JOURNAL OF EMERGING TECHNOLOGY AND DIGITAL TRANSFORMATION

ONLINE ISSN
3006-9726
PRINT ISSN
3006-9718

https://doi.org/10.5281/zenodo.17036934

VOLUME . 4 ISSUE . 1 (2025)

AI-POWERED LIGHTWEIGHT FRAMEWORK FOR ANOMALY DETECTION AND DDOS PREVENTION IN SDN

Muhammad Shabbir*
Department of Computer Science , Sindh Madresstual Islam University, Karachi Pakistan
Email: m.shabbir1047@gmail.com
Mehfooz Ali
Department of Computer Science, Sindh Madresstual Islam University, Karachi Pakistan
Email: mehfooz.connect@gmail.com
Muhammad Owais Siyal
Department of Computer Science, Fast National University, Karachi Pakistan
Email: owaissiyal29@gmail.com
Mudassir Iftikhar
Department of Computer Science, Sindh Madresstual Islam University, Karachi Pakistan
Email: kb41495@gmail.com

RECEIVED	***ACCEPTED***	***PUBLISHED***
15 July 2025	*06 Aug 2025*	*02 Sep 2025*

Abstract:
The current dated of your focus on networking systems has been extraordinarily supportive to frequent field such as education, medicine, finance, government, etc. It has also been observed that there is an growing demand for dependable, swift, and productive automated systems. As a result, there is a increasing interest in, and broad application of, SDN. Like other networking systems, SDNs allow for central control of networked devices, setting them apart from traditional networking systems SDNs hold the advantage of programmability and custom control. However, freedoms of programmability still retain fewer security challenges and make SDN systems more appealing to certain methods of cyberattacks, particularly the distributed denial of service. So This Paper presents a lightweight framework for detecting and mitigating Distributed Denial of Service (DDoS) attacks in Software-Defined Networking (SDN) environments. SDN enhances network management but introduces vulnerabilities that make it susceptible to DDoS attacks. The framework includes flow collection, feature extension, anomaly detection, and mitigation modules. The Naïve Bayes model achieved 93.67% accuracy, with a recall of 1.00 and precision of 0.91. The logistic regression model showed 97.08% accuracy, with a recall of 0.99 and precision of 0.97. The framework was validated using Mininet and the Ryu controller, with traffic data collected via the SDN controller. This framework contributes to network security by offering an effective solution for DDoS detection and mitigation in SDN environments. Future work will enhance the mitigation module and refine the user interface.

Keywords: NLP, AI, ML, DDOS, Mitigating Framework, Network Security

Introduction

Software-Defined Networking (SDN) stands at the forefront of a technological revolution, reshaping the landscape of network architecture by decoupling control logic from forwarding logic. Through the abstraction of control[1] logic into a central controller and communication facilitated by southbound Application Programming Interfaces (APIs), notably employing the OpenFlow protocol, SDN provides a comprehensive view of the network, thereby enhancing decision-making capabilities.[2] This transformative approach allows network administrators to program and manage the entire network efficiently, addressing issues that once consumed substantial time with remarkable speed.

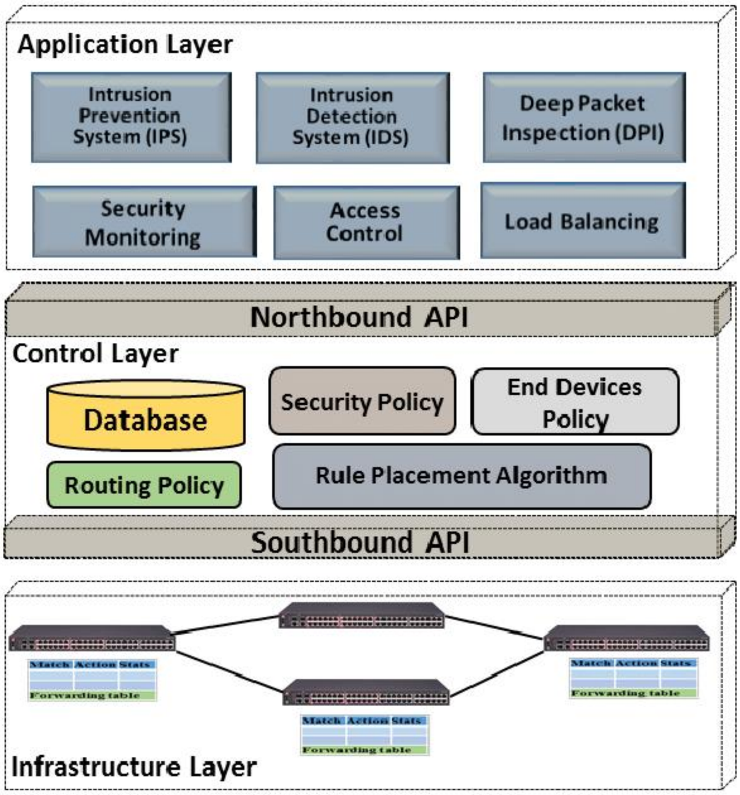


Figure 1 Simplified SDN Architecture

While industry behemoths like Google, Microsoft, and HP explore and adopt SDN integration, the benefits it brings to network management are indisputable[3]. However, amid the positive transformations, challenges emerge, with Distributed Denial of Service (DDoS) attacks standing out as a persistent threat. Despite the evolution of detection and mitigation solutions, DDoS attacks continue to escalate in power, frequency, and severity, creating an urgent need for a highly well-organized Interference Identification System (IDS) framework. A Distributed Denial of Service (DDoS) attack represents a spiteful try to interrupt ordinary traffic, devastating a directed server, service, or system with a flood of internet traffic[4]. This disruptive tactic utilizes multiple compromised computer systems as sources of attack traffic, including computers and networked resources like IoT devices[5]. Analogous to an unexpected traffic jam on a highway, a DDoS attack impedes regular traffic from reaching its destination. Immediate action upon detection is imperative, as delayed response can lead to server crashes and prolonged recovery times[6].

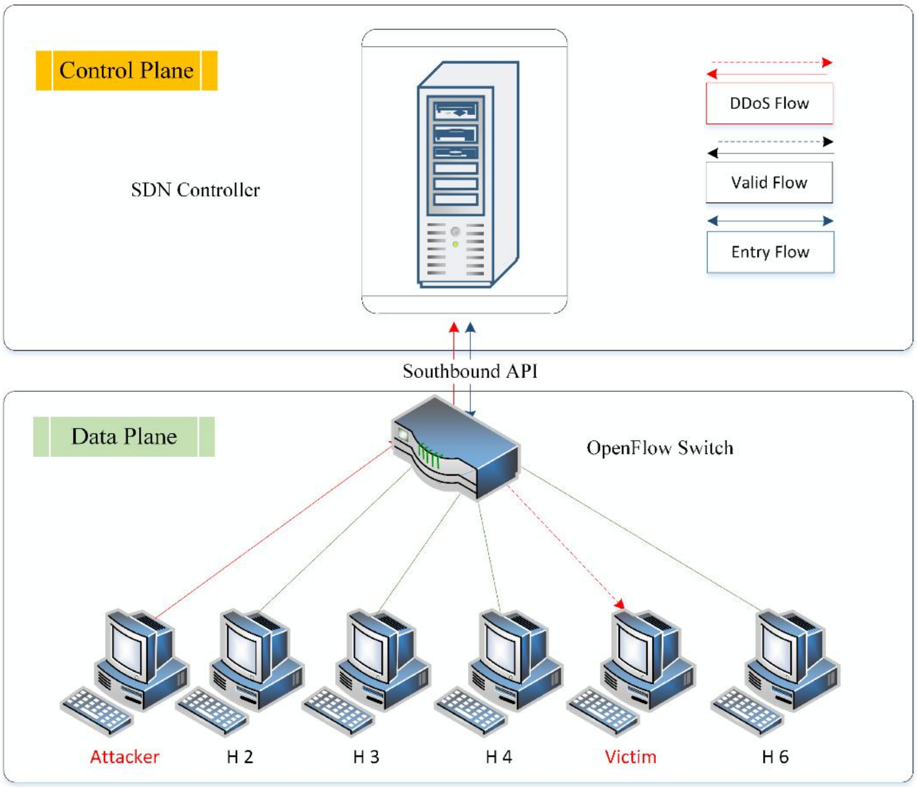


Figure 2 A simplified DDoS attack in an SDN environment.

Mitigating DDoS attacks poses a unique challenge, as attackers employ sophisticated techniques to masquerade fake traffic as legitimate. The criticality of swift response is emphasized for enterprises operating at the edge, engaging in mission-critical activities that cannot afford downtime. DDoS mitigation emerges as a protective layer within SDN[7], ensuring the ongoing availability of essential activities and services. SDN not only revolutionizes network architecture but also plays a pivotal role in fortifying network security. The dynamic nature of SDN allows for the design, construction, and operation of networks, yet it also exposes vulnerabilities. Particularly[8], the emergence of new security concerns, exemplified by the frequent launch of DDoS attacks against SDN networks, underscores the need for robust security measures. As SDN transforms the network landscape, the persistent threat of DDoS attacks necessitates a proactive approach to network security. The integration of SDN and the development of efficient IDS frameworks become imperative[9] in countering evolving threats. While SDN empowers network administrators with unprecedented control and efficiency, the journey towards its effective deployment requires a comprehensive understanding of emerging challenges, particularly those posed by DDoS attacks. The intricate dance between innovation and security underscores the ongoing evolution of network management in the era of Software-Defined Networking[10].

