IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.125359

"INTELLIGENT DDOS ATTACK:LEVEARGING RANDOM FOREST CLASSIFICATION"

Darshan J Baligeri[1], Dhruva K[2], Gagan K[3] ,Varshith GR[4], Meenakshi H[5]

Student, Department of CS&E, Maharaja Institute of Technology Mysore, India[1]

Student, Department of CS&E, Maharaja Institute of Technology Mysore, India[2]

Student, Department of CS&E, Maharaja Institute of Technology Mysore, India[3]

Student, Department of CS&E, Maharaja Institute of Technology Mysore, India[4]

Assistant Professor, Department of CS&E, Maharaja Institute of Technology Mysore, India[5]

Abstract: This project presents an AI-driven Intrusion Detection System (IDS) designed to detect Distributed Denial-of-Service (DDoS) attacks using advanced machine learning techniques. By leveraging Random Forest, Neural Networks, and Logistic Regression, the system effectively classifies network traffic to distinguish between legitimate and malicious activity. The model is trained on a labeled dataset and optimized for high accuracy and low false positive rates. Through rigorous testing and evaluation, the Random Forest classifier demonstrated superior performance in real-time detection scenarios. The project highlights the potential of machine learning in enhancing cybersecurity and offers a scalable, efficient solution for detecting network-based threats. The system also addresses critical challenges such as minimizing false alarms, handling high-volume traffic, and ensuring fast inference speeds, making it a practical tool for modern network security environments.

INTRODUCTION

In the modern digital landscape, Distributed Denial of Service (DDoS) attacks pose a significant threat to network security by overwhelming systems with excessive traffic, leading to service disruptions. By leveraging artificial intelligence (AI), specifically machine learning algorithms, an advanced Intrusion Detection System (IDS) can be developed to detect and mitigate DDoS attacks more effectively.This AI-driven IDS can analyze vast amounts of network data in real-time, identifying patterns and anomalies indicative of DDoS attacks with high accuracy. The synergy between AI not only improves the detection rates but also reduces false positives and response times, offering a robust solution to safeguard network infrastructure against evolving cyber threats.

LITERATURE SURVEY

The literature survey explores various research studies focused on the detection of Distributed Denial of Service (DDoS) attacks through Intrusion Detection Systems (IDS). Traditional IDS approaches, such as signature-based and anomaly-based detection, have long been used but exhibit limitations in detecting novel or evolving threats. To address these limitations, recent studies have integrated machine learning (ML) techniques to enhance detection performance and reduce false positives. Among these, ensemble models like Random Forest have gained popularity due to their robustness, accuracy, and ability to handle high-dimensional network traffic data. This review highlights different ML-based approaches, analyzes their strengths and weaknesses, and emphasizes the importance of feature selection, real-time analysis, and algorithm evaluation for effective DDoS mitigation.

SOFTWARE REQUIRMENT SPECIFICATIONS

DDoS Detection: The system must accurately detect Distributed Denial of Service (DDoS) attacks using machine learning algorithms, including Random Forest, Logistic Regression, and Neural Networks, trained on diverse network traffic datasets.

Real-Time Analysis: The system should process incoming network traffic in real time, identifying potential threats and providing immediate feedback to the SDN controller for mitigation.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.125359

Algorithm Selection and Integration: The system must evaluate the performance of all three algorithms based on accuracy, precision, recall, and F1- score. The best performing algorithm will be seamlessly integrated with the SDN controller for real time threat response.

Traffic Management and Mitigation: Once an attack is detected, the system should dynamically reconfigure network flows via the SDN controller to mitigate the impact of the attack, ensuring minimal service disruption.

Dynamic Adaptability: The system must adapt to evolving DDoS attack patterns by continuously updating detection thresholds and response strategies, leveraging SDN's programmability and central control features.

Performance Monitoring: Provide a comprehensive dashboard for administrators, displaying real-time network traffic patterns, detection alerts, system performance metrics, and algorithm evaluations. **User Authentication and Security:** Implement secure login, role-based access, and OTP verification for system access, ensuring that only authorized users can interact with the system.

Scalability: The system should handle high volumes of network traffic and support scalability to accommodate growing network infrastructure and increasing attack complexities.

