FDRP Journal’s

INDJCST-0000905

03

FDRP

Journal’s

Document Details

Submission ID

trn:oid:::1:3171284198

Submission Date

Mar 3, 2025, 11:14 AM GMT+7

Download Date

Mar 3, 2025, 11:21 AM GMT+7

File Name

INDJCST-0000905.docx

File Size

544.4 KB

10 Pages

3,173 Words

19,715 Characters

8% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

23 Not Cited or Quoted 7%
Matches with neither in-text citation nor quotation marks

1 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

6%	Internet sources
4%	Publications
3%	Submitted works (Student Papers)

Match Groups

23 Not Cited or Quoted 7%
Matches with neither in-text citation nor quotation marks

1 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

6%	Internet sources
4%	Publications
3%	Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	**www.ncbi.nlm.nih.gov**	<1%
2	Publication	**"Computing and Machine Learning", Springer Science and Business Media LLC, 20...**	<1%
3	Internet	**sairamce.edu.in**	<1%
4	Internet	**www.biorxiv.org**	<1%
5	Internet	**www.gsconlinepress.com**	<1%
6	Student papers	**University of Hertfordshire**	<1%
7	Student papers	**University of Central Lancashire**	<1%
8	Student papers	**University of Nottingham**	<1%
9	Publication	**Osama S. Younes. "A hybrid deep learning model for detecting DDoS flooding att...**	<1%
10	Internet	**www.engpaper.com**	<1%

11	Internet	papers.academic-conferences.org	<1%
12	Internet	www.frontiersin.org	<1%
13	Internet	link.springer.com	<1%
14	Internet	nccur.lib.nccu.edu.tw	<1%
15	Internet	www.mdpi.com	<1%
16	Publication	Anna N. Gummadi, Osvaldo Arreche, Mustafa Abdallah. "A systematic evaluation ...	<1%

Adaptive AI Framework for Anomaly Detection and DDoS Mitigation in Distributed Systems

Karthik Kamarapu[1], Kali Rama Krishna Vucha[2]
[1]Independent Software Researcher, Osmania University, Hyderabad, India.
[2]Independent Software Researcher, Acharya Nagarjuna University, Guntur, India.
[1]karthikkamarapu@ieee.org

Abstract - *Distributed Denial of Service (DDoS) attacks remain a major challenge for distributed systems, given the traditional detection mechanisms often fail to address the scalability and complexity of modern network architecture. This paper explores the integration of Artificial Intelligence (AI) techniques for anomaly detection and the proposed approach leverages machine learning models to detect and respond to malicious traffic patterns in real-time. This study presents a comprehensive review of AI based techniques, their application within distributed systems, and comparison against conventional methods. This study also demonstrates that AI driven anomaly detection offers effective defense against evolving DDoS threats.*

Key Word: *Anomaly Detection, Distributed Systems, DDoS Mitigation, Machine Learning, Network Security*

I. Introduction

The decentralized nature of the distributed systems provides improved scalability and fault tolerance but also has complex security challenges. One of the most disruptive threats is the Distributed Denial of Service (DDoS) attack that aims to deplete system resources and cause significant financial and operational impacts [1], [2]. Conventional DDoS detection methods such as signature-based and threshold-based techniques have shown significant limitations against evolving attack patterns. Signature-based methods are highly dependent on known patterns and ineffective against new type of threats [3]. Similarly, threshold-based methods can have high false positive rates and may not be effective against legitimate traffic surges [4]. As a result, there is an urgent need for more advanced and intelligent detection systems that can work effectively within a distributed architecture.

Artificial Intelligence has emerged as one of the solutions for enhanced security in the distributed systems. AI driven anomaly detection uses patterns in the network traffic to identify any deviations that can signal potential attacks. By continuously learning from the incoming data, these models can adjust to new attack vectors and provide a proactive defence against DDoS threats. However, existing AI based methods face several challenges. The lack of explainability in many models reduces their trust factor and makes it difficult for administrators to understand act on the predictions [5]. Moreover, suboptimal feature selection hinders the efficiency and accuracy of the detection system and increases resource overhead [6]. Real-time detection remains significant hurdle as many AI techniques require high compute and fail to meet latency requirements of the large-scale distributed systems [7]. Additionally, most existing frameworks are not explicitly designed for cloud and edge computing architectures which limits their scalability and robustness [8].