2. Problem Statement

The rapid growth of data transfer on the internet has led to the widespread placement of full high- speed Internet systems in marketable and learning institutions. In today's system environments, the identification of potential Distributed Denial of Service (DDoS) attacks is crucial to ensure the availability and integrity of network services[11]. Traditional Intrusion Detection Systems (IDS) fail to deal with the problems given by high traffic volumes, resulting in packet losses and decreased detection accuracy. This constraint makes IDS unsuitable for usage in high-speed networks. Existing research has focused on the performance issues of IDS on high-speed networks[12], highlighting the significance of efficient packet capture and data processing techniques. While these studies have improved IDS performance, issues remain in guaranteeing low latency for legal traffic and maintaining high detection accuracy in the face of increasing DDoS attacks [13]. To address the aforementioned challenges, this research aims to leverage machine learning (ML) models for rapid detection of Distributed Denial of Service (DDoS) attacks in high-speed networks. By integrating ML algorithms into the Intrusion Detection System (IDS), we intend to enhance the system's ability to identify and respond to malicious traffic patterns promptly and accurately. Furthermore, this study proposes the development[14] of a user-friendly interface for realtime packet visualization and analysis. This interface will provide network administrators with intuitive tools to monitor network traffic, detect anomalies, and take immediate action when necessary. Through the visualization of packet flows and the utilization of ML-based anomaly detection techniques, our system aims to empower network operators to effectively manage network security threats while minimizing disruption to legitimate network traffic. By combining efficient packet capturing techniques, advanced ML models, and a userfriendly interface for visualization and analysis, this research endeavors to address the performance limitations of traditional IDS in high-speed networks[15]. Ultimately, the goal is to ensure the availability, integrity, and security of network services in the face of evolving DDoS threats.

3. Related Work

In response to the exponential surge in internet data transmission, organizations and academic institutions have increasingly embraced high-speed network connections, ushering in an era where the role of Intrusion Detection Systems (IDS)[16] is nothing short of pivotal. These systems play a crucial role in identifying and mitigating potential network threats within the dynamic and evolving digital landscape. However, the relentless growth in network traffic volume presents formidable challenges for IDS, manifesting in packet losses and a notable reduction in accuracy. These challenges arise from the intricate processing demands imposed[17] by the diverse and high-volume nature of network traffic, creating significant roadblocks to the effective deployment of IDS in high-speed network environments. Addressing these challenges head-on, Software-Defined Networking (SDN) has emerged as a robust and adaptive solution[18]. By segregating the control plane and data plane, SDN offers a comprehensive set of advantages, including enhanced manageability, control, dynamic rule updating, comprehensive network analysis, and a unified global view facilitated by a centralized controller. While these advantages position SDN as a promising solution, it is essential to acknowledge the concomitant challenges, encompassing security vulnerabilities and deployment intricacies[19].

The research landscape has witnessed extensive efforts focused on assessing the efficacy of IDS in high-speed networks, with a specific emphasis on the challenges posed by heavy traffic loads. A seminal study by Hu et al meticulously[20] outlines the hurdles associated with packet capture systems, proposing innovative solutions through the implementation of multithreaded architectures. This architectural approach aims to optimize IDS performance by mitigating overload, thereby reducing packet losses, and enhancing CPU utilization. Furthermore, additional studies accentuate the intricate correlation between packet loss rates and IDS[21] effectiveness, shedding light on the dual influence of packet capture and inspection mechanisms . A more granular exploration conducted by Hu et al. delves into Suricata and Snort, both open-source IDSs, with the primary objective of augmenting their performance in high-speed networks. The study meticulously examines various factors impeding[22] IDS utilization, offering valuable insights into memory usage, CPU utilization, packet loss rates, and detection accuracy. These findings provide a nuanced understanding that significantly contributes to the ongoing development of novel IDS systems tailored explicitly for high-speed networks. In the realm of attack detection, our proposed Model transcends conventional features by incorporating additional metrics aimed at enhancing precision. These supplementary metrics encompass the average flow packet size, counts of flows directed to the same host within the past 5 seconds, and counts of flows targeting the same host and port within the last 5 seconds. The effectiveness of attack detection[23] is further augmented by the utilization of six machine learning algorithms, introducing a multifaceted approach to fortify network defenses against evolving threats

Introducing the Distributed DOS Mitigation Tree Architecture (SDMTA), we plan a revolutionary DDoS mitigation strategy tailored explicitly[24] for fusion cloud environments. The SDMTA seamlessly integrates network monitoring into detection procedures, providing a comprehensive evaluation of detection rates over diverse input datasets[25]. This integrative approach seeks to address the multifaceted nature of modern cyber threats and fortify network security in hybrid cloud environments.

While SDN introduces groundbreaking changes to networking paradigms, it simultaneously introduces new security threats, most notably distributed denial-of-service (DDoS) attacks[26]. The centralized controller in SDN becomes a potential single point of attack and failure. Additionally, the integration of the Internet of Things (IoT) into networks poses unprecedented security challenges, prompting the proposal of a Danger Info and Incident Managing based IoT botnet DDoS attack recognition and mitigation system[27]. This system stands as a testament to the proactive approach required to safeguard network integrity in the face of emerging threats associated with IoT integration.

The transition to Internet Protocol version six (IPv6), while alleviating the issue of IPv4 address depletion, brings forth new challenges, particularly in the form of ICMPv6-based[28] Denial of Service (DoS) and DDoS attacks. IDSs tailored to combat these specific security issues are instrumental in fortifying network security, contributing to the continual evolution and enhancement of network defense mechanisms. As the digital landscape evolves, it becomes imperative to continually refine and bolster security frameworks to effectively counter emerging threats and vulnerabilities[29].

Machine learning algorithms provide convincing performance for detecting DDoS assaults in SDN. The ML approaches efficiently identify attacks on the SDN controller's control plane 11[30]. This section briefly summarizes current efforts to identify DDoS assaults in SDN using machine learning techniques. Furthermore, the section [31] examines recent research on features selection-based ML models and approaches. The paper proposes a statistical and machine learning-based technique. It proposes a hybrid model based on K-means and K closest neighbors (KNN). DDoS detection in SDN was achieved in Kernel principal component analysis (KPCA), Genetic algorithm (GA)[32], and SVM-based technique is described. An entropy-based approach that utilizes Flow samples are supplied for traffic categorization, which focuses solely on a standard distribution of traffic. [33] has a COFFEE model that collects information from the flow for attack detection. The suspected flow is transmitted to the controller in order to extract more characteristics. Machine learning algorithms use various features to identify attacks.

Traffic data is analyzed and extracted using the Self-organizing Map (SOM). The Artificial Neural Network is used to detect DDoS attacks after features have been extracted. In order to identify the assault, the researchers suggested a k-nearest neighbor-based approach that makes advantage of the abstract distance between the traffic elements [35]. This technique lowers the false alarm rate and provides useful results for detecting irregular flow. Although the researchers suggested a number of machine learning-based methods for identifying DDoS attacks, these methods have several drawbacks, including poor efficiency, low accuracy, and poor feature selection. To detect DDoS attacks, a strategy based on Kmean clustering and Naïve Bayes (NB) was suggested [36].The technique known as Nave Bayes divides the cluster data into the norm and attack traffic after the K-mean cluster method groups the traffic data that exhibit comparable patterns. Techniques based on artificial neural networks are suggested in order to identify both known and unknown DDoS attacks [37].

A dynamic MLP (Multi Layer Perceptron) with a feedback mechanism is used by the scholar in the control system to identify DDoS attacks [38]. They employ a few specific features that are unable to differentiate between attack and normal traffic flows. The trigger method, which the authors introduced, lessens the stress on the switches and detects DDoS attacks more quickly. Although it increases the controller's workload, the trigger mechanism placed on the controller's control plane successfully attacks [39]. A more granular technique that employs the flow characteristics to identify an attack was put forth by Zang et al. It

ONLINE ISSN
3006-9726
PRINT ISSN
3006-9718
https://doi.org/10.5281/zenodo.17036934
VOLUME . 4 ISSUE . 1 (2025)

increases the accuracy of detection by extracting the flow's 39 distinct traffic features.

Table 1 : Summary of Previous Work and their contributions

PAPERS	FOCUS	CONTRIBUTION
Hu et al.	Encounters of packet capture Methods in high-speed nets	● Proposes multithreaded architectures. ● Aims to optimize IDS performance. ● Reduces packet losses. ● Enhances CPU utilization.
Hu et al.	Performance enhancement of Suricata and Snort in highspeed networks.	● Examines factors affecting IDS utilization. ● Memory usage ● CPU utilization ● Packet loss rates ● Detection accuracy
SDMTA	DDoS mitigation strategy for hybrid cloud environments.	● Introduces SDMTA. ● Integrates network monitoring into detection procedures. ● - Aims for comprehensive evaluation of detection rates.
IPv6-specific IDS	Security challenges in IPv6 networks, specifically ICMPv6-based attacks.	● Proposes IDSs tailored for IPv6 transition. ● Focuses on combating security issues related to ICMPv6-based attacks.
Machine Learning in SDN	Various machine learningbased methods for DDoS detection in SDN.	● Includes multiple methods ● Support Vector Machine (SVM) ● Kernel PCA ● Genetic Algorithm (GA) ● Self-organizing map (SOM) ● Naïve Bayes (NB) ● Artificial Neural Network-based methods

4. Methodology

4.1 System Requirements

4.1.1 Functional Requirements

The system must fulfill the following functional requirements:

- DDoS Attack Detection: The machine learning model must accurately identify Distributed Denial of Service (DDoS) attacks based on network traffic data.