System Requirements

The hardware specifications for the system should support efficient processing and real- time performance:

Processor: A multi-core processor (minimum Intel i5 or equivalent) for smooth real time processing of OCR and TTS tasks

RAM: At least 8 GB of RAM to handle large image files and real-time text-to-speech conversion.

Storage: A minimum of 100 GB of storage for image and audio file storage, along with backup storage.

Input Devices: A standard mouse and keyboard for user interaction, with an optional webcam for real-time image capture.

Output Devices: Speakers or headphones for text-to-speech output.

Software Requirements

The software requirements for this system include the operating system, development platforms, libraries, and frameworks necessary to run the OCR and TTS modules:

Operating System:

Windows 10/11 or Linux-based systems (Ubuntu preferred) for compatibility with the development environment.

Programming Languages: Python:

Core development language for implementing detection algorithms (Random Forest, Logistic Regression, and Neural Network).

Used for real-time traffic analysis, integrating machine learning models, and interacting with SDN controllers (e.g., OpenFlow).

JavaScript:

For frontend development to visualize real-time network traffic, detected anomalies, and mitigation status dynamically.

Used in SDN controllers like ONOS or OpenDaylight for extending functionalities via custom scripts.

HTML/CSS:building a responsive and intuitive user interface for administrators to monitor and control system parameters.

Machine Learning Algorithms: Random Forest:

To identify attack patterns based on traffic features such as packet size, rate, and source distribution.

Handles high-dimensional data effectively.

Logistic Regression:

For baseline classification of normal versus anomalous traffic.

Lightweight and interpretable model for quick decision-making.Neural Network

SYSTEM ANALYSIS

1) Traffic Behavior Understanding:

Real-time traffic is collected from multiple sources to understand normal vs. abnormal patterns. Key data points

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.125359

include IP addresses, port numbers, and packet transmission rates.

2) Data Cleaning and Feature Identification:

Preprocessing is performed to clean and normalize raw traffic data. Important features such as packet size, flow rate, and source/destination information are extracted to help differentiate between benign and malicious traffic.

3) Detection Strategy Using AI/ML:

Multiple machine learning models (Random Forest, Logistic Regression, and Neural Networks) are considered. These models are trained to recognize patterns associated with DDoS attacks and flag anomalies in incoming traffic.

4) Response Mechanism:

On detecting malicious traffic, the system is designed to trigger alerts and take preventive actions such as IP blocking or throttling using programmable network tools like SDN.

5) Monitoring Interface:

A user-friendly interface is planned using Node.js and React.js, allowing administrators to track, manage, and respond to DDoS threats efficiently in real-time.

High Level Design

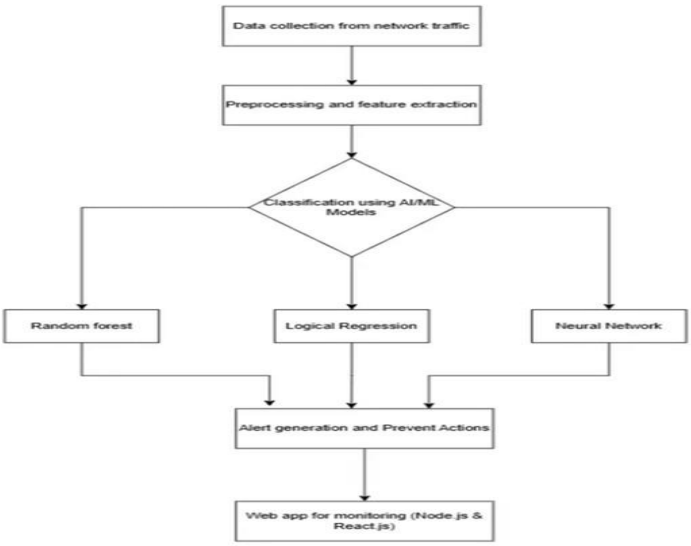


Fig 4.1: System Architecture

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.125359

Low Level Design

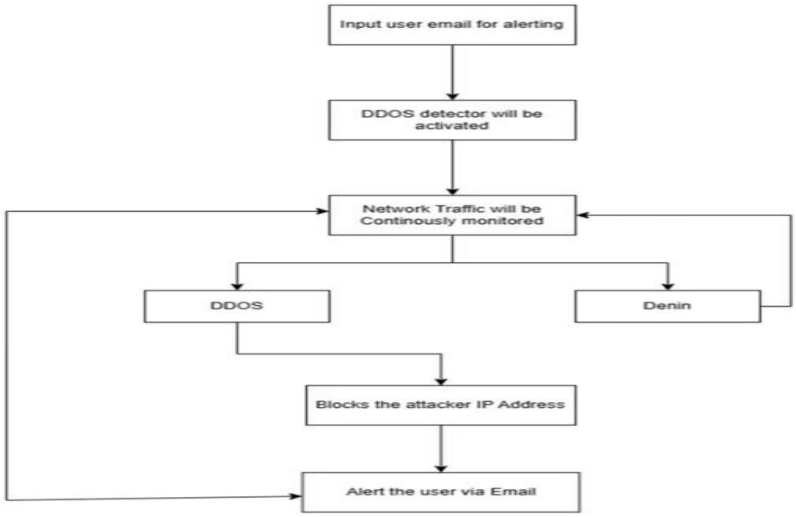
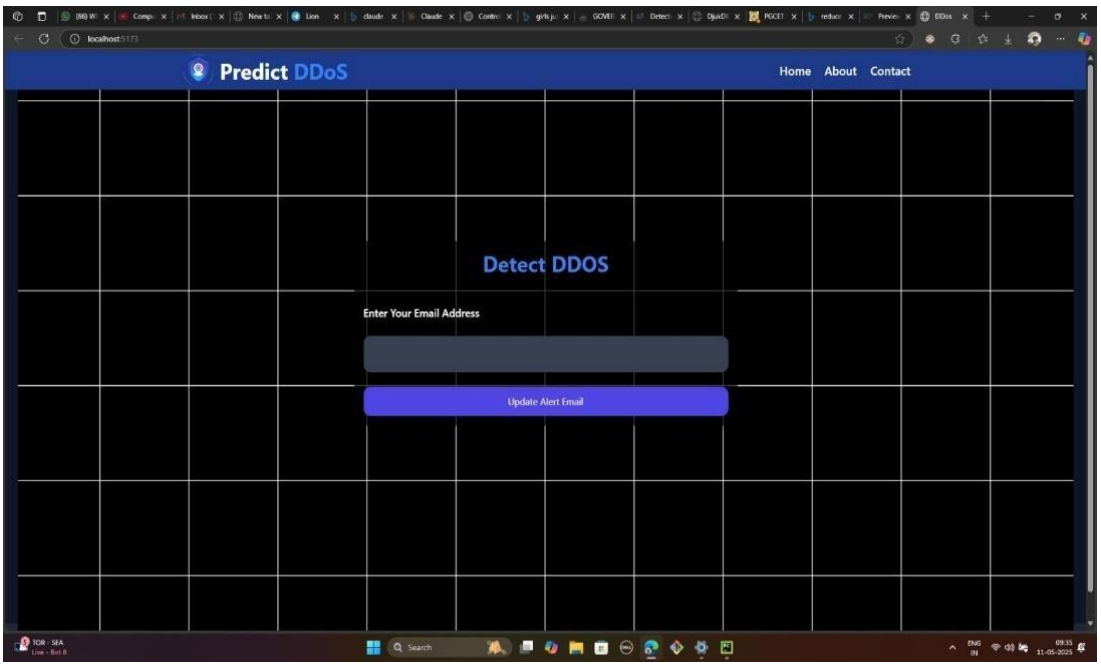


Fig 4.2: System Design Workflow

METHODOLOGY

Objective: Develop and compare multiple learning modelsMethod: Implement three algorithms—Random Forest (RF), Logistic Regression (LR), Neural Network (NN)—using Python's scikit-learn and TensorFlow libraries.

