SDN Based DDOS Detection and Mitigation

Anirudh C M
Department of Computer Science and Engineering Sri Siddhartha Institute of Technology
Tumkur, Karnataka, India cmanirudh03@gmail.com

Chandan T O
Department of Computer Science and Engineering Sri Siddhartha Institute of Technology
Tumkur, Karnataka, India chandanto9036@gmail.com

Chirag V
Department of Computer Science and Engineering Sri Siddhartha Institute of Technology
Tumkur, Karnataka, India chiragchiru2108@gmail.com

Likhith Gowda S R
Department of Computer Science and Engineering Sri Siddhartha Institute of Technology
Tumkur, Karnataka, India likhithgowdasr21@gmail.com

Dr V Raviram
Prof and HOD, Department of Computer Science and Engineering Sri Siddhartha Institute of Technology
Tumkur, Karnataka, India raviramv@ssit.edu.in

***Abstract*—Distributed Denial of Service (DDoS) attacks continue to threaten the performance and availability of today's network infrastructure. Traditional security solutions often fall short in dynamic environments, as they rely on fixed rules and lack the flexibility to respond to evolving threats in real time. This paper introduces a smart and adaptive solution that combines Software Defined Networking (SDN) and Machine Learning (ML) to detect and counter DDoS threats efficiently. Leveraging the RYU SDN controller, Mininet for emulated environments, and a Random Forest classifier for traffic analysis, the system dynamically identifies and mitigates abnormal traffic patterns. Experimental evaluation shows that the framework achieves a detection accuracy of 97.2%, maintains a low false positive rate of 3.1%, and introduces minimal delay, confirming the practicality and effectiveness of the proposed approach.**

***Index Terms*—SDN, DDoS Attack, Machine Learning, Random Forest, RYU Controller, Mininet**

I. INTRODUCTION

In today's interconnected world, Distributed Denial of Service (DDoS) attacks remain one of the most destructive forms of cyberattacks. By flooding targeted servers or networks with an overwhelming volume of traffic from multiple compromised sources, attackers can cause severe service disruptions. These attacks are often launched with malicious intent—whether for political motives, financial gain, or personal vendettas—and can lead to significant downtime, financial losses, and reputa- tional damage.

Traditional network security measures, which rely heavily on static rules and signature-based detection, are often ill-equipped to handle sophisticated and evolving threats. To address these limitations, this work explores a modern approach that combines Software Defined Networking (SDN) and Machine Learning (ML) to deliver a dynamic and intelligent DDoS defense mechanism. SDN offers a modern network design by separating the control plane from the data plane, enabling centralized control and flexible, software-based traffic management. This flexibil- ity is particularly useful for responding to real-time threats like DDoS attacks. On the other hand, ML enables systems to learn from data patterns and make intelligent, real-time decisions—an essential feature for identifying and responding to anomalies in network traffic.

In this research, we present an SDN-based intrusion detec- tion and prevention framework that integrates machine learn- ing for enhanced responsiveness. We tested various classifiers such as K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Trees (DT), and Random Forest (RF). The Random Forest algorithm was found to outperform the others in terms of accuracy and efficiency.

The framework is implemented using Mininet, a widely- used SDN emulator, and the RYU controller, which facilitates real-time traffic management and rule enforcement. This paper highlights the system's ability to adaptively detect and miti- gate DDoS attacks while maintaining low false alarms and preserving legitimate network traffic.

II. BACKGROUND AND KEY TECHNOLOGIES

- **Distributed Denial-of-Service (DDoS)**

DDoS attacks are malicious attempts to render a service, server, or network resource unavailable to legitimate users by overwhelming it with a massive influx of traffic. Unlike traditional Denial of Service (DoS) attacks, which typically originate from a single source, DDoS attacks utilize multiple sources—often organized into a bot- net—to generate disruptive traffic. These attacks aim to exhaust the target's bandwidth, CPU, or memory, causing slowdowns or complete service outages. Any publicfacing network infrastructure is potentially vulnerable to such attacks, making timely detection and mitigation crucial.