- Controller Communication: The system should maintain a robust communication protocol with the network controller to obtain traffic data as needed.
- Traffic Mitigation: Upon detection of a DDoS attack, the system must initiate a mitigation process that specifically targets malicious traffic without 13 affecting normal network operations.

4.1.2 Software And Hardware Requirements

The system's performance is characterized by the following non-functional requirements:

- Response Time: The system must detect DDoS attacks within 20 seconds of their initiation.
- Traffic Analysis Frequency: It should analyze incoming traffic data every 5 seconds to classify it as normal or anomalous.
- Mitigation Timeframe: The system must be capable of mitigating identified malicious traffic within 2 minutes, including pinpointing the compromised port number.

4.1.3 Libararies

- To support the intended functionalities, the system requires the following software and hardware:
- Operating Systems: Compatible with Windows, Linux, and macOS.
- Virtualization Software: VMware or VirtualBox to simulate network environments.
- Network Tools:

  Ryu Controller: For network management and control based on OpenFlow protocols.

  Mininet: To create a realistic virtual network for testing and development.
- Development Tools:
- Node.js: For building scalable network applications.
- Flask: A lightweight WSGI web application framework used for service integration.

4.1.4 Libraries

The implementation will utilize various Python libraries to support machine learning and data manipulation functionalities:

- Scikit-learn: For implementing machine learning algorithms.
- Numpy: Essential for handling large, multi-dimensional arrays and matrices.
- Pandas: Provides high-performance, easy-to-use data structures.

4.2 Design

The design of the DDoS detection and mitigation framework involves four core components, each playing a critical role in ensuring effective detection and mitigation of malicious traffic within an SDN environment:

4.2.1 Flow Collector Module:

- **Purpose:** To gather traffic flow data from the network switches controlled by the SDN controller.
- **Process:** This module collects real-time traffic data, including flow identifiers (such as source/destination IP, source/destination ports, protocol) and flow counters (such as packet and byte counts).
- **Communication:** Utilizes the Open Flow protocol to request flow statistics from switches via the SDN controller, which then aggregates this data and sends it to the Flow Collector.

ONLINE ISSN
3006-9726
PRINT ISSN
3006-9718
https://doi.org/10.5281/zenodo.17036934
VOLUME . 4 ISSUE . 1 (2025)

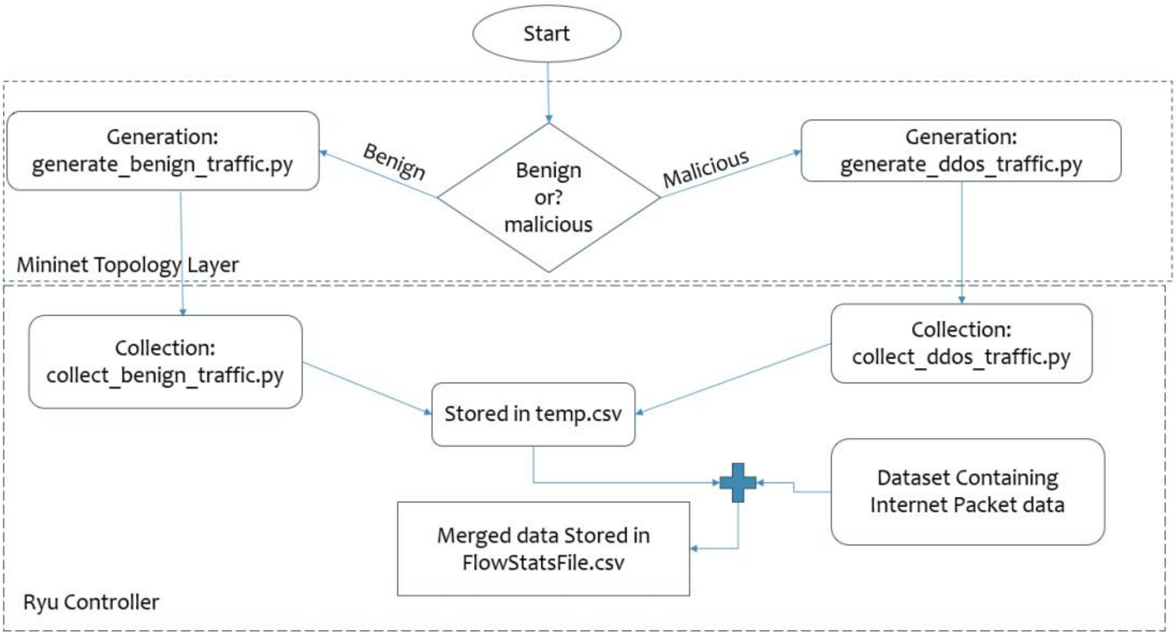

Start
Benign or? malicious
Benign
Malicious
Generation: generate_benign_traffic.py
Generation: generate_ddos_traffic.py
Mininet Topology Layer
Collection: collect_benign_traffic.py
Collection: collect_ddos_traffic.py
Stored in temp.csv
Dataset Containing Internet Packet data
Merged data Stored in FlowStatsFile.csv
Ryu Controller

Figure 3 Flow collection model diagram

4.2.2 Feature Engineering and Model Training:

- **Purpose:** Enhance the ability of the machine learning model to accurately identify and respond to DDoS threats in network traffic and to refine the input data to improve model efficiency and accuracy.
- **Process:** The dataset is processed through several stages of transformation 15 to isolate the most informative features and prepare them for model training. This systematic refinement helps in reducing the dimensionality of the data, which is crucial for effective model training and performance.
- **Model Selection:** Logistic Regression was chosen for its high detection accuracy (97.08%) and a high recall (0.99) and precision (0.97) and efficient training time (~40 seconds). Naïve Bayes was also considered but lower accuracy (93.67%) had a relatively similar training time (30 seconds).

ONLINE ISSN
3006-9726
PRINT ISSN
3006-9718
https://doi.org/10.5281/zenodo.17036934
VOLUME . 4 ISSUE . 1 (2025)

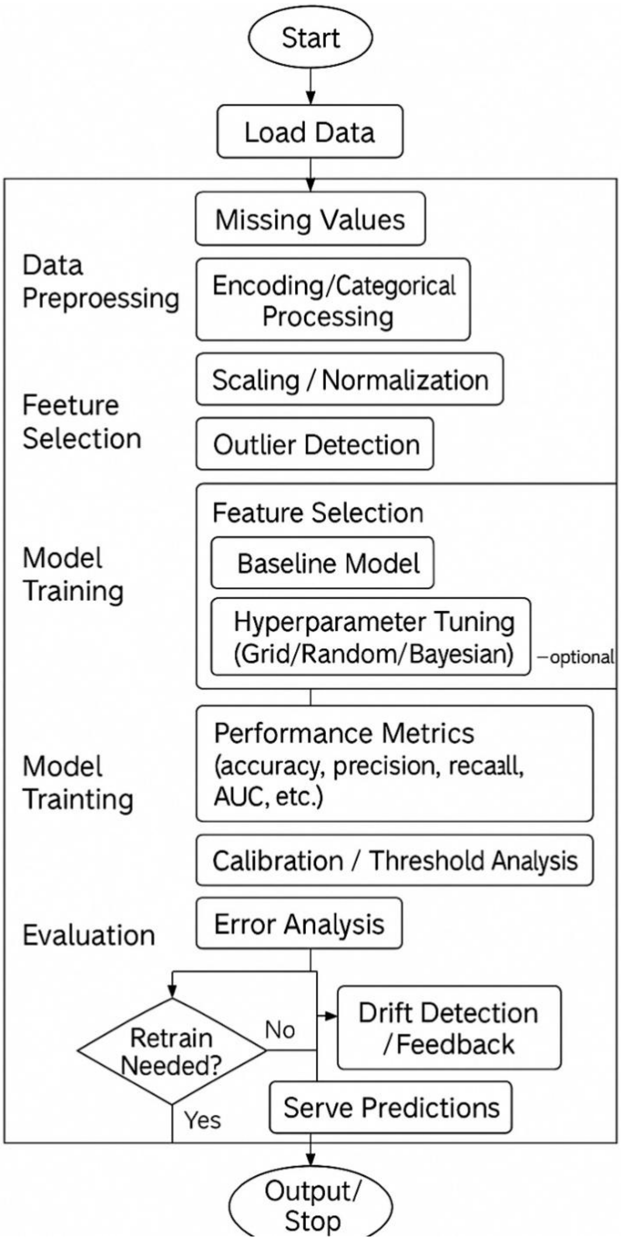


Figure 4 Feature Selection and Model Training Model.

4.2.3 Anomaly Detection Module:

- **Purpose:** To identify anomalies in network traffic indicative of DDoS attacks using machine learning.
- **Process:** The incomming traffic is collected on the switches at set intervals by the controller and is then passed to the trained model, the model first 16 preprocesses the packets and then labels them.
- **Detection:** Once the model is loaded into the controller, the model monitors incoming traffic in real-time, classifying each flow as either normal or anomalous. The model keeps a threshold value of 20%, i.e. for the traffic in the network it checks and labels all the traffic and sees if the Malicious traffic exceeds 20%, if it does it triggers a DDOS attack

warning, this approach allows us to minimize false alarms.