To address these challenges, this paper presents a new AI-driven framework for real-time anomaly detection and mitigation in the distributed systems. This framework integrates explainable AI techniques that provides clarity to the detection process and enables more informed decision-making. By leveraging optimized feature selection methods, the proposed solution improves detection accuracy and operational efficiency. Furthermore, the architecture is explicitly designed to operate effectively in cloud and edge environments. A comparison of the framework against both traditional and AI-based methods demonstrate how it is superior in terms of detection accuracy, scalability and adaptability.

II. Proposed Framework

This section introduces the proposed AI-driven framework for anomaly detection and mitigation of DDoS attacks in distributed systems. This framework integrates ML models for real time anomaly detection, optimized feature selection and explainable AI techniques to improve transparency. It is specifically designed for hybrid cloud and edge computing environments.

Data Collection Layer

The data collection layer is responsible for monitoring and capturing network traffic across distributed nodes. Sensors at key network points such as gateway, routers and edge devices ensure traffic monitoring that collect essential features including packet size, source and destination IP addresses, protocol types and flow duration which forms the dataset for anomaly detection.

Preprocessing tasks are performed at edge nodes to handle huge volumes of data generated by distributed systems which include filtering not relevant data, aggregating flows into sessions and compressing data for efficient storage and transfer. This data is then transmitted to central cloud storage for further analysis. The use of distributed sensors and edge-based preprocessing reduces network overhead and ensures minimal latency.

Processing Layer

The processing layer is the core of the framework. This layer performs feature selection, anomaly detection and explainability tasks. Feature selection techniques are applied to ensure only the most relevant features are used for anomaly detection. By selecting features such as traffic intensity pattern, abnormal packet intervals and flow irregularities, the framework optimizes computation efficiency and improves accuracy of detection.

Machine learning models are the core of the anomaly detection process. Models such as Random Forest and Gradient Boosted Trees are trained on historical traffic datasets to identify patterns that can indicate DDoS attacks. These models are capable of classifying anomalies in real-time and enables the framework to respond dynamically to evolving threats. Explainable AI (XAI) methods such as SHAP (SHapley Additive exPlanations) are integrated to improve the interpretability of the models. XAI provides insights into the overall importance of the features and explains individual predictions that enables network administrators to understand why specific traffic flows were flagged as anomalous.

Mitigation layer

The mitigation layer manages the response to detected anomalies and implements automated strategies to minimize the impact of DDoS attacks. When an anomaly is detected, the system initiates mitigation actions such as rate-limiting suspicious traffic, blacklisting malicious IPs, or redirecting attack traffic to a scrubbing center. These actions are executed dynamically and updated continuously based on feedback from detection models.

To ensure scalability, the mitigation layer distributes tasks between cloud and edge nodes. Lightweight models operate on edge devices for immediate and low-latency responses, while more complex analyses are conducted in the cloud. This hybrid approach balances speed and computational efficiency and ensures that the system remains effective in large-scale and latency-sensitive environments.

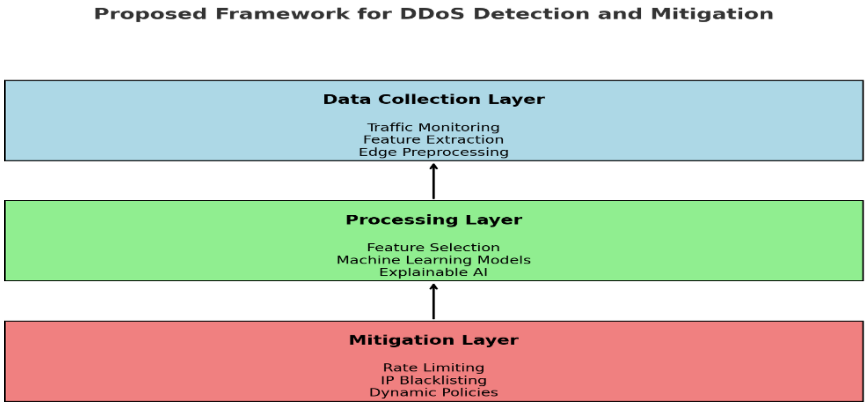

Fig. 1 Proposed Framework layers

2.1 Traffic Monitoring and Data Collection

The section goes into the details of its structure and functionality.

