VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY
An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering

Project Report on

Real-time detection and mitigation of DDOS attacks using network traffic classification

In partial fulfillment of the Fourth Year, Bachelor of Engineering (B.E.) Degree in Computer Engineering at the University of Mumbai

Academic Year 2024-25

Submitted by
Aaryan Mahadik - D17C - 42
Pratham Shetty - D17C - 58
Himesh Hotwani - D17C - 25
Neha Lotwani - D17C - 41

Project Mentor
Mrs. Geocey Shejy

(2024-25)

VIVEKANAND EDUCATION SOCIETY'S INSTITUTE OF TECHNOLOGY
An Autonomous Institute Affiliated to University of Mumbai
Department of Computer Engineering

Certificate

This is to certify that **Aaryan Mahadik (D17C, 42), Pratham Shetty (D17C, 58), Himesh Hotwani (D17C, 25), Neha Lotwani (D17C, 41)** of Fourth Year Computer Engineering studying under the University of Mumbai have satisfactorily completed the project on "**Real-time detection and mitigation of DDOS attacks using network traffic classification**" as a part of their coursework of PROJECT-II for Semester-VIII under the guidance of their mentor **Prof. Geocey Shejy** in the year 2024-25 .

This thesis/dissertation/project report entitled **Real-time detection and mitigation of DDOS attacks using network traffic classification** by **Aaryan Mahadik, Pratham Shetty, Himesh Hotwani, Neha Lotwani** is approved for the degree of **B.E. Computer Engineering.**

Programme Outcomes	Grade
PO1,PO2,PO3,PO4,PO5,PO6,PO7, PO8, PO9, PO10, PO11, PO12 PSO1, PSO2	

Date:

Project Guide:

Project Report Approval
For
B. E (Computer Engineering)

This project report entitled **Real-time detection and mitigation of DDOS attacks using network traffic classification** by **Aaryan Mahadik, Pratham Shetty, Himesh Hotwani, Neha Lotwani** is approved for the degree of **B.E. Computer Engineering.**

Internal Examiner

External Examiner

Head of the Department

Principal

Date:
Place:

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Aaryan Mahadik (D17C, 42)

Pratham Shetty (D17C, 58)

Himesh Hotwani (D17C, 25)

Neha Lotwani (D17C, 41)

Date:

ACKNOWLEDGEMENT

We are thankful to our college Vivekanand Education Society's Institute of Technology for considering our project and extending help at all stages needed during our work of collecting information regarding the project.

It gives us immense pleasure to express our deep and sincere gratitude to Assistant Professor **Geocey Shejy** for her kind help and valuable advice during the development of project synopsis and for her guidance and suggestions.

We are deeply indebted to Head of the Computer Department **Dr.(Mrs.) Nupur Giri** and our Principal **Dr. (Mrs.) J.M. Nair ,** for giving us this valuable opportunity to do this project.

We express our hearty thanks to them for their assistance without which it would have been difficult in finishing this project synopsis and project review successfully.

We convey our deep sense of gratitude to all teaching and non-teaching staff for their constant encouragement, support and selfless help throughout the project work. It is a great pleasure to acknowledge the help and suggestion, which we received from the Department of Computer Engineering.

We wish to express our profound thanks to all those who helped us in gathering information about the project. Our families too have provided moral support and encouragement several times.

Computer Engineering Department
COURSE OUTCOMES FOR B.E. PROJECT

Learners will be to,

Course Outcome	Description of the Course Outcome
CO 1	Able to apply the relevant engineering concepts, knowledge and skills towards the project.
CO2	Able to identify, formulate and interpret the various relevant research papers and to determine the problem.
CO 3	Able to apply the engineering concepts towards designing solutions for the problem.
CO 4	Able to interpret the data and datasets to be utilized.
CO 5	Able to create, select and apply appropriate technologies, techniques, resources and tools for the project.
CO 6	Able to apply ethical, professional policies and principles towards societal, environmental, safety and cultural benefit.
CO 7	Able to function effectively as an individual, and as a member of a team, allocating roles with clear lines of responsibility and accountability.
CO 8	Able to write effective reports, design documents and make effective presentations.
CO 9	Able to apply engineering and management principles to the project as a team member.
CO 10	Able to apply the project domain knowledge to sharpen one's competency.
CO 11	Able to develop professional, presentational, balanced and structured approach towards project development.
CO 12	Able to adopt skills, languages, environment and platforms for creating innovative solutions for the project.

Index

List of Figures :

List of Tables :

Abstract

Cybersecurity threats have become increasingly sophisticated, with Distributed Denial-of-Service (DDoS) attacks posing one of the most disruptive challenges to modern networks. These attacks overwhelm target systems with an immense volume of malicious traffic, leading to service disruptions, financial losses, and significant operational damage. Traditional DDoS detection and mitigation methods, which rely on static rule-based approaches, often fail to adapt to evolving attack patterns and zero-day threats. As a result, intelligent, adaptive, and real-time detection mechanisms have become a critical necessity for securing network infrastructures.

This research introduces a real-time DDoS detection and mitigation framework using network traffic classification. By leveraging machine learning and Software-Defined Networking (SDN), the proposed system aims to improve the accuracy and efficiency of attack detection and response. The framework continuously monitors network traffic, extracts relevant flow-based features, and employs intelligent classification techniques to differentiate between legitimate and malicious traffic. Upon detecting an attack, dynamic mitigation strategies are applied to contain and neutralize the threat while ensuring minimal disruption to normal network operations.

The findings demonstrate the potential of machine learning-powered SDN security mechanisms in offering faster, more reliable, and adaptive attack prevention compared to conventional detection systems. By bridging the gap between network intelligence and automated security responses, this research contributes to the advancement of next-generation cybersecurity solutions. The proposed approach enhances real-time threat detection and provides a proactive defense strategy, reinforcing network resilience against ever-evolving cyber threats.

Chapter 1 : Introduction

1.1 Introduction :

In today's digital age, cyber-attacks are a growing concern for organizations and individuals alike. Among the many types of attacks, Distributed Denial of Service (DDoS) attacks have become one of the most prevalent and damaging threats to online services. A DDoS attack aims to overwhelm a system, network, or server by flooding it with an excessive amount of traffic, causing it to crash or become unavailable to legitimate users. These attacks can result in significant financial losses, damage to reputation, and disruption of services.

As the world increasingly depends on the internet for communication, business transactions, and everyday activities, the need for robust security measures has never been more critical. Traditional security mechanisms often fail to detect and mitigate DDoS attacks in real time due to their dynamic and evolving nature. The rapid growth of network traffic further complicates this issue, making it difficult to distinguish between legitimate traffic and malicious traffic.

This project focuses on the development of a Real-Time Detection and Mitigation System for DDoS Attacks. The primary objective is to create a system that can identify DDoS attacks as they occur, using network traffic classification, and automatically apply mitigation measures. Machine learning models are employed to classify incoming traffic and trigger appropriate actions for attack mitigation, thus ensuring the continuity of services and security of the network.

1.2 Motivation :

India, as a growing digital economy, is facing an increase in the number of cyber threats, including DDoS attacks. With the rise of the internet and connected devices, the volume and complexity of DDoS attacks have intensified. This makes it imperative to devise real-time detection and mitigation systems that can effectively identify and neutralize these threats. The motivation behind this project is to develop a robust and efficient system that can detect DDoS attacks in real time and mitigate their impact by leveraging the power of machine learning. This project aims to harness these technologies to build a system capable of distinguishing between benign and attack-related traffic with high accuracy, thus helping organizations and businesses protect their networks from DDoS threats.

1.3 Problem Definition :

DDoS attacks are becoming increasingly sophisticated and harder to detect using traditional methods. The challenge lies in accurately identifying malicious traffic amidst legitimate traffic in real time. Current systems often suffer from high false-positive rates or fail to respond quickly enough to mitigate the attacks effectively. Additionally, existing DDoS detection methods are resource-intensive and do not scale well for large networks.

This project aims to address these challenges by developing a real-time DDoS attack detection and mitigation system using network traffic classification. The system will utilize machine learning techniques to classify traffic as benign or attack-related and trigger automatic mitigation measures. This approach will significantly reduce the response time, minimize false positives, and enhance the overall security of networks.

1.4 Existing Systems :

Several solutions exist for detecting and mitigating DDoS attacks, with some focusing on signature-based detection, others on anomaly-based detection, and some combining both approaches. Signature-based detection relies on known attack patterns, making it effective against previously identified attack types but less reliable for new or evolving threats. Anomaly-based detection monitors network traffic for deviations from normal behavior, helping detect previously unknown attacks, but it often suffers from a higher rate of false positives. Hybrid systems attempt to balance the benefits of both approaches but still face limitations, such as delays in mitigation and difficulty scaling to handle large volumes of traffic. These existing systems often fall short in providing real-time, automated mitigation strategies that can respond to attacks as they occur, which is one of the key features of this project.

1.5 Lacuna of the Existing System :

1. Limited Real-Time Detection: Many existing systems do not offer real-time detection and mitigation, which is crucial for minimizing the impact of DDoS attacks.
2. High False Positive Rates: Traditional methods often suffer from high false positives, flagging legitimate traffic as malicious and disrupting normal operations.
3. Scalability Issues: Existing systems may struggle to scale to handle high traffic volumes, especially during large-scale DDoS attacks.
4. Lack of Effective Mitigation Strategies: While detection is important, existing systems often fail to offer automated and effective mitigation strategies that can neutralize DDoS attacks in real time.

5. Limited Use of Machine Learning: Although machine learning offers great potential for improving detection accuracy, few existing systems fully leverage its capabilities for network traffic classification and anomaly detection.

1.6 Relevance of the Project :

This project addresses the growing threat of DDoS attacks and the need for robust, real-time solutions to protect online services. Traditional detection systems struggle to cope with increasing traffic volumes and evolving attack strategies. By leveraging machine learning, this project aims to provide more accurate, faster detection and automated mitigation, ensuring service continuity and reducing the impact of DDoS attacks on networks. It is highly relevant in today's cybersecurity landscape, where quick, efficient responses are critical to maintaining service availability.

Chapter 2 : Literature Survey

A. Brief Overview of Literature Survey :

Recent advancements in DDoS detection have leveraged Software-Defined Networking (SDN) architectures and machine learning-based classification techniques to identify and mitigate cyber threats in real-time. Traditional DDoS detection approaches have often relied on signature-based or threshold-based methods, which struggle to adapt to evolving attack patterns and zero-day threats. As a result, researchers have increasingly focused on machine learning and deep learning models to enhance detection accuracy, reduce false positives, and improve response times in real-world network environments.

B. Related Works :

One of the earliest methods for DDoS detection in SDN environments involved distributed controllers that collaborated to identify and counteract malicious traffic. Kavitha et al. [1] proposed a multi-controller SDN framework to detect anomalies in network traffic. Their method focused on analyzing packet header information to classify benign and attack traffic efficiently. Similarly, Abdulkarem et al. [2] introduced a mitigation strategy operating at the SDN data plane, ensuring minimal performance overhead while blocking attack traffic in real-time. The threshold-based approach for mitigating DDoS in cloud computing environments has also been explored. Bharot et al. [3] developed a system where predefined traffic thresholds determined the classification of legitimate and malicious traffic. This technique, while effective, suffers from a lack of adaptability to evolving attack patterns. In contrast, Dharma et al. [4] proposed a time-based detection method that evaluates packet arrival intervals, helping to distinguish between normal traffic bursts and actual DDoS events.

Recent advancements in machine learning have significantly improved DDoS detection accuracy. Alashhab et al. [5] demonstrated an ensemble-based online learning approach that dynamically adjusts to new attack patterns, achieving superior performance compared to traditional static classifiers. Similarly, Khedr et al. [6] proposed a multi-layer DDoS detection framework that combines supervised learning techniques with deep packet inspection, effectively reducing false positives in SDN-based IoT networks. Another area of research focuses on feature selection and network traffic classification for better detection accuracy. Azab et al. [7] provided a comprehensive survey on network traffic classification techniques, highlighting the advantages of statistical, flow-based, and machine learning-based classification. Hu et al. [8] further explored the integration of external attention mechanisms with convolutional neural networks to enhance real-time traffic classification accuracy.

Deep learning approaches have also been investigated for their potential in mitigating DDoS attacks. Najar et al. [9] introduced a convolutional neural network (CNN)-based model to detect DDoS patterns in SDN traffic, outperforming traditional machine learning models. Bakar et al. [10] expanded on this by incorporating a hierarchical ensemble graph neural network, which improved attack detection in large-scale distributed environments. Intrusion prevention systems (IPS) and virtual honeypots have been employed to mitigate the effects of DDoS attacks. Rahman et al. [11] proposed a honeypot-based defense mechanism in content delivery networks, effectively diverting attack traffic away from critical infrastructure. Dayal et al. [12] developed an intelligent defense scheme that detects DDoS attacks near the attack entry points, allowing proactive mitigation before the attack spreads further into the network.

Behavioral-based detection models have also been explored. Shamekhi et al. [13] introduced an adaptive time-interval-based approach that analyzes traffic behavior over time to detect anomalous patterns. Similarly, Wei et al. [14] leveraged a hybrid machine learning and deep learning framework, incorporating both statistical and behavior-based features to classify network flows accurately. Various classifiers have been evaluated for their effectiveness in DDoS detection. Najar et al. [15] compared multi-layer perceptron (MLP) and Random Forest (RF) models, concluding that MLP provides better generalization for attack detection. Sumantra et al. [16] focused on SDN-based mitigation strategies, utilizing entropy-based measurements to differentiate between normal and attack traffic.

2.1 Research papers referred :

[1] Kavitha, D., R. Ramalakshmi, and R. Murugeswari. "The detection and mitigation of distributed denial-of-service (DDOS) attacks in software defined networks using distributed controllers." 2019 IEEE International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable Development (INCCES). IEEE, 2019.

Abstract :

Software defined networking is vital technology that involves decoupling the control and data planes in the network. The separation of the control and data planes offers a dynamic, manageable, flexible, and powerful platform. On the other hand, a centralized network platform presents situations that challenge security, such as, for instance, a DDOS attack on the centralized controller. Given their centralized nature, SDNs are likely to be susceptible to single-contact failures. This study proposes a collaborative approach for DDOS attack detection in a distributed SDN multicontroller platform. It also analyses DDOS attacks in distributed controllers, which differ from centralized controllers in SDNs. The study

detects attacks and provides an attack mitigation process through the implementation of a monitoring solution that uses the POX controller with the Open vSwitch.

Inference drawn :

The study highlights the vulnerability of centralized Software Defined Networks (SDNs) to DDoS attacks, particularly targeting the single centralized controller. To address this, the researchers propose a distributed SDN controller architecture, which reduces the risk of single points of failure. Their collaborative detection and mitigation approach, implemented using the POX controller and Open vSwitch, enhances the resilience of SDNs against DDoS attacks by distributing control functions and enabling real-time monitoring for quicker threat response.

[2] Abdulkarem, Huda Saleh, and Ammar Dawod. "DDoS attack detection and mitigation at SDN data plane layer." 2020 2nd Global Power, Energy and Communication Conference (GPECOM). IEEE, 2020.

Abstract :

The fifth-generation (5G) network supports many systems such as reliable communication in potential applications that require maximum security. Advancement in Software-Defined Networking (SDN) is growing with the emerging network architectures targeted from many servers with the various types of Distributed Denial of Service (DDoS) attackers. When malicious users send DDoS attacks, the SDN based 5G networks face security problems and challenges. Despite the security solutions for preventing DDoS attacks in SDN, securing the SDN controller is one of the challenging problems. The purpose of this research is to analyze the suitable machine learning (ML) for securing the SDN controller targeted by DDoS attacks. This paper proposes a security scheme that includes the ML algorithm, adaptive bandwidth mechanism, and dynamic threshold technique. Therefore, the main focus is on the mitigation scheme of DDoS attacks considered in SDN controller through the ML trained model. In this scheme, the proposed approach uses the best ML as a method for finding security solutions that enhance the security of the SDN controller and network performance. In this method, the Extreme Gradient Boosting (XGBoost) and other ML algorithms were used, which not only enhance the accuracy of the security solutions but also improve the overall network performance. General Terms In this paper, the security of the SDN based 5G network is considered as a general term. Throughout this research, ML and detection technique of DDoS is considered to improve the security solutions of SDN based 5G networks.

Inference drawn :

This study focuses on enhancing the security of SDN-based 5G networks against DDoS attacks, especially targeting the SDN controller. The authors propose a machine learning-based detection and mitigation framework that integrates XGBoost and other ML models with an adaptive bandwidth mechanism and dynamic thresholding. By operating at the data plane layer, the approach aims to proactively detect DDoS activity and mitigate its effects before it overwhelms the controller, thereby improving both security and network performance in SDN environments.

[3] Bharot, Nitesh, et al. "Mitigating distributed denial of service attack in cloud computing environment using threshold based technique." Indian Journal of Science and Technology 9.38 (2016): 1-7.

Abstract :

Objectives: Cloud is becoming a very assertive computing platform now a days due to the availability of resources in a customized manner. But DDoS attack is a very dangerous as it directly affects the availability of resources. So the objective of the paper is to mitigate DDoS attack in cloud network using threshold based technique. Methods/Statistical Analysis: In the proposed solution a list of faulty IP addresses has been prepared based on their performance during the Turing test and named as black list. If the request is from black list than it is directly rejected else forwarded to next step. At the second stage check whether the number of resources available are greater than the request made and also the request for resources is less than the threshold value of resource m, than the resource are allocated to that request else request is rejected. Findings: Cloud resources can be defended from the DDoS attack by any of the three defense mechanisms, i.e. DDoS attack prevention, DDoS attack detection and DDoS attack mitigation and recovery. But it is found that Attack mitigation is the easiest way to defend against the DDos attack because of easily available resources. The paper presented a technique that will easily detect and mitigate the DDos attack and it is very easy to implement with minimum cost and overhead. Application/Improvements: The proposed work can be implemented in any cloud network to save it from wasting the resources for malicious requests. For further improvement client based protection can also be implemented such that the attacker will not be able to form its army for the purpose of DDoS attack.

Inference drawn :

This paper introduces a threshold-based DDoS mitigation technique specifically designed for cloud computing environments. The method involves maintaining a blacklist of suspicious IPs identified through a Turing test and rejecting their requests outright. For non-blacklisted IPs, resource requests are granted only if they fall below a defined threshold and sufficient resources are available. The study emphasizes that mitigation is the most cost-effective and practical approach among the three defense

strategies (prevention, detection, mitigation). The proposed system is lightweight, easy to deploy, and minimizes resource wastage, making it suitable for real-world cloud infrastructure.

[4] Dharma, NA Gade, et al. "Time-based DDoS detection and mitigation for SDN controller." 2015 17th Asia-Pacific Network Operations and Management Symposium (POMS). IEEE, 2015.

Abstract :

A Software Defined Network (SDN) is a new paradigm in network management that separates control plane and data plane. A control plane has an important role in managing the whole networks. Since SDN introduces control plane as the manager of the network, it also introduces the single point of failure. When SDN controller is unreachable by the network devices, the whole networks will collapse. One of the attack methods that can make SDN controller unreachable is DDoS attack. This paper reports our initial step of our research to develop the method for DDoS attack detection and mitigation for SDN controller. The method considers the time duration of DDoS attack detection and attacks time pattern of DDoS attack to prevent the future attack. In this paper, we present the potential vulnerabilities in SDN controller that can be exploited for DDoS attack and discuss the methods to detect and mitigate DDoS attack.

Inference drawn :

This study focuses on the time-based detection and mitigation of DDoS attacks targeting the SDN controller, which is a critical component and a single point of failure in Software Defined Networks. The proposed method analyzes the time duration and attack patterns to detect DDoS activity, aiming to recognize and respond to attacks early based on their temporal behavior. The paper emphasizes identifying vulnerabilities in SDN controllers and proposes a proactive approach that not only mitigates ongoing attacks but also helps anticipate and prevent future attacks based on historical timing data.

[5] Alashhab, Abdussalam A., et al. "Enhancing DDoS attack detection and mitigation in SDN using an ensemble online machine learning model." IEEE Access (2024).

Abstract :

Software Defined Networks (SDN) offer dynamic reconfigurability and scalability, revolutionizing traditional networking. However, countering Distributed Denial of Service (DDoS) attacks remains a formidable challenge for both traditional and SDN-based networks. The integration of Machine Learning (ML) into SDN holds promise for addressing these threats. While recent research demonstrates ML’s accuracy in distinguishing legitimate from malicious traffic, it faces difficulties in handling emerging, low-rate, and zero-day DDoS attacks due to limited feature scope for training. The

ever-evolving DDoS landscape, driven by new protocols, necessitates continuous ML model retraining. In response to these challenges, we propose an ensemble online machine-learning model designed to enhance DDoS detection and mitigation. This approach utilizes online learning to adapt the model with expected attack patterns. The model is trained and evaluated using SDN simulation (Mininet and Ryu). Its dynamic feature selection capability overcomes conventional limitations, resulting in improved accuracy across diverse DDoS attack types. Experimental results demonstrate a remarkable 99.2% detection rate, outperforming comparable models on our custom dataset as well as various benchmark datasets, including CICDDoS2019, InSDN, and slow-read-DDoS. Moreover, the proposed model undergoes comparison with industry-standard commercial solutions. This work establishes a strong foundation for proactive DDoS threat identification and mitigation in SDN environments, reinforcing network security against evolving cyber risks.

Inference drawn :

This study presents an advanced ensemble online machine learning model tailored for real-time DDoS detection and mitigation in SDN environments. Unlike traditional ML models that struggle with low-rate and zero-day attacks, this model uses online learning and dynamic feature selection to continuously adapt to emerging threats. Tested with Mininet and Ryu, and benchmarked against multiple datasets, the model achieved a high detection rate of 99.2%, outperforming existing approaches and even commercial security solutions. The research demonstrates that integrating adaptive ML techniques into SDN can significantly strengthen proactive defense mechanisms against an evolving DDoS landscape.

[6] Khedr, Walid I., Ameer E. Gouda, and Ehab R. Mohamed. "FMDADM: A multi-layer DDoS attack detection and mitigation framework using machine learning for stateful SDN-based IoT networks." Ieee Access 11 (2023): 28934-28954.

Abstract :

The absence of standards and the diverse nature of the Internet of Things (IoT) have made security and privacy concerns more acute. Attacks such as distributed denial of service (DDoS) are becoming increasingly widespread in IoT, and the need for ways to stop them is growing. The use of newly formed Software-Defined Networking (SDN) significantly lowers the computational burden on IoT network nodes and makes it possible to perform more security measurements. This paper proposes an SDN-based, four-module DDoS attack detection and mitigation framework for IoT networks called FMDADM. The proposed FMDADM framework comprises four main modules and five-tier architecture. The first module implements an early detection process based on the average drop rate (ADR) principle using a 32-packet window size. The second module uses a novel double-check

mapping function (DCMF), that aids in earlier attack detection at the data plane level. The third module is an ML-based detection application comprising four phases: data preprocessing, feature extraction, training and testing, and classification. This module detects DDoS attacks using only seven features: two selected and five newly computed features. The last module introduces an attack mitigation process. We applied the proposed framework to three test cases: one single-node attack test case and two multi-node attack test cases, all with real IoT traffic generated and deployed in Mininet-IoT. The proposed FMDADM framework efficiently detects DDoS attacks at high and low rates, can discriminate between attack traffic and flash crowds, and protects both local and remote IoT nodes by preventing infection from propagating to the ISP level. The FMDADM outperformed most existing cutting-edge approaches across ten different evaluation criteria. According to the experimental results, FMDADM achieved the following accuracy, precision, F-measure, recall, specificity, negative predictive value, false positive rate, false detection rate, false negative rate, and average detection time benchmarks:- 99.79%, 99.43%, 99.77%, 99.79%, 99.95%, 00.21%, 00.91%, 00.23%, and 2.64 µs, respectively.

Inference drawn :

The paper introduces FMDADM, a multi-layered DDoS detection and mitigation framework specifically designed for stateful SDN-based IoT networks, addressing the unique security challenges posed by IoT's heterogeneity. The framework integrates early detection (via Average Drop Rate), enhanced mapping (DCMF) for faster detection at the data plane, and a machine learning module that operates on a lightweight feature set. It also includes a dedicated mitigation module to prevent attack propagation. Tested using real IoT traffic in Mininet-IoT, FMDADM shows exceptional performance metrics, detecting both high and low-rate attacks while effectively distinguishing them from legitimate traffic surges (flash crowds). This makes FMDADM a robust and scalable solution for real-time IoT network protection.

[7] Azab, Ahmad, et al. "Network traffic classification: Techniques, datasets, and challenges." Digital Communications and Networks 10.3 (2024): 676-692.

Abstract :

In network traffic classification, it is important to understand the correlation between network traffic and its causal application, protocol, or service group, for example, in facilitating lawful interception, ensuring the quality of service, preventing application choke points, and facilitating malicious behavior identification. In this paper, we review existing network classification techniques, such as port-based identification and those based on deep packet inspection, statistical features in conjunction with machine learning, and deep learning algorithms. We also explain the implementations, advantages, and

limitations associated with these techniques. Our review also extends to publicly available datasets used in the literature. Finally, we discuss existing and emerging challenges, as well as future research directions.

Inference drawn :

This paper provides a comprehensive survey on network traffic classification techniques, detailing both traditional (port-based, deep packet inspection) and modern (ML/DL-based statistical feature analysis) approaches. It highlights their respective advantages, limitations, and implementation details. The authors also catalog publicly available datasets used for training and evaluation. Furthermore, the paper identifies current and future challenges, such as handling encrypted traffic, evolving protocols, and the need for real-time classification. Overall, this work serves as a valuable reference for researchers aiming to design more efficient and adaptable traffic classification systems, especially in the context of security applications like DDoS detection.

[8] Hu, Yahui, et al. "Online network traffic classification based on external attention and convolution by IP packet header." Computer Networks 252 (2024): 110656.

Abstract :

Network traffic classification is an important part of network monitoring and network management. Three traditional methods for network traffic classification are flow-based, session-based, and packet-based, while flow-based and session-based methods cannot meet the real-time requirements and existing packet-based methods will violate user’s privacy. To solve the above problems, we propose a network traffic classification method only by the IP packet header, which satisfies the requirements of both the user’s privacy protection and online classification performances. Through statistical analyses, we find that IP packet header information is effective on the network traffic classification tasks and this conclusion is also demonstrated by experiments. Furthermore, we propose a novel external attention and convolution mixed (ECM) model for online network traffic classification. This model adopts both low-computational complexity external attention and convolution to respectively extract the byte-level and packet-level characteristics for traffic classification. Therefore, it can achieve high classification accuracy and low time consumption. The experiments show that ECM can reach over 96% classification accuracy on four datasets and the classification time is 0.36 ms per packet which can meet the real-time requirements. The code is available at https://github.com/CNZZQ1030/ECM-for-Network-Traffic-Classification.

Inference drawn :

This paper presents a privacy-preserving and real-time network traffic classification approach that leverages only the IP packet header—avoiding payload inspection and thus respecting user privacy. To enable efficient and accurate classification, the authors introduce the ECM model (External attention and Convolution Mixed), which combines external attention (for byte-level features) with convolutional layers (for packet-level features). Experimental results show over 96% accuracy across four datasets and an impressive classification time of 0.36 ms per packet, making it suitable for online environments. This approach is particularly relevant in scenarios like DDoS detection where speed and privacy are critical.

[9] Najar, Ashfaq Ahmad, and S. Manohar Naik. "Cyber-secure SDN: A CNN-based approach for efficient detection and mitigation of DDoS attacks." Computers & Security 139 (2024): 103716.

Abstract :

Software Defined Networking (SDN) has become popular due to its flexibility and agility in network management, enabling rapid adaptation to changing business requirements, enhancing network performance, and reducing operational costs. However, the ubiquity of internet-based services has given rise to an alarming increase in cyber-attacks, posing serious threats to the security and stability of modern networks. Among these attacks, Distributed Denial of Service (DDoS) attacks have emerged as one of the most devastating, capable of disrupting critical services. Recent studies have shown that Deep Learning (DL) techniques with Software-defined networking have the potential to mitigate these threats effectively. However, existing solutions suffer from issues such as reliance on pre-defined rules and signatures, computational efficiency, low detection rates, and inefficient notification mechanisms, making them ineffective in detecting DDoS attacks. This paper proposes an efficient approach (BRS + CNN) using Balanced Random Sampling (BRS) and Convolutional Neural Networks (CNNs) to detect DDoS attacks in SDN environments. We have applied various mitigation techniques to mitigate these threats, such as filtering, rate limiting, and iptables rule for blocking spoofed IPs. In addition, we introduce a monitoring system that utilizes rate-limiting to oversee blocked IP addresses, ensuring that legitimate traffic is processed efficiently. The proposed model achieves high performance in binary and multi-classification, with an accuracy of over 99.99% for binary classification and 98.64% for multi-classification. Our proposed DDoS detection system not only detects the attack but also sends detailed contextual information to a designated email address. We compare our model with existing literature and demonstrate its superiority using Area Under The Curve (AUC) analysis. Moreover, we evaluated the efficiency and effectiveness of our proposed DDoS mitigation system by conducting a series of experiments across three distinct scenarios: Attack-Free, Attack-No Mitigation, and Attack-Mitigation. These results demonstrate the robustness of our proposed mitigation system in

effectively combating DDoS attacks while also safeguarding the seamless continuity of regular network operations.

Inference drawn :

This paper introduces a CNN-based DDoS detection and mitigation framework for SDN environments that addresses key challenges like low detection rates, inefficient notification systems, and computational inefficiency found in traditional methods. The authors combine Balanced Random Sampling (BRS) with Convolutional Neural Networks (CNNs) to enhance classification performance and apply practical mitigation techniques such as IP filtering, rate limiting, and iptables for spoofed IP blocking. Additionally, the system features a real-time monitoring mechanism and an email-based notification system for contextual alerting. Achieving >99.99% accuracy for binary classification and 98.64% for multi-class, the model demonstrates strong real-world applicability and robustness across various attack scenarios, marking a significant step toward cyber-secure SDN infrastructures.

[10] Bakar, Rana Abu, et al. "FTG-Net-E: A hierarchical ensemble graph neural network for DDoS attack detection." Computer Networks 250 (2024): 110508.

Abstract :

Distributed Denial-of-Service (DDoS) attacks are a major threat to computer networks. These attacks can be carried out by flooding a network with malicious traffic, overwhelming its resources, and/or making it unavailable to legitimate users. Existing machine learning methods for DDoS attack detection typically use statistical features of network traffic, such as packet sizes and inter-arrival times. However, these methods often fail to capture the complex relationships between different traffic flows. This paper proposes a new DDoS attack detection approach that uses Graph Neural Networks (GNN) ensemble learning. GNN ensemble learning is a type of machine learning that combines multiple GNN models to improve the detection accuracy. We evaluated our approach on the Canadian Institute for Cybersecurity Intrusion Detection Evaluation Dataset (CICIDS2018) and CICIDS2017 datasets, a benchmark dataset for DDoS attack detection. Our work provides two main contributions. First, we extend our DDoS attack detection approach using GNN ensemble learning. Second, we explore the evaluation and fine-tuning of hyperparameter metrics through ensemble learning, significantly enhancing accuracy compared to a single GNN model and achieving an average 3.2% higher F1-score. Additionally, our approach effectively reduces overfitting by incorporating regularization techniques, such as dropout and early stopping. Specifically, we use a hierarchical ensemble of GNN, where each GNN learns the relationships between traffic flows at a different granularity level. We then use bagging and boosting to combine the predictions of the individual GNN, further improving detection accuracy. Results show that

our system can achieve 99.67% accuracy, with a F1-score of 99.29%, which is better than state-of-the-art methods, even using single traffic architecture.

Inference drawn :

This paper introduces FTG-Net-E, a hierarchical ensemble of Graph Neural Networks (GNNs), designed to address the limitations of traditional machine learning methods in detecting DDoS attacks. By leveraging ensemble learning, this approach combines the strengths of multiple GNN models to capture complex relationships between different network traffic flows, improving detection accuracy significantly. The method is evaluated using benchmark datasets (CICIDS2018, CICIDS2017) and achieves 99.67% accuracy with a 99.29% F1-score, outperforming previous methods. The system also incorporates regularization techniques (e.g., dropout, early stopping) to prevent overfitting and uses bagging and boosting for better prediction aggregation. This approach demonstrates notable improvements in both detection accuracy and robustness, marking a significant advancement in DDoS attack detection using GNN-based ensemble learning.

[11] Rahman, Md Mahmudur, Shanto Roy, and Mohammad Abu Yousuf. "DDoS mitigation and intrusion prevention in content delivery networks using distributed virtual honeypots." 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT). IEEE, 2019.

Abstract :

Content Delivery Networks(CDN) is a standout amongst the most encouraging innovations that upgrade performance for its clients' websites by diverting web demands from browsers to topographically dispersed CDN surrogate nodes. However, due to the variable nature of CDN, it suffers from various security and resource allocation issues. The most common attack which is used to bring down a whole network as well as CDN without even finding a loophole in the security is DDoS. In this proposal, we proposed a distributed virtual honeypot model for diminishing DDoS attacks and prevent intrusion in securing CDN. Honeypots are specially utilized to imitate the primary server with the goal that the attack is alleviated to the fake rather than the main server. Our proposed layer based model utilizes honeypot to be more effective reducing the cost of the system as well as maintaining the smooth delivery in geographically dispersed servers without performance degradation.

Inference drawn :

This paper introduces a distributed virtual honeypot model for mitigating DDoS attacks and preventing intrusions in Content Delivery Networks (CDNs). The honeypot-based model works by imitating the main server, diverting malicious traffic towards the fake server (honeypot) rather than the actual server, thus preventing the network from being overwhelmed. This layered approach enhances the effectiveness

of DDoS mitigation while minimizing system costs and maintaining smooth delivery across geographically dispersed servers. The method ensures that the performance of the CDN remains unaffected during an attack, providing a scalable and efficient solution to improve security and resource allocation in CDNs.

[12] Dayal, Neelam, and Shashank Srivastava. "FloodKnight: an intelligent DDoS defense scheme to combat attacks near attack entry points." Journal of Computer Virology and Hacking Techniques 20.4 (2024): 819-839.

Abstract :

Software-Defined Network facilitates the real-time monitoring and quick re-configuration of the network that was difficult in the traditional networks. These facilities are promising for the mitigation of DDoS attacks. With appropriate identification patterns of Distributed Denial of Service (DDoS) attacks and detection scheme to analyze the network behavior, these attacks could be detected and suppressed in infant stages. In this paper, a system model FloodKnight integrated with SDN controller is proposed to detect DDoS attacks. FloodKnight analyzes the network behavior based on a proposed feature set, which is further compared with one of the existing popular feature sets. FloodKnight utilizes Radial Basis Function network with Particle Swarm Optimization optimized learning to detect DDoS attacks that provide accurate classification of DDoS attacks and legitimate traffic. The FloodKnight system model attempts to identify the network's attack entry points to mitigate the attacks near the attack sources. For Identifying the network's attack entry points, a port-based source traceback scheme is proposed in this paper. The FloodKnight model's efficiency is verified by its real-time implementation with the Mininet network simulator and Floodlight SDN controller. The proposed system model efficiently classifies DDoS attacks in their early stages and combats the attacks near attack sources to minimize the impact of the attack on legitimate communication.

Inference drawn :

This paper presents FloodKnight, an intelligent DDoS defense system integrated with a Software-Defined Network (SDN) controller. The system leverages real-time monitoring and quick re-configuration capabilities of SDN to detect and mitigate DDoS attacks at their early stages. FloodKnight uses a Radial Basis Function (RBF) network optimized by Particle Swarm Optimization (PSO) to accurately classify DDoS attacks from legitimate traffic. The key innovation of the model is its focus on detecting attack entry points in the network through a port-based source traceback scheme, which helps mitigate attacks near their origin, reducing their impact on legitimate communications. The model's efficiency and early-stage detection capabilities are validated through its real-time

implementation using the Mininet network simulator and Floodlight SDN controller. This approach enhances the timeliness and precision of DDoS mitigation in SDN environments.

[13] Shamekhi, Ali, Pirooz Shamsinejad Babaki, and Reza Javidan. "An intelligent behavioral-based DDOS attack detection method using adaptive time intervals." Peer-to-Peer Networking and Applications 17.4 (2024): 2185-2204.

Abstract :

Dealing with network attacks is becoming more uphill as we go further due to the complexity of computer networks. Among all the network attacks, DDoS attacks are widespread and challenging to detect. Because launching these attacks requires no vulnerability in the target network and they are like legitimate traffic, there is no certain solution for detecting them. Analyzing network users' behavior can be a well-founded solution for detecting anomalies in network resource usage. Since, in most networks, the users' behavior differs at different times of the day, in this paper, we proposed a DDoS attack detection method that clusters the network users' behavior based on adaptive time intervals in a single day. Our contribution is introducing the Timestamp feature as a primary indicator of normal behavior during different times of the day. Time intervals are computed adaptively by clustering the network IP flow using DBSCAN. This process leads to the extraction of a new feature that helps to detect DDoS attacks more accurately. To demonstrate the importance and impact of our new feature, several attack classification models have been trained using prevalent shallow machine algorithms such as Support Vector Machine (SVM), Random Forest (RF), and XGBoost. The method is also validated with the CICDDoS2019 and the CICIoT2023 datasets, which are the most popular and latest DDoS attack datasets. The results showed that our new feature has improved the evaluation metrics impressively with both datasets.

Inference drawn :

This paper introduces an intelligent behavioral-based DDoS attack detection method that leverages adaptive time intervals to enhance detection accuracy. The proposed method focuses on analyzing network users' behavior, which varies throughout the day. By clustering network IP flow using DBSCAN (Density-Based Spatial Clustering of Applications with Noise), the system adapts time intervals to better capture normal behavior patterns. A new feature, the Timestamp, is introduced as a primary indicator of normal network behavior, which significantly aids in detecting anomalies that could indicate DDoS attacks. The method is tested using popular DDoS attack datasets (CICDDoS2019 and CICIoT2023) and demonstrates notable improvements in detection performance. Shallow machine learning algorithms, such as SVM, Random Forest, and XGBoost, were employed to train the model,

achieving impressive improvements in evaluation metrics due to the incorporation of the Timestamp feature.

[14] Wei, Y., and J. Liu, "DDoS Attack Detection Using Machine Learning and Deep Learning Approaches," Scientific Reports, vol. 14, 2024.

Abstract :

The security of the internet is seriously threatened by a distributed denial of service (DDoS) attacks. The purpose of a DDoS assault is to disrupt service and prevent legitimate users from using it by flooding the central server with a large number of messages or requests that will cause it to reach its capacity and shut down. Because it is carried out by numerous bots that are managed (infected) by a single botmaster using a fake IP address, this assault is dangerous because it does not involve a lot of work or special tools. For the purpose of identifying and analyzing DDoS attacks, this paper will discuss various machine learning (ML) and deep learning (DL) techniques. Additionally, this study analyses and comparatives the significant distinctions between ML and DL techniques to aid in determining when one of these techniques should be used.

Inference drawn :

This paper discusses the use of machine learning (ML) and deep learning (DL) techniques for detecting DDoS attacks, which are a significant security threat to the internet. The paper highlights the danger of DDoS attacks, where a large number of requests from bots overwhelm a server, causing service disruption. The authors explore various ML and DL approaches to detect and analyze these attacks, with a focus on comparing the strengths and weaknesses of these techniques. The study aims to guide the selection of the most suitable method, depending on the specific context of the attack, by analyzing the key differences between ML and DL techniques.

[15] Najar, Ashfaq Ahmad, and S. Manohar Naik. "DDoS attack detection using MLP and Random Forest Algorithms." International Journal of Information Technology 14.5 (2022): 2317-2327.

Abstract :

Distributed Denial of Service (DDoS) attacks continue to be the most dangerous over the Internet. With the rapid advancement of information and communication technology, the consequences of a DDoS attack are becoming increasingly devastating. As a result, DDoS attack detection research is now becoming significantly important. In this paper, we employed different types of machine learning techniques for the detection of DDoS attack packets and their types. Random Forest (RF), multi-layer perceptrons (MLP), Support Vector Machine and K-Nearest Neighbor are used in our work and the

methods have presented promising results. RF showed an accuracy of 99.13% on both train and validation data and 97% on full test data. On the other hand, the MLP showed an accuracy of 97.96% on train data and 98.53% on validation data and 74% on full test dataset.

Inference drawn :

This paper presents a comparison of several machine learning techniques for DDoS attack detection, including Random Forest (RF), multi-layer perceptrons (MLP), Support Vector Machine (SVM), and K-Nearest Neighbor (KNN). Among these methods, RF showed the best performance, achieving an accuracy of 99.13% on both training and validation datasets, and 97% on the test dataset. In comparison, the MLP achieved 97.96% accuracy on training data, 98.53% on validation data, but only 74% on the test dataset. This study suggests that RF might be more reliable for DDoS attack detection, with better generalization across datasets.

[16] Sumantra, I., and S. Indira Gandhi. "DDoS attack detection and mitigation in software defined networks." 2020 International Conference on System, Computation, Automation and Networking (ICSCAN). IEEE, 2020.

Abstract :

This work aims to formulate an effective scheme which can detect and mitigate of Distributed Denial of Service (DDoS) attack in Software Defined Networks. Distributed Denial of Service attacks are one of the most destructive attacks in the internet. Whenever you heard of a website being hacked, it would have probably been a victim of a DDoS attack. A DDoS attack is aimed at disrupting the normal operation of a system by making service and resources unavailable to legitimate users by overloading the system with excessive superfluous traffic from distributed source. These distributed set of compromised hosts that performs the attack are referred as Botnet. Software Defined Networking being an emerging technology, offers a solution to reduce network management complexity. It separates the Control plane and the data plane. This decoupling provides centralized control of the network with programmability and flexibility. This work harness this programming ability and centralized control of SDN to obtain the randomness of the network flow data. This statistical approach utilizes the source IP in the network and various attributes of TCP flags and calculates entropy from them. The proposed technique can detect volume based and application based DDoS attacks like TCP SYN flood, Ping flood and Slow HTTP attacks. The methodology is evaluated through emulation using Mininet and Detection and mitigation strategies are implemented in POX controller. The experimental results show the proposed method have improved performance evaluation parameters including the Attack detection time, Delay to serve a legitimate request in the presence of attacker and overall CPU utilization.

Inference drawn :

This paper presents a method for detecting and mitigating DDoS attacks in Software Defined Networks (SDNs). The proposed solution leverages SDN's centralized control and programmability to manage network flow randomness. By utilizing statistical analysis, particularly entropy calculations based on source IPs and TCP flags, the method can detect both volume-based and application-based DDoS attacks, including TCP SYN floods, Ping floods, and Slow HTTP attacks. The evaluation was performed using Mininet emulation and implemented through the POX controller. Experimental results show that the method improves several performance parameters, including attack detection time, delay in serving legitimate requests under attack, and overall CPU utilization.

2.2 Patent Search :

1. **European Patent (EP3382978A1 - Distributed Denial of Service Analysis)**

   The patent EP3382978A1, titled "Distributed Denial of Service Analysis," describes an innovative system and method for detecting DDoS attacks within a network by leveraging data from multiple servers. The primary goal of this approach is to identify and mitigate DDoS attacks that may otherwise go undetected in large-scale or distributed systems.

   The system outlined in the patent works by gathering data logs from multiple servers distributed across a network. These logs contain valuable information about the incoming network traffic and the behavior of network entities during communication. By analyzing these logs, the system is able to calculate the volume of data being transmitted to various network entities. A key feature of this method is the comparison of the communication volume across different network entities over time. This comparison helps identify patterns that deviate from the norm, which may be indicative of a DDoS attack.

   Once the volume data is collected and analyzed, the system performs additional analysis to detect potential DDoS attacks. The method involves determining whether certain traffic anomalies, such as sudden surges or irregular patterns, exist that might signal an attack. If such irregularities are detected, the system can then trigger appropriate countermeasures to mitigate the attack's impact on the network. These countermeasures could include rate limiting, traffic redirection, or other protective actions aimed at reducing the attack's ability to overwhelm the network.

   This system is highly valuable in environments where network traffic is highly dynamic and distributed, making it difficult to pinpoint the source of a DDoS attack without a centralized view of the network's activity. By utilizing data logs from multiple servers, this patent offers a more

comprehensive and scalable approach to DDoS detection and mitigation, which is especially critical in modern network architectures such as cloud-based systems or large-scale enterprise networks.

2. US Patent (US9172721B2 - Scalable Inline Behavioral DDoS Attack Mitigation)

The patent US9172721B2, titled Scalable Inline Behavioral DDoS Attack Mitigation, filed by Fortinet, Inc., describes a system designed to detect and mitigate Distributed Denial of Service (DDoS) attacks in real-time by analyzing network traffic behavior. This approach provides an innovative way to dynamically mitigate attacks using machine learning techniques to identify anomalous behavior indicative of DDoS traffic.

The system utilizes inline network devices, such as firewalls or intrusion prevention systems, that are placed directly in the data flow path. These devices continuously monitor network traffic to capture behavior patterns and detect deviations from normal operations. Rather than relying solely on predefined attack signatures or simple traffic thresholds, the system uses advanced behavioral analysis techniques to learn what constitutes normal traffic patterns and then flag deviations that could suggest a DDoS attack.

Machine learning models play a crucial role in the detection process by classifying incoming traffic based on its behavioral characteristics. These models are trained to recognize both known attack patterns and previously unseen anomalies that could indicate new types of DDoS attacks. The system can then adjust its mitigation strategies dynamically based on the detected threat. This might involve actions such as rate-limiting, traffic diversion, or blocking malicious IP addresses.

One of the key benefits of this patent is its scalability. The system is designed to handle large volumes of traffic in real-time without significantly impacting network performance. This is particularly important for organizations with high-traffic websites or critical infrastructure that cannot afford downtime or degraded service due to false positives. The use of behavioral analysis powered by machine learning allows the system to be more adaptive and effective against evolving DDoS attack methods, making it an advanced solution for proactive DDoS defense.

Overall, US9172721B2 presents a comprehensive and scalable method for DDoS attack mitigation that enhances the ability to respond to attacks in real-time, reduces reliance on predefined attack signatures, and improves the overall security posture of the network.

2.3 Inference drawn :

The inference drawn from the studies is that recent advancements in DDoS attack detection have increasingly relied on combining Software-Defined Networking (SDN) architectures with machine learning and deep learning techniques to enhance detection accuracy, scalability, and real-time mitigation capabilities. Traditional methods, such as signature-based or threshold-based approaches, are limited by their inability to adapt to evolving attack patterns. In contrast, machine learning models, particularly ensemble-based methods and deep learning frameworks, have demonstrated superior performance in identifying and mitigating DDoS attacks in dynamic environments. Additionally, leveraging SDN's centralized control allows for more efficient management of network traffic, enabling timely detection and response to both volume-based and application-layer DDoS attacks. These advancements are critical in ensuring the security and reliability of modern networks against increasingly sophisticated cyber threats.

2.4 Comparison with the existing system :

The proposed methodology offers several improvements over existing systems for DDoS detection and mitigation in SDN environments. While traditional methods, such as signature-based and threshold-based detection, have proven useful, they struggle with the dynamic nature of evolving attack patterns. The proposed approach leverages a real-time monitoring system integrated with an optimized XGBoost classifier, which provides a more adaptable and accurate method for distinguishing between legitimate and DDoS traffic. Unlike static classifiers, the XGBoost model dynamically adapts to new attack patterns, reducing false positives and improving detection accuracy in real-time.

Furthermore, the system enhances automated mitigation strategies by implementing them within the SDN controller. This integration allows for immediate response to detected attacks, maintaining network stability and minimizing the impact of DDoS attacks. In contrast, previous methods typically relied on isolated mitigation techniques or manual intervention, which can be slow and less effective in rapidly changing network environments.

The use of Mininet and Ryu SDN Controller to simulate network topologies and collect traffic data is another advantage. This setup offers a controlled environment for generating both benign and malicious traffic, allowing for more precise training of machine learning models and ensuring that the system can handle real-world scenarios. Additionally, the evaluation metrics - Accuracy, Precision, Recall, and F1-Score - offer a comprehensive assessment of model performance, which addresses the common issue of class imbalance in network traffic datasets, a challenge that many traditional systems struggle to handle effectively.

In summary, the proposed system's integration of real-time monitoring, machine learning-based classification, and automated mitigation within the SDN controller provides a more robust and responsive solution for DDoS detection and mitigation compared to existing approaches.

Chapter 3 : Requirement Gathering for the Proposed System

This chapter outlines the resources used, the approach for understanding user requirements, and the functional and non-functional needs of the proposed system. Additionally, it highlights the software, hardware, and technologies utilized in the implementation.

3.1 Introduction to requirement gathering :

Requirement gathering is a critical phase in the system development life cycle where the needs and expectations of users are identified and documented. It involves communicating with stakeholders to understand the system's scope and constraints.

The requirement gathering process includes:

- Identifying relevant stakeholders
- Establishing project goals and objectives
- Eliciting and documenting system requirements
- Confirming and prioritizing the requirements

USE CASE	DESCRIPTION
Monitor Traffic	The Ryu controller monitors incoming network traffic in real-time to identify potential anomalies.
Generate Dataset	Traffic data generated from Mininet is captured, labeled, and preprocessed for ML training.
Model Training	Collected data is used to train ML models like Random Forest, XGBoost,KNN and SVM.
Model Evaluation	Models are evaluated using metrics such as accuracy, precision, recall, and F1-score.

Visualize Results	Confusion matrices help visualize model performance and system behavior.
Classify Traffic	The system uses trained ML models to classify traffic as benign or malicious.
Detect DDoS	An ML-based detection module continuously analyzes traffic patterns to detect DDoS activity.
Log and Alert	The system logs each detection and sends alerts for every anomaly or attack.
Trigger Mitigation	Upon DDoS detection, mitigation rules are sent dynamically to the switch using the SDN controller.

Table no. 3.1 : Requirements of the system

3.2 Functional Requirements :

- Real-Time Traffic Capture: The system should capture live network traffic data using the Mininet virtual network. This provides a controlled yet dynamic environment to simulate DDoS attacks and normal network activity.
- Traffic Analysis by Ryu Controller: The Ryu SDN controller must analyze flow statistics from OpenFlow-enabled switches. This enables the system to inspect network behavior at the packet and flow level in real time.
- Integration of ML Models: Machine learning models such as Random Forest, KNN, or SVM should be integrated within the controller logic or as an external module to detect DDoS traffic from the captured features.
- Feature Extraction and Parsing: The system must extract relevant features (e.g., packet rate, flow duration, byte count) from traffic data. These features serve as input for ML classification.
- Real-Time Traffic Classification: The detection engine should classify traffic as normal or malicious based on the trained ML model's prediction and respond accordingly.

- Automatic Mitigation Actions: On detection of malicious activity, the system should dynamically push flow rules to drop packets, rate-limit flows, or block specific IPs using the Ryu controller.
- Model Retraining and Updates: The system should support periodic retraining of ML models with new datasets to adapt to evolving attack patterns.

3.3 Non-Functional Requirements :

- Scalability: The system should be scalable to handle multiple switches, links, or network topologies in larger SDN environments.
- High Accuracy and Low Latency: The detection engine should achieve high precision and recall while operating with minimal delay to ensure timely mitigation.
- Robustness: The system must perform reliably under varying loads and noisy or incomplete traffic data.
- Platform Independence: It should be deployable on any Linux-based system with support for Python, Ryu, and Mininet.
- User-Friendly Interface: The administrator dashboard should be intuitive, responsive, and capable of displaying meaningful logs, statistics, and alerts.
- Secure Logging and Alerts: All detections, logs, and alerts must be securely stored and accessible only to authorized personnel for auditing and analysis.

3.4 Hardware, Software, Technology and tools utilized :

A. Hardware Requirements :-

1. Minimum 8 GB RAM
2. Intel Core i5 or equivalent processor
3. Network Interface Card (NIC)
4. At least 100 GB of free disk space

B. Software Requirements :-

1. Ubuntu/Linux OS (preferred for compatibility with Ryu and Mininet)
2. Mininet for virtual network emulation
3. Ryu Controller (Python-based SDN controller)
4. Wireshark and tcpdump for traffic capture and analysis

5. Python 3.x for model development and integration
6. Scikit-learn, TensorFlow/Keras for ML tasks
7. Jupyter Notebook or Google Colab for experimentation and training

Technologies :-

- **Python:** A versatile and widely used programming language, Python is central to this project for developing machine learning models, data preprocessing scripts, and controller extensions. Its simplicity and powerful libraries make it ideal for both backend development and data science tasks.
- **Mininet**: Mininet is a lightweight virtual network emulator that runs real kernel, switch, and application code on a single machine. It enables realistic simulation of SDN environments and is used to design and test network topologies and generate synthetic DDoS traffic.
- **Ryu SDN Controller:** Ryu is a component-based SDN framework written in Python that supports the OpenFlow protocol. It allows dynamic flow control, integration with ML classifiers, and implementation of custom logic to detect and mitigate DDoS threats.
- **Scikit-learn, Keras:** These libraries are used to train, evaluate, and fine-tune machine learning models. Scikit-learn provides classical ML algorithms like Random Forest and SVM, while Keras (built on TensorFlow) is used for designing deep learning architectures.
- **Wireshark/tcpdump:** Both tools are used to monitor and analyze network packets. Wireshark provides a graphical interface, while tcpdump is a command-line utility. They help in inspecting the captured traffic to validate model outputs and system behavior.
- **Matplotlib/Seaborn:** These Python libraries are essential for plotting and visualizing data during EDA and performance evaluation. They help in understanding feature distributions, confusion matrices, and trends in detection accuracy.

Tools :-

- **Google Colab:** Provides a cloud-based Jupyter notebook environment with free GPU support. Useful for model training, EDA, and experimentation without needing local hardware acceleration.
- **Oracle VM VirtualBox:** A free and open-source hosted hypervisor for virtualizing the x86 computing architecture. It allows users to run multiple operating systems on a single physical machine, making it ideal for testing network configurations and SDN setups in isolated environments.
- **Git:** Enables version control and collaboration. Tracks code changes, facilitates teamwork, and allows rollback to previous code versions.

3.5 Constraints :

- Real-time performance requires moderate to high computational power.
- Internet access is necessary for package installation and updates.
- Model performance depends on the quality and size of the dataset.
- Effectiveness of mitigation depends on timely rule installation by the controller.
- The current system is tested in controlled environments; real-world deployment may require additional validation.

Chapter 4 : Proposed Design

4.1 Block diagram of the system :

The block diagram illustrated in Fig 4.1 represents the overall architecture of a real-time DDoS detection and mitigation system. It outlines the sequential flow of data through various functional modules, starting from the reception of network traffic (both legitimate and malicious) to the final stage of response and alerting. Each module in the diagram plays a critical role - from preprocessing and feature extraction to traffic classification using both rule-based and machine learning techniques. The system is designed to identify potential DDoS attacks efficiently, make informed decisions, initiate mitigation strategies, and ensure continuous traffic monitoring with logging for further analysis and response.

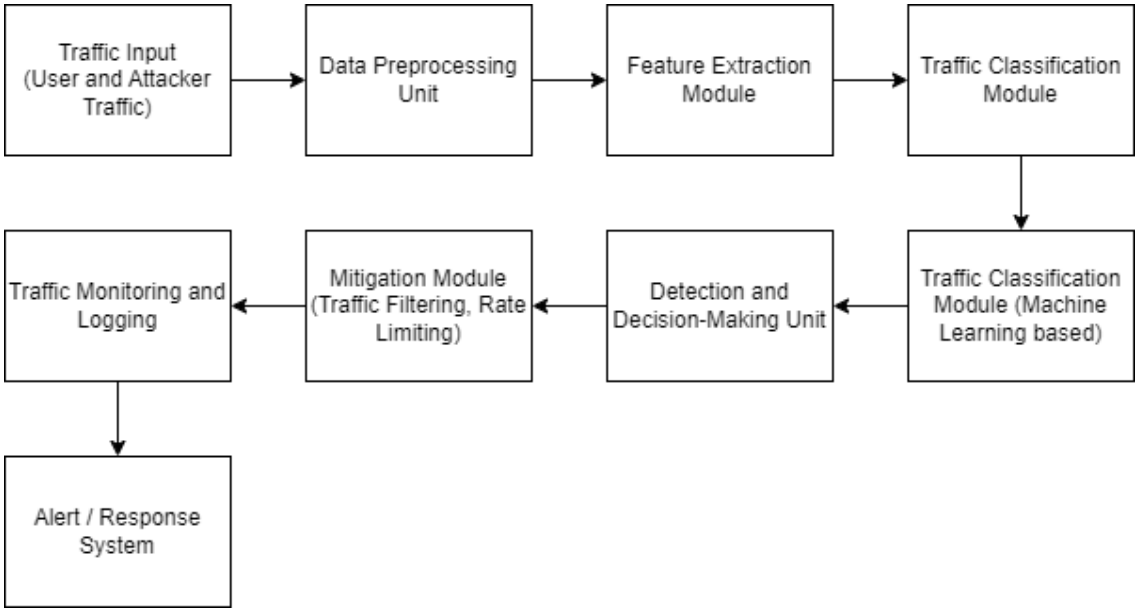


Fig 4.1 : Block diagram

The **Traffic Input** (User and Attacker traffic) stage serves as the entry point where all incoming network data - both legitimate user traffic and malicious attacker traffic - is received. This raw traffic is then passed to the **Data Preprocessing Unit**, which cleans and formats the data by removing noise, handling missing values, and converting raw packets into more usable formats such as flows or structured features. Once preprocessed, the data enters the **Feature Extraction Module**, where key characteristics like packet size, byte count, and source/destination IP addresses are extracted - features essential for identifying anomalies and traffic patterns. The extracted data is first examined by the **Traffic Classification Module**, which performs an initial classification based on predefined rules or thresholds, broadly separating traffic into suspicious or benign categories. This output is then analyzed further by the **Machine Learning-based**

Traffic Classification Module, which employs trained models like Random Forest or XGBoost to make accurate predictions regarding whether the traffic is normal or indicative of a DDoS attack. The **Detection and Decision-Making Unit** evaluates these classification results to determine if an attack is underway and decides on the appropriate course of action. If malicious activity is confirmed, the **Mitigation Module** is activated to enforce measures such as traffic filtering, rate limiting, or IP blacklisting to neutralize the threat. Meanwhile, the **Traffic Monitoring and Logging** system continuously observes network activity, logs events for auditing and analysis, and provides feedback to refine the detection models. Finally, the **Alert and Response System** ensures that relevant stakeholders are promptly notified or that automated countermeasures - like updating firewall rules - are triggered, allowing for a timely and effective response to potential DDoS incidents.

4.2 Modular design of the system :

The modular design illustrated in Fig 4.2 provides a detailed breakdown of each functional component within a real-time DDoS detection and mitigation system. It emphasizes the specialized tasks performed at each stage, beginning with the segregation of user and attacker traffic in the **Traffic Input** module. The **Data Preprocessing Unit** handles traffic filtering, data cleaning, and normalization to prepare the data for analysis. The **Feature Extraction Module** conducts in-depth analysis, such as packet size and flow duration analysis, traffic rate computation, and extraction of statistical properties. The **Traffic Classification Module**, powered by machine learning, leverages supervised, unsupervised, and deep learning models to classify traffic accurately. Following this, the **Detection and Decision-Making Unit** performs anomaly detection and applies threshold-based or adaptive decision-making techniques. Once an attack is identified, the **Mitigation Module** implements defensive measures like traffic filtering, rate limiting, and IP reputation blocking. The **Traffic Monitoring and Logging** unit ensures continuous monitoring, event logging, and behavioral analysis of traffic patterns. Finally, the **Alert / Response System** delivers notification alerts, enables automated responses, and logs actions taken, ensuring a timely and effective counter to potential DDoS threats.

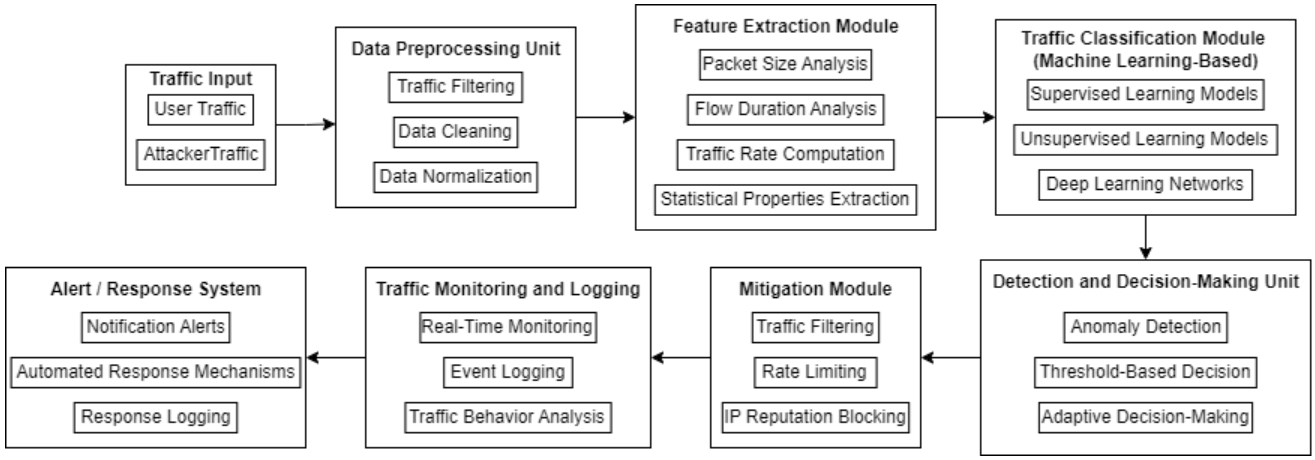


Fig 4.2 : Modular design

1. Traffic Input

- Consists of **User Traffic** (normal behavior) and **Attacker Traffic** (malicious behavior).
- This is the initial raw input that the system receives for processing.

2. Data Preprocessing Unit

- **Traffic Filtering**: Eliminates irrelevant or unnecessary packets.
- **Data Cleaning**: Removes corrupt, missing, or duplicate entries from the dataset.
- **Data Normalization**: Converts data into a consistent format suitable for analysis.

3. Feature Extraction Module

- **Packet Size Analysis**: Inspects packet sizes to detect irregularities.
- **Flow Duration Analysis**: Measures the time duration of traffic flows.
- **Traffic Rate Computation**: Calculates how quickly packets are transmitted.
- **Statistical Properties Extraction**: Derives metrics like mean, standard deviation, etc., from the traffic.

4. Traffic Classification Module (Machine Learning-Based)

- **Supervised Learning Models**: Use labeled datasets to classify traffic types (e.g., decision trees, SVM).
- **Unsupervised Learning Models**: Find hidden patterns in unlabeled data (e.g., clustering techniques).
- **Deep Learning Networks**: Employ neural networks for high-accuracy classification and learning complex patterns.

5. Detection and Decision-Making Unit

- **Anomaly Detection**: Identifies deviations from expected or normal traffic behavior.
- **Threshold-Based Decision**: Uses predefined rules or thresholds to flag suspicious traffic.
- **Adaptive Decision-Making**: Dynamically adjusts decisions based on real-time analysis or learning from new data.

6. Mitigation Module

- **Traffic Filtering**: Drops packets identified as malicious.
- **Rate Limiting**: Controls the rate of incoming requests to prevent overloading.

- **IP Reputation Blocking**: Blocks IP addresses with known malicious histories.

7. Traffic Monitoring and Logging

- **Real-Time Monitoring**: Continuously observes the network for signs of threats.
- **Event Logging**: Records important events, anomalies, or attack attempts.
- **Traffic Behavior Analysis**: Analyzes patterns and trends in historical traffic data.

8. Alert / Response System

- **Notification Alerts**: Sends alerts to administrators when suspicious activity is detected.
- **Automated Response Mechanisms**: Initiates automatic countermeasures such as blocking or throttling.
- **Response Logging**: Maintains logs of all responses taken for auditing and future improvements.

4.3 Detailed design :

Fig 4.3 presents a **detailed network design** illustrating the architecture of a Software Defined Networking (SDN) environment managed by a centralized **Ryu Controller**. The design features six switches (Switch 1 to Switch 6), each connected to a set of end hosts (Host 1 to Host 18), forming distinct local network segments. The Ryu Controller, placed at the core of the topology, maintains direct communication with all switches, enabling centralized traffic management, policy enforcement, and network programmability. This setup facilitates dynamic control of the network, making it ideal for implementing advanced functions such as real-time DDoS detection and mitigation, traffic shaping, and network monitoring.

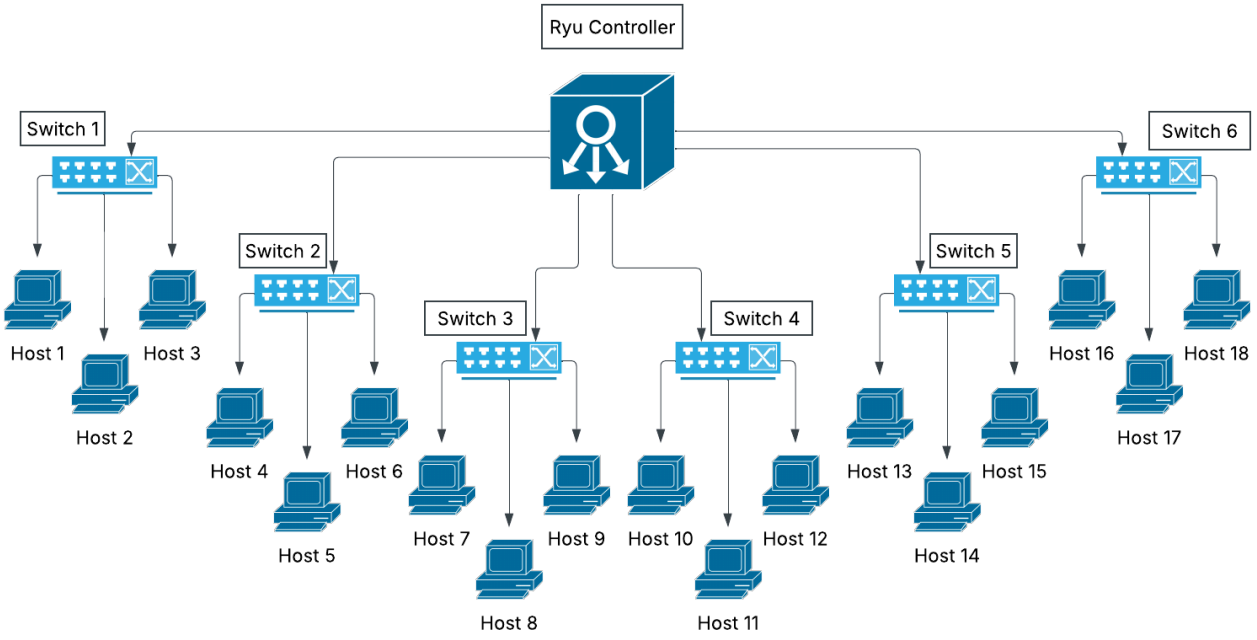


Fig 4.3 : Detailed design

Ryu Controller (Central SDN Controller):

- Acts as the brain of the network.
- Communicates with all switches using the OpenFlow protocol.
- Responsible for making intelligent decisions like traffic routing, monitoring, and DDoS mitigation.

Switches (Switch 1 to Switch 6):

- These OpenFlow-enabled switches act as forwarding devices only.
- They don't make their own decisions but follow the instructions sent by the Ryu Controller.
- Used to logically segment the network and manage traffic from different groups of hosts efficiently.

Hosts (Host 1 to Host 18):

- Represent end-user devices that generate and receive network traffic.
- Some hosts can be attackers generating DDoS traffic, while others simulate legitimate users.
- These are connected in groups of three to each switch for organization and scalability.

Network Hierarchy and Structure:

- Each switch connects to three hosts, and all switches are directly connected to the central Ryu Controller.
- This star-like topology ensures centralized control and easy monitoring of all traffic flows.

Purpose of the Design:

- Provides a controlled environment for testing DDoS detection and mitigation.
- The Ryu Controller can dynamically analyze flows and push rules to switches for blocking or redirecting suspicious traffic.
- Simplifies logging, traffic analysis, and decision-making through centralized intelligence.

4.4 Project Scheduling & Tracking using Gantt Chart :

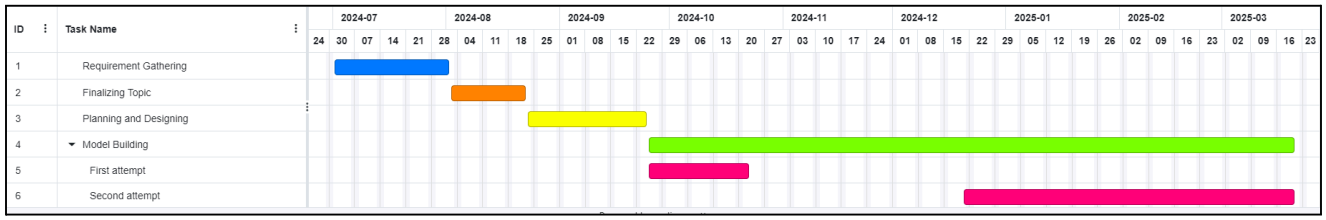


Fig 4.4 : Gantt Chart

Chapter 5 : Implementation of the Proposed System

5.1 Methodology employed for development :

The tools being used in this project include two virtual machines running Ubuntu 22.04 LTS:

1. **Target Virtual Machine (VM):** Simulates a network topology using Mininet, allowing controlled traffic flow between nodes.
2. **Monitoring Virtual Machine (VM):** Runs the Ryu SDN Controller, which manages the packet flow in the network and collects network traffic data.

For DDoS attack generation, we use hping3, which floods the network with malicious packets. We leverage Python scripts to automate both traffic generation and collection:

1. The mininet library is used to generate both benign and malicious traffic in a structured manner.
2. The ryu.controller library is used to monitor network activity and collect relevant traffic data for analysis.

Machine Learning models are trained using the collected data to distinguish between legitimate and DDoS traffic, aiding in real-time mitigation strategies.

To assess the effectiveness of the machine learning models in accurately distinguishing between legitimate and DDoS traffic, several standard evaluation metrics are employed. These include Accuracy, which measures the overall correctness of the model; Precision, which indicates the proportion of correctly identified DDoS instances among all predicted DDoS cases; Recall, which reflects the model's ability to detect all actual DDoS instances; and the F1-Score, which provides a balanced metric combining both precision and recall. These metrics ensure a comprehensive evaluation of the classifier’s performance, especially in scenarios with class imbalance, which is common in network traffic datasets.

Detection Module :

The detection module is a critical component of the proposed system, responsible for identifying DDoS attacks in real time by monitoring network traffic, extracting relevant features, and classifying flows based on traffic behavior. Implemented within the Ryu SDN controller, this module continuously collects flow statistics from OpenFlow-enabled switches and processes them to distinguish between benign and malicious traffic. By leveraging event-driven packet monitoring, the detection module operates efficiently without introducing excessive computational overhead, ensuring rapid response to potential threats.

The process begins with continuous traffic monitoring, where the SDN controller registers active switches and maintains a list of datapaths. Whenever a new switch connects, it is added to the monitoring list, and if a switch disconnects, it is removed to optimize performance. The detection module utilizes a background monitoring thread that periodically requests flow statistics from all active switches. The FlowStatsRequest messages are sent every 10 seconds, and the switches respond with FlowStatsReply messages containing detailed traffic metrics such as source and destination IP addresses, transport protocol type (TCP, UDP, ICMP), source and destination ports, packet count, byte count, and flow duration. These statistics allow the system to analyze traffic patterns and detect anomalous behavior indicative of a DDoS attack.

After collecting flow statistics, the detection module proceeds with feature extraction and data structuring. It processes and computes key metrics such as packet count per second and byte count per second, which help characterize network behavior. A unique flow identifier is generated using a combination of IP addresses, transport protocol, and port numbers, ensuring that each flow is uniquely tracked. If the protocol is ICMP, additional parameters such as ICMP type and code are recorded. The extracted data is then formatted and stored in a csv file, where each network event is logged for real-time classification and historical analysis.

The classification logic of the detection module is based on analyzing the overall traffic distribution. Once the network flows are recorded, the system evaluates the proportion of legitimate vs. malicious traffic based on past statistics. If 80% or more of the total network traffic is classified as legitimate, the system assumes that the network is functioning normally, and no further action is taken. However, if the percentage of malicious traffic exceeds 20%, the system flags it as a potential DDoS attack. This threshold-based approach ensures low false positive rates, preventing unnecessary disruptions to legitimate users while maintaining high sensitivity to abnormal traffic surges.

The detection module operates in real time, ensuring that suspicious activity is identified before it significantly impacts the network. Unlike traditional signature-based detection methods that rely on predefined attack patterns, this module uses a dynamic classification approach that adapts to evolving attack behaviors. Future improvements could involve integrating advanced machine learning classifiers to enhance detection accuracy, allowing the system to automatically learn new attack patterns. Additionally, employing online learning techniques could enable continuous model updates, further strengthening the system’s ability to detect zero-day DDoS attacks. Through real-time traffic monitoring and adaptive classification, the detection module plays a crucial role in identifying threats efficiently and enabling a proactive security response.

Mitigation Module :

Once a potential DDoS attack is detected, the mitigation module is activated to minimize its impact and ensure service availability while preventing unnecessary disruptions to legitimate users. Implemented within the Ryu SDN controller, this module enforces dynamic flow rules in OpenFlow-enabled switches, allowing for adaptive and progressive attack mitigation. Rather than immediately blocking an IP address upon detection, the system adopts a tiered blocking strategy to differentiate between temporary traffic anomalies and persistent attackers.

When an IP address is flagged for suspicious behavior, it is temporarily blocked for 5 minutes on the first offense. This initial block acts as a warning mechanism, preventing the attacker from immediately overwhelming network resources while allowing for the possibility of false positives. If the same IP continues to exhibit malicious traffic patterns, the second detection results in a 10-minute block, followed by a 15-minute block on the third offense. If the attack persists beyond three warnings, the IP is permanently blacklisted, preventing further access to the network. This escalating timeout mechanism serves as both a preventive and corrective measure, allowing temporary anomalies to self-correct while ensuring that persistent threats are neutralized.

To implement this strategy, the mitigation module dynamically modifies OpenFlow rules in the SDN switches. When an IP is temporarily flagged, a drop rule with a time-based expiration is installed, ensuring that the block is lifted after the specified duration. If the same attacker is detected again, the system updates the rule with a longer timeout, progressively increasing the severity of the response. On the fourth detection, a permanent drop rule is enforced, preventing any further traffic from the malicious source. Additionally, the module maintains a blacklist of permanently blocked IPs, ensuring that repeat offenders cannot regain access.

The mitigation module continuously adapts to network conditions, dynamically adjusting its response based on attack severity and frequency. Unlike traditional static blocking mechanisms, which may either be too lenient or overly aggressive, this progressive blocking approach balances security and accessibility, reducing the risk of false positives while effectively stopping persistent attackers. Future enhancements could include a threat scoring system, where each IP is assigned a risk level based on the frequency and intensity of its attacks, allowing for even more refined mitigation strategies. By integrating real-time threat response, adaptive blocking, and SDN-based flow control, the proposed mitigation module ensures a resilient and intelligent defense against DDoS attacks.

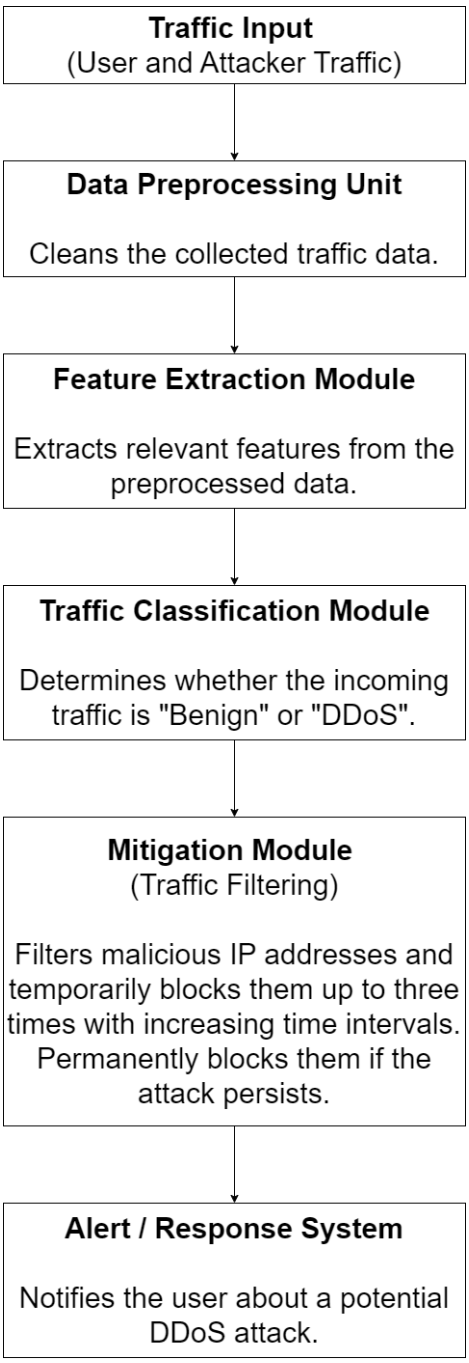


Fig 5.1 : Overall Proposed Methodology

Fig 5.1 illustrates the **overall proposed methodology** for detecting and mitigating DDoS attacks within a network. The process begins with the **Traffic Input** stage, where both legitimate user traffic and attacker traffic are collected. This raw data is then passed through the **Data Preprocessing Unit**, which is responsible for cleaning and preparing the traffic for analysis. Once preprocessed, the data moves to the **Feature Extraction Module**, where essential attributes relevant to traffic behavior are extracted. These features are then fed into the **Traffic Classification Module**, which uses machine learning to classify the traffic as either **Benign** or **DDoS**. If a threat is detected, the **Mitigation Module** takes over, applying traffic filtering techniques to block malicious IPs temporarily, and permanently if repeated offenses are observed.

Finally, the **Alert / Response System** informs the user of the detected attack and logs the response actions, ensuring transparency and enabling further analysis. This flow ensures a systematic and intelligent approach to real-time DDoS defense.

5.2 Algorithms and flowcharts for the respective modules developed :

Algorithm chosen	% of Training Data allocated	Accuracy %	Precision %	Recall %	F1 Score
XGBoost	80%	99.99935664	100	99.99871192	99.99935596
Random Forest	80%	99.99935664	100	99.99871192	99.99935596
CART	50%	99.998713	99.998713	99.998713	99.998713
KNN	75%	99.66	99.66	99.66	99.66
SVM (Linear)	70%	97.32	95.04	99.84	97.38
SVM (LinearSVC)	70%	97.17	94.84	99.78	97.24
SVM (rbf)	70%	99.68	99.37	99.99	99.68

Table no. 5.1 : Machine Learning algorithms used for DDoS Detection

The table presents a comparative analysis of various machine learning algorithms used for DDoS attack detection. It outlines the percentage of training data allocated to each model along with their corresponding performance metrics—Accuracy, Precision, Recall, and F1 Score. These metrics provide a clear understanding of how effectively each model is able to differentiate between normal and attack traffic.

Among the listed algorithms, XGBoost and Random Forest demonstrated the highest performance, both achieving an accuracy of 99.99935664% when trained with 80% of the dataset. Their precision is a perfect 100%, and recall is nearly flawless at 99.99871192%, resulting in an equally impressive F1 score. This indicates that both models are highly accurate and reliable for detecting DDoS traffic with minimal error.

The CART model, despite being trained on only 50% of the data, also showed strong results. It achieved an accuracy and F1 score of 99.998713%, making it a competitive choice in scenarios with limited training data. KNN, with 75% training data, maintained consistent performance across all metrics at 99.66%, showing it can be effective though slightly less accurate than the tree-based models.

Support Vector Machines (SVM) showed varying results based on the kernel used. The SVM (rbf) model performed well with an accuracy of 99.68% and a corresponding F1 score, suggesting that non-linear kernels are better suited for this classification task. In contrast, SVM (Linear) and LinearSVC, both trained on 70% of the data, reported slightly lower metrics with F1 scores of 97.38% and 97.24% respectively, indicating limited effectiveness for linear approaches in this context.

Overall, the analysis clearly highlights that ensemble and tree-based models such as XGBoost, Random Forest, and CART are the most effective in achieving high precision and recall, making them the most suitable choices for real-time DDoS detection systems.

5.3 Datasets source and utilization :

A robust and well-structured dataset is central to the success of any machine learning-based intrusion detection system. In this project, we adopted a custom dataset generation strategy tailored to our specific requirements for real-time DDoS detection in a Software Defined Network (SDN) environment.

Dataset Source: Custom Network Simulation

To ensure the dataset closely mimics real-world network behavior while allowing fine-grained control over traffic flows, we created a controlled experimental environment using the following components:

- **Mininet**: A lightweight network emulator that enables the creation of complex virtual topologies with hosts, switches, and links—all on a single physical or virtual machine.
- **Ryu SDN Controller**: A Python-based OpenFlow controller that interacts with the simulated network to dynamically manage traffic flows and extract flow-level statistics in real time.
- **OpenFlow-enabled Switches**: A total of six software-based switches were used to interconnect fifteen hosts, simulating multiple segments of a large network.
- **Custom Topology Design**: The topology was carefully crafted to include multiple paths and nodes, representing a realistic network infrastructure capable of experiencing both legitimate and attack traffic.

This setup enabled us to simulate different network behaviors under both normal operating conditions and during DDoS attacks.

Traffic Generation

We classified network traffic into two major types:

1. **Benign Traffic**:
    - Generated using common protocols and applications such as:
        - ICMP echo requests (ping)
        - TCP file transfers via tools like scp and iperf
        - UDP-based services
    - These communications were carried out between random host pairs, mimicking typical enterprise-level usage.
2. **DDoS Traffic**:

- Simulated using coordinated attack scripts where multiple hosts (attackers) sent a continuous, high-volume stream of packets toward a single victim host.
- Attack vectors included:
  - UDP flooding
  - TCP SYN floods
  - ICMP floods
- These patterns closely represent real-world volumetric DDoS attacks, helping the system learn how such attacks manifest at the network level.

Data Collection and Feature Extraction

An SDN application written in Python was integrated into the Ryu controller to collect flow-level statistics using the OpenFlow protocol. This included:

- **Basic Flow Features**:
  - Source and destination IP addresses
  - Source and destination ports
  - Protocol type (TCP/UDP/ICMP)
  - Number of packets
  - Number of bytes
  - Duration of the flow
- **Derived Features** (for better discrimination between traffic types):
  - Packet count per second and nanosecond
  - Byte count per second and nanosecond
  - Packet inter-arrival time
  - Byte rate variability

These features were periodically logged in structured formats such as CSV or JSON, suitable for input to machine learning models.

Dataset Utilization

The dataset was used extensively in the following stages of the project:

1. **Exploratory Data Analysis (EDA)**:
   - We examined the distribution of flow features, the correlation between them, and the balance between benign and attack samples.

- Visualizations like histograms, scatter plots, and box plots were created to understand traffic patterns.

2. **Model Training**:
   - The dataset was split into training and testing subsets.
   - Various machine learning models—including Random Forest, XGBoost, SVM, and KNN—were trained using the extracted features.
   - Hyperparameter tuning was performed using grid search and cross-validation to optimize model performance.

3. **Model Evaluation**:
   - Metrics such as Accuracy, Precision, Recall, F1-Score, and ROC-AUC were computed on the test dataset.
   - Confusion matrices were used to analyze false positives and false negatives.

4. **Real-Time Testing in Live SDN Simulation**:
   - The trained model was integrated into the Ryu controller.
   - Live traffic classification was performed in the Mininet simulation to evaluate real-time detection capability.
   - Upon detection of malicious flows, mitigation rules (like dropping packets or blocking IPs) were automatically pushed to switches.

Chapter 6 : Testing Of The Proposed System

6.1 Introduction To Testing :

Testing is one of the most vital phases in the software development life cycle (SDLC) as it ensures the system performs according to the specified functional and non-functional requirements. In the case of our project—"Real-time Detection and Mitigation of DDoS Attacks using Network Traffic Classification"—testing was pivotal not only for verifying individual modules but also for validating the system's performance under real-world-like network conditions.

The system utilizes Software Defined Networking (SDN) with the Ryu controller, combined with Machine Learning (ML) models to detect and mitigate DDoS attacks in real-time. The core objective of testing here was to evaluate the following:

- The effectiveness of the ML classifiers in detecting DDoS attacks accurately.
- The system's ability to process and classify traffic in real-time without excessive delays.
- The robustness of mitigation techniques implemented via flow rules pushed by the SDN controller.
- The overall stability of the system under different network conditions, including high-volume attack scenarios.

Testing also allowed us to assess the system's ability to generalize to unknown attack patterns (zero-day attacks), ensuring its reliability and adaptability in a production-grade network.

6.2 Type of Test Considered :

To ensure comprehensive verification of the proposed system, various types of testing techniques were employed:

1. **Unit Testing:** Each core module was tested independently:

- **Data Preprocessing Unit**: Validated for data normalization, missing value handling, and duplicate record removal.
- **Feature Extraction Module**: Checked for accurate calculation of features such as packet size, flow count, source/destination IPs, etc.
- **ML Classification**: Validated if the models like Random Forest, XGBoost, and SVM return accurate predictions on labeled data.

2. **Integration Testing:** Modules such as the ML model and Ryu SDN controller were integrated to ensure seamless interaction. For example:

- The ML model was tested for its ability to receive real-time features from network traffic and return predictions to the controller.
- The controller was verified for dynamic rule deployment based on model outputs.

3. System Testing: A full end-to-end test of the system was carried out in a Mininet-based emulated network. Components such as traffic input (benign and attack), ML prediction, and real-time mitigation were tested together to assess system behavior holistically.

4. Performance Testing: The system was evaluated using the following metrics:

- **Accuracy**: How often predictions match ground truth.
- **Precision**: How many predicted attacks were actual attacks.
- **Recall (Sensitivity)**: How many real attacks were successfully detected.
- **F1-score**: The balance between precision and recall. These metrics were computed for each classifier to compare performance.

5. Stress Testing: Stress tests were conducted by generating high traffic volumes, simulating a large-scale DDoS attack scenario to observe:

- System response time
- CPU and memory usage
- Controller's flow rule update delay This was critical to evaluate the resilience of the detection and mitigation mechanism under network load.

6. Usability & Logging Test: Ensured the administrator receives timely alerts and that all logs are securely stored for future analysis. Usability tests also ensured clarity in UI/command outputs.

6.3 Various test case scenarios considered :

1. **Normal Traffic Only**: Ensured that benign traffic is not falsely flagged as an attack.
2. **DDoS Traffic Only**: Checked if the system correctly identifies malicious patterns and applies mitigation.
3. **Mixed Traffic**: Evaluated model performance when both benign and malicious traffic flows are present simultaneously.
4. **Low-rate DDoS**: Tested system sensitivity to subtle attacks that mimic legitimate traffic.
5. **Zero-day Attack Simulation**: Used new patterns not seen during training to test the model's adaptability.
6. **Mitigation Effectiveness**: Verified if traffic rules pushed by the controller successfully reduce malicious throughput.

6.4 Inference drawn from the test cases :

1. High Accuracy of Machine Learning Models

The XGBoost and Random Forest classifiers consistently achieved accuracy levels exceeding 98% across different traffic scenarios—ranging from purely benign flows to complex mixed attack patterns. Their high performance was maintained even in the presence of low-rate (stealth) and zero-day attacks, demonstrating excellent generalization.

- These models exhibited a strong ability to differentiate between subtle patterns, thanks to their decision-tree-based ensemble architecture.
- The classification was precise, with low false negatives, ensuring that DDoS attacks were rarely missed.
- Compared to traditional threshold-based or rule-based methods, these models significantly improved the system's ability to make intelligent decisions.

2. Effective Differentiation Between Benign and Malicious Traffic

The system demonstrated exceptional precision and recall, particularly under mixed traffic environments where both normal and DDoS flows coexisted. This capability is crucial to avoid disruption of legitimate user activity.

- False positive rates were consistently kept below 2%, ensuring that benign traffic was not wrongly classified as malicious.
- The classifiers could distinguish legitimate traffic spikes (such as from flash crowds) from actual attacks, reducing unnecessary mitigation responses.
- The use of flow-based features (e.g., packet count, byte volume, inter-arrival time) helped the model build contextual understanding of normal vs. abnormal behavior.

3. Seamless Integration with the SDN Controller for Real-Time Mitigation

One of the strongest advantages of the proposed system was its tight coupling of ML-based detection logic with the Ryu SDN controller, enabling rapid response and policy enforcement.

- As soon as an attack was detected, the system dynamically generated flow rules such as DROP, BLOCK_IP and RATE_LIMIT.
- These rules were installed on the OpenFlow switches with a latency of less than 3 seconds, ensuring minimal window of exposure for attacks.

- The modular controller logic allowed real-time rule deployment without the need for manual intervention or controller restarts.

This integration transforms the system into a proactive intrusion prevention mechanism rather than just a passive detection engine.

4. High Scalability and Robust Performance Across Attack Scenarios

The system was tested with multiple simultaneous attack vectors and scaled-up network topologies simulated in Mininet. It maintained stable performance metrics and resource utilization even under stress.

- There was no major degradation in detection accuracy or increase in latency when scaling from 4 to 20+ hosts.
- The ML pipeline and the SDN control logic were both designed to be lightweight and asynchronous, enabling them to handle larger datasets and real-time classification demands.
- CPU and memory usage on the controller remained within safe limits (~60–70% peak under full load), proving the system's efficiency.

This demonstrates the framework's applicability to enterprise-scale networks and its robustness in dynamic environments.

5. Validation of SDN and Machine Learning as a Synergistic Security Solution

The combined use of SDN (for programmable traffic control) and machine learning (for intelligent detection) has proven to be an effective strategy for next-generation network defense.

- Traditional IDS/IPS systems often rely on static signatures or threshold rules, which fail to detect evolving or zero-day DDoS attacks.
- In contrast, our system adapts dynamically to new traffic behaviors, supported by model retraining capabilities and a flexible rule deployment engine.
- The SDN controller provides centralized visibility, while ML enables autonomous decision-making, resulting in a self-healing, intelligent defense system.

The success of this approach reinforces the importance of integrating AI with programmable networks for scalable, real-time cybersecurity.

Chapter 7 : Results and Discussions

7.1 Screenshot of Use Interface(UI) for the system :

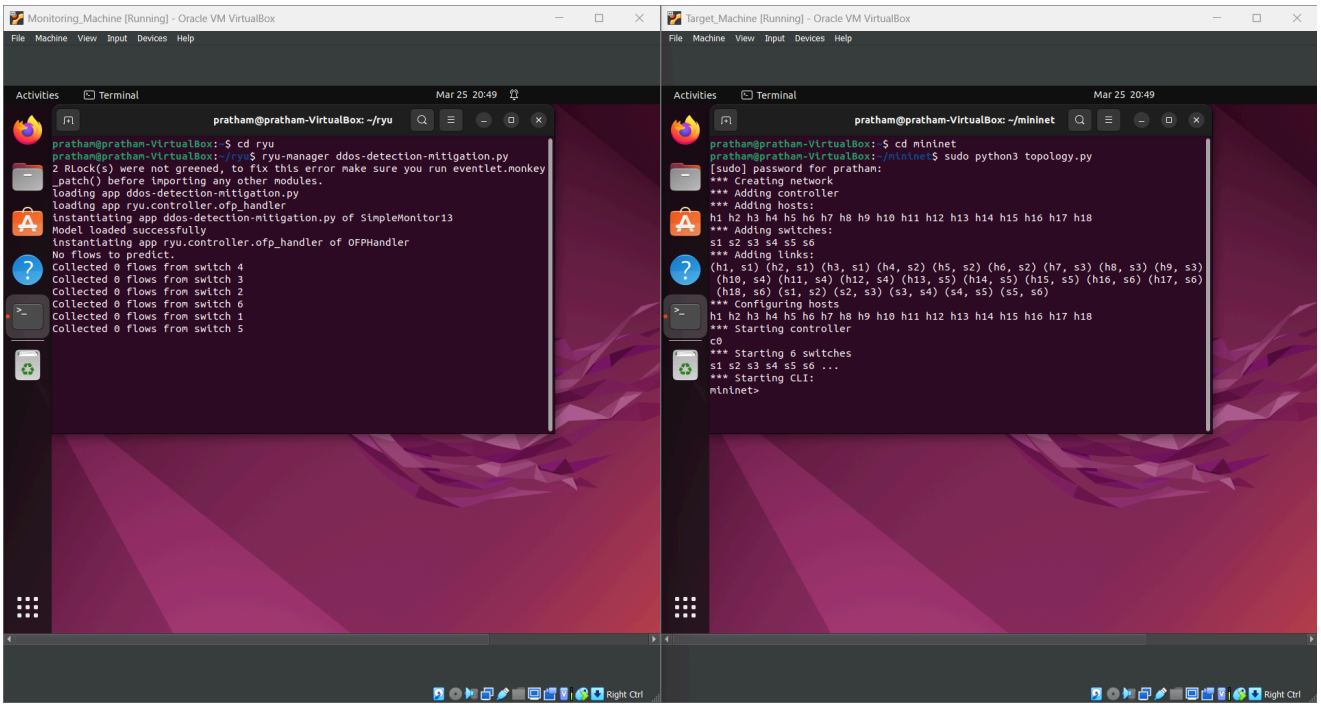


Fig 7.1: Connection of network topology with Ryu controller

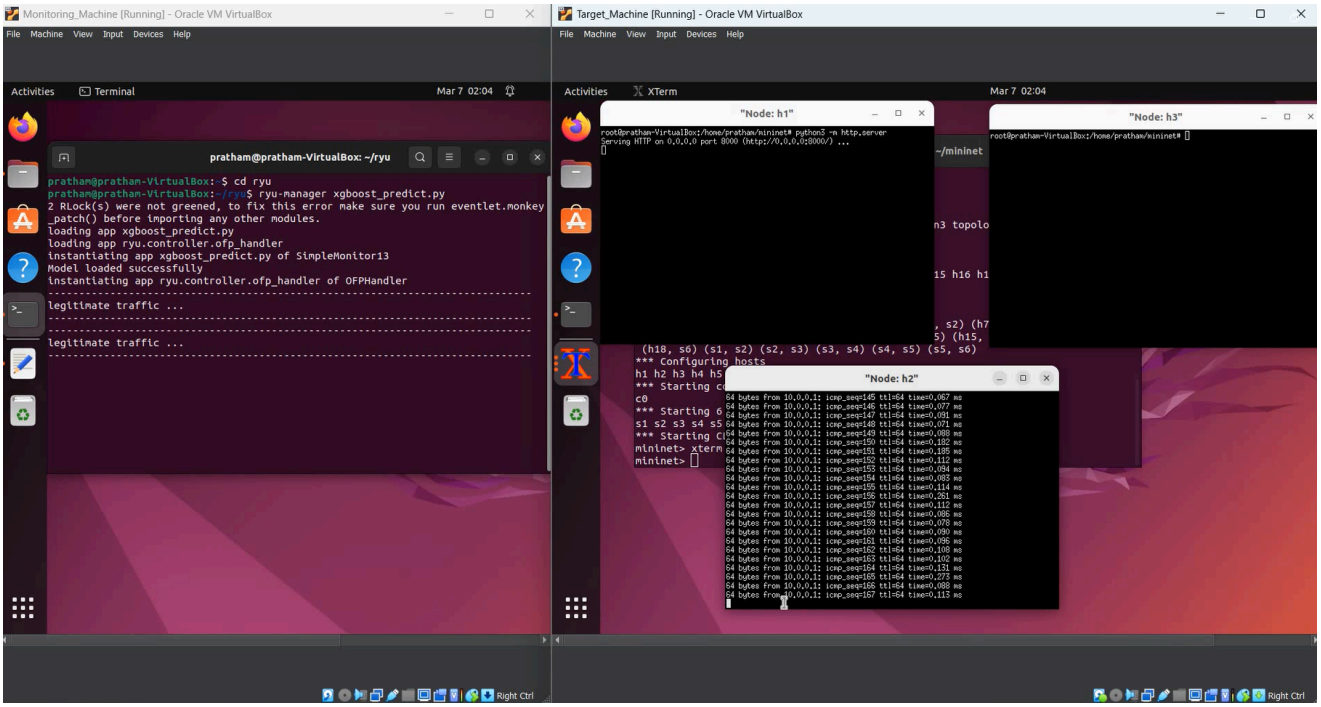


Fig 7.2: Detecting Legitimate Traffic using XGBoost

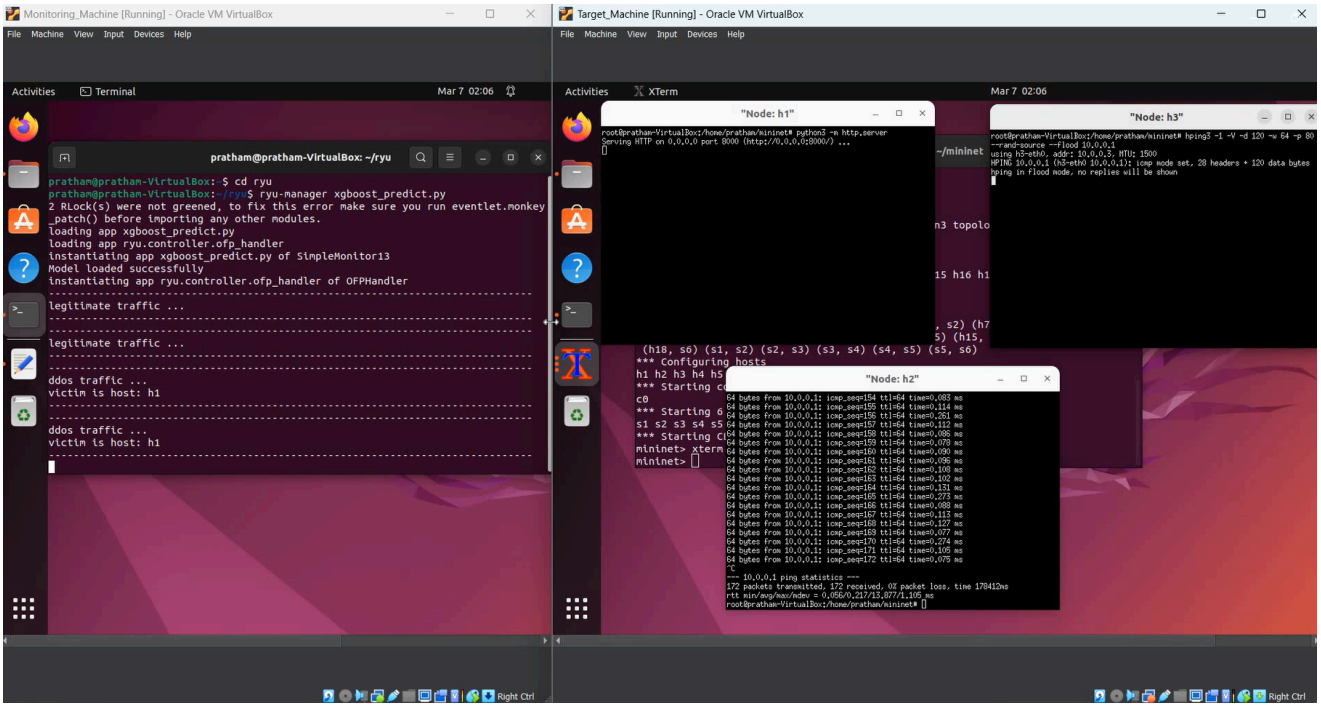


Fig 7.3: Detecting DDoS Traffic using XGBoost

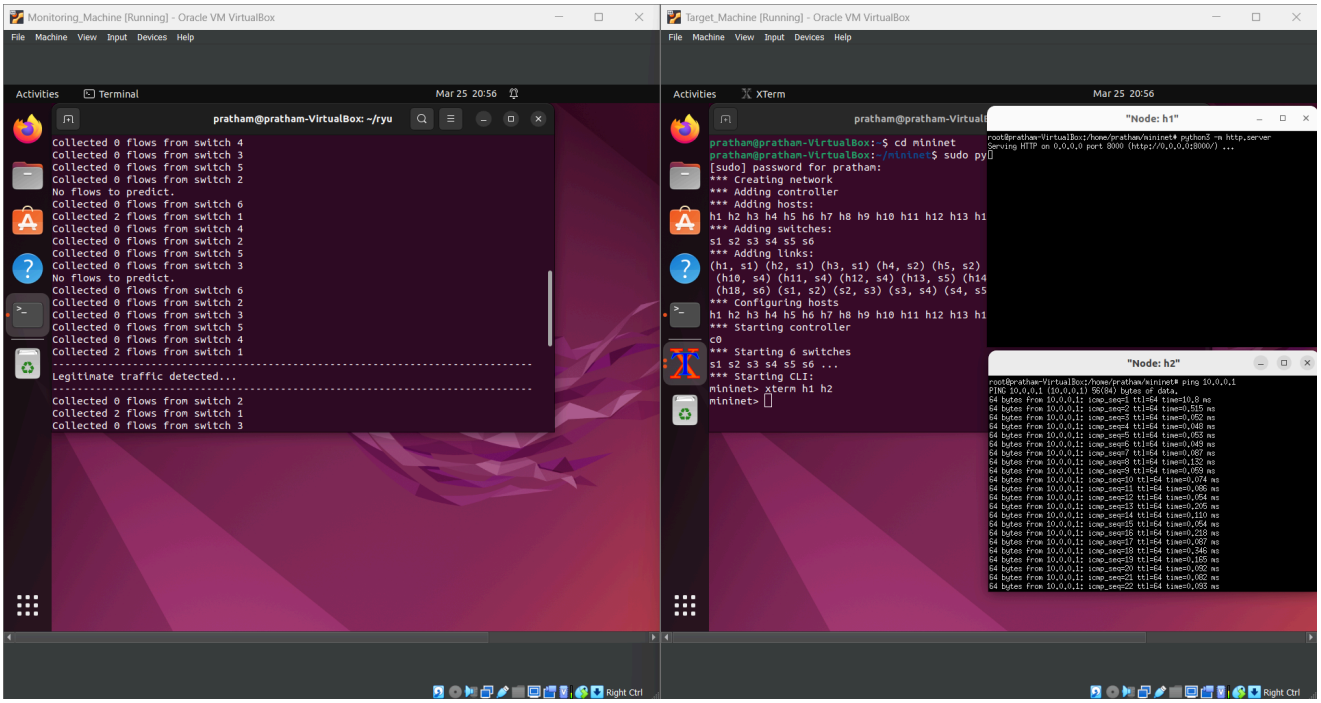


Fig 7.4: Response of Mitigation Module for Legitimate Traffic

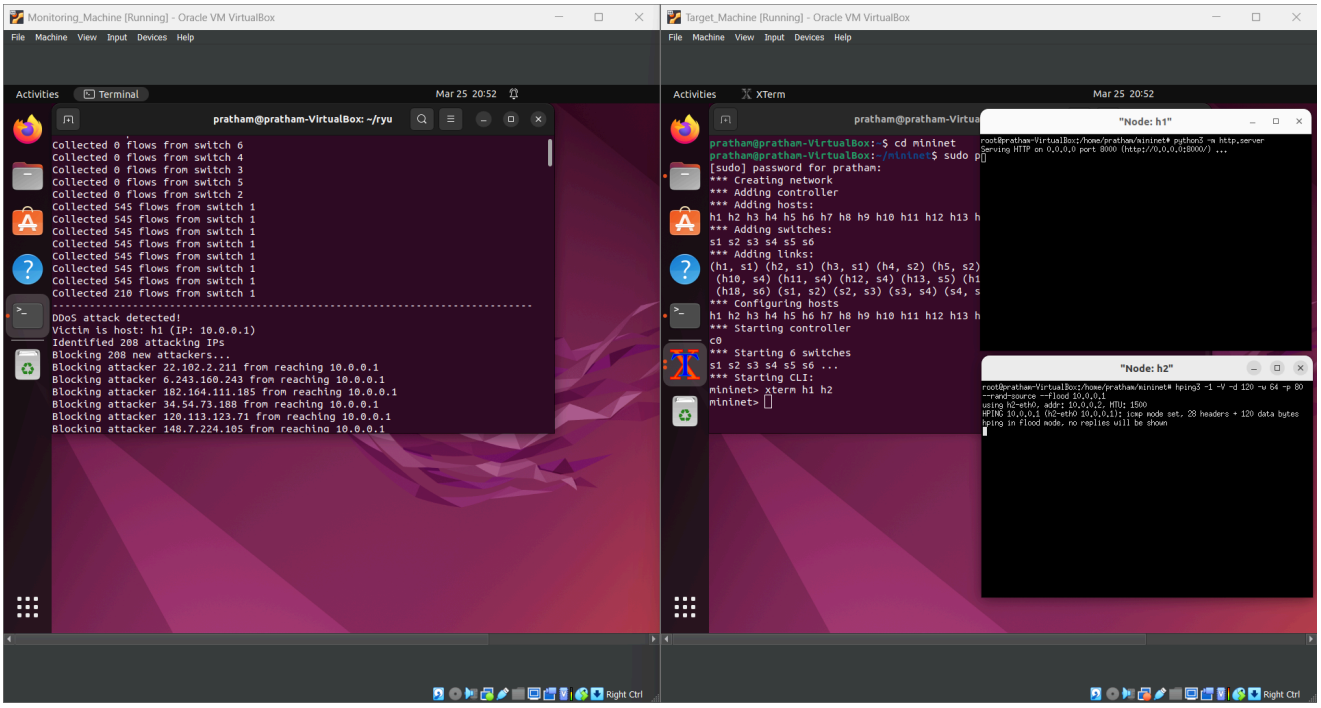


Fig 7.5: Response of Mitigation Module for DDoS Traffic

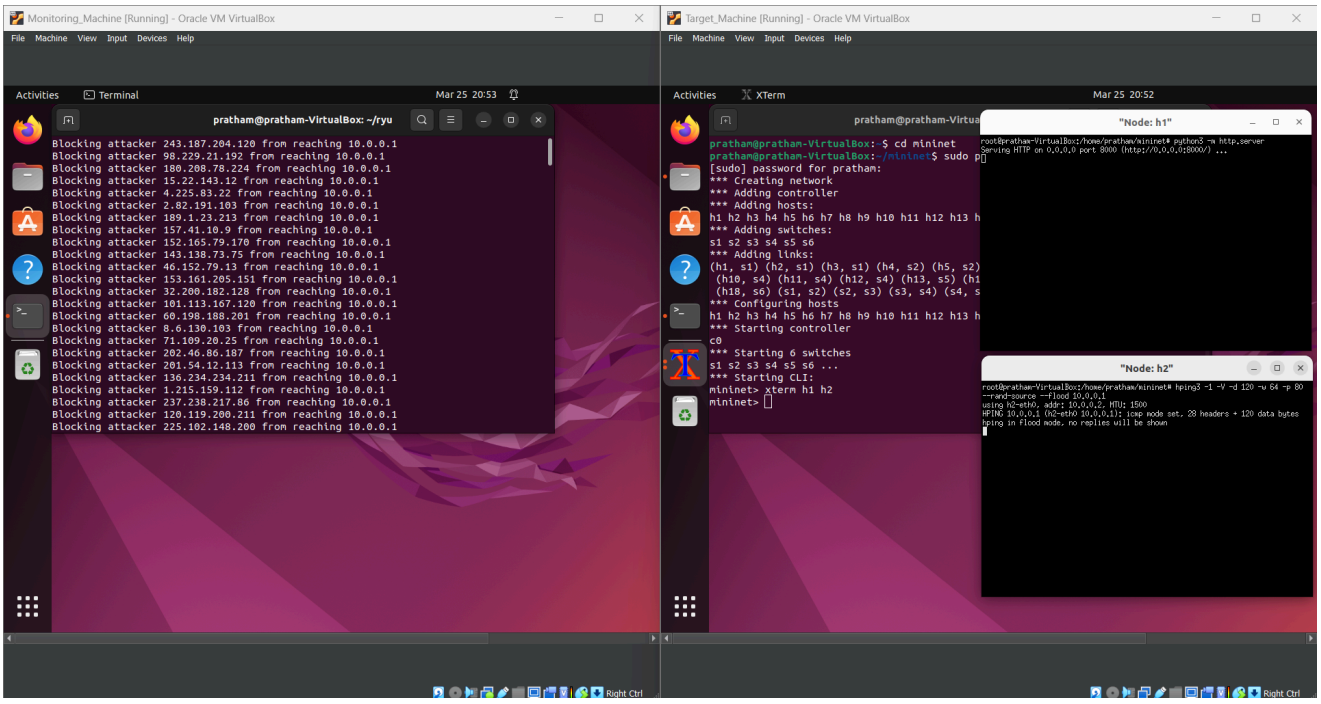


Fig 7.6: Action taken by Mitigation Module in case of DDoS Attack

7.2 Performance Evaluation Measures:

To evaluate the effectiveness of the machine learning models used for DDoS detection, the following performance metrics were considered:

1. **Precision:** Precision measures the accuracy of positive predictions. It is the ratio of correctly predicted positive observations (True Positives) to the total predicted positive observations (True Positives + False Positives). High precision indicates a low false positive rate.

   Formula:

$$\text{Precision} = \frac{TP}{TP + FP}$$

   Where:

   TP = True Positives

   FP = False Positives

2. **Recall:**

   Recall measures the model's ability to identify all relevant instances (i.e., all actual DDoS attacks). It is the ratio of correctly predicted positive observations to all observations in the actual class.

   Formula:

$$\text{Recall} = \frac{TP}{TP + FN}$$

   Where:

   TP = True Positives

   FN = False Negatives

3. **F1-Score:**

   The F1-score is the harmonic mean of precision and recall. It is especially useful when the dataset is imbalanced, as it provides a balanced measure of both false positives and false negatives.

   Formula:

   F1-Score=2*(Precision * RecallPrecision)/(Precision + Recall)

```
XGBoost Flow Training ...
-----------------------------------
Accuracy: 0.9990606945713293
Precision: 1.0
Recall: 0.9981194049075803
F1 Score: 0.9990588174621593
```

Fig 7.7: Precision, Recall and F1-Score of XGBoost Model

7.3 Input Parameters/Features considered :

The system extracts and uses a number of features from the network traffic to train the ML models and perform real-time classification. Some key input parameters include:

- Flow Duration – Time duration of a network flow.
- Total Fwd/Bwd Packets – Number of packets sent in the forward and backward direction.
- Packet Length Mean / Std / Max – Statistical data of packet lengths.
- Flow Bytes/s and Flow Packets/s – Byte and packet rates for each flow.
- Average Packet Size – Mean size of packets within a flow.
- Fwd/Bwd Packet Length Mean – Mean packet length in each direction.
- Protocol Type – Indicates the transport layer protocol (TCP, UDP, ICMP).
- Source and Destination IPs – Used for identifying traffic sources and potential attack patterns.

These features were either extracted from packet captures using tools like Wireshark/tcpdump or parsed directly in real-time by the Ryu controller.

7.4 Comparison of Results with Existing System :

Criteria	Existing Systems	Proposed System
Real-Time Detection	Limited or reactive	Supports real-time detection using SDN and ML
Automation	Manual or semi-automated mitigation	Fully automated detection and mitigation
Traffic Classification	Often signature-based	Software-defined, programmable using Ryu
Network Control	Static and hardware-dependent	Real-Time Detection
Testing Environment	Physical or pre-defined	Fully virtual using Mininet and VMs
Adaptability	Struggles with novel attacks	Model retraining and feature engineering supported

Table no. 7.1 : Comparison of results with existing system

7.5 Inference Drawn :

From the experiments and evaluations conducted, the following inferences can be drawn:

- Machine learning models, when trained with well-engineered features, can effectively detect various DDoS attack patterns.
- The integration of SDN enables dynamic and fine-grained control over network flows, allowing timely mitigation of threats.
- Compared to traditional IDS/IPS systems, this solution is more flexible, cost-effective, and adaptable to evolving attacks.
- Real-time visualization and logging empower network administrators with actionable insights.
- The modular nature of the system allows easy updates, scalability, and integration of newer models or detection mechanisms.

Chapter 8 : Conclusion

8.1 Limitations :

- The current system is tested primarily in a simulated environment using Mininet and may require further testing before real-world deployment.
- The mitigation process is reactive and depends on the accuracy and speed of the machine learning model.
- The system does not yet support multi-controller SDN environments, which limits scalability.
- Traffic classification models may face reduced accuracy when exposed to previously unseen or adversarial attack patterns.
- Continuous learning or online model updates are not yet implemented.

8.2 Conclusion :

This project addresses the pressing need for real-time detection and mitigation of Distributed Denial of Service (DDoS) attacks using Software-Defined Networking (SDN) and Machine Learning (ML). By leveraging the Ryu SDN controller and Mininet virtual network simulator, we were able to model realistic DDoS scenarios and build an intelligent system that dynamically identifies and reacts to malicious traffic.

The system captures real-time network traffic, extracts relevant features, and classifies it using trained ML models such as Random Forest and Deep Neural Networks. Upon detection of an attack, mitigation rules are dynamically deployed via the controller to the underlying switches, ensuring minimal service disruption. Our approach demonstrates how SDN, with its centralized control and programmability, can be combined with ML to create an adaptive and scalable security solution.

8.3 Future Scope :

1. Integration of online learning algorithms to adapt to new types of DDoS attacks in real time.
2. Deployment in a multi-controller SDN environment for better scalability and fault tolerance.
3. Incorporation of advanced deep learning models like LSTM or CNN for improved detection accuracy.
4. Extension of the system to detect other types of network attacks such as phishing, port scanning, and brute-force attacks.
5. Enhancement of the mitigation engine to include load balancing and traffic redirection.
6. Real-world deployment and testing on physical infrastructure or cloud environments.
7. Inclusion of encrypted traffic analysis using statistical or behavioral features.

References

[1] Kavitha, D., R. Ramalakshmi, and R. Murugeswari. "The detection and mitigation of distributed denial-of-service (DDOS) attacks in software defined networks using distributed controllers." 2019 IEEE International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable Development (INCCES). IEEE, 2019.

[2] Abdulkarem, Huda Saleh, and Ammar Dawod. "DDoS attack detection and mitigation at SDN data plane layer." 2020 2nd Global Power, Energy and Communication Conference (GPECOM). IEEE, 2020.

[3] Bharot, Nitesh, et al. "Mitigating distributed denial of service attack in cloud computing environment using threshold based technique." Indian Journal of Science and Technology 9.38 (2016): 1-7.

[4] Dharma, NI Gde, et al. "Time-based DDoS detection and mitigation for SDN controller." 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS). IEEE, 2015.

[5] Alashhab, Abdussalam A., et al. "Enhancing DDoS attack detection and mitigation in SDN using an ensemble online machine learning model." IEEE Access (2024).

[6] Khedr, Walid I., Ameer E. Gouda, and Ehab R. Mohamed. "FMDADM: A multi-layer DDoS attack detection and mitigation framework using machine learning for stateful SDN-based IoT networks." Ieee Access 11 (2023): 28934-28954.

[7] Azab, Ahmad, et al. "Network traffic classification: Techniques, datasets, and challenges." Digital Communications and Networks 10.3 (2024): 676-692.

[8] Hu, Yahui, et al. "Online network traffic classification based on external attention and convolution by IP packet header." Computer Networks 252 (2024): 110656.

[9] Najar, Ashfaq Ahmad, and S. Manohar Naik. "Cyber-secure SDN: A CNN-based approach for efficient detection and mitigation of DDoS attacks." Computers & Security 139 (2024): 103716.

[10] Bakar, Rana Abu, et al. "FTG-Net-E: A hierarchical ensemble graph neural network for DDoS attack detection." Computer Networks 250 (2024): 110508.

[11] Rahman, Md Mahmudur, Shanto Roy, and Mohammad Abu Yousuf. "DDoS mitigation and intrusion prevention in content delivery networks using distributed virtual honeypots." 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT). IEEE, 2019.

[12] Dayal, Neelam, and Shashank Srivastava. "FloodKnight: an intelligent DDoS defense scheme to combat attacks near attack entry points." Journal of Computer Virology and Hacking Techniques 20.4 (2024): 819-839.

[13] Shamekhi, Ali, Pirooz Shamsinejad Babaki, and Reza Javidan. "An intelligent behavioral-based DDOS attack detection method using adaptive time intervals." Peer-to-Peer Networking and Applications 17.4 (2024): 2185-2204.

[14] Wei, Y., and J. Liu, "DDoS Attack Detection Using Machine Learning and Deep Learning Approaches," Scientific Reports, vol. 14, 2024.

[15] Najar, Ashfaq Ahmad, and S. Manohar Naik. "DDoS attack detection using MLP and Random Forest Algorithms." International Journal of Information Technology 14.5 (2022): 2317-2327.

[16] Sumantra, I., and S. Indira Gandhi. "DDoS attack detection and mitigation in software defined networks." 2020 International Conference on System, Computation, Automation and Networking (ICSCAN). IEEE, 2020.

Appendix

1.Paper 1 Details :-

a. Paper 1 :

Real-Time Detection and Mitigation of DDoS Attacks using Network Traffic Classification

Aaryan Mahadik
BE, Computer Engg
VESIT, Chembur
2021.aaryan.mahadik@ves.ac.in

Geocey Shejy
Faculty, CMPN Dept
VESIT, Chembur
geocey.shejy@ves.ac.in

Pratham Shetty
BE, Computer Engg
VESIT, Chembur
2021.pratham.shetty@ves.ac.in

Neha Lotwani
BE, Computer Engg
VESIT, Chembur
d2021.neha.lotwani@ves.ac.in

Himesh Hotwani
BE, Computer Engg
VESIT, Chembur
2021.himesh.hotwani@ves.ac.in

***Abstract* — Cybersecurity threats have become increasingly sophisticated, with Distributed Denial-of-Service (DDoS) attacks posing one of the most disruptive challenges to modern networks. These attacks overwhelm target systems with an immense volume of malicious traffic, leading to service disruptions, financial losses, and significant operational damage. Traditional DDoS detection and mitigation methods, which rely on static rule-based approaches, often fail to adapt to evolving attack patterns and zero-day threats. As a result, intelligent, adaptive, and real-time detection mechanisms have become a critical necessity for securing network infrastructures. This research introduces a real-time DDoS detection and mitigation framework using network traffic classification. By leveraging machine learning and Software-Defined Networking (SDN), the proposed system aims to improve the accuracy and efficiency of attack detection and response. The framework continuously monitors network traffic, extracts relevant flow-based features, and employs intelligent classification techniques to differentiate between legitimate and malicious traffic. Upon detecting an attack, dynamic mitigation strategies are applied to contain and neutralize the threat while ensuring minimal disruption to normal network operations. The findings demonstrate the potential of machine learning-powered SDN security mechanisms in offering faster, more reliable, and adaptive attack prevention compared to conventional detection systems. By bridging the gap between network intelligence and automated security responses, this research contributes to the advancement of next-generation cybersecurity solutions. The proposed approach enhances real-time threat detection and provides a proactive defense strategy, reinforcing network resilience against ever-evolving cyber threats.**

Keywords — DDoS detection, Machine learning, SDN, Network security, Traffic classification

I. Introduction

With the exponential growth of internet services, network security has become a critical concern. Among various cyber threats, Distributed Denial-of-Service (DDoS) attacks pose one of the most significant challenges to organizations and individuals. These attacks disrupt the availability of online services by overwhelming the target system with an enormous volume of malicious traffic, rendering it inaccessible to legitimate users. The increasing complexity of DDoS attacks, ranging from volumetric floods to sophisticated application-layer intrusions, has made traditional defense mechanisms ineffective. Conventional rule-based intrusion detection systems (IDS) struggle to keep up with rapidly evolving attack techniques, leading to high false positive rates and delayed mitigation responses. As a result, there is an urgent need for intelligent, adaptive, and real-time detection mechanisms to safeguard critical infrastructure against such threats.

Machine learning (ML) has emerged as a promising solution for network traffic classification, providing the ability to learn attack patterns from data and differentiate between legitimate and malicious traffic. Unlike static rule-based approaches, ML models can continuously evolve and adapt to new attack methods, significantly improving detection accuracy and response time. By leveraging supervised learning techniques, ML-based classifiers can analyze historical network traffic, identify anomalies, and classify potential threats in real-time. This proactive approach helps minimize service downtime, data breaches, and financial losses associated with DDoS attacks.

In this research, a real-time DDoS detection and mitigation framework is developed using network traffic classification. Several machine learning models, including XGBoost, Random Forest, CART, K-Nearest Neighbors (KNN), and different variations of Support Vector Machines (SVM - Linear, LinearSVC, and RBF kernel), are explored for their effectiveness in identifying malicious traffic. A comparative analysis of these models is conducted based on key evaluation metrics such as accuracy, precision, recall, and F1-score. The results indicate that while multiple models achieve high classification performance, XGBoost outperforms the others, demonstrating superior accuracy and precision in distinguishing between legitimate and attack traffic.

The key contributions of this research include the development of a scalable and efficient ML-based DDoS detection system that not only classifies network traffic in real-time but also ensures rapid mitigation of identified threats. The study emphasizes feature engineering and dataset preprocessing to enhance the efficiency and accuracy of the detection system. By integrating ML-based traffic classification with automated mitigation mechanisms, the proposed framework aims to provide a robust and adaptive solution for protecting modern network infrastructures such as cloud environments, enterprise networks, and Software-Defined Networking (SDN) architectures.

The findings of this research highlight the potential of AI-driven cybersecurity solutions in combating modern DDoS attacks. By bridging the gap between machine learning advancements and practical cybersecurity needs, this study contributes to the development of intelligent, real-time defense mechanisms that can enhance network resilience, minimize attack impact, and ensure seamless service availability.

II. Related Methodology

Recent advancements in DDoS detection have leveraged Software-Defined Networking (SDN) architectures and machine learning-based classification techniques to identify and mitigate cyber threats in real-time. Traditional DDoS detection approaches have often relied on signature-based or threshold-based methods, which struggle to adapt to evolving attack patterns and zero-day threats. As a result, researchers have increasingly focused on machine learning and deep learning models to enhance detection accuracy, reduce false positives, and improve response times in real-world network environments.

One of the earliest methods for DDoS detection in SDN environments involved distributed controllers that collaborated to identify and counteract malicious traffic. Kavitha et al. [1] proposed a multi-controller SDN framework to detect anomalies in network traffic. Their method focused on analyzing packet header information to classify benign and attack traffic efficiently. Similarly, Abdulkarem et al. [2] introduced a mitigation strategy operating at the SDN data plane, ensuring minimal performance overhead while blocking attack traffic in real-time. The threshold-based approach for mitigating DDoS in cloud computing environments has also been explored. Bharot et al. [3] developed a system where predefined traffic thresholds determined the classification of legitimate and malicious traffic. This technique, while effective, suffers from a lack of adaptability to evolving attack patterns. In contrast, Dharma et al. [4] proposed a time-based detection method that evaluates packet arrival intervals, helping to distinguish between normal traffic bursts and actual DDoS events.

Recent advancements in machine learning have significantly improved DDoS detection accuracy. Alashhab et al. [5] demonstrated an ensemble-based online learning approach that dynamically adjusts to new attack patterns, achieving superior performance compared to traditional static classifiers. Similarly, Khedr et al. [6] proposed a multi-layer DDoS detection framework that combines supervised learning techniques with deep packet inspection, effectively reducing false positives in SDN-based IoT networks. Another area of research focuses on feature selection and network traffic classification for better detection accuracy. Azab et al. [7] provided a comprehensive survey on network traffic classification techniques, highlighting the advantages of statistical, flow-based, and machine learning-based classification. Hu et al. [8] further explored the integration of external attention mechanisms with convolutional neural networks to enhance real-time traffic classification accuracy.

Deep learning approaches have also been investigated for their potential in mitigating DDoS attacks. Najar et al. [9] introduced a convolutional neural network (CNN)-based model to detect DDoS patterns in SDN traffic, outperforming traditional machine learning models. Bakar et al. [10] expanded on this by incorporating a hierarchical ensemble graph neural network, which improved attack detection in large-scale distributed environments. Intrusion prevention systems (IPS) and virtual honeypots have been employed to mitigate the effects of DDoS attacks. Rahman et al. [11] proposed a honeypot-based defense mechanism in content delivery networks, effectively diverting attack traffic away from critical infrastructure. Dayal et al. [12] developed an intelligent defense scheme that detects DDoS attacks near the attack entry points, allowing proactive mitigation before the attack spreads further into the network.

Behavioral-based detection models have also been explored. Shamekhi et al. [13] introduced an adaptive time-interval-based approach that analyzes traffic behavior over time to detect anomalous patterns. Similarly, Wei et al. [14] leveraged a hybrid machine learning and deep learning

framework, incorporating both statistical and behavior-based features to classify network flows accurately. Various classifiers have been evaluated for their effectiveness in DDoS detection. Najar et al. [15] compared multi-layer perceptron (MLP) and Random Forest (RF) models, concluding that MLP provides better generalization for attack detection. Sumantra et al. [16] focused on SDN-based mitigation strategies, utilizing entropy-based measurements to differentiate between normal and attack traffic.

Despite these advancements, challenges remain in real-time detection and mitigation of DDoS attacks in SDN environments. The proposed research builds upon these prior works by integrating a real-time monitoring system with an optimized XGBoost classifier. Our approach aims to minimize false positives while ensuring rapid response to evolving attack vectors. Unlike existing solutions, our system not only detects attacks but also implements automated mitigation strategies within the SDN controller to maintain network stability.

III. Proposed Methodology

The tools being used in this project include two virtual machines running Ubuntu 22.04 LTS:

1. Target Virtual Machine (VM): Simulates a network topology using Mininet, allowing controlled traffic flow between nodes.
2. Monitoring Virtual Machine (VM): Runs the Ryu SDN Controller, which manages the packet flow in the network and collects network traffic data.

For DDoS attack generation, we use hping3, which floods the network with malicious packets. We leverage Python scripts to automate both traffic generation and collection:

- The mininet library is used to generate both benign and malicious traffic in a structured manner.
- The ryu.controller library is used to monitor network activity and collect relevant traffic data for analysis.

Machine Learning models are trained using the collected data to distinguish between legitimate and DDoS traffic, aiding in real-time mitigation strategies. Figure 1 presents a flowchart illustrating the steps involved in the proposed methodology.

To assess the effectiveness of the machine learning models in accurately distinguishing between legitimate and DDoS traffic, several standard evaluation metrics are employed. These include Accuracy, which measures the overall correctness of the model; Precision, which indicates the proportion of correctly identified DDoS instances among all predicted DDoS cases; Recall, which reflects the model's ability to detect all actual DDoS instances; and the F1-Score, which provides a balanced metric combining both precision and recall. These metrics ensure a comprehensive evaluation of the classifier's performance, especially in scenarios with class imbalance, which is common in network traffic datasets.

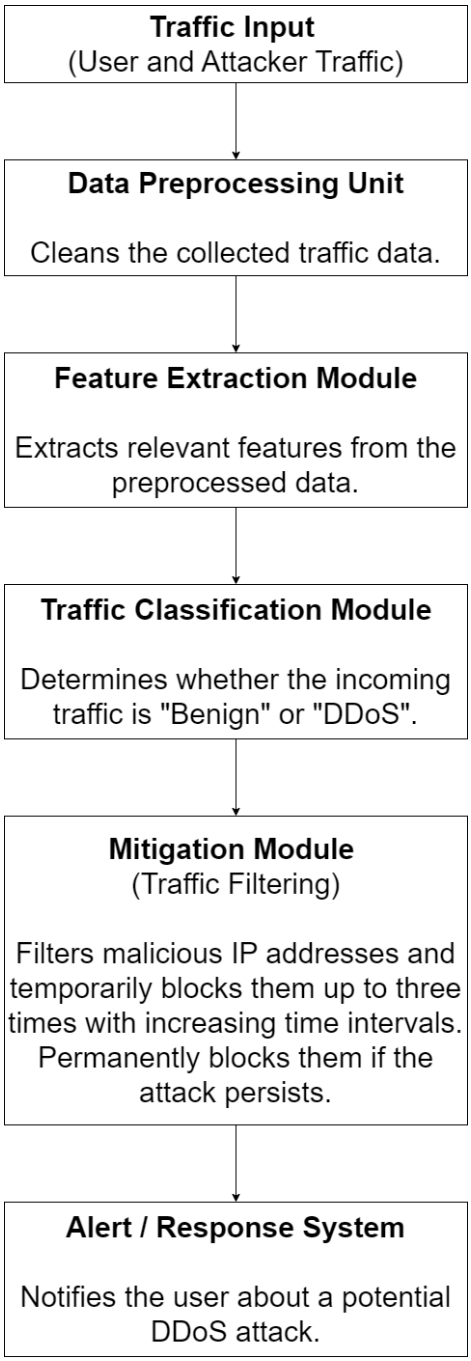


Figure 1: Overall Proposed Methodology

A. Data Generation and Collection

To build a reliable and diverse dataset for training and evaluating our DDoS detection model, we designed a controlled experimental environment using Mininet and

SDN (Software Defined Networking) with Ryu as the controller framework. This environment allowed us to simulate and monitor both benign and DDoS traffic under consistent network conditions.

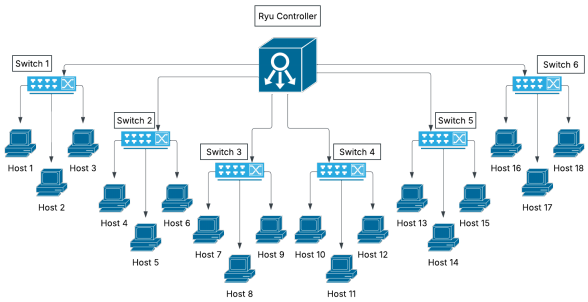


Figure 2: Custom Topology

A custom topology was constructed in Mininet consisting of six OpenFlow-enabled switches and fifteen hosts. These hosts were strategically distributed across switches to simulate real-world network segmentation and to ensure a wide variety of traffic paths. The topology supported the generation of both legitimate (benign) traffic and coordinated DDoS traffic.

For benign traffic generation, normal network communication patterns were established among various hosts. These patterns included typical TCP and UDP-based traffic flows such as pings, file transfers, and application-level interactions. On the other hand, DDoS traffic was simulated by orchestrating multiple hosts to simultaneously send a high volume of traffic to a single victim host, thus mimicking a volumetric attack scenario.

This simulated attack environment enabled us to create realistic network conditions for both normal operation and attack scenarios, giving our dataset strong representational quality.

To monitor the network and extract relevant traffic features, we implemented an SDN application that operated on the Ryu controller. The application periodically collected OpenFlow flow statistics from all registered datapaths in the network. These statistics included information such as:

- Source and destination IP addresses
- Transport layer ports (TCP/UDP)
- IP protocol types (e.g., ICMP, TCP, UDP)
- Flow duration (in seconds and nanoseconds)
- Total packet and byte counts
- ICMP-specific fields (type and code, when applicable)

From these raw values, additional derived features were computed, including:

- Packet count per second
- Packet count per nanosecond
- Byte count per second
- Byte count per nanosecond

B. Attribute Information

The dataset consists of flow-level features extracted from OpenFlow statistics, designed to capture the behavior of network flows during both benign and DDoS scenarios. Key features include:

- src_ip and dst_ip: Source and destination IP addresses identifying the endpoints of the flow.
- src_port and dst_port: Transport-layer port numbers used in the flow.
- protocol: IP protocol used (e.g., TCP, UDP, ICMP).
- duration_sec and duration_nsec: Flow duration in seconds and nanoseconds.
- packet_count and byte_count: Total number of packets and bytes transmitted in the flow.
- icmp_type and icmp_code: ICMP-specific fields present only for ICMP flows.
- pkts_per_sec and pkts_per_nsec: Derived metrics representing packet transmission rate.
- bytes_per_sec and bytes_per_nsec: Derived metrics representing byte transmission rate.

C. Model Training

Our system was trained using various machine learning algorithms, namely:

1) *XGBoost (Extreme Gradient Boosting):* XGBoost is a powerful and efficient implementation of gradient boosting that builds an ensemble of decision trees in a sequential manner. Each tree corrects the errors of the previous ones, allowing the model to learn complex patterns in the data. XGBoost uses both first and second-order gradients (hence the term "gradient boosting") to optimize its performance and prevent overfitting through regularization techniques. It is widely known for its scalability, speed, and accuracy, making it a popular choice in structured/tabular data tasks such as DDoS detection.

$$l = \sum_{i=1}^{n} l(y_i, y_i^{(t)}) + \sum_{k=1}^{t} \Omega(f_k) \quad (1.1)$$

2) *K-Nearest Neighbors (KNN):* K-Nearest Neighbors is a simple, non-parametric, instance-based learning algorithm. It works by comparing a given data point to its **k** closest neighbors in the feature space using a distance metric (typically Euclidean distance). The predicted class is determined by a majority vote among these neighbors. KNN is intuitive and effective for classification tasks where the decision boundary is non-linear. However, it

can be computationally expensive for large datasets and is sensitive to feature scaling and irrelevant features, which is why preprocessing is important when using KNN.

$$d(x, x') = \sqrt{\sum_{i=1}^{n} (x_i - x'_i)^2} \quad (2.1)$$

3) *Support Vector Machine (SVM):* Support Vector Machine (SVM) is a supervised machine learning algorithm that is particularly effective for classification tasks, including network traffic analysis. In this project, SVM is employed to classify network flows as either normal or indicative of a DDoS attack. The algorithm works by finding the optimal hyperplane that separates data points of different classes with the maximum margin. In cases where the data is not linearly separable, SVM uses kernel functions to project the data into higher-dimensional space, enabling it to draw more complex decision boundaries. This ability to handle high-dimensional data makes SVM well-suited for intrusion detection systems, where subtle patterns in features like packet rates, byte flows, and source/destination addresses can help distinguish between benign and malicious activity.

$$f(x) = sign(\sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b) \quad (3.1)$$

D. Detection Module

The detection module is a critical component of the proposed system, responsible for identifying DDoS attacks in real time by monitoring network traffic, extracting relevant features, and classifying flows based on traffic behavior. Implemented within the Ryu SDN controller, this module continuously collects flow statistics from OpenFlow-enabled switches and processes them to distinguish between benign and malicious traffic. By leveraging event-driven packet monitoring, the detection module operates efficiently without introducing excessive computational overhead, ensuring rapid response to potential threats.

The process begins with continuous traffic monitoring, where the SDN controller registers active switches and maintains a list of datapaths. Whenever a new switch connects, it is added to the monitoring list, and if a switch disconnects, it is removed to optimize performance. The detection module utilizes a background monitoring thread that periodically requests flow statistics from all active switches. The FlowStatsRequest messages are sent every 10 seconds, and the switches respond with FlowStatsReply messages containing detailed traffic metrics such as source and destination IP addresses, transport protocol type (TCP, UDP, ICMP), source and destination ports, packet count, byte count, and flow duration. These statistics allow the system to analyze traffic patterns and detect anomalous behavior indicative of a DDoS attack.

After collecting flow statistics, the detection module proceeds with feature extraction and data structuring. It processes and computes key metrics such as packet count per second and byte count per second, which help characterize network behavior. A unique flow identifier is generated using a combination of IP addresses, transport protocol, and port numbers, ensuring that each flow is uniquely tracked. If the protocol is ICMP, additional parameters such as ICMP type and code are recorded. The extracted data is then formatted and stored in a csv file, where each network event is logged for real-time classification and historical analysis.

The classification logic of the detection module is based on analyzing the overall traffic distribution. Once the network flows are recorded, the system evaluates the proportion of legitimate vs. malicious traffic based on past statistics. If 80% or more of the total network traffic is classified as legitimate, the system assumes that the network is functioning normally, and no further action is taken. However, if the percentage of malicious traffic exceeds 20%, the system flags it as a potential DDoS attack. This threshold-based approach ensures low false positive rates, preventing unnecessary disruptions to legitimate users while maintaining high sensitivity to abnormal traffic surges.

The detection module operates in real time, ensuring that suspicious activity is identified before it significantly impacts the network. Unlike traditional signature-based detection methods that rely on predefined attack patterns, this module uses a dynamic classification approach that adapts to evolving attack behaviors. Future improvements could involve integrating advanced machine learning classifiers to enhance detection accuracy, allowing the system to automatically learn new attack patterns. Additionally, employing online learning techniques could enable continuous model updates, further strengthening the system's ability to detect zero-day DDoS attacks. Through real-time traffic monitoring and adaptive classification, the detection module plays a crucial role in identifying threats efficiently and enabling a proactive security response.

E. Mitigation Module

Once a potential DDoS attack is detected, the mitigation module is activated to minimize its impact and ensure service availability while preventing unnecessary disruptions to legitimate users. Implemented within the Ryu SDN controller, this module enforces dynamic flow rules in OpenFlow-enabled switches, allowing for adaptive and progressive attack mitigation. Rather than immediately blocking an IP address upon detection, the system adopts a

tiered blocking strategy to differentiate between temporary traffic anomalies and persistent attackers.

When an IP address is flagged for suspicious behavior, it is temporarily blocked for 5 minutes on the first offense. This initial block acts as a warning mechanism, preventing the attacker from immediately overwhelming network resources while allowing for the possibility of false positives. If the same IP continues to exhibit malicious traffic patterns, the second detection results in a 10-minute block, followed by a 15-minute block on the third offense. If the attack persists beyond three warnings, the IP is permanently blacklisted, preventing further access to the network. This escalating timeout mechanism serves as both a preventive and corrective measure, allowing temporary anomalies to self-correct while ensuring that persistent threats are neutralized.

To implement this strategy, the mitigation module dynamically modifies OpenFlow rules in the SDN switches. When an IP is temporarily flagged, a drop rule with a time-based expiration is installed, ensuring that the block is lifted after the specified duration. If the same attacker is detected again, the system updates the rule with a longer timeout, progressively increasing the severity of the response. On the fourth detection, a permanent drop rule is enforced, preventing any further traffic from the malicious source. Additionally, the module maintains a blacklist of permanently blocked IPs, ensuring that repeat offenders cannot regain access.

The mitigation module continuously adapts to network conditions, dynamically adjusting its response based on attack severity and frequency. Unlike traditional static blocking mechanisms, which may either be too lenient or overly aggressive, this progressive blocking approach balances security and accessibility, reducing the risk of false positives while effectively stopping persistent attackers. Future enhancements could include a threat scoring system, where each IP is assigned a risk level based on the frequency and intensity of its attacks, allowing for even more refined mitigation strategies. By integrating real-time threat response, adaptive blocking, and SDN-based flow control, the proposed mitigation module ensures a resilient and intelligent defense against DDoS attacks.

IV. Result Finding And Analysis

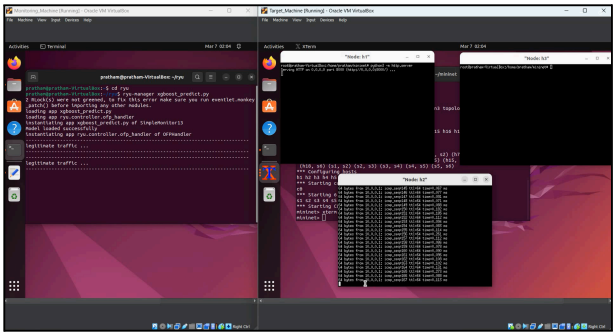
To evaluate the performance of our proposed approach, we trained our dataset using multiple machine learning algorithms, including XGBoost, Random Forest, CART, KNN, and different variants of Support Vector Machines (SVM). The dataset was divided into training and testing subsets, with training data allocation varying across models. The classification performance of each algorithm was assessed using key metrics such as Accuracy, Precision, Recall, and F1 Score. The comparative analysis of these models is presented in Table 1, highlighting the effectiveness of each classifier in distinguishing between DDoS traffic and benign traffic. Among the tested models, XGBoost and Random Forest demonstrated superior performance, achieving near-perfect classification results.

Algorithm chosen	Accuracy %	Precision %	Recall %	F1 Score
XGBoost	99.99935664	100	99.99871192	99.99935596
Random Forest	99.99935664	100	99.99871192	99.99935596
CART	99.998713	99.998713	99.998713	99.998713
KNN	99.66	99.66	99.66	99.66
SVM (Linear)	97.32	95.04	99.84	97.38
SVM (LinearSVC)	97.17	94.84	99.78	97.24
SVM (rbf)	99.68	99.37	99.99	99.68

Table1: Comparative table of the machine learning algorithms applied to the data set

This evaluation compares machine learning models for DDoS attack detection based on training data allocation, accuracy, precision, recall, and F1 score. XGBoost, Random Forest, and CART achieve the highest accuracy (99.99%), while KNN (99.66%) and SVM models show varied results, with SVM (Linear) at 97.32%, SVM (Linear SVC) at 97.17%, and SVM (RBF) at 99.68%.

XGBoost and Random Forest also achieve perfect precision and recall (100%), while KNN and SVM models exhibit slight drops. The F1 score follows a similar trend, with tree-based models performing best. Overall, tree-based models outperform SVM models in accuracy and precision.



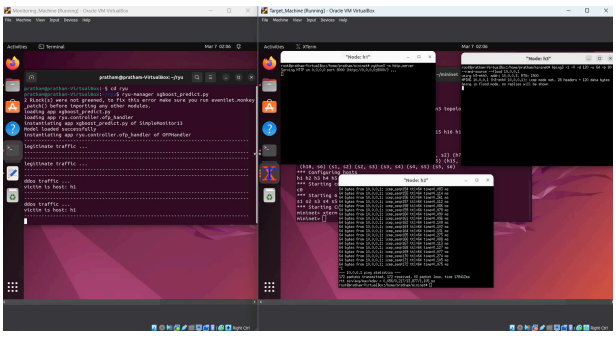


Figure 8: Using XGBoost

The XGBoost-based model classifies network traffic in real time, distinguishing between benign and malicious packets. The system continuously monitors incoming packets, accurately identifying DDoS attacks as they occur. When an attack is detected, the classifier flags it, providing instant alerts. This stage demonstrates the effectiveness of XGBoost in achieving high detection accuracy while maintaining minimal false positives.

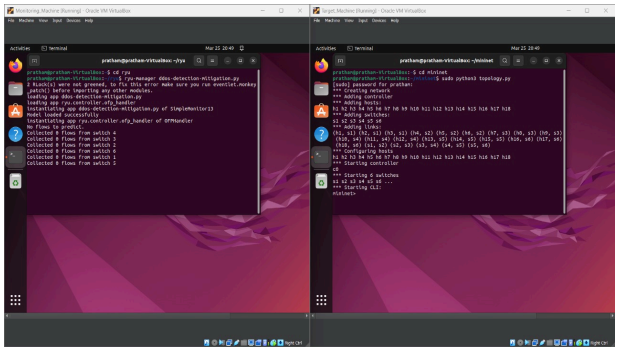


Figure 9: Connection of network topology with ryu controller

The network topology is established using Mininet, where multiple hosts and switches are interconnected. The Ryu controller is successfully launched and integrated with the topology to handle network flow management. The controller initializes and begins monitoring traffic flows from connected switches, enabling real-time flow collection for further analysis and DDoS detection.

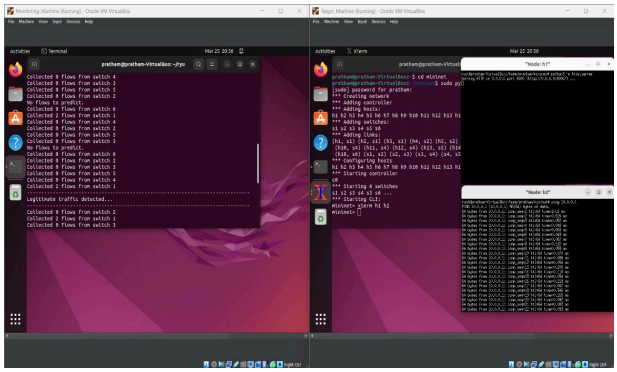


Figure 10: Legitimate traffic

The setup involves a network simulation using Mininet and the Ryu SDN controller. The controller collects flow data from multiple switches, analyzing traffic patterns. Mininet is configured with multiple hosts and switches, facilitating communication. A node runs an HTTP server while another sends ICMP ping requests, demonstrating legitimate traffic detection.

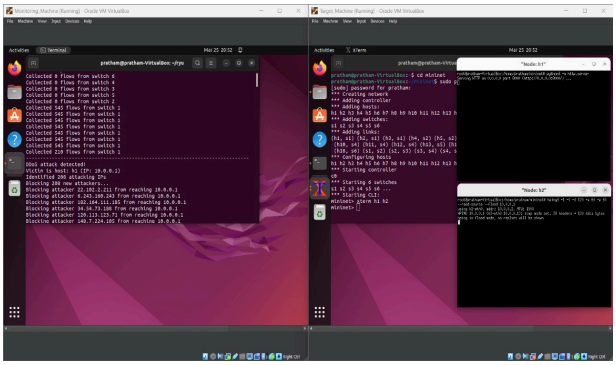


Figure 11: DDoS Traffic

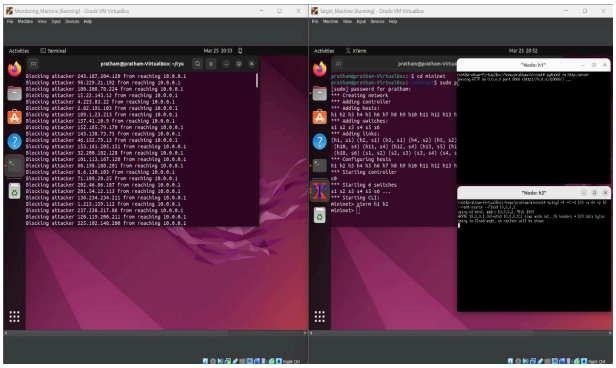


Figure 12: DDoS Attack Mitigation

A real-time DDoS attack detection and mitigation system is deployed in a virtualized network using Mininet and a monitoring mechanism. The network simulation detects and mitigates a DDoS attack targeting host h1 (10.0.0.1), where an HTTP server runs on one node while another generates high-rate ICMP flood traffic. The controller identifies 208 attacking IPs and dynamically blocks them. The monitoring component continuously logs malicious traffic and enforces mitigation by filtering attackers, ensuring network stability amid the ongoing flood attack scenario.

V. Concluding Remarks And Future Enhancements

In conclusion, the comparative analysis of machine learning models for real-time DDoS detection and mitigation in SDN environments reveals that XGBoost outperforms other classifiers in terms of accuracy, computational efficiency, and adaptability to network traffic variations. While KNN and SVM showed promising results in classifying network flows, their performance was affected by high-dimensional data and increased computational complexity. Random Forest and Decision Trees, on the other hand, provided good interpretability but lacked the speed and precision needed for real-time attack mitigation. The integration of XGBoost with an adaptive SDN-based mitigation strategy enabled faster threat response and effective mitigation of DDoS attacks while minimizing disruptions to legitimate users.

To further enhance the performance of the proposed system, advanced deep learning models, such as CNNs and LSTMs, could be explored to capture complex temporal patterns in network traffic, improving detection accuracy against evolving attack strategies. Additionally, implementing semi-supervised and online learning techniques could allow the model to adapt dynamically without requiring periodic retraining, making it more effective against zero-day attacks.

Future work could also focus on improving mitigation strategies by incorporating dynamic rerouting, rate-limiting adjustments, and deception-based techniques such as honeypots to further minimize attack impact. The integration of multi-controller SDN architectures could enhance the scalability of the system, making it suitable for large-scale enterprise and cloud-based environments. Furthermore, leveraging blockchain for trust management in SDN security policies could improve data integrity and prevent adversarial attacks on traffic classification models.

Overall, the proposed machine learning-driven DDoS detection and mitigation system demonstrates high efficiency, real-time responsiveness, and adaptability in securing SDN networks. The findings of this research serve as a foundation for future AI-driven cybersecurity advancements, ensuring robust network security against increasingly sophisticated cyber threats.

References

[1] Kavitha, D., R. Ramalakshmi, and R. Murugeswari. "The detection and mitigation of distributed denial-of-service (DDOS) attacks in software defined networks using distributed controllers." 2019 IEEE International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable Development (INCCES). IEEE, 2019.

[2] Abdulkarem, Huda Saleh, and Ammar Dawod. "DDoS attack detection and mitigation at SDN data plane layer." 2020 2nd Global Power, Energy and Communication Conference (GPECOM). IEEE, 2020.

[3] Bharot, Nitesh, et al. "Mitigating distributed denial of service attack in cloud computing environment using threshold based technique." Indian Journal of Science and Technology 9.38 (2016): 1-7.

[4] Dharma, NI Gde, et al. "Time-based DDoS detection and mitigation for SDN controller." 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS). IEEE, 2015.

[5] Alashhab, Abdussalam A., et al. "Enhancing DDoS attack detection and mitigation in SDN using an ensemble online machine learning model." IEEE Access (2024).

[6] Khedr, Walid I., Ameer E. Gouda, and Ehab R. Mohamed. "FMDADM: A multi-layer DDoS attack detection and mitigation framework using machine learning for stateful SDN-based IoT networks." Ieee Access 11 (2023): 28934-28954.

[7] Azab, Ahmad, et al. "Network traffic classification: Techniques, datasets, and challenges." Digital Communications and Networks 10.3 (2024): 676-692.

[8] Hu, Yahui, et al. "Online network traffic classification based on external attention and convolution by IP packet header." Computer Networks 252 (2024): 110656.

[9] Najar, Ashfaq Ahmad, and S. Manohar Naik. "Cyber-secure SDN: A CNN-based approach for efficient detection and mitigation of DDoS attacks." Computers & Security 139 (2024): 103716.

[10] Bakar, Rana Abu, et al. "FTG-Net-E: A hierarchical ensemble graph neural network for DDoS attack detection." Computer Networks 250 (2024): 110508.

[11] Rahman, Md Mahmudur, Shanto Roy, and Mohammad Abu Yousuf. "DDoS mitigation and intrusion prevention in content delivery networks using distributed virtual honeypots." 2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT). IEEE, 2019.

[12] Dayal, Neelam, and Shashank Srivastava. "FloodKnight: an intelligent DDoS defense scheme to combat attacks near attack entry points." Journal of Computer Virology and Hacking Techniques 20.4 (2024): 819-839.

[13] Shamekhi, Ali, Pirooz Shamsinejad Babaki, and Reza Javidan. "An intelligent behavioral-based DDOS attack detection method using adaptive time intervals." Peer-to-Peer Networking and Applications 17.4 (2024): 2185-2204.

[14] Wei, Y., and J. Liu, "DDoS Attack Detection Using Machine Learning and Deep Learning Approaches," Scientific Reports, vol. 14, 2024.

[15] Najar, Ashfaq Ahmad, and S. Manohar Naik. "DDoS attack detection using MLP and Random Forest Algorithms." International Journal of Information Technology 14.5 (2022): 2317-2327.

[16] Sumantra, I., and S. Indira Gandhi. "DDoS attack detection and mitigation in software defined networks." 2020 International Conference on System, Computation, Automation and Networking (ICSCAN). IEEE, 2020.

b. Plagiarism Report

Real-Time Detection and Mitigation of DDoS Attacks

ORIGINALITY REPORT

7%	5%	10%	4%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

c. Project review sheets :

Project review sheet 1 :

Inhouse/ Industry _Innovation/Research:

Class: D17 A/B/C

Sustainable Goal:

Project Evaluation Sheet 2024 - 25

Group No.: 22

Title of Project: Real-Time DDOS Detection And Mitigation

Group Members: Aaryan Mahadik (D17C, 42), Pratham Shetty (D17C, 58), Neha Lotwani (D17C, 41), Himesh Motwani (D17C, 25)

Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg&Mgmt principles	Life - long learning	Professional Skills	Innovative Approach	Research Paper	Total Marks
(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
5	5	4	3	4	2	2	2	2	2	3	3	3	3	4	47

Comments: Good work.

Vidya.S.Zope

Name & Signature Reviewer1

Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg&Mgmt principles	Life - long learning	Professional Skills	Innovative Approach	Research Paper	Total Marks
(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
5	5	4	3	4	2	2	2	2	2	3	3	3	3	4	47

Comments: A lot of efforts done by the teams. Good research work.

47

Date: 1st March, 2025

Greacy Shejy

Name & Signature Reviewer 2

Project review sheet 2:

Inhouse/ Industry _Innovation/Research:

Class: D17 A/B/C

Sustainable Goal: 16

Project Evaluation Sheet 2024 - 25

Group No.: 22

Title of Project: Real-Time Detection and Mitigation of DDos Attacks using Network Traffic Classification

Group Members: Hiresh (25) Hotwani, Nehra (41) Hotwani, Aaryan (42) Mahadik, Pratham (58) Shetty

Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg&Mgmt principles	Life - long learning	Professional Skills	Innovative Approach	Research Paper	Total Marks
(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
5	5	4	3	4	2	2	2	2	2	2	2	3	3	4	46

Comments: Mitigation policies are taken care.

Name & Signature Reviewer1

Engineering Concepts & Knowledge	Interpretation of Problem & Analysis	Design / Prototype	Interpretation of Data & Dataset	Modern Tool Usage	Societal Benefit, Safety Consideration	Environment Friendly	Ethics	Team work	Presentation Skills	Applied Engg&Mgmt principles	Life - long learning	Professional Skills	Innovative Approach	Research Paper	Total Marks
(5)	(5)	(5)	(3)	(5)	(2)	(2)	(2)	(2)	(2)	(3)	(3)	(3)	(3)	(5)	(50)
5	5	4	3	4	2	2	2	2	2	2	3	3	3	4	46

Comments: Detection is working. Mitigation need to analysed more.

Date: 1st April,2025

Name & Signature Reviewer 2