- **Software Defined Networking (SDN)**

SDN is an advanced networking approach that decouples control functions from the physical hardware, allowing for centralized oversight and customizable network management through software. Instead of relying on tradi- tional hardware-configured routers and switches, SDN allows administrators to manage network behavior using software applications via open interfaces like OpenFlow. This programmability not only enhances visibility and control but also provides the agility needed to respond to fast-changing network threats such as DDoS attacks. [2]

- **RYU Controller**

RYU is a Python-based, open-source framework designed for building and managing Software Defined Networking (SDN) controllers. It supports a range of networking protocols, including OpenFlow, and offers developers a simple interface to design and deploy custom SDN applications. In our implementation, RYU is responsible for monitoring network flows, extracting traffic features, and applying flow control rules based on predictions from the machine learning model.

- **Mininet**

Mininet is a lightweight, open-source emulator used to prototype SDN environments on a single machine. It can simulate network components like switches, hosts, and links, offering a practical testbed for evaluating SDN applications before deploying them in real networks. Its simplicity and flexibility make it an ideal tool for research and educational purposes.

- **Random Forest Algorithm**

Random Forest is a powerful ensemble machine learning algorithm that builds multiple decision trees and com- bines their outputs to make more accurate predictions. It is particularly effective in handling large and com- plex datasets with high-dimensional features. For DDoS detection, Random Forest provides high classification accuracy, resilience to overfitting, and robustness against noisy data, making it well-suited for analyzing real-time network traffic.

III. Related Work

Several studies have explored the use of ML in SDN environments to detect malicious traffic.

- Dong Li et al. (2018) compared various classifiers and found Random Forest to outperform others in SDN setups [1].
- Yijie Li et al. (2019) proposed Deep Belief Networks, showing strong performance but at the cost of computa- tional complexity [2].
- Other research such as Peng Xiao et al. (2015) introduced correlation-based methods for data center DDoS detection [3].
- Fatima Khashab et al. (2021) combined entropy analysis with SVM for real-time classification in SDN [4].

While effective, these approaches either lack scalability or require significant computational resources. Our work differentiates by using a lightweight yet robust Random Forest model integrated directly within the SDN controller for real-time classification and mitigation, ensuring low overhead and high accuracy.

IV. System Architecture

The proposed architecture is designed with modularity and scalability in mind and is structured into three distinct layers that work together to enable real-time DDoS detection and mitigation.

A. Architectural Layers

Software-Defined Networking (SDN) is a network archi- tecture approach that enables the network to be intelligently and centrally controlled, or ’programmed,’ using software applications.
SDN architecture includes three layers

- **Application Layer:** This topmost layer focuses on providing intelligence and policy-making capabilities. It hosts the machine learning logic—including the pre- trained Random Forest model—that analyzes traffic flow features and makes classification decisions. Additionally, it supports tasks such as network virtualization, traffic monitoring, and the enforcement of security policies.
- **Control Layer:** This middle layer contains the SDN controller, specifically the RYU controller in our implementation. It acts as the intermediary between the appli- cation layer and the underlying network infrastructure. The controller manages communication with switches, collects flow statistics, processes Packet-In messages, and enforces decisions made by the ML model by installing or modifying flow rules in real-time.
- **Data Layer:** This foundational layer consists of OpenFlow-compatible SDN switches and network de- vices. It is responsible for forwarding packets based on the flow rules provided by the controller. In our setup, Mininet emulates this environment with hosts, virtual switches, and traffic flows representing both benign and attack traffic.