Traffic Monitoring

This step is initiated at distributed nodes, including IoT devices, edge nodes, and gateways to capture network traffic. Network sensors collect package-level details such as source and destination IP addresses, protocol type, packet size and timestamps. To facilitate this, tools like Scapy, Wireshark and tcpdump are employed for capturing packets.

Edge-Based Preprocessing

Data is pre-processed at edge nodes to minimize the usage of bandwidth and ensure low-latency anomaly detection. This task includes filtering irrelevant data and compressing data for efficient storage and transmission.

Data Transmission

Pre-processed features are serialized to JSON and is transmitted from edge nodes to centralized cloud storage for aggregation and deeper analysis. Real-time streaming tools Apache Kafka facilitates efficient transfer of this data that ensure low-latency communication between distributed nodes and the cloud.

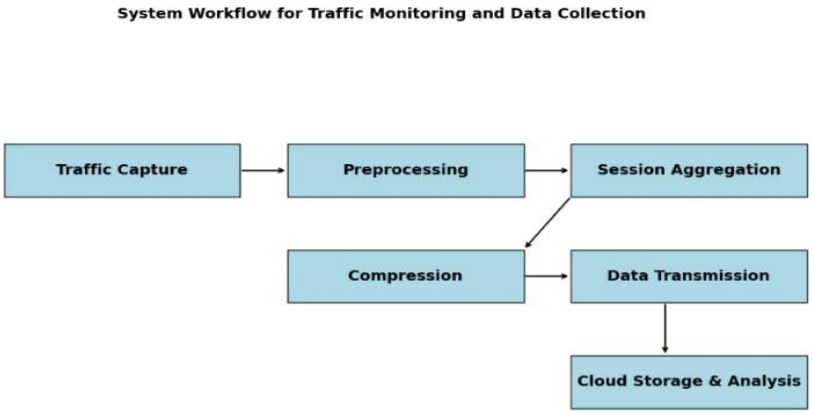

Fig. 2 Traffic monitoring and data collection

Table no 1 Tools used for data collection, processing and transmission

Function	Tool/Framework	Role
Packet Capture	Tcpdump, Wireshark	Capture raw traffic packets
Data Preprocessing	Scapy, PyShark	Filters and aggregates traffic
Transmission	Apache Kafka	Streams data to cloud

2.2 Processing Layer

This layer performs critical tasks such as feature selection, anomaly detection using machine learning models and explainability to improve decision-making. This layer bridges traffic data and actionable insights that enable real-time DDoS detection and response.

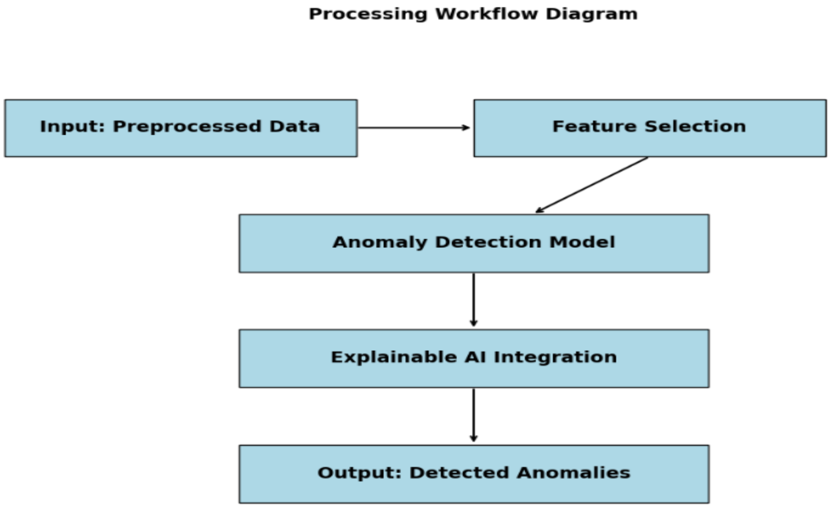
Processing Workflow Diagram
Input: Preprocessed Data
Feature Selection
Anomaly Detection Model
Explainable AI Integration
Output: Detected Anomalies

Fig.3 Processing workflow diagram

Feature Selection

This step is to identify the most relevant features in pre-processed data that ensures computational efficiency and improves model accuracy by removing irrelevant or redundant features. The below code identifies most important features such as packet size, source IP diversity, flow duration and protocol type which are used for training the anomaly detection model.

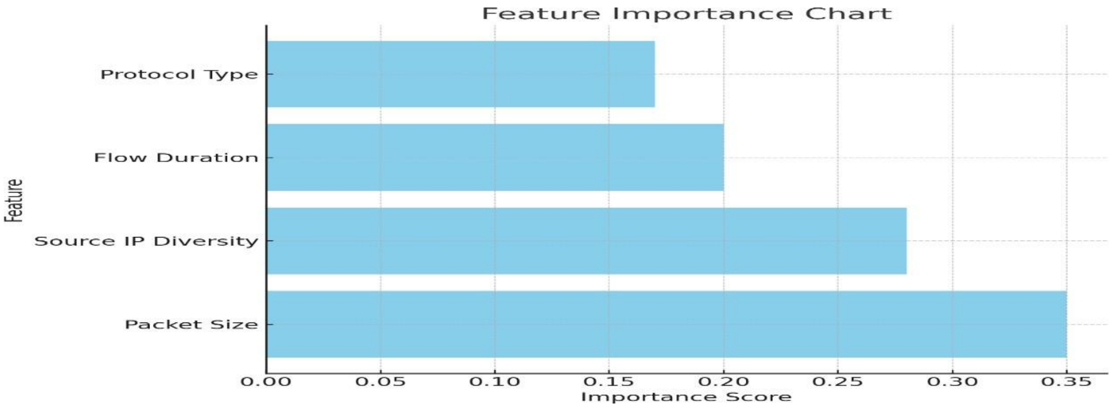

Fig.4 Feature Importance Chart

Anomaly Detection

Once the features are selected, machine learning models are trained to detect anomalies in the data. These models classify traffic into normal or attach categories. Algorithms RandomForest is used for anomaly detection tasksdue to their ability to handle complex patterns in data. The trained model is deployed for real-time detection tasks and classifies incoming traffic as normal or malicious.

Explainability with SHAP

Explainable AI(XAI) techniques such as SHAP (SHapley Additive exPlanations) are integrated for machine learning models that provide global and local insights on how features influence model predictions. This improves trust and interpretability. The below code generates summary plot showing the contribution of each feature to the model's prediction that helps administrators with the decision-making process.

2.3 Explainable AI Integration

This layer provides transparency into decision-making process of machine learning models. This integration ensures that predictions made by anomaly detection models are interpretable and actionable, improves trust, debugging and operational decision-making.

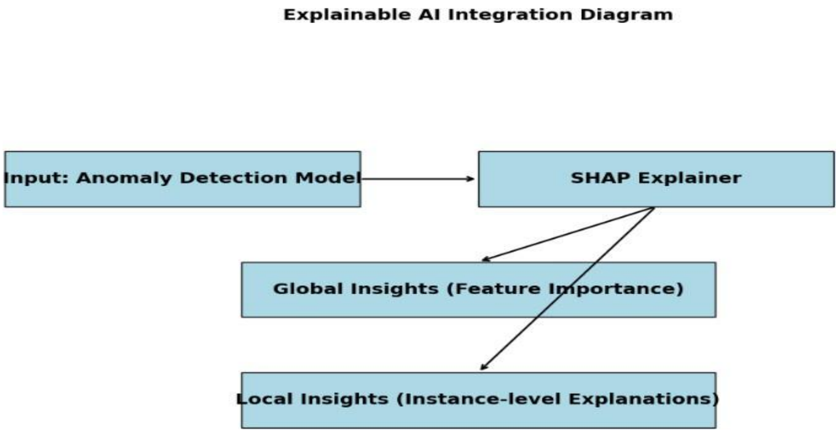
Explainable AI Integration Diagram
Input: Anomaly Detection Model
SHAP Explainer
Global Insights (Feature Importance)
Local Insights (Instance-level Explanations)

Fig. 5 Explainable AI Integration

Global Explainability

This focuses on understanding the overall behaviour of machine learning model. It identifies the features that most influence the predictions across the dataset. For instance, traffic attributes such as packet size, source IP diversity, or protocol type may become key indicators of anomalies. Global explainability highlights the patterns and trends the model learns during training. These insights help network administrators in verifying the model's behaviour and refining the feature selection processes for improved performance.

SHAP (SHapley Additive exPlanations) is used to compute feature importance. It assigns each feature a contribution score based on its influence on model's predictions. For example, SHAP generates summary plots where features are ranked by their impact that enables admins to visualize the model's decision priorities. This transparency ensure that critical features are identified, and irrelevant or redundant ones are excluded.

Local Explainability

This focuses on individual predictions that offer granular insights into why a specific traffic instance was classified as normal or anomalous. This level of details is important in DDoS detection where rapid and accurate responses depend on understanding the logic behind predictions. Local explainability generates explanations for single predictions and shows how feature values contribute to the outcome. For instance, if traffic flow is flagged as anomalous, SHAP can reveal that unusually high packet rates and frequent source IP changes are the main contributors.

Table no 2. Individual Predictions

Instance	Prediction	Top Features Contributing	Explanation
1	Attack	Packet Size, Flow Duration	High packet rate and long flow duration detected.
2	Normal	Protocol Type, Source IP Count	Common protocol with consistent IP diversity.

Implementation with SHAP

The XAI integration in this framework leverages SHAP which is a well-known framework that explains predictions by approximating Shapley values. These values are derived from cooperative game theory and represent the contribution of each feature to the model's output. SHAP operates on a variety of machine learning models including Random Forest and Gradient Boosted Trees.

The implementation begins by trading the anomaly detection model on preprocesses traffic data. The training model is then passed to a SHAP explainer which calculates global and local explanations. Global explanations are used to generate important features, and local explanations are used for instance level visualizations. These outputs are presented to admins through dashboards or reports which enable them to act on model's predictions.

Significance of Explainability

The integration of XAI transforms the anomaly detection into a collaborative tool that combines automated precision with human interpretability. In the context of DDoS detection, XAI's role is pivotal in ensuring operational confidence and accuracy. The dual perspective of global and local explainability enables admins to optimize the framework's performance while making informed and timely decisions during real-world deployments.

2.4 Mitigation Layer

The layer responds to the detected anomalies such as DDoS attaches. This layer translates the outputs of anomaly detection system into actionable countermeasures and ensures the system remains operational even under malicious traffic. It dynamically applies preconfigured strategies to mitigate the impact of the attacks.

Anomaly Response

When the anomaly detection system identifies suspicious activity, the mitigation layer applies the appropriate response based on the predefined rules. Actions such as rate limiting, IP blacklisting etc. are executed to reduce the impact of the anomaly. For example, traffic coming from an IP address associated with unusually high packet rates may be throttled or blocked entirely.

Policy Decision Engine

This is the core of the mitigation layer that converts anomaly detection into actionable mitigation plans. This engine evaluates various parameters such as the type and severity of the anomaly and current network conditions to determine the effective response. For instance, if an attack persists even after initial mitigation efforts, the engine can escalate the responses by tightening rate limits.

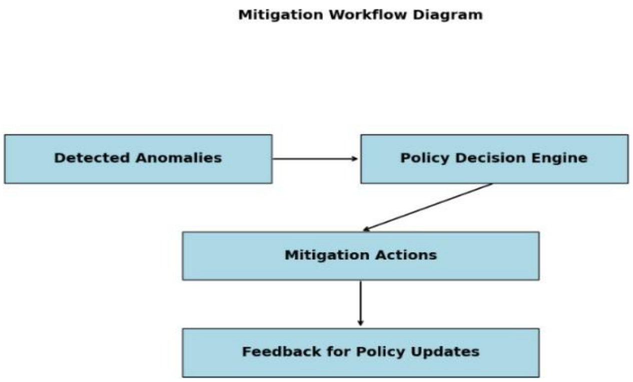

Fig.6 Mitigation workflow diagram

Dynamic updates and feedback

This layer operates as a feedback-driven system that continuously monitors the effectiveness of applied responses and updating policies in real time. This feedback loop ensures that the system remains responsive to evolving attacks and if the mitigation actions fail, the policy decision engine revisits the strategy and adapts accordingly. For instance, an ongoing volumetric attack might prompt the system to lower the traffic thresholds, deploy additional edge nodes for mitigation or prioritize critical services over less essential ones.

Operational workflow

The mitigation layer operates in a cyclical workflow. Inputs the anomaly detection system trigger the policy decision engine which determines the appropriate action. These actions are implemented across the network with continuous monitoring. The feedback loop provides real-time updates to the policy decision engine that enables it to adapt to persistent or evolving threats.

III. Experimental Setup and Results

This framework is tested in a controlled environment simulating a real-world distributed system. The setup consists of both edge and cloud components that are designed to evaluate the three layers of data, processing and mitigation.

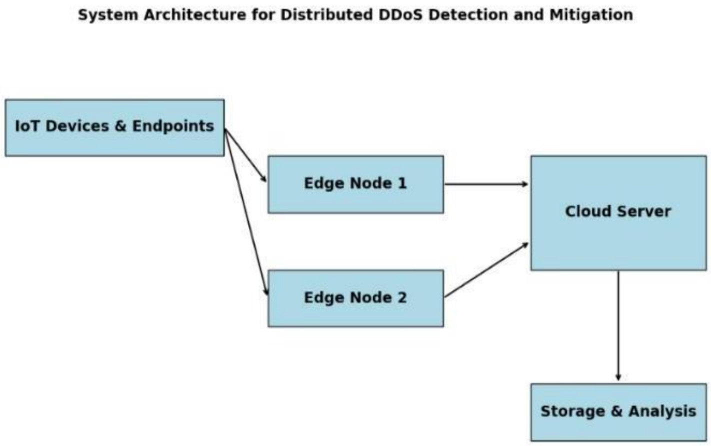

Fig.7 Architecture for DDoS Detection and Mitigation

Environment Configuration

The testbed includes multiple edge nodes deployed on virtual machines and a centralized cloud server. Edge nodes are responsible for traffic monitoring, pre-processing and lightweight anomaly detection and the cloud server performs deep analysis, training and policy management. The environment is orchestrated using Kubernetes which ensures scalable and distributed deployment.

Datasets

The evaluation leverages both real-world and synthetic datasets. The primary dataset used is the CICDDoS2019 dataset which contains labeled traffic data that represents various types of DDoS attacks e.g., UDP flood, SYN flood, HTTP flood and normal traffic. Synthetic traffic data is also generated using tools Hping3. This allows testing of the framework under diverse and evolving attack scenarios.

Traffic Simulation and Evaluation Metrics

Traffic is simulated across edge nodes that resemble real-world scenarios. Edge nodes capture and pre-process traffic before transmitting it to the cloud. Simulating DDoS attacks are introduced intermittently to evaluate the detection and mitigation response of the framework. The framework's performance is measure using detection metrics, mitigation metrics and data efficiency

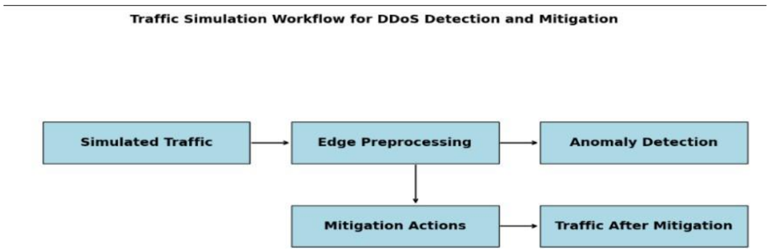

Fig.8 Traffic simulation

Results

The results validate the effectiveness of the proposed framework across all three layers.

Data Layer**:** On average the pre-processing pipeline reduces raw traffic data size by 40% reducing the volume of data sent to the cloud.

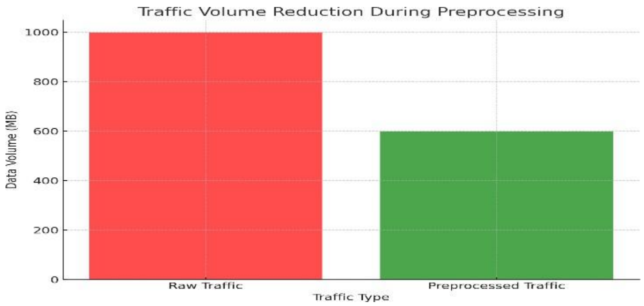

Fig.9 Traffic volume reduction during preprocessing

The pre-processing latency increases linearly with incoming traffic flows. For smaller traffic, say 100 flows per second, the average latency is as low as 5 milli seconds and goes up to 13 milli seconds for 500 flows per second.

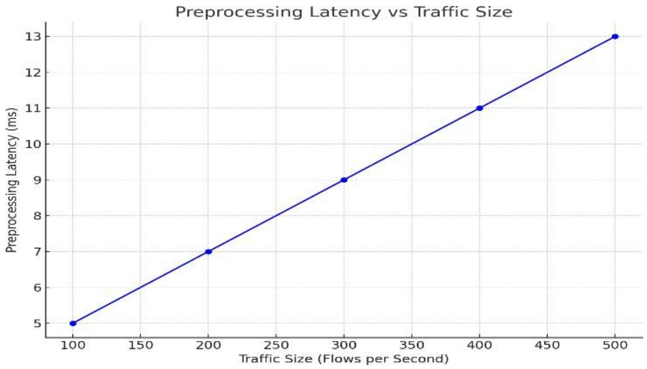

Fig.10 Preprocessing Latency vs Traffic Size

1 Processing Layer: The anomaly detection system demonstrates high accuracy in classifying the traffic. The Random Forest model achieves an accuracy of 98.5% with a precision of 97.2% and a recall of 96.8%. The false positive rate is maintained at 2.1%.

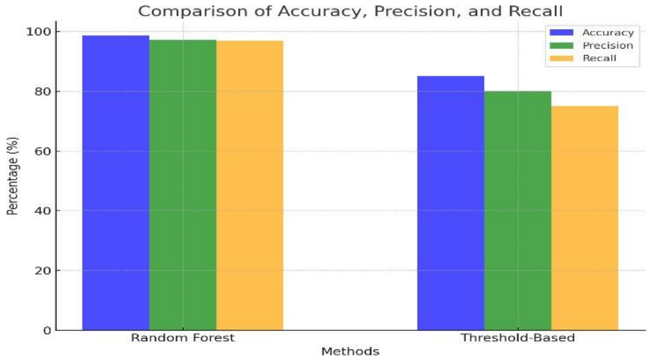

Fig.11 Comparison of Anomaly detection performance

Explainability through SHAP reveals the features like packet size and source IP diversity contribute significantly to anomaly detection.

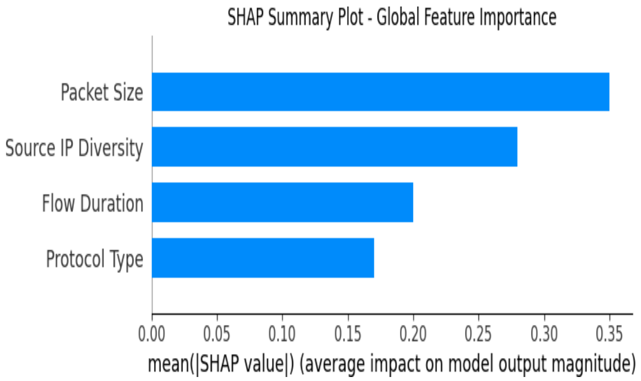

Fig. 12 SHAP Summary Plot

The SHAP decision plot provides a step-by-step breakdown of how individual features cumulatively influences prediction.

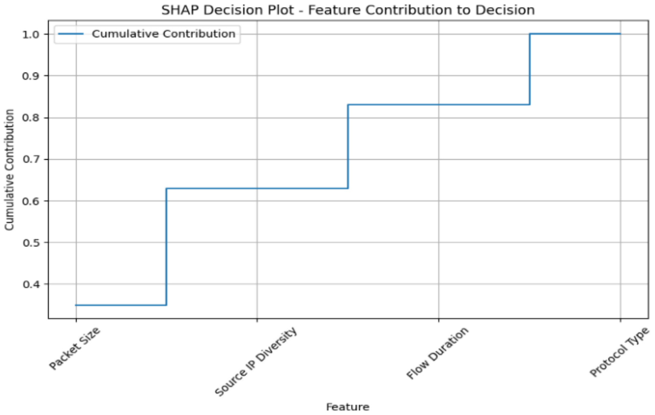

Fig. 13 SHAP Decision Plot

Detection latency for processing layers against traffic size is indicated in the chart below:

