonstrated resilience to evolving attack patterns and provided better overall protection against DDOS attacks.

SDN-Specific Solutions:

1. Jafarian et al. (2015):

- a. **Contribution:** Investigated the use of SDN controllers for DDOS mitigation.

- b. Findings:** Their study highlighted the advantages of using SDN's centralized control for quick and efficient response to DDOS attacks, emphasizing the need for robust controller security mechanisms.

2. Bhatia et al. (2016):

- a. Contribution:** Examined the role of OpenFlow in facilitating DDOS mitigation in SDNs.
- b. Findings:** OpenFlow's fine-grained control over traffic flows was shown to be effective in isolating and mitigating malicious traffic, though it required careful configuration to avoid bottlenecks.

Simulation and Evaluation Tools:

Lantz et al. (2010):

- **Contribution:** Introduced Mininet, a network emulator for testing SDN applications.
- **Findings:** Mininet provided a flexible and scalable platform for simulating SDN environments and evaluating the performance of DDOS detection and mitigation strategies.

Handigol et al. (2012):

- **Contribution:** Demonstrated the use of Mininet in conjunction with OpenFlow and SDN controllers for real-time network testing.
- **Findings:** The combination of these tools allowed for comprehensive evaluation of SDN-based security solutions in a controlled environment.

Emerging Trends and Future Directions:

Kouzehkanan and St-Hilaire (2021):

- **Contribution:** Explored the integration of blockchain technology with SDN for enhanced security.
- **Findings:** Blockchain provided a decentralized and tamper-proof mechanism for managing network security policies, offering promising results for mitigating DDOS attacks.

Kim et al. (2022):

- **Contribution:** Investigated the use of Network Function Virtualization (NFV) in conjunction with SDN for scalable DDOS mitigation.
- **Findings:** NFV allowed for dynamic deployment of security functions, improving the scalability and flexibility of the mitigation system.

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 Mininet	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.

Table 1

3. Literature Review

3.1 Related Theories and Algorithms

The research project is grounded in several fundamental theories that provide the basis for understanding and developing effective detection and mitigation strategies. Here are the key theories:

Traffic Pattern Analysis:

- **Description:** This theory focuses on the analysis of network traffic patterns to identify normal and abnormal behaviours. Traffic patterns are studied to detect deviations that could indicate potential DDOS attacks.
- **Relevance:** In SDN, traffic pattern analysis is crucial because it helps in distinguishing between legitimate high-volume traffic and malicious traffic. By analysing patterns, it's possible to identify anomalies and take preventive measures.

Anomaly Detection:

- **Description:** Anomaly detection involves identifying data points, events, or observations that deviate significantly from the majority of the data, thus signalling a potential security threat.
- **Relevance:** Anomaly detection algorithms are used in SDNs to monitor traffic flows and detect unusual activities that may suggest a DDOS attack. This theory underpins many of the machine learning and statistical techniques used for DDOS detection.

Statistical Analysis:

- **Description:** Statistical analysis involves the collection, analysis, interpretation, and presentation of masses of numerical data. In the context of network security, it includes the use of metrics like mean, variance, and entropy to analyse traffic data.
- **Relevance:** Statistical methods provide a quantitative approach to identifying traffic anomalies. They help in establishing baselines for normal traffic and detecting significant deviations that could indicate an attack.

Machine Learning Theory:

- **Description:** Machine learning involves algorithms that can learn from and make predictions on data. It includes supervised learning, where models are trained on labelled data, and unsupervised learning, where models identify patterns in data without labels.
- **Relevance:** Machine learning algorithms are used to develop models that can classify network traffic as normal or malicious. These models can learn from historical attack data and adapt to new attack patterns, improving the accuracy and efficiency of DDOS detection.

Control Theory:

- **Description:** Control theory deals with the behaviour of dynamical systems with inputs and how their behaviour is modified by feedback. It is used to design systems that maintain desired outputs despite external disturbances.

- **Relevance:** In SDN, control theory is applied to manage and control network traffic dynamically. It helps in implementing effective response strategies to mitigate DDOS attacks by adjusting network configurations in real-time.

Game Theory:

- **Description:** Game theory is the study of mathematical models of strategic interaction among rational decision-makers. It is used to analyse competitive situations where the outcomes depend on the actions of multiple agents.
- **Relevance:** Game theory can be applied to model the interactions between attackers and defenders in a network. It helps in developing strategies that anticipate and counteract the moves of potential attackers.

Queuing Theory:

- **Description:** Queuing theory is the mathematical study of waiting lines, or queues. It is used to model and analyse the behaviour of queues to predict traffic congestion and system performance.
- **Relevance:** In the context of SDN, queuing theory helps in understanding traffic congestion and optimizing resource allocation to ensure smooth network performance even during DDOS attacks.

These theories provide a comprehensive framework for understanding the complexities of DDOS detection and mitigation in SDN environments. They guide the development of algorithms and strategies to enhance network security and resilience against attacks.

3.2 Fundamental algorithms

The research project on DDOS detection and mitigation in SDN environments leverages several key algorithms. These algorithms form the backbone of the system's ability to detect and mitigate malicious traffic effectively. Here are the fundamental algorithms:

Support Vector Machines (SVM):

- **Description:** SVM is a supervised learning algorithm used for classification and regression. It works by finding the hyperplane that best separates different classes in the feature space.
- **Application:** SVM is used to classify network traffic as normal or malicious. By training on historical attack data, the SVM model can effectively distinguish between benign and attack traffic.

Decision Trees:

- **Description:** Decision trees are a non-parametric supervised learning method used for classification and regression. They create a model that predicts the value of a target variable by learning simple decision rules inferred from data features.
- **Application:** Decision trees are employed to create a model that can predict whether a given traffic flow is part of a DDOS attack. They are particularly useful for their interpretability and ease of implementation.

K-Nearest Neighbours (KNN):

- **Description:** KNN is a simple, instance-based learning algorithm that classifies a data point based on the majority class among its k-nearest neighbours.
- **Application:** KNN is used to classify network traffic by comparing new traffic patterns to historical patterns. It is effective in scenarios where the distribution of attack and normal traffic is well-defined.

Naive Bayes:

- **Description:** Naive Bayes is a probabilistic classifier based on Bayes' theorem, assuming independence between features.
- **Application:** This algorithm predicts the probability of a traffic flow being part of a DDOS attack by considering the likelihood of each feature independently. It is efficient and requires relatively small amounts of training data.

Clustering Algorithms:

K-Means:

- **Description:** K-Means is an unsupervised learning algorithm that partitions data into k clusters based on feature similarity.
- **Application:** K-Means is used to identify clusters of normal and attack traffic, aiding in the detection of anomalies.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise):

- **Description:** DBSCAN is a clustering algorithm that groups together points that are closely packed and marks points in low-density regions as outliers.
- **Application:** DBSCAN is useful for identifying outliers in network traffic that may represent DDOS attacks.

Neural Networks:

- **Description:** Neural networks are a set of algorithms, modelled loosely after the human brain, designed to recognize patterns.
- **Application:** Deep learning models, such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs), can be trained to detect complex patterns associated with DDOS attacks. They are particularly effective in capturing non-linear relationships in traffic data.

Entropy-Based Algorithms:

- **Description:** These algorithms measure the randomness or unpredictability in the network traffic.
- **Application:** Entropy-based methods are used to quickly identify changes in traffic patterns that may suggest a DDOS attack. They help in detecting sudden spikes in traffic entropy, which can indicate an attack.

Flow-Based Detection Algorithms:

- **Description:** These algorithms analyse traffic flows (sequences of packets sharing common attributes) to detect anomalies.
- **Application:** Flow-based detection is particularly suitable for SDNs as it leverages the controller's global view of the network to monitor and analyse flows in real-time.

These algorithms provide a comprehensive toolkit for detecting and mitigating DDOS attacks in SDN environments. They enable the system to analyse traffic patterns, classify traffic, identify anomalies, and take appropriate action to mitigate attacks effectively.

4. Proposed model/algorithm

4.1 Proposed model

The proposed model for DDOS detection and mitigation in SDN environments is designed to effectively identify and respond to malicious traffic patterns. Here is a detailed description of the proposed model:

Traffic Monitoring and Data Collection:

- **Description:** The model begins with continuous monitoring of network traffic using the SDN controller. Data from various network flows is collected, including packet size, inter-arrival times, and other relevant metrics.
- **Components:** SDN controller, network switches, traffic flow data.

Feature Extraction:

- **Description:** Key features are extracted from the collected traffic data. These features include statistical metrics like mean, variance, and entropy, as well as more complex features derived from machine learning algorithms.
- **Components:** Feature extraction module, traffic flow data.

Anomaly Detection:

- **Description:** The anomaly detection module uses machine learning algorithms to analyse the extracted features and identify traffic patterns that deviate from normal behaviour. This involves the use of algorithms such as Support Vector Machines (SVM), Decision Trees, and K-Nearest Neighbours (KNN).
- **Components:** Anomaly detection module, machine learning algorithms, historical traffic data.

Classification of Traffic:

- **Description:** Traffic is classified into normal and malicious categories based on the results of the anomaly detection. This step involves further analysis and validation of the identified anomalies to reduce false positives.
- **Components:** Classification module, anomaly detection results.

Real-Time Mitigation:

- **Description:** Upon detecting malicious traffic, the model triggers mitigation actions in real-time. This may involve rerouting traffic, rate limiting, or blocking malicious IP addresses. The SDN controller dynamically adjusts network configurations to implement these actions.
- **Components:** Mitigation module, SDN controller, network switches.

Feedback Loop and Model Updating:

- **Description:** The system includes a feedback loop that continuously updates the machine learning models based on new traffic data and mitigation outcomes. This ensures that the model adapts to evolving attack patterns.
- **Components:** Feedback loop, machine learning model updates, continuous data collection.

Components and Workflow

Data Collection:

- **Tools:** Network monitoring tools integrated with SDN.
- **Process:** Continuous collection of traffic data.

Feature Extraction:

- **Tools:** Statistical analysis tools, custom feature extraction scripts.
- **Process:** Extraction of relevant features from traffic data.

Anomaly Detection:

- **Tools:** Machine learning libraries (e.g., scikit-learn, TensorFlow).
- **Process:** Application of ML algorithms to detect anomalies.

Classification:

- **Tools:** Classification algorithms, validation tools.
- **Process:** Categorization of traffic as normal or malicious.

Mitigation:

- **Tools:** SDN controller APIs, network configuration scripts.
- **Process:** Implementation of real-time mitigation actions.

Feedback Loop:

- **Tools:** Data analysis tools, model training frameworks.
- **Process:** Continuous improvement of the model based on new data.

The proposed model integrates advanced machine learning techniques with the dynamic capabilities of SDN to provide a robust solution for DDOS detection and mitigation. It leverages real-time data collection and analysis to detect and respond to threats effectively, ensuring network security and stability.

4.2 Proposed algorithms

The research project proposes several key algorithms to enhance DDOS detection and mitigation. These algorithms are designed to work in tandem to provide a comprehensive solution for identifying and responding to malicious traffic. Here are the details of the proposed algorithms:

Feature Extraction Algorithm:

- **Description:** This algorithm focuses on extracting relevant features from the network traffic data. These features include statistical measures (mean, variance, standard deviation) and more complex attributes derived from traffic patterns.
- **Process:**
 - a. Collect raw traffic data from the SDN controller.
 - b. Compute statistical features such as packet count, byte count, and flow duration.
 - c. Derive advanced features like entropy, flow inter-arrival time, and packet size distribution.
- **Purpose:** To transform raw traffic data into a set of features that can be used by machine learning models for anomaly detection.

Anomaly Detection Algorithm:

- **Description:** This algorithm uses machine learning techniques to detect anomalies in the network traffic. It applies supervised learning methods to classify traffic as normal or abnormal based on the extracted features.
- **Process:**
 - a. Train the model on historical traffic data, labelled as normal or attack traffic.
 - b. Use algorithms such as Support Vector Machines (SVM), Decision Trees, and K-Nearest Neighbours (KNN) for training.
 - c. Apply the trained model to incoming traffic data to identify anomalies.
- **Purpose:** To detect potential DDOS attacks by identifying deviations from normal traffic patterns.

Classification Algorithm:

- **Description:** Once anomalies are detected, this algorithm classifies the traffic into specific types of attacks or normal traffic. It uses the outputs from the anomaly detection algorithm to make detailed classifications.
- **Process:**
 - a. Input detected anomalies into the classification model.
 - b. Use a multi-class classification approach to distinguish between different types of DDOS attacks.
 - c. Validate and refine the classification results to ensure accuracy.
- **Purpose:** To provide detailed identification of the type of attack, enabling targeted mitigation strategies.

Mitigation Algorithm:

- **Description:** This algorithm dynamically responds to detected DDOS attacks by adjusting network configurations through the SDN controller. It aims to minimize the impact of the attack on the network.
- **Process:**
 - a. Receive classified attack information from the classification algorithm.

- b. Determine the appropriate mitigation action (e.g., rate limiting, traffic rerouting, IP blocking).
- c. Implement the mitigation action using SDN controller commands.
- d. Monitor the network to assess the effectiveness of the mitigation and adjust as necessary.
- **Purpose:** To protect the network from ongoing attacks and maintain service availability.

Feedback Loop Algorithm:

- **Description:** This algorithm continuously updates the machine learning models based on new traffic data and the outcomes of mitigation actions. It ensures that the system adapts to evolving attack patterns.
- **Process:**
 - a. Collect new traffic data and mitigation outcomes.
 - b. Retrain the machine learning models periodically with the updated data.
 - c. Integrate feedback into the feature extraction and anomaly detection processes.
- **Purpose:** To improve the accuracy and responsiveness of the detection and mitigation system over time.

Integration and Workflow

Data Collection and Feature Extraction:

Tools: Network monitoring tools, feature extraction scripts.

Flow: Continuous data collection → Feature extraction.

Anomaly Detection and Classification:

Tools: Machine learning libraries (e.g., scikit-learn, TensorFlow).

Flow: Feature input → Anomaly detection → Classification.

Mitigation and Feedback:

Tools: SDN controller APIs, mitigation scripts.