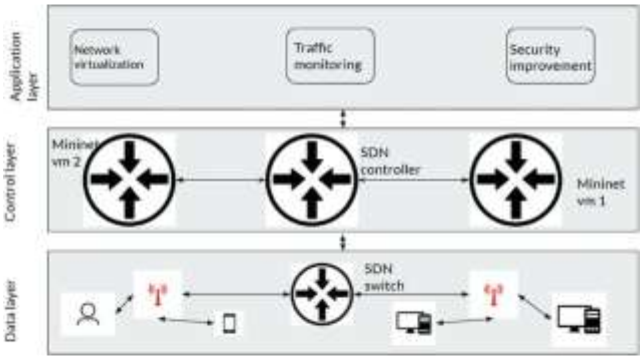


Fig. 1: SDN-based Architecture for DDoS Mitigation

V. Workflow and Methodology

The complete detection and mitigation pipeline involves the following stages:

- **Traffic Generation:** Benign and malicious traffic are generated using Scapy and Hping3, simulating real-world scenarios including various types of DDoS attacks such as SYN floods and UDP floods.
- **Dataset Generation and Feature Extraction:** Traffic data is generated using Python scripts to simulate both normal and DDoS attack scenarios. The RYU controller collects flow statistics using the OpenFlow protocol. The following features are extracted from each flow:

1) Source IP
2) Destination IP
3) Packet Count
4) Byte Count
5) Flow Duration
6) Packet Rate

These features are stored in CSV format, and labeled as ’normal’ or ’attack’ based on known traffic behavior during simulation.

- **Traffic Classification:** The extracted features are fed into a pre-trained Random Forest classifier, which is opti- mized for high-speed prediction. The classifier determines whether a flow is malicious or benign. This decision is made in near real-time, allowing the controller to im- mediately enforce flow rules based on the classification. The model’s strength lies in its ability to identify patterns in both simple and complex attack signatures without requiring manual tuning or heuristic thresholds.
- **Integration with RYU Controller:** The trained Random Forest model is serialized and loaded into the RYU controller. For every new Packet-In event:the controller extracts the relevant features.

– The controller extracts the relevant features.

– The model predicts whether the flow is malicious or benign.

– Based on the prediction:

∗ **If malicious:** a blocking rule is installed on the switch to drop traffic.

∗ **If normal:** traffic is allowed to pass without disruption.

This setup ensures that malicious traffic is detected and blocked in real-time, enhancing responsiveness.

- **Mitigation Strategy:** The system applies differentiated mitigation strategies depending on the type of attack:

– **For DoS Attacks:** Packets from a single repeated source IP are blocked directly at the switch level.

– **For DDoS Attacks:** Instead of blocking individual IPs, the system disables the port from which the majority of attack traffic originates.

This dual-level mitigation minimizes collateral damage and avoids interfering with legitimate traffic.

- **Visualization and Monitoring:** Tools like Wireshark and RYU’s built-in GUI tools are used for monitoring traffic trends, visualizing anomalies, and confirming the effectiveness of mitigation.
- **Logging and Learning:** Logs are maintained for each classification decision, which can later be used for re- training or enhancing the ML model.

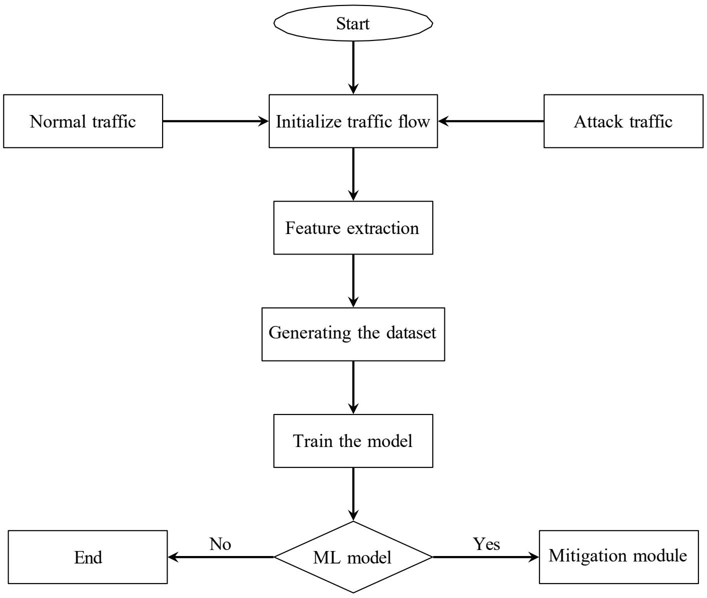


Fig. 2: flow chart

A. Justification of Chosen Technologies

1) **RYU** was chosen due to its Python-based architecture, ease of integration with ML libraries like Scikit-learn, and strong OpenFlow support.

2) **Mininet** provides a cost-effective and flexible testbed to emulate real-time SDN environments without requiring physical hardware.

3) **Random Forest** was selected after comparing its performance with other algorithms such as KNN, SVM, and Decision Tree. It provided the best balance of speed, accuracy, and low false positives.

This layered and systematic architecture ensures that the system can be easily extended, tested, and deployed in various

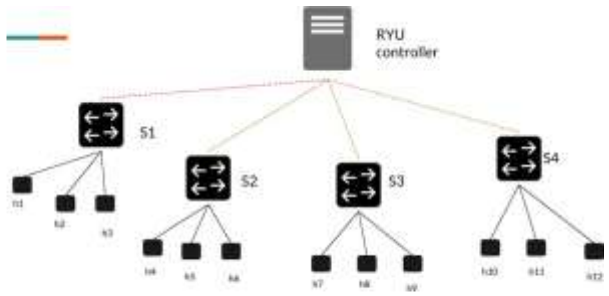


Fig. 3: Network Topology

network environments while maintaining performance and reliability.

VI. Implementation

The proposed SDN-based DDoS detection and mitigation system was implemented using a combination of open-source tools, network emulation environments, and machine learning libraries. This section provides an overview of the experimental setup, model training process, and integration details.

A. Environment Setup

The implementation was carried out in a virtualized environment using the following tools and configurations:

- **Operating System:** Ubuntu 22.04 LTS
- **SDN Controller:** RYU (Python-based)
- **SDN Controller:** RYU (Python-based)
- **Network Emulator:** Mininet 2.3.0
- **Programming Language:** Python 3.10
- **Machine Learning Library:** Scikit-learn
- **Traffic Tools:** Scapy and Hping3 (for generating SYN, UDP, and ICMP floods)
- **Analysis Tool:** Wireshark (for real-time packet inspection)

The virtual topology was created using Mininet, consisting of multiple hosts connected through OpenFlow switches. Some hosts simulated legitimate traffic, while others were designated to perform DDoS-style attacks. The RYU controller was configured to monitor traffic, extract flow features, classify flows, and update switch rules accordingly.

B. Model Training and Testing

The machine learning component, a Random Forest classifier, was trained offline using data collected from simulated traffic scenarios. The training process included the following steps:

1) **Data Collection:** Flow statistics were collected by the RYU controller during traffic simulation in Mininet.
2) **Feature Extraction:** The following features were recorded: Source IP, Destination IP, Packet Count, Byte Count, Flow Duration, and Packet Rate.
3) **Labeling:** Data samples were labeled as “normal” or “attack” based on their behavior during the emulation.
4) **Pre-processing:** The dataset was cleaned and normal- ized to improve model training.
5) **Training:** A `Random Forest Classifier` from Scikit-learn was trained using 80% of the data, with the remaining 20% used for testing.
6) **Evaluation:** The model achieved 97.2% accuracy with a 3.1% false positive rate.

Once validated, the trained model was serialized using joblib and integrated into the RYU controller.

C. Controller Integration and Real-Time Operation

The trained Random Forest model was loaded into the RYU controller to enable real-time classification. When the controller receives a Packet-In message from a switch:

- It extracts relevant flow features.
- Passes them to the ML model for prediction.
- Based on the result:

– Malicious flows are blocked by installing drop rules.
– Benign flows are forwarded as normal.

This ensures minimal delay (average prediction time ¡ 50 ms) and enables dynamic response to ongoing attacks.

D. Visualization and Monitoring

To validate the effectiveness of the system:

- Wireshark was used to monitor packet flows before and after mitigation.
- Visual trends confirmed a significant drop in malicious traffic.
- Controller logs captured rule enforcement actions and classification outcomes for later analysis.

VII. Results and Evaluation

The proposed system was tested in a simulated SDN environment using Mininet and RYU to evaluate its accuracy, speed, and effectiveness in mitigating DDoS attacks.

A. Key Performance Metrics

Metric	Value
Detection Accuracy	97.2%
False Positive Rate	3.1%
Avg. Mitigation Delay	~1.5 sec
Prediction Time per Flow	< 50 ms
Malicious Traffic Dropped	~80%

Detection and Mitigation Metrics

B. Summary

The Random Forest classifier provided high detection ac- curacy with minimal delay, making it suitable for real-time network protection. The system effectively distinguished be- tween normal and malicious flows, and dynamically applied mitigation strategies with low false positives. Compared to traditional rule-based systems and heavier ML models, this approach offers a fast, lightweight, and scalable solution for DDoS defense.

C. Snapshots

Fig. 4: Mini-net Topology Setup

Fig. 5: SDN Controller Launch

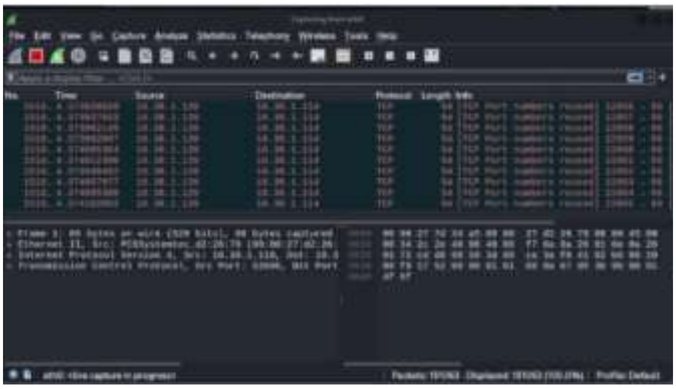
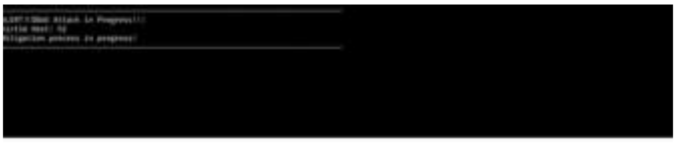


Fig. 6: Traffic Monitoring Interface

Fig. 7: Legitimate Traffic Detected



```
ALERT!!DDoS Attack in Progress!!!
Victim Host: h[illegible]
Mitigation process in progress!
```

Fig. 8: DDoS Attack Detected

Fig. 9: Mitigation Flow Rules Applied

VIII. Discussion

The results obtained from the experimental setup validate the feasibility and efficiency of integrating Machine Learning with Software Defined Networking for real-time DDoS attack detection and mitigation.

The high detection accuracy of 97.2% and a low false positive rate of 3.1% demonstrate the reliability of the Ran- dom Forest classifier in classifying network flows accurately. Additionally, the system's ability to respond in under 50 ms per flow makes it suitable for real-time applications where minimal latency is critical.

The use of RYU as the SDN controller and Mininet for emulation provided a flexible and low-cost environment to validate the solution. The layered architecture proved effective in isolating responsibilities between the detection, control, and mitigation processes.

Compared to existing approaches that use static rules or require high computation for deep learning models, our system offers a lightweight and practical alternative. While other stud- ies may leverage deep neural networks for enhanced feature learning, they typically suffer from longer training times and the need for powerful GPUs—something our solution avoids.

However, the current implementation also has limitations:

- The model is trained offline and requires periodic updates to adapt to new attack patterns.
- The mitigation strategy can be improved to avoid over-blocking in complex DDoS scenarios.
- More advanced traffic classification techniques like online learning, unsupervised anomaly detection, or federated learning could be explored in future work.

Overall, the framework offers an effective combination of performance, scalability, and ease of implementation, positioning it as a strong solution for deployment in practical SDN environments.

IX. Conclusion

This paper focused on the design and implementation of a real-time framework for detecting and mitigating Distributed Denial of Service (DDoS) attacks by integrating Software Defined Networking (SDN) with Machine Learning (ML). A Random Forest classifier was embedded into the RYU controller to enable intelligent, flow-based traffic classification, effectively identifying both DoS and DDoS attacks with high accuracy.

The framework was tested in a simulated SDN environment using Mininet, with traffic generated via tools such as Hping3 and Scapy. The ML model was trained on labeled traffic data, and detection was executed in real time at the data plane. Upon identifying malicious behavior, the system dynamically updated flow rules within switches to drop packets either by source IP or by port.

Key outcomes of the system include:

- High detection accuracy exceeding 97% with a low false positive rate
- Quick response time, with average prediction time below 50 ms
- Minimal disruption to legitimate users and low resource consumption
- Successful mitigation of both DoS and DDoS attack scenarios

The results demonstrate that the combination of SDN's programmability with ML's predictive intelligence can serve as a powerful, adaptive, and efficient defense mechanism for securing modern networks.

X. Future Enhancement

While the proposed system demonstrated strong performance in a controlled emulation environment, there are several key areas where it can be improved or extended for greater scalability, adaptability, and real-world applicability:

- **Deep Learning Integration:** Incorporating advanced deep learning models such as Long Short-Term Memory (LSTM), Convolutional Neural Networks (CNN), or Autoencoders could enhance detection accuracy and adaptability. These models can learn complex patterns over time and reduce the need for manual feature engineering.
- **Multi-Controller SDN Support:** To improve scalability and fault tolerance, the framework can be expanded to operate across multiple distributed SDN controllers. This would allow it to protect larger and more complex network topologies.
- **Real-World Deployment:** Testing the system on hardware-based SDN switches or enterprise-grade routers would help assess its performance in live production environments, including throughput, latency, and compatibility under realistic loads.
- **Online Learning and Adaptive Models:** Future versions of the system could incorporate continuous learning by updating the ML model with live traffic data. Such an approach enhances the identification of zero-day threats and evolving patterns of DDoS attacks.
- **Broader Threat Detection:** The current focus on DDoS can be broadened to detect additional threats such as port scanning, brute-force login attempts, ARP spoofing, and other reconnaissance or exploitation techniques.
- **Enhanced Visualization and Monitoring:** Integrating real-time dashboards using tools like Grafana or the ELK Stack (Elasticsearch, Logstash, Kibana) can provide live insights into network health, threat levels, and mitigation actions for administrators and researchers.

These enhancements would not only increase the system's robustness but also make it suitable for enterprise-scale deployment and long-term adaptability in evolving cyber threat landscapes.

References

[1] J. Mirkovic and P. Reiher, ”A taxonomy of DDoS attack and DDoS defense mechanisms,” ACM SIGCOMM Computer Communication Review, vol. 34, no. 2, pp. 39–53, Apr. 2004.

[2] B. V. Baiju, S. M. Yahiya, P. Akash Raj, and S. S. Farooq, “DDoS Attack Detection Using SDN Techniques,” International Journal of Advanced Research in Engineering and Technology (IJARET), vol. 12, no. 2, pp. 402–411, Apr. 2021.

[3] D. Li, C. Yu, Q. Zhou, and J. Yu, “Using SVM to Detect DDoS Attacks in SDN Network,” *IOP Conference Series: Materials Science and Engineering*, vol. 466, p. 012003, 2018.

[4] Y. Li, B. Liu, S. Zhai, and M. Chen, “DDoS Attack Detection Method Based on Feature Extraction of Deep Belief Networks,” *IOP Conference Series: Earth and Environmental Science*, vol. 252, no. 3, 2019.

[5] P. Xiao, W. Qu, H. Qi, and Z. Li, “Detecting DDoS Attacks Against Data Centers With Correlation Analysis,” *Computer Communications*, vol. 67, pp. 66–74, 2015.

[6] F. Khashab, J. Moubarak, A. Feghali, and C. Bassil, “DDoS Attack Detection and Mitigation in SDN Using Machine Learning,” in *Proc. IEEE 7th Int. Conf. on Network Softwarization (NetSoft)*, 2021.

[7] C. Biradar, “DDoS Attack Detection and Mitigation using SDN,” GitHub repository, 2020. [Online]. Available: https://github.com/chiragbiradar/ DDoS-Attack-Detection-and-Mitigation