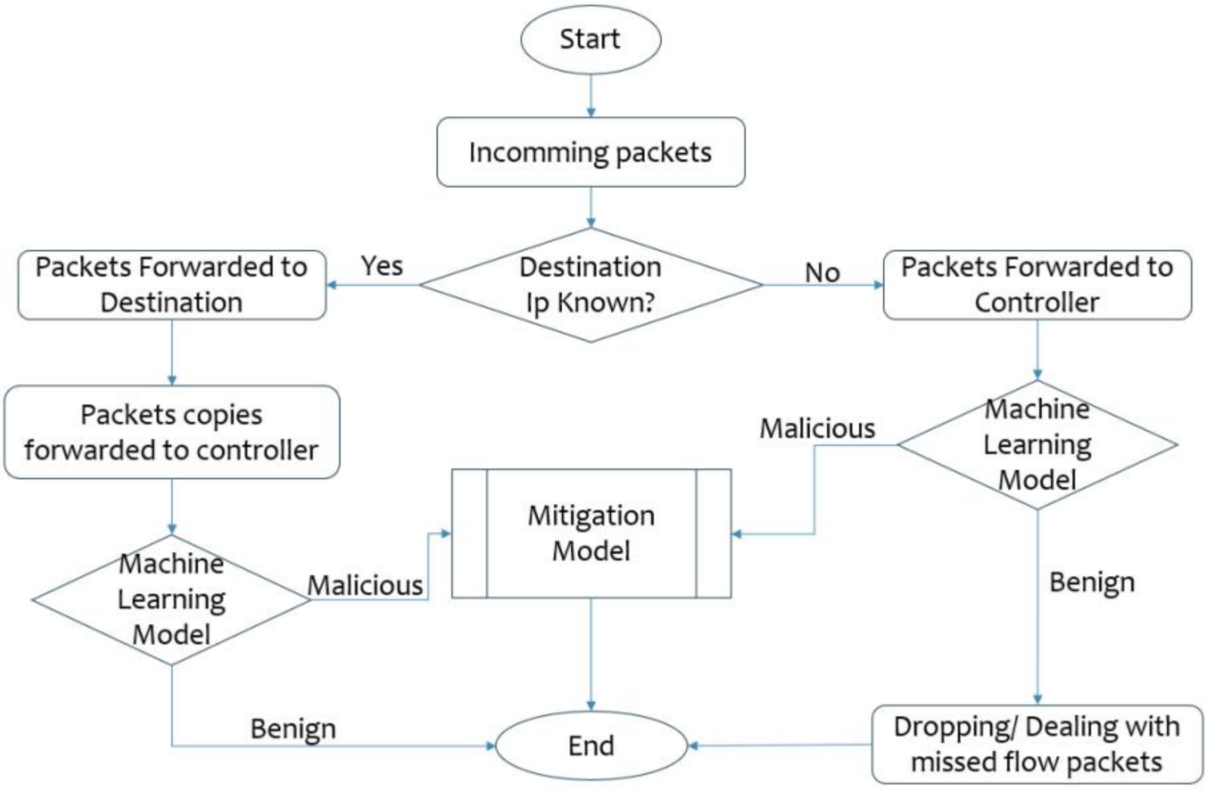

Start
Incomming packets
Destination Ip Known?
Yes
No
Packets Forwarded to Destination
Packets Forwarded to Controller
Packets copies forwarded to controller
Machine Learning Model
Malicious
Mitigation Model
Malicious
Machine Learning Model
Benign
Benign
End
Dropping/ Dealing with missed flow packets

Figure 5 Anomaly Detection Module

4.2.4 Mitigation Module:

- **Purpose:** To act against detected anomalies to protect the network from DDoS attacks.
- **Process:** Based on the classification from the Anomaly Detection module, this module takes immediate action to mitigate the detected attack. Mitigation strategies include isolating or rerouting suspicious traffic, updating flow tables to block malicious traffic, or adjusting network policies dynamically.
- **Customization:** Allows administrators to define specific mitigation rules and strategies based on the network's security requirements, ensuring minimal impact on legitimate traffic.

4.3. Overall System Design

The overall design integrates these modules into a cohesive framework that operates in real-time to detect and respond to DDoS attacks efficiently:

- ● System Integration: Each module communicates seamlessly with others, allowing for a continuous flow of data from collection to mitigation.
- ● Scalability and Efficiency: The design leverages lightweight machine learning models and modular architecture to ensure scalability and efficiency, crucial for large-scale SDN deployments.

- ● Flexibility: The framework is adaptable to various network configurations, supporting a wide range of feature sets and attack scenarios.

The design ensures that the framework provides a robust solution for DDoS detection and mitigation, tailored specifically for the dynamic requirements of SDN environments.

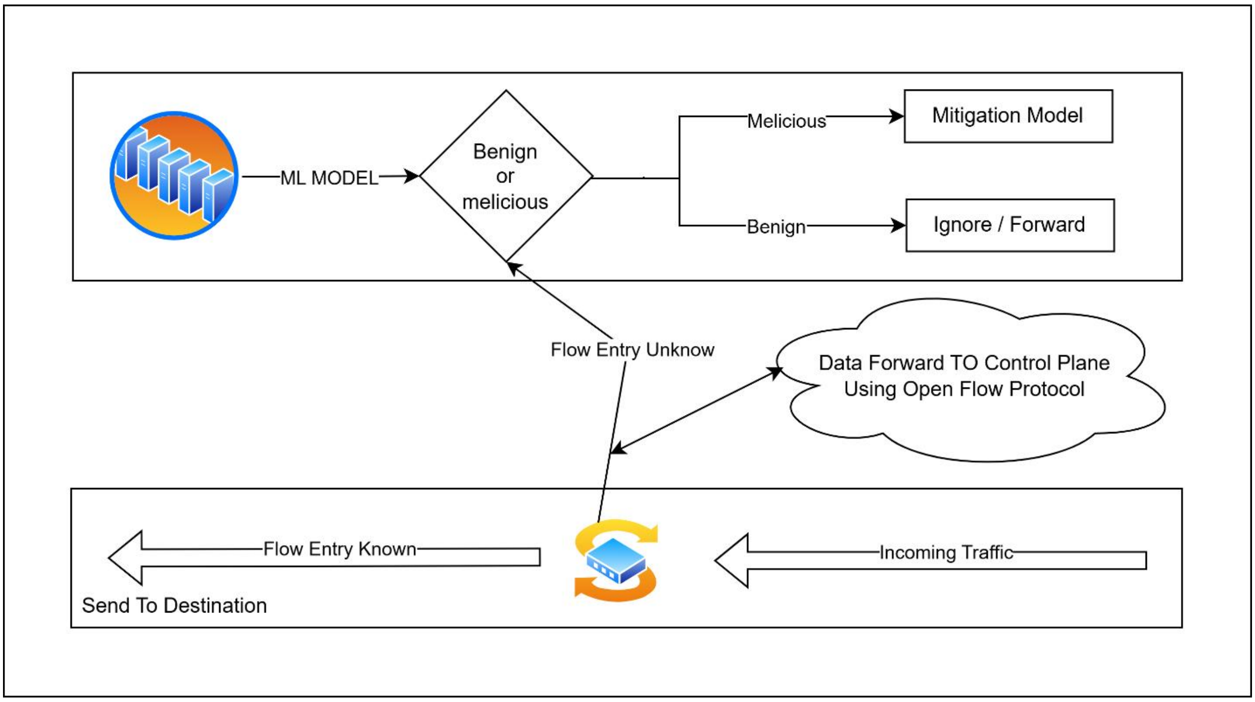
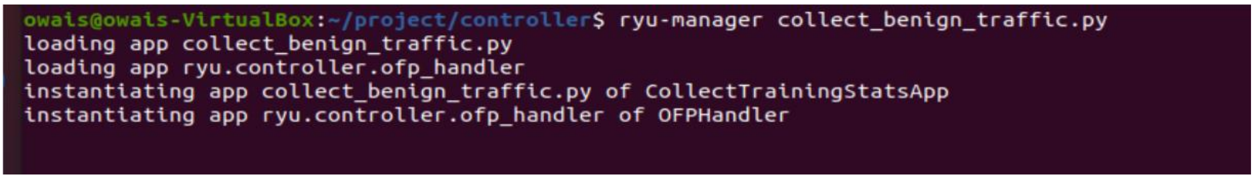


Figure 6 System Architecture Design

5. Implementation

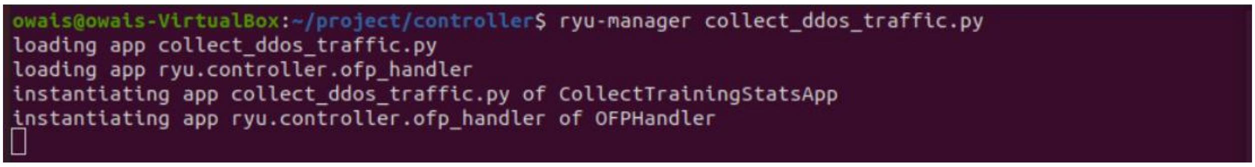
The implementation of the DDoS detection and mitigation framework is meticulously structured to ensure seamless integration within a Software-Defined Networking (SDN) environment[40]. This section delves into the specific technologies, tools, and methodologies employed across the core modules of the system. 7.1. Flow Collector Module:

- **Technology Stack:** Python programming language, coupled with the Ryu SDN controller.
- **Functionality:** This module utilizes the Ryu controller to send OpenFlow messages to network switches, requesting the current state of their flow tables.
- **Implementation Details:** The flow collector module periodically sends a FlowStatsRequest to each switch. When the switches reply with a FlowStatsReply, this module parses the flow data, including metrics like packet counts and byte counts. This data is then forwarded to the Feature Extender Module.