User Interface Design



IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.125359

Control Flow

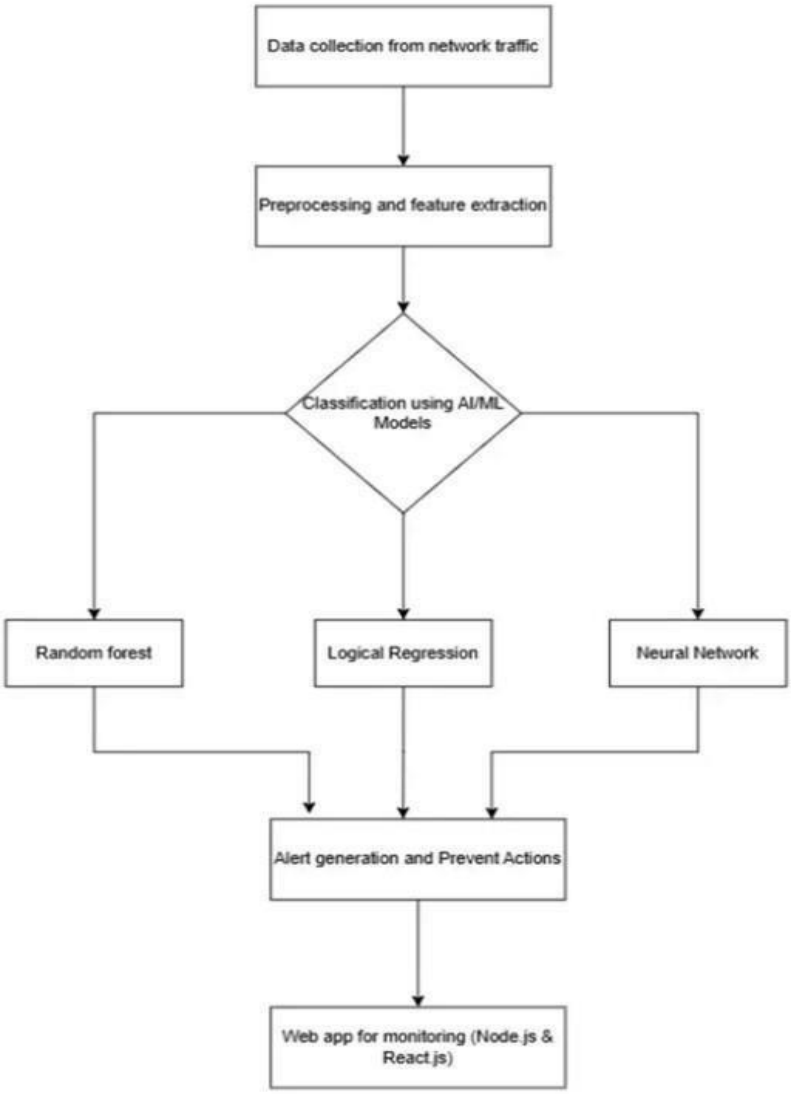


Fig 5.1: System Architecture

TESTING DETAILS

Unit Testing

- Mocking External Dependencies: Network traffic data sources and APIs are mocked to simulate various DDoS scenarios without relying on live data, ensuring reproducible unit test results.
- Validation of Preprocessing Logic: Unit tests ensure that feature extraction (e.g., packet count, flow duration, byte size) is accurate and consistent across different input formats.
- Model Input Consistency Checks: Tests verify that the input data format and feature vector passed to the Random Forest classifier are correct and match the model's expectations.
- Threshold Testing: Unit tests validate decision boundaries or thresholds used in classifying normal vs. malicious traffic.
- Performance Regression Testing: Lightweight performance tests are added to monitor execution time of key components and detect potential slowdowns after code changes.
- Integration with CI/CD Pipeline: Unit tests are integrated into a continuous integration pipeline (e.g., GitHub Actions) to ensure tests are run on every commit and pull request.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.125359

- Random Forest Output Validation: Output from the model (e.g., probability scores or class labels) is tested against known benchmarks to ensure classifier stability.
- Edge Case Handling: Tests are written for edge cases like empty datasets, corrupted packets, or missing values to ensure graceful handling.
- False Positive/Negative Test Simulation: Unit tests simulate specific known attack vectors to ensure proper classification and evaluate misclassification scenarios.

Integration Testing:

End-to-End Data Pipeline Testing: Validates the integration between data ingestion, preprocessing, model inference, and result reporting modules.