Flow: Classification results → Mitigation actions → Feedback loop.

The proposed algorithms collectively enhance the system's ability to detect and mitigate DDOS attacks by leveraging advanced machine learning techniques and real-time network control capabilities.

5. Simulation Results

5.1 Experimental set up

This section outlines the configuration and environment used to test and evaluate the proposed DDOS detection and mitigation model in an SDN environment. Here is a detailed description:

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 such 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:

$$SSIP = \frac{SumIPsrc}{T}$$

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.

Simulation Environment:

- **Tools and Platforms:** The experiments were conducted using a simulation environment based on Mininet, an emulator that creates a virtual network, running on a standard Linux platform. Mininet is chosen for its ability to simulate a realistic SDN environment, including switches, hosts, and controllers.
- **SDN Controller:** The OpenFlow-based SDN controller, such as OpenDaylight or Ryu, was deployed to manage the network. The controller plays a crucial role in the collection of traffic data and the implementation of mitigation strategies.
- **Network Topology:** A custom network topology was created within Mininet to represent a typical network structure vulnerable to DDOS attacks. The topology included multiple switches and hosts to simulate various traffic scenarios.

Traffic Generation:

- **Normal Traffic:** Normal network traffic was generated using tools like Iperf and Ping to simulate regular user behaviour. This included a mix of web browsing, file transfers, and streaming activities to create a baseline of normal traffic patterns.
- **Attack Traffic:** DDOS attack traffic was generated using tools like LOIC (Low Orbit Ion Cannon) and Hping3 to simulate different types of DDOS attacks. These tools allowed for the creation of high-volume traffic directed at specific targets within the network.

Feature Extraction and Data Collection:

- **Traffic Data Collection:** The SDN controller collected traffic flow data from the network switches. This data included various metrics such as packet count, byte count, flow duration, and inter-arrival times.
- **Feature Extraction:** Key features were extracted from the collected data using custom scripts. These features were essential for training and testing the machine learning models used in the anomaly detection and classification stages.

Machine Learning Model Training:

- **Data Set Preparation:** The collected traffic data was split into training and testing datasets. The training dataset included labelled instances of normal and attack traffic, which were used to train the machine learning models.
- **Algorithms Used:** Various machine learning algorithms, including Support Vector Machines (SVM), Decision Trees, and K-Nearest Neighbours (KNN), were trained on the extracted features. Cross-validation techniques were used to ensure the robustness and accuracy of the models.

Development and simulation Platform Tools:

In this work, 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.

Simulation Execution:

Real-Time Detection and Mitigation: The trained models were deployed within the simulation environment to perform real-time detection and mitigation of DDOS attacks. The SDN controller continuously monitored traffic flows, detected anomalies, and executed mitigation actions based on the classification results.

Performance Metrics: The effectiveness of the proposed model was evaluated using metrics such as detection accuracy, false positive rate, mitigation response time, and overall network performance.

Design Specification:

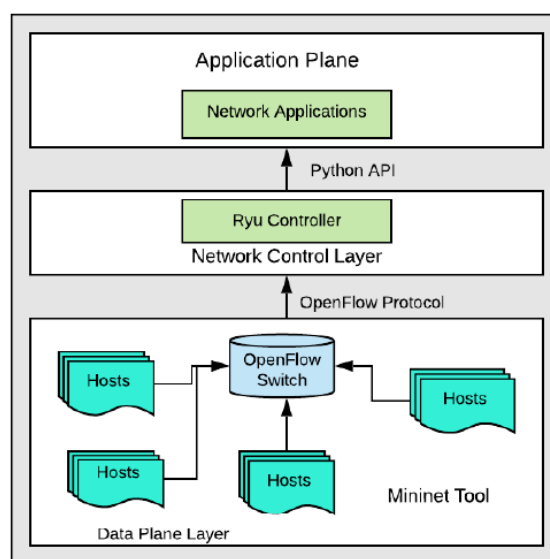


Figure 1: SDN Framework

The presented SDN framework, data plane has multiple node/hosts virtually created using mininet and all these are connected to the Openflow switch which 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 python 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.

Network Design:

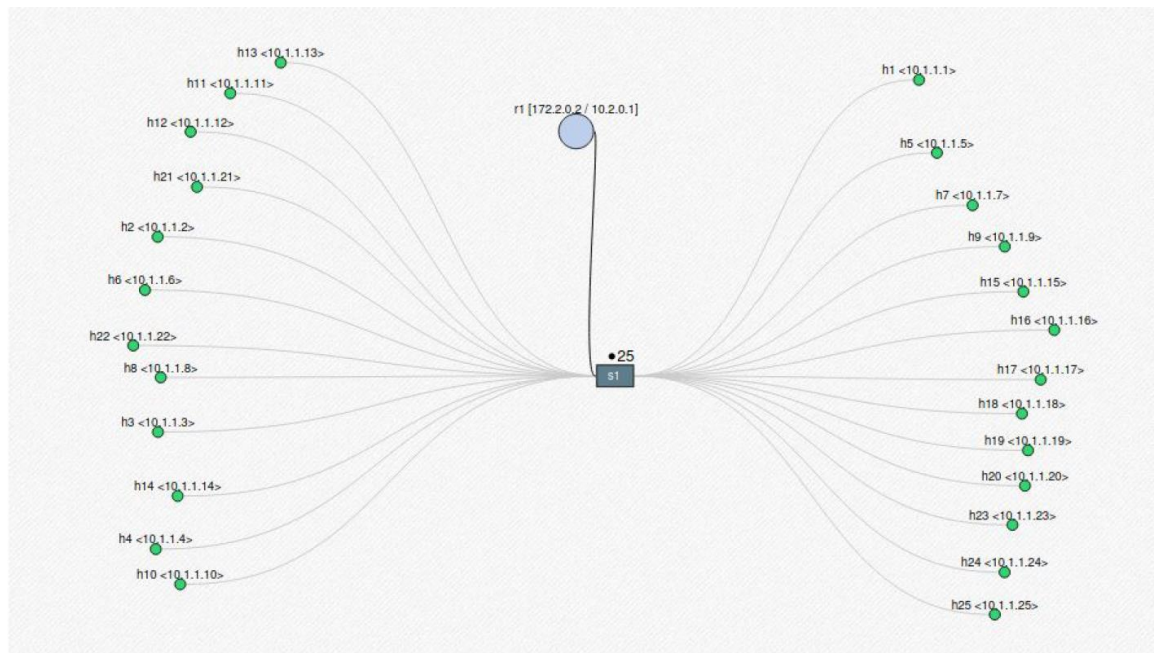


Figure 2: Mininet Network Design

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.

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 be 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.

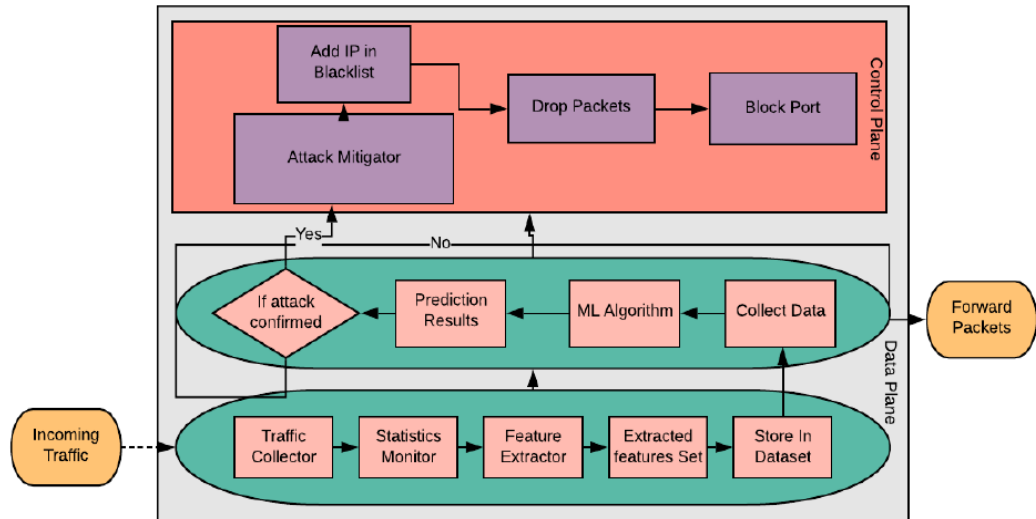


Figure 3: Flowchart of the presented method.

Software defined networks have various protocols and controllers, each of those are designed to perform in a certain way and provide efficiency and edibility 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 includes parameters such as speed of source IP, speed of flow entries and ratio of flowpair entries, all of this 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.

Evaluation and Analysis:

- **Results Compilation:** The simulation results were compiled and analysed to assess the performance of the proposed DDOS detection and mitigation model. Graphs and charts were generated to visualize the detection accuracy, the impact of mitigation strategies, and the overall network stability during attack scenarios.
- **Comparative Analysis:** The results were compared with baseline scenarios and other existing DDOS detection methods to highlight the advantages and improvements offered by the proposed model.

By setting up a comprehensive simulation environment and employing realistic traffic patterns, the experimental setup ensures a thorough evaluation of the proposed DDOS detection and mitigation model in SDN environments. This approach provides valuable insights into the model's effectiveness and potential areas for further improvement

5.2 Experimental results

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.

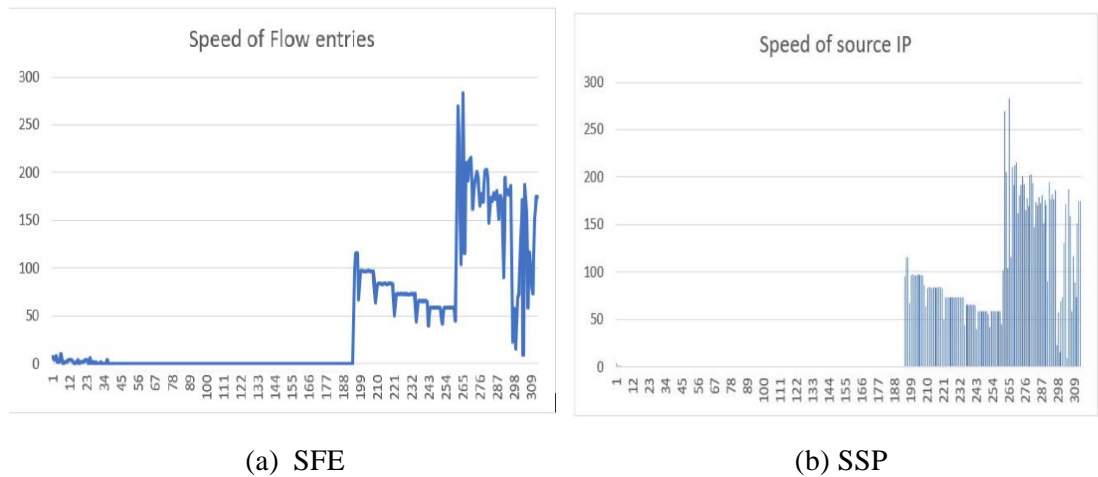


Figure 4

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.

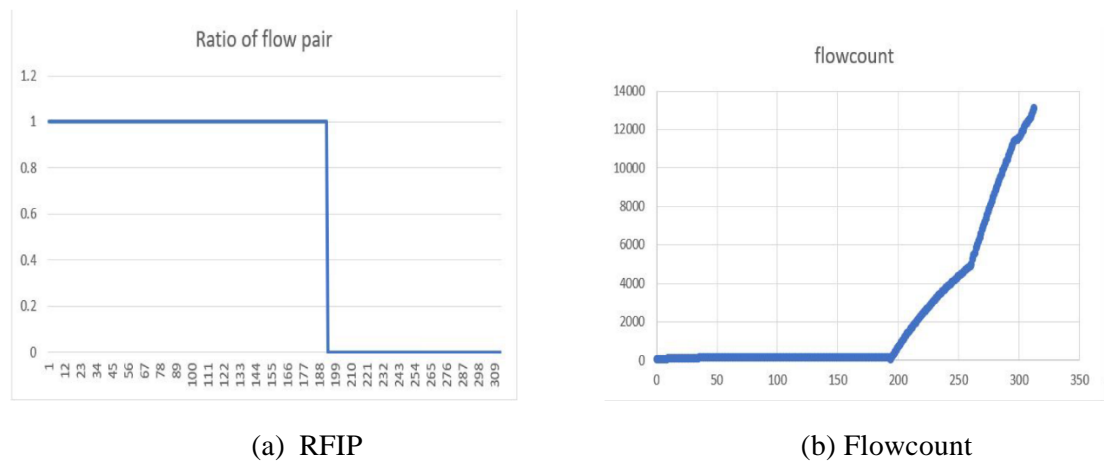


Figure 5

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.


```

mousam@mousam-VirtualBox: ~/Desktop/DDoS_final_yea...
$ ryu-manager controller.py
loading app controller.py
loading app ryu.controller.ofp_handler
instantiating app controller.py of SimpleSwitch13
instantiating app ryu.controller.ofp_handler of OFPHandle
SVM input data [11, 7, 1.0] prediction result ['0']
It's Normal Traffic
SVM input data [8, 1, 1.0] prediction result ['0']
It's Normal Traffic
SVM input data [12, 1, 1.0] prediction result ['0']
It's Normal Traffic
SVM input data [4, 0, 1.0] prediction result ['0']
It's Normal Traffic
SVM input data [2, 0, 1.0] prediction result ['0']
It's Normal Traffic
SVM input data [4, 0, 1.0] prediction result ['0']
It's Normal Traffic
SVM input data [6, 0, 1.0] prediction result ['0']
It's Normal Traffic
SVM input data [4, 0, 1.0] prediction result ['0']
It's Normal Traffic