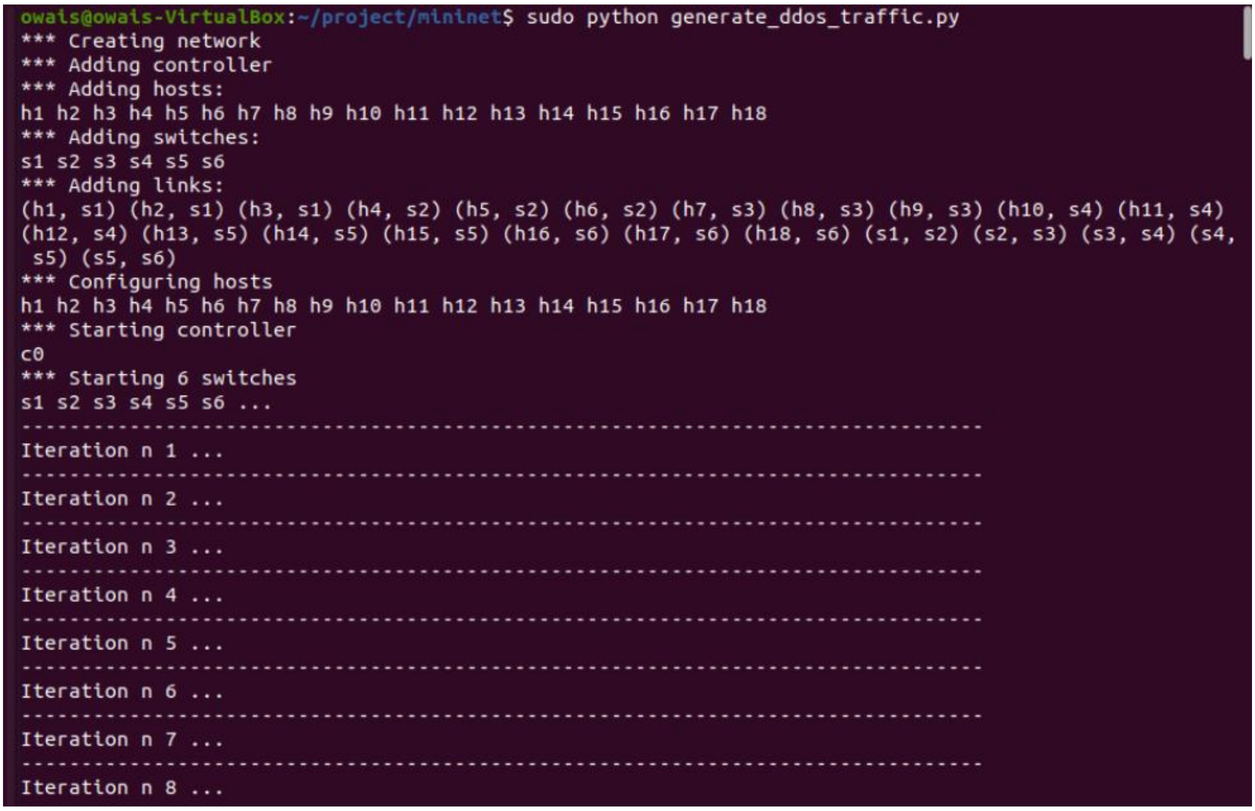
```
owais@owais-VirtualBox:~/project/controller$ ryu-manager collect_benign_traffic.py
loading app collect_benign_traffic.py
loading app ryu.controller.ofp_handler
instantiating app collect_benign_traffic.py of CollectTrainingStatsApp
instantiating app ryu.controller.ofp_handler of OFPHandler
```

Figure 7 Collect Normal Traffic



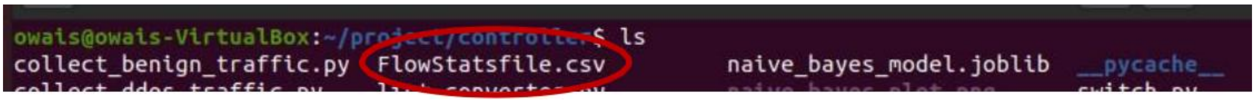
```
owais@owais-VirtualBox:~/project/controller$ ryu-manager collect_ddos_traffic.py
loading app collect_ddos_traffic.py
loading app ryu.controller.ofp_handler
instantiating app collect_ddos_traffic.py of CollectTrainingStatsApp
instantiating app ryu.controller.ofp_handler of OFPHandler
```

Figure 8 DDOS Collection Controller



```
owais@owais-VirtualBox:~/project/mininet$ sudo python generate_ddos_traffic.py
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
*** Adding switches:
s1 s2 s3 s4 s5 s6
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s2) (h5, s2) (h6, s2) (h7, s3) (h8, s3) (h9, s3) (h10, s4) (h11, s4)
(h12, s4) (h13, s5) (h14, s5) (h15, s5) (h16, s6) (h17, s6) (h18, s6) (s1, s2) (s2, s3) (s3, s4) (s4,
 s5) (s5, s6)
*** Configuring hosts
h1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
*** Starting controller
c0
*** Starting 6 switches
s1 s2 s3 s4 s5 s6 ...
--------------------------------------------------------------------------------
Iteration n 1 ...
--------------------------------------------------------------------------------
Iteration n 2 ...
--------------------------------------------------------------------------------
Iteration n 3 ...
--------------------------------------------------------------------------------
Iteration n 4 ...
--------------------------------------------------------------------------------
Iteration n 5 ...
--------------------------------------------------------------------------------
Iteration n 6 ...
--------------------------------------------------------------------------------
Iteration n 7 ...
--------------------------------------------------------------------------------
Iteration n 8 ...
```

Figure 9 Starting Controller



```
owais@owais-VirtualBox:~/project/controller$ ls
collect_benign_traffic.py  FlowStatsfile.csv      naive_bayes_model.joblib  __pycache__
```

Figure 10 Flow State File

5.1 Feature Engineering and Model Training:

- **Technology Stack:** Python scripting for data manipulation, using libraries such as scikit-learn for implementing ML algorithms,

ONLINE ISSN
3006-9726
PRINT ISSN
3006-9718
https://doi.org/10.5281/zenodo.17036934
VOLUME . 4 ISSUE . 1 (2025)

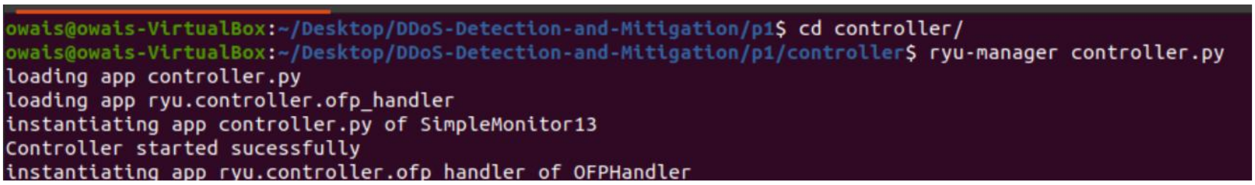
feature selection and Pandas for handling large datasets efficiently.

- **Functionality:** Standardizes the data and splits it into training and testing sets to ensure the model is not biased toward the structure of the data it was trained on[41]. Divides the dataset into features (‘X_flow’) and labels (‘y_flow’), separating inputs from the target variable. Applies variance Figure SEQ Figure ARABIC 10:Starting DDOS traffic generation. Figure SEQ Figure 11: generated dataset file. 21 thresholding and ANOVA F-test to retain features with the highest statistical significance.
- **Implementation Details:** Variance Thresholding technique is applied to remove features with zero variance, which are non-contributive to model predictions. The SelectKBest method then refines this further by selecting the top 15 features based on ANOVA F-test scores, focusing on the most relevant attributes for the model. Data normalization through Standard Scaling ensures that the Gaussian Naive Bayes classifier[42] performs optimally under the assumptions of a standard normal distribution. The model is trained on this scaled data and evaluated based on accuracy, precision, and recall metrics derived from the confusion matrix. Additionally, performance metrics such as execution time, memory, CPU, and disk usage are meticulously tracked to optimize and assess the efficiency of the training process.