- Component Interaction Verification: Ensures smooth communication between modules such as packet capture → feature extraction → classification → logging/alert system.
- Live Traffic Simulation: Synthetic network traffic (including normal and DDoS patterns) is injected to validate real-time processing and classification accuracy.
- Model Integration Check: Confirms that the Random Forest classifier loads correctly and functions within the overall detection pipeline, including retraining/updating mechanisms.
- Database/API Integration: Tests integration with databases or APIs used for storing results or fetching external threat intelligence.
- Alert Generation Verification: Ensures alerts are triggered when the system detects DDoS behavior and that alerting mechanisms (emails, logs, dashboards) function as intended.
- Error Propagation Handling: Verifies that errors in one module (e.g., malformed data packets) do not crash the system but are logged and handled gracefully.
- Real-time Performance Validation: Tests whether all integrated modules work under load conditions and meet timing constraints for live DDoS detection.
- Security & Permission Check: Ensures all inter-module data exchanges follow access permissions and prevent unauthorized data leakage.

- Dependency Version Compatibility: Validates that all libraries and dependencies (e.g., scikit-learn, pandas, socket) work together without conflict across environments.

User Testing

- Stakeholder Feedback Sessions: Conducted sessions with potential users (e.g., network admins, security analysts) to gather feedback on usability, clarity of alerts, and interpretability of results.
- Interface Usability Testing: Evaluated the front-end/dashboard (if applicable) for intuitiveness, accessibility, and clarity in displaying threat classifications and logs.
- Scenario-Based Testing: Users were given specific scenarios (e.g., under attack, normal traffic, false alarms) to test how they interpret system responses and act upon alerts.
- Alert Comprehension Validation: Tested how easily users could understand and respond to DDoS alerts generated by the system—ensuring that terminology and risk levels are user-friendly.
- Feedback on False Positives/Negatives: Collected user opinions on instances where normal traffic was flagged (false positive) or attacks went unnoticed (false negative) to refine model thresholds or retraining logic.
- User Training & Documentation Review: Assessed the effectiveness of user guides, documentation, and onboarding materials by observing how quickly a new user could set up and run the system.
- Error Reporting Mechanism: Verified that users could easily report bugs or anomalies through the interface or logs, supporting ongoing improvement.
- Custom Configuration Testing: Users tested options like setting alert thresholds, enabling/disabling modules, or tuning detection sensitivity to see if configurations met their operational needs.
- Satisfaction Surveys & Feedback Forms: Distributed surveys after hands-on use to collect qualitative and quantitative feedback about system satisfaction, usability, and trustworthiness.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.125359

RESULTS DISCUSSION

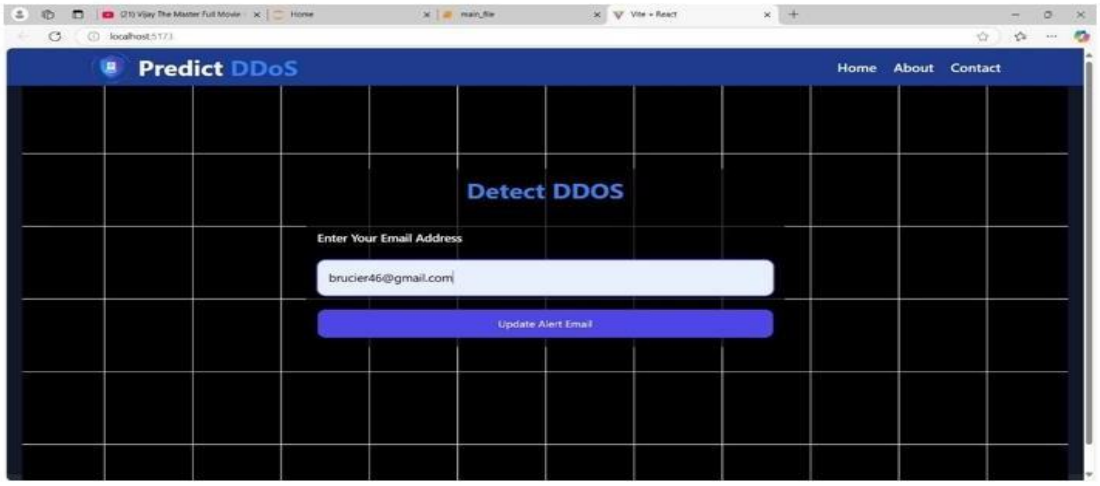


Fig 7.1: Predict DDoS: A Web Interface for DDoS Attack Detection and Alert System – Featuring Email Alert Configuration.