```

Figure 6: Normal Traffic Prediction

SVM machine learning algorithm predicting the traffic as normal traffic.

```

mousam@mousam-VirtualBox: ~/Desktop/DDoS_final_year_project$ ryu-manager controll
er.py
loading app controller.py
loading app ryu.controller.ofp_handler
instantiating app controller.py of SimpleSwitch13
instantiating app ryu.controller.ofp_handler of OFPHandler
SVM input data [13, 12, 0.16666666666666666] prediction result ['0']
It's Normal Traffic
SVM input data [276, 276, 0.006944444444444444] prediction result ['1']
Attack Traffic detected
Mitigation Started
attack detected from port 1
Block the port 1
attack detected from port 1
Block the port 1
attack detected from port 1
Block the port 1
attack detected from port 1
Block the port 1
attack detected from port 1
Block the port 1
attack detected from port 1
Block the port 1
SVM input data [1, 0, 0.006920415224913495] prediction result ['0']
It's Normal Traffic
SVM input data [0, 0, 0.006920415224913495] prediction result ['0']
It's Normal Traffic
SVM input data [0, 0, 0.006920415224913495] prediction result ['0']
It's Normal Traffic
SVM input data [0, 0, 0.006920415224913495] prediction result ['0']
It's Normal Traffic
SVM input data [0, 0, 0.006920415224913495] prediction result ['0']
It's Normal Traffic

```

Figure 7: 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.

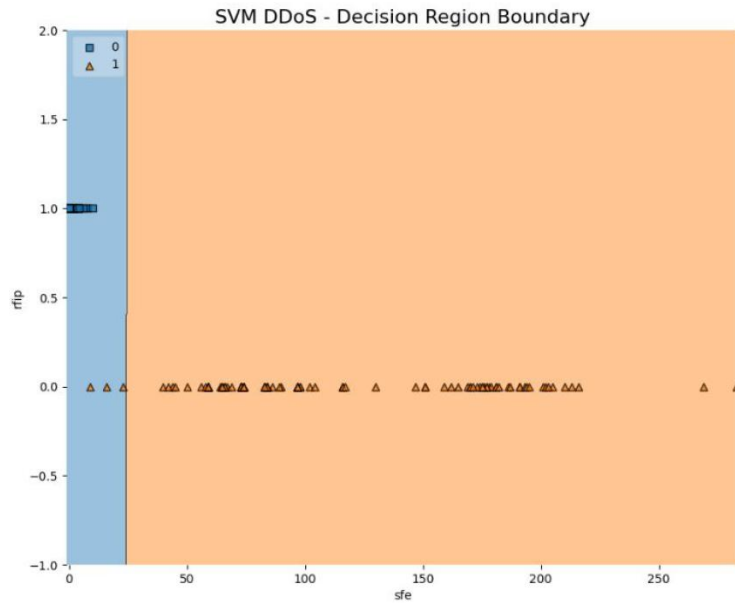


Figure 8: SVM Decision Boundary

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.

```
mousam@mousam-VirtualBox: ~/Desktop/DDoS_final_year_project/analysis
mousam@mousam-VirtualBox:~/Desktop/DDoS_final_year_project/analysis
$ python accuracy_score.py
Accuracy is 98.71794871794873
cross-validation score 0.9957446808510639
```

Figure 9: 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%.

```
mousam@mousam-VirtualBox: ~/Desktop/DDoS_finalYe...
mousam@mousam-VirtualBox:~/Desktop/DDoS_final_year_project/analysis$
python accuracy_score.py
Calculating Detection Ratio & False
Detection rate 1.0
False Alarm rate 0.0
```

Figure 10: Detection 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.

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.