5.2 Anomaly Detection Module:

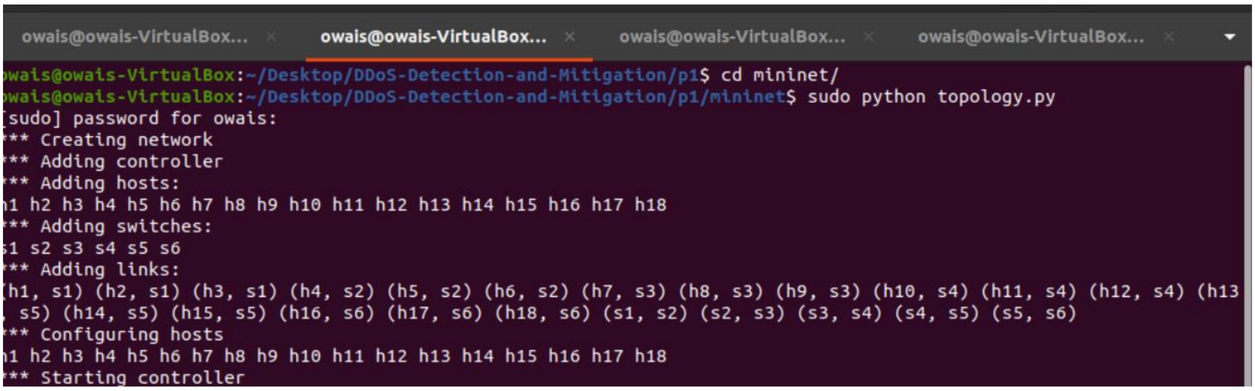
- **Technology Stack:** Scikit-learn for implementing machine learning models.
- **Functionality:** Utilizes trained machine learning models to classify traffic flows based on the features extended by the previous module.
- **Implementation Details:** This module trains the Naïve Bayes classifier using a dataset labeled with instances of normal and DDoS traffic. The training process involves feature selection and model validation to optimize performance and accuracy. After training, the model continuously receives new traffic data, classifying it in real-time to detect potential DDoS attacks.

- **Feature Implementation Snapshots:**



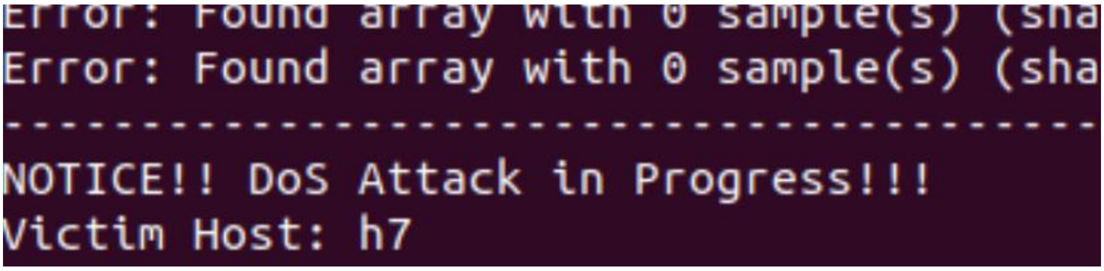
```
owais@owais-VirtualBox:~/Desktop/DDoS-Detection-and-Mitigation/p1$ cd controller/
owais@owais-VirtualBox:~/Desktop/DDoS-Detection-and-Mitigation/p1/controller$ ryu-manager controller.py
loading app controller.py
loading app ryu.controller.ofp_handler
instantiating app controller.py of SimpleMonitor13
Controller started sucessfully
instantiating app ryu.controller.ofp_handler of OFPHandler
```

Figure 11 Anomaly Starting Controller


owais@owais-VirtualBox...
owais@owais-VirtualBox...
owais@owais-VirtualBox...
owais@owais-VirtualBox...

```
wais@owais-VirtualBox:~/Desktop/DDoS-Detection-and-Mitigation/p1$ cd mininet/
wais@owais-VirtualBox:~/Desktop/DDoS-Detection-and-Mitigation/p1/mininet$ sudo python topology.py
[sudo] password for owais:
** Creating network
** Adding controller
** Adding hosts:
1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
** Adding switches:
1 s2 s3 s4 s5 s6
** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s2) (h5, s2) (h6, s2) (h7, s3) (h8, s3) (h9, s3) (h10, s4) (h11, s4) (h12, s4) (h13
, s5) (h14, s5) (h15, s5) (h16, s6) (h17, s6) (h18, s6) (s1, s2) (s2, s3) (s3, s4) (s4, s5) (s5, s6)
** Configuring hosts
1 h2 h3 h4 h5 h6 h7 h8 h9 h10 h11 h12 h13 h14 h15 h16 h17 h18
** Starting controller
```

Figure 12 Starting Mininet



```
Error: Found array with 0 sample(s) (sha
Error: Found array with 0 sample(s) (sha
----------------------------------------
NOTICE!! DoS Attack in Progress!!!
Victim Host: h7
```

Figure 13 Dos Detected

5.3 Mitigation Module

- **Technology Stack:** Python with integration into the Ryu controller for executing network commands.
- **Functionality:** Responds to detected threats by implementing predefined mitigation strategies.
- **Implementation Details:** In the development of our DDoS mitigation strategy, one of the key techniques involved the detection and isolation of the specific incoming port through which the DDoS attack was being perpetrated. By accurately identifying the compromised port, our system was able to implement a targeted timeout on this port, effectively halting the DDoS traffic without impacting other unaffected traffic channels. This selective approach not only minimized downtime for legitimate network users but also enhanced the overall security posture by swiftly neutralizing the threat at its source[43]. The advantage of this method lies in its precision and efficiency. By focusing directly on the source of the attack, the system avoids the broader network disruptions typically associated with more general DDoS mitigation techniques, such as total bandwidth throttling or blanket IP blocking. Additionally, this targeted timeout approach conserves network resources, maintains optimal network performance for legitimate users, and significantly reduces the window of vulnerability during an attack, thereby enhancing both the responsiveness and resilience of the network infrastructure[44].

6. System Integration and Testing

6.1 Integration:

All modules are integrated within a single application framework running atop the Ryu controller, facilitating direct communication between components and synchronous operations.

6.2 Testing Environment:

Mininet is used to create a virtual network that mimics the deployment environment, enabling comprehensive testing of the system under controlled conditions. This setup allows for the simulation of both benign and DDoS traffic to test the system's responsiveness and effectiveness.

- **Performance Metrics:** The system's performance is evaluated based on detection accuracy, mitigation effectiveness, and resource efficiency (CPU, memory usage). Automated scripts monitor these metrics during test scenarios to ensure the system meets predefined performance standards

7. Challenges and Solutions

- **Data Handling:** Managing large volumes of traffic data efficiently was achieved through optimized data structures and processing algorithms.
- **Model Accuracy:** Multiple models were tested, and Naïve Bayes was selected due to its balance between detection accuracy and training time.

8. Testing and Evaluation

The testing and evaluation phase of the DDoS detection framework is crucial for validating the system's effectiveness in detecting and mitigating malicious network traffic. This phase involves:

8.1 Testing:

8.1.1. Dataset Creation:

- **Traffic Generation:** Using Mininet, both benign and DDoS traffic scenarios were generated. Normal traffic included basic ping commands and HTTP requests, while DDoS traffic involved attack vectors like ICMP floods, SYN floods, UDP floods, and Smurf attacks. The hping3 tool was used to simulate these attacks.
- **Traffic Collection:** The Ryu controller collected traffic data, storing it in a structured format for further analysis. Flow statistics were captured to provide features such as source/destination IPs, port numbers, protocol types, and packet/byte counts.

8.2.2 Feature Engineering and Model Training:

- **Feature Extraction:** Features were extracted using custom Python scripts to compute additional metrics like packet count per second, flow duration, and average packet size. These features were crucial in distinguishing normal traffic from attacks.
- **Model Training**: Naïve Bayes and Logistic Regression models were trained using the scikit-learn library. The dataset was split into training and test sets to validate the models' performance.

8.3.3 Model Testing and Real-Time Analysis:

- **Offline Testing:** Models were first tested on the test dataset to assess accuracy, precision, recall, and F1 score. This ensured that models could effectively distinguish between benign and malicious traffic.

ONLINE ISSN
3006-9726
PRINT ISSN
3006-9718
https://doi.org/10.5281/zenodo.17036934
VOLUME . 4 ISSUE . 1 (2025)

- **Real-Time Testing:** The selected Naïve Bayes model was deployed on the Ryu controller to classify traffic in real-time. Generated traffic was then analyzed to determine if the model could accurately identify DDoS attacks.

8.4.4 Mitigation Testing:

- 30 Rule Implementation: Custom mitigation rules were implemented to block or isolate malicious traffic based on model predictions.
- Effectiveness Testing: Various DDoS attack scenarios were simulated to evaluate how quickly and accurately the system could respond and mitigate attacks.

9. Evaluation:

9.1. Accuracy Metrics:

- **Detection Accuracy:** The ability of the model to correctly identify benign and malicious traffic was measured. The Naïve Bayes model achieved an accuracy of 93.72%.
- **False Positives/Negatives:** The rate of incorrectly classified traffic was measured to evaluate the model's reliability.

9.2. Performance Metrics:

- **Execution Time:** The time taken to train and deploy the model was measured. Naïve Bayes had a training time of about 20 seconds, whereas Logistic Regression took 3-5 minutes.
- **Resource Usage:** The framework's impact on CPU, memory, and disk usage was monitored to ensure it remained lightweight and suitable for deployment in resource-constrained environments.

9.3. Mitigation Effectiveness:

- **Response Time:** The time taken for the system to detect an attack and implement mitigation actions was measured.
- **Traffic Impact:** The system's impact on legitimate traffic during attack scenarios was assessed to ensure that mitigation did not disrupt normal network operations.