Fig 7.4: AI-Powered DDoS Detection System Monitoring in Action

RESULT Discussion

The DDoS detection system based on the Random Forest classification algorithm demonstrated promising results in accurately identifying distributed denial-of-service attacks from normal network traffic. The model was trained on a labeled dataset comprising both benign and malicious traffic patterns. The major outcomes are asfollows: After training, the models are tested using the 30% test data, and their performance is measured using key metrics:

- Accuracy: Measures how often the model makes correct predictions.
- F1 Score: A balance between precision and recall.
- Precision: Measures how many detected DDoS attacks are actually correct.
- Recall: Measures how well the model detects actual DDoS attacks.

The performance of each model is recorded:

(i) Random Forest(Best Performance)

- Accuracy: 99.95%

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 5, May 2025

DOI: 10.17148/IARJSET.2025.125359

- F1 Score: 99.95%
- Precision: 100%
- Recall: 99.90%

(ii) Logistic Regressi

- Accuracy: 94.48%
- F1 Score: 95.00%
- Precision: 90.98%
- Recall: 99.39%

(iii) Neural Network

- Accuracy: 98.50%
- F1 Score: 98.57%
- Precision: 98.50%
- Recall: 98.64% Observation:

- Random Forest performs the best with nearly 100% accuracy.
- Neural Network shows high accuracy but slightly lower than Random Forest.

CONCLUSION

The Intrusion Detection System for DDoS Attack Detection successfully integrates advanced machine learning techniques to enhance network security and real-time threat detection. By leveraging Neural Networks, Random Forest, and Logistic Regression, the system efficiently classifies network traffic and identifies potential DDoS attacks with high accuracy and minimal false positives. This project demonstrates the effectiveness of AI-driven cybersecurity solutions in addressing real-world challenges such as network intrusion detection and proactive threat mitigation. The system's robust architecture ensures scalability, low-latency detection, making it adaptable for dynamic network environments. Through this initiative, we have gained handson experience in machine learning model optimization, Challenges such as minimizing false negatives, optimizing model inference speed, and handling high-volume network traffic provided valuable insights into the complexities of deploying AI-based cybersecurity solutions in production environments. Overall, this system lays the foundation for future enhancements, including multilayered attack detection, adaptive learning models, and real-time response automation. It highlights the transformative potential of AI in strengthening cybersecurity frameworks and ensuring resilient network infrastructures against evolving cyber threats

REFERENCES

[1]. Al-Razib, M., & Islam, S. (2020). A Survey of DDoS Attack Detection Methods in SDN: Challenges and Future Directions. Journal of Computer Networks and Communications.
[2]. This paper surveys DDoS attack detection techniques in environments and discusses the challenges and potential future solutions.
[3]. Wang, T., & Wang, J. (2019). DDoS Attack Detection Using Machine Learning in : A Survey. International Journal of Computer Science and Network Security, 19(2), 12-19.
[4]. This survey focuses on machine learning-based methods to detect DDoS attacks in SDN networks.
[5]. Rashid, A., & Islam, M. R. (2021). A Hybrid Deep Learning Model for DDoS Attack Detection in Software Defined Networks. Journal of Network and Computer Applications, 79, 102880.
[6]. This paper proposes a hybrid deep learning model to effectively detect DDoS attacks in SDN environments.
[7]. Hussein, A. M., & Saeed, A. H. (2020). DDoS Detection in SDN-Based Networks Using Machine Learning Algorithms. Proceedings of the International Conference on Networking and Communications Systems, 98-104.
[8]. This study explores various machine learning algorithms for detecting DDoS attacks in SDN based networks.
[9]. Sharma, V., & Sharma, S. (2021). Real-Time DDoS Attack Detection and Mitigation in SDN Using AI Techniques. Journal of Cyber Security Technology, 5(3), 156-173.
[10]. This paper presents real-time DDoS attack detection using AI techniques in SDN networks, along with potential mitigation strategies.
[11]. Nguyen, D. C., & Zhang, Y. (2021). Deep Learning-Based Intrusion Detection System for SDN Against DDoS Attacks. IEEE Transactions on Network and Service Management, 18(4), 3456-3470.
[12]. This work investigates the application of deep learning in detecting DDoS attacks within SDN environments.