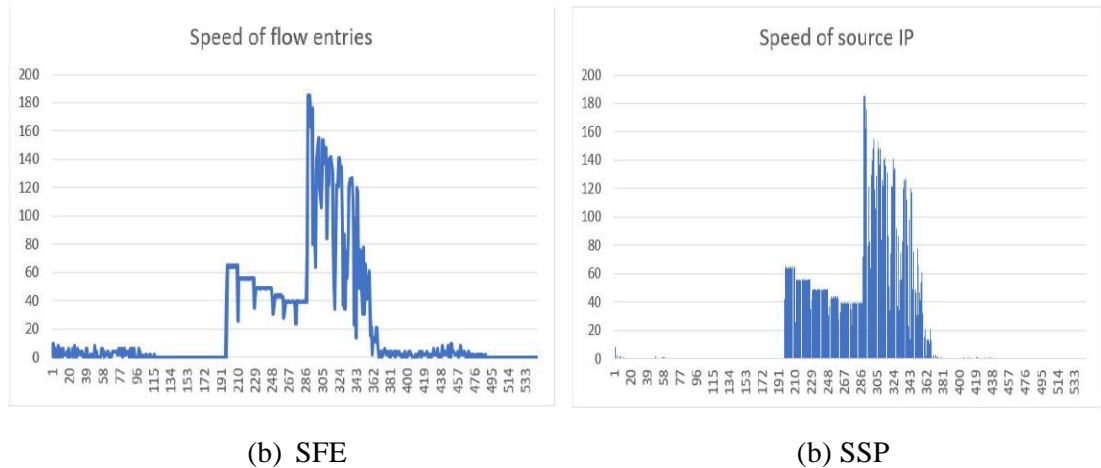


Figure 11

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

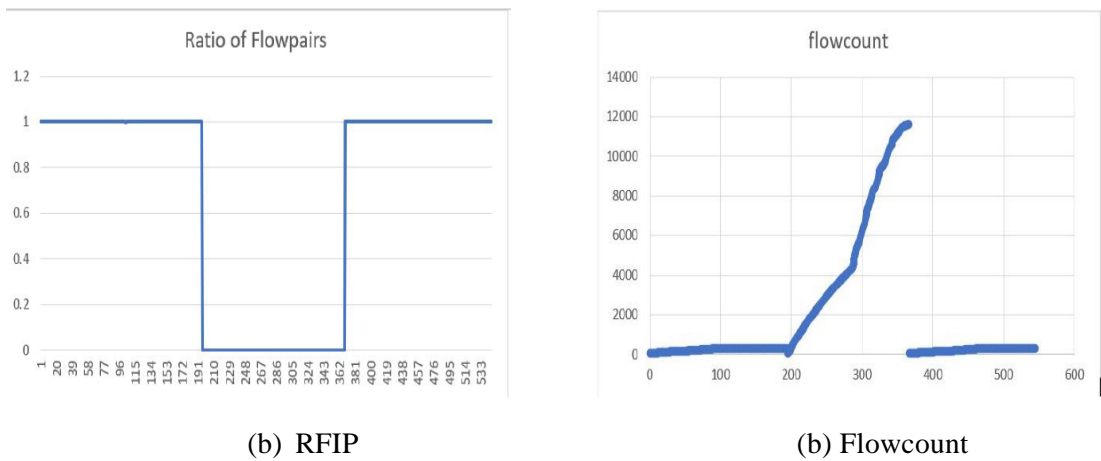


Figure 12

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 13: 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.

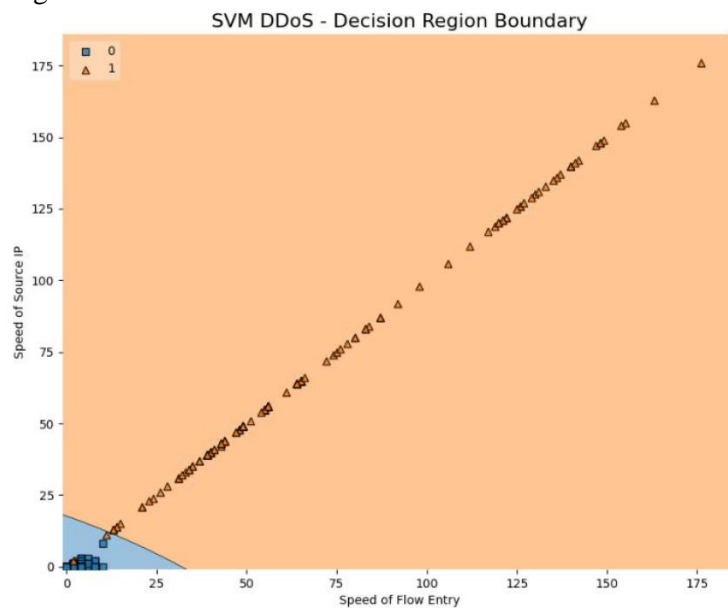


Figure 14: SVM Decision Boundary

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.

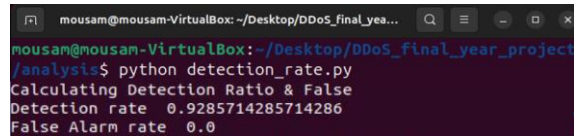
```

mousam@mousam-VirtualBox: ~/Desktop/DDoS_final_ye...
mousam@mousam-VirtualBox: ~/Desktop/DDoS_final_year_project/analysis$
python accuracy_score.py
Accuracy is 99.26470588235294
cross-validation score 0.9975308641975309

```

Figure 15: 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%.

A terminal window with a dark background and light-colored text. The window title is "mousam@mousam-VirtualBox: ~/Desktop/DDoS_final_yea...". The prompt is "mousam@mousam-VirtualBox:~/Desktop/DDoS_final_year_project/analysis\$". The command "python detection_rate.py" has been executed. The output shows "Calculating Detection Ratio & False", "Detection rate 0.9285714285714286", and "False Alarm rate 0.0".

```
mousam@mousam-VirtualBox: ~/Desktop/DDoS_final_yea...
mousam@mousam-VirtualBox:~/Desktop/DDoS_final_year_project/analysis$ python detection_rate.py
Calculating Detection Ratio & False
Detection rate 0.9285714285714286
False Alarm rate 0.0
```

Figure 16: Detection 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. Discussion and Conclusion

6.1 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 are 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.

6.2 Future work

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.

6.3 Conclusion

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 show the presented method can provide accuracy Of 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.