ONLINE ISSN
3006-9726
PRINT ISSN
3006-9718
https://doi.org/10.5281/zenodo.17036934
VOLUME . 4 ISSUE . 1 (2025)

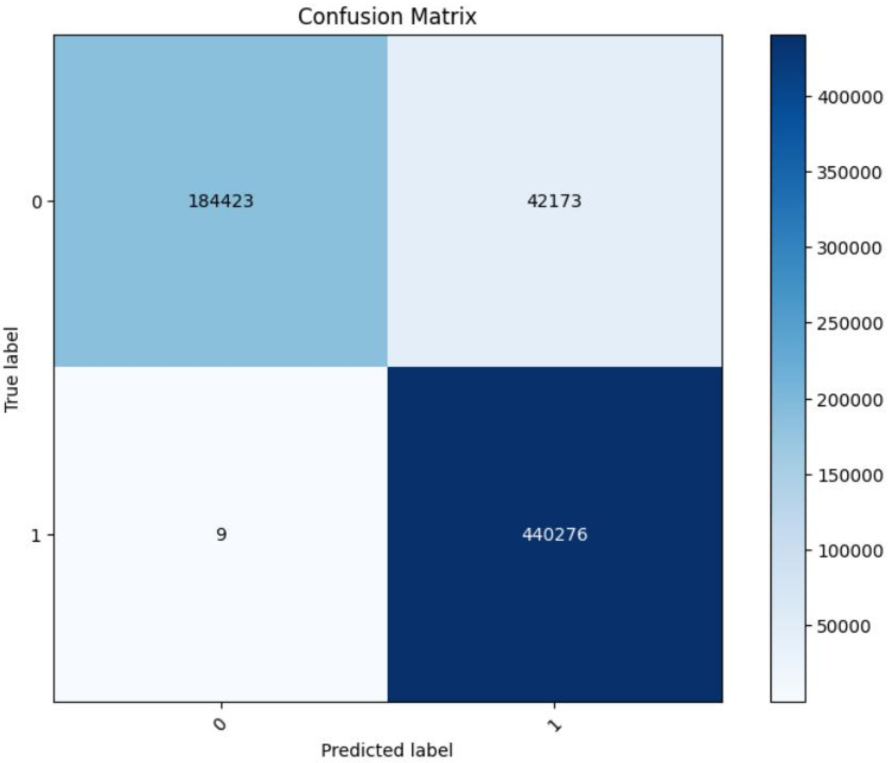


Figure 14 Confusion Matrix

Model	Accuracy	Precision	Recall	F1-Score	Time to Detect	Peak Memory Usage (k-best)	Peak Memory Usage (w/o k-best)
Naive Bayes	93.67%	0.91	0.99	0.95	8.37 sec	933.18 MB	1541.69 MB
Logistic Regression	97.08%	0.99	0.97	0.98	5.08 sec	812.23 MB	1340.10 MB

Table 2 Summery Of Model Performance

12. Conclusion

The designed DDoS detection and mitigation framework for Software-Defined Networking (SDN) environments successfully balances detection accuracy, response time, and resource efficiency. The comprehensive framework integrates several innovative elements to enhance network security in real-time:

1. **Framework Capabilities:** The framework leverages modular components for traffic collection, feature engineering, anomaly detection, and mitigation. This modularity 32 ensures adaptability to various network configurations and allows for future enhancements. The flow collector efficiently gathers traffic data, while the feature extender augments this data for

ONLINE ISSN
3006-9726
PRINT ISSN
3006-9718
https://doi.org/10.5281/zenodo.17036934
VOLUME . 4 ISSUE . 1 (2025)

better classification. The anomaly detection module uses machine learning to identify abnormal traffic patterns, and the mitigation module swiftly responds to mitigate attacks.

2. **Machine Learning Effectiveness:** The Naïve Bayes machine learning model provided effective DDoS detection with a 93.72% accuracy rate, balancing between accuracy and efficiency. The training time of under 20 seconds indicates that this approach can be rapidly deployed and updated, which is crucial for evolving threats.

3. **Mitigation Strategies:** The framework's mitigation module provides actionable responses to detected DDoS attacks, isolating and blocking malicious traffic in realtime. The flexibility in defining mitigation rules allows network administrators to customize responses based on the specific network environment.

4. **Performance and Resource Efficiency:** The framework demonstrated efficient usage of computational resources, ensuring that its deployment does not hinder normal network operations. The testing and evaluation phase confirmed that the framework could operate in a real-world network without significant latency or resource overhead.

5. **Comprehensive Testing:** The framework was tested in a simulated SDN environment using Mininet and the Ryu controller, ensuring realistic traffic conditions. The testing demonstrated the framework's ability to detect and mitigate a variety of DDoS attacks, providing practical validation of its effectiveness.

13. Future Work

While the framework demonstrates potential in mitigating DDoS attacks within SDN environments, several enhancements could further bolster its effectiveness:

- **Advanced Machine Learning Models:** Investigating more sophisticated machine learning and deep learning algorithms may improve the precision and reliability of attack detection.
- **Feature Expansion:** Adding a broader set of network metrics can enrich the analysis of traffic patterns, thereby boosting the framework's capability to identify anomalies.
- **Real-World Deployment:** Deploying the framework in a live production environment will enable performance evaluation under authentic traffic conditions, offering valuable insights into its operational efficacy.

Final Thoughts

This framework marks a substantial advancement in network security for SDN contexts, delivering a potent means for detecting and counteracting DDoS threats. Its modular architecture, streamlined design, and integration of machine learning technologies render it an adaptable and powerful tool for network defense. With ongoing enhancements, this framework is poised to set a new benchmark in protecting SDN-based networks from DDoS dangers.

REFRENCES

[1] N. Aslam, S. Srivastava, and M. M. Gore, “A Comprehensive Analysis of Machine Learning- and Deep Learning-Based Solutions for DDoS Attack Detection in SDN,” *Arab. J. Sci. Eng.*, vol. 49, no. 3, pp. 3533–3573, Mar. 2024, doi: 10.1007/s13369-023-08075-2.

[2] N. N. Tuan, P. H. Hung, N. D. Nghia, N. V. Tho, T. V. Phan, and N. H. Thanh, “A DDoS attack mitigation scheme in ISP networks using machine learning based on

https://doi.org/10.5281/zenodo.17036934
VOLUME . 4 ISSUE . 1 (2025)

SDN," *Electronics*, vol. 9, no. 3, p. 413, 2020.

[3] J. A. Perez-Diaz, I. A. Valdovinos, K.-K. R. Choo, and D. Zhu, "A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning," *IEEe Access*, vol. 8, pp. 155859–155872, 2020.

[4] M. Revathi, V. V. Ramalingam, and B. Amutha, "A Machine Learning Based Detection and Mitigation of the DDOS Attack by Using SDN Controller Framework," *Wirel. Pers. Commun.*, vol. 127, no. 3, pp. 2417–2441, Dec. 2022, doi: 10.1007/s11277-021-09071-1.

[5] S. S. Mohammed *et al.*, "A new machine learning-based collaborative DDoS mitigation mechanism in software-defined network," in *2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, IEEE, 2018, pp. 1–8. Accessed: Aug. 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8589104/

[6] A. Singh, H. Kaur, and N. Kaur, "A novel DDoS detection and mitigation technique using hybrid machine learning model and redirect illegitimate traffic in SDN network," *Clust. Comput.*, vol. 27, no. 3, pp. 3537–3557, June 2024, doi: 10.1007/s10586-023-04152-1.

[7] M. Aslam *et al.*, "Adaptive machine learning based distributed denial-of-services attacks detection and mitigation system for SDN-enabled IoT," *Sensors*, vol. 22, no. 7, p. 2697, 2022.

[8] S. Arora, P. Khare, and S. Gupta, "AI-driven DDoS mitigation at the edge: Leveraging machine learning for real-time threat detection and response," in *2024 International Conference on Data Science and Network Security (ICDSNS)*, IEEE, 2024, pp. 1–7. Accessed: Aug. 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10690930/

[9] B. K. Devi, G. Preetha, G. Selvaram, and S. M. Shalinie, "An impact analysis: Real time DDoS attack detection and mitigation using machine learning," in *2014 International Conference on Recent Trends in Information Technology*, IEEE, 2014, pp. 1–7. Accessed: Aug. 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6996133/

[10] J. D. Gadze, A. A. Bamfo-Asante, J. O. Agyemang, H. Nunoo-Mensah, and K. A.-B. Opare, "An investigation into the application of deep learning in the detection and mitigation of DDOS attack on SDN controllers," *Technologies*, vol. 9, no. 1, p. 14, 2021.

[11] A. R. Wani, Q. P. Rana, U. Saxena, and N. Pandey, "Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques," in *2019 Amity International conference on artificial intelligence (AICAI)*, IEEE, 2019, pp. 870–875. Accessed: Aug. 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8701238/

[12] D. Manikumar and B. U. Maheswari, "Blockchain based DDoS mitigation using machine learning techniques," in *2020 Second international conference on inventive research in computing applications (ICIRCA)*, IEEE, 2020, pp. 794–800. Accessed: Aug. 13, 2025. [Online]. Available:

https://ieeexplore.ieee.org/abstract/document/9183092/

[13] F. Khashab, J. Moubarak, A. Feghali, and C. Bassil, "DDoS attack detection and mitigation in SDN using machine learning," in *2021 IEEE 7th International Conference on Network Softwarization (NetSoft)*, IEEE, 2021, pp. 395–401. Accessed: Aug. 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9492558/

[14] S. Vattikuti, M. R. Hegde, M. Manish, V. Bodduvaram, and V. Sarasvathi, "DDoS attack detection and mitigation using anomaly detection and machine learning models," in *2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, IEEE, 2021, pp. 1–6. Accessed: Aug. 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9683214/

[15] V. Hnamte, A. A. Najar, H. Nhung-Nguyen, J. Hussain, and M. N. Sugali, "DDoS attack detection and mitigation using deep neural network in SDN environment," *Comput. Secur.*, vol. 138, p. 103661, 2024.

[16] N. Z. Bawany, J. A. Shamsi, and K. Salah, "DDoS Attack Detection and Mitigation Using SDN: Methods, Practices, and Solutions," *Arab. J. Sci. Eng.*, vol. 42, no. 2, pp. 425–441, Feb. 2017, doi: 10.1007/s13369-017-2414-5.

[17] M. Zekri, S. El Kafhali, N. Aboutabit, and Y. Saadi, "DDoS attack detection using machine learning techniques in cloud computing environments," in *2017 3rd international conference of cloud computing technologies and applications (CloudTech)*, IEEE, 2017, pp. 1–7. Accessed: Aug. 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8284731/

[18] O. Rahman, M. A. G. Quraishi, and C.-H. Lung, "DDoS attacks detection and mitigation in SDN using machine learning," in *2019 IEEE world congress on services (SERVICES)*, IEEE, 2019, pp. 184–189. Accessed: Aug. 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8817237/

[19] Q. Li, L. Meng, Y. Zhang, and J. Yan, "DDoS Attacks Detection Using Machine Learning Algorithms," in *Digital TV and Multimedia Communication*, vol. 1009, G. Zhai, J. Zhou, P. An, and X. Yang, Eds., in Communications in Computer and Information Science, vol. 1009. , Singapore: Springer Singapore, 2019, pp. 205–216. doi: 10.1007/978-981-13-8138-6_17.

[20] M. A. Al-Shareeda, S. Manickam, and M. A. Saare, "DDoS attacks detection using machine learning and deep learning techniques: analysis and comparison," *Bull. Electr. Eng. Inform.*, vol. 12, no. 2, pp. 930–939, 2023.

[21] A. R. Gawande, "DDoS detection and mitigation using machine learning," PhD Thesis, Rutgers University-Camden Graduate School, 2018. Accessed: Aug. 13, 2025. [Online]. Available: https://rucore.libraries.rutgers.edu/rutgers-lib/57074/

[22] D. Kumar, R. K. Pateriya, R. K. Gupta, V. Dehalwar, and A. Sharma, "DDoS detection using deep learning," *Procedia Comput. Sci.*, vol. 218, pp. 2420–2429, 2023.

[23] S. Pande, A. Khamparia, D. Gupta, and D. N. H. Thanh, "DDOS

Detection Using Machine Learning Technique," in *Recent Studies on Computational Intelligence*, vol. 921, A. Khanna, A. K. Singh, and A. Swaroop, Eds., in Studies in Computational Intelligence, vol. 921. , Singapore: Springer Singapore, 2021, pp. 59–68. doi: 10.1007/978-981-15-8469-5_5.

[24] R. Amrish, K. Bavapriyan, V. Gopinaath, A. Jawahar, and C. V. Kumar, "DDoS detection using machine learning techniques," *J. IoT Soc. Mob. Anal. Cloud*, vol. 4, no. 1, pp. 24–32, 2022.

[25] A. I. El Sayed, M. Abdelaziz, M. Hussein, and A. D. Elbayoumy, "DDoS mitigation in IoT using machine learning and blockchain integration," *IEEE Netw. Lett.*, vol. 6, no. 2, pp. 152–155, 2024.

[26] A. Jawahar *et al.*, "DDoS mitigation using blockchain and machine learning techniques," *Multimed. Tools Appl.*, vol. 83, no. 21, pp. 60265–60278, 2024.

[27] A. Mihoub, O. B. Fredj, O. Cheikhrouhou, A. Derhab, and M. Krichen, "Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques," *Comput. Electr. Eng.*, vol. 98, p. 107716, 2022.

[28] Y. I. Aljanabi, A. A. Majeed, K. H. Jihad, and B. A. Qader, "Detect and mitigate blockchain-based DDoS attacks using machine learning and smart contracts," *Informatica*, vol. 46, no. 7, 2022, Accessed: Aug. 13, 2025. [Online]. Available: https://informatica.si/index.php/informatica/article/view/4033

[29] R. Sanjeetha, A. Raj, K. Saivenu, M. I. Ahmed, B. Sathvik, and A. Kanavalli, "Detection and mitigation of botnet based DDoS attacks using catboost machine learning algorithm in SDN environment," *Int. J. Adv. Technol. Eng. Explor.*, vol. 8, no. 76, p. 445, 2021.

[30] T. Alyas, "Detection and mitigation of DDoS attack in cloud computing using machine learning algorithm," *ICST Trans. Scalable Inf. Syst.*, 2018, Accessed: Aug. 13, 2025. [Online]. Available: https://www.academia.edu/download/85179730/eai.29-7-2019.pdf

[31] D. Satyanarayana and A. S. Alasmi, "Detection and mitigation of DDOS based attacks using machine learning algorithm," in *2022 International Conference on Cyber Resilience (ICCR)*, IEEE, 2022, pp. 1–5. Accessed: Aug. 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9995773/

[32] S. Ahmed *et al.*, "Effective and efficient DDoS attack detection using deep learning algorithm, multi-layer perceptron," *Future Internet*, vol. 15, no. 2, p. 76, 2023.

[33] I. Ko, D. Chambers, and E. Barrett, "Feature dynamic deep learning approach for DDoS mitigation within the ISP domain," *Int. J. Inf. Secur.*, vol. 19, no. 1, pp. 53–70, Feb. 2020, doi: 10.1007/s10207-019-00453-y.

[34] K. M. Sudar and P. Deepalakshmi, "Flow-Based Detection and Mitigation of Low-Rate DDOS Attack in SDN Environment Using Machine Learning Techniques," in *IoT and Analytics for Sensor Networks*, vol. 244, P. Nayak, S. Pal, and S.-L. Peng, Eds., in Lecture Notes in Networks and Systems, vol. 244. , Singapore: Springer Singapore, 2022, pp. 193–205. doi: 10.1007/978-981-16-2919-8_18.

[35] W. I. Khedr, A. E. Gouda, and E. R. Mohamed, "FMDADM: A multi-layer DDoS attack detection and mitigation framework using machine learning for stateful SDN-based IoT networks," *Ieee Access*, vol. 11, pp. 28934–28954, 2023.

[36] R. Santos, D. Souza, W. Santo, A. Ribeiro, and E. Moreno, "Machine learning algorithms to detect DDoS attacks in SDN," *Concurr. Comput. Pract. Exp.*, vol. 32, no. 16, p. e5402, Aug. 2020, doi: 10.1002/cpe.5402.

[37] A. Aljuhani, "Machine learning approaches for combating distributed denial of service attacks in modern networking environments," *IEEE Access*, vol. 9, pp. 42236–42264, 2021.

[38] M. Aslam, D. Ye, M. Hanif, and M. Asad, "Machine Learning Based SDN-enabled Distributed Denial-of-Services Attacks Detection and Mitigation System for Internet of Things," in *Machine Learning for Cyber Security*, vol. 12486, X. Chen, H. Yan, Q. Yan, and X. Zhang, Eds., in Lecture Notes in Computer Science, vol. 12486. , Cham: Springer International Publishing, 2020, pp. 180–194. doi: 10.1007/978-3-030-62223-7_16.

[39] T. E. Ali, Y.-W. Chong, and S. Manickam, "Machine learning techniques to detect a DDoS attack in SDN: A systematic review," *Appl. Sci.*, vol. 13, no. 5, p. 3183, 2023.

[40] F. Musumeci, A. C. Fidanci, F. Paolucci, F. Cugini, and M. Tornatore, "Machine-Learning-Enabled DDoS Attacks Detection in P4 Programmable Networks," *J. Netw. Syst. Manag.*, vol. 30, no. 1, p. 21, Jan. 2022, doi: 10.1007/s10922-021-09633-5.

[41] M. E. Ahmed, H. Kim, and M. Park, "Mitigating DNS query-based DDoS attacks with machine learning on software-defined networking," in *MILCOM 2017-2017 IEEE Military Communications Conference (MILCOM)*, IEEE, 2017, pp. 11–16. Accessed: Aug. 13, 2025. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8170802/

[42] J. Ramprasath, N. Krishnaraj, and V. Seethalakshmi, "Mitigation Services on SDN for Distributed Denial of Service and Denial of Service Attacks Using Machine Learning Techniques," *IETE J. Res.*, vol. 70, no. 1, pp. 70–81, Jan. 2024, doi: 10.1080/03772063.2022.2142163.

[43] K. A. Simpson, S. Rogers, and D. P. Pezaros, "Per-host DDoS mitigation by direct-control reinforcement learning," *IEEE Trans. Netw. Serv. Manag.*, vol. 17, no. 1, pp. 103–117, 2019.

[44] T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. K. Son, "Performance evaluation of Botnet DDoS attack detection using machine learning," *Evol. Intell.*, vol. 13, no. 2, pp. 283–294, June 2020, doi: 10.1007/s12065-019-00310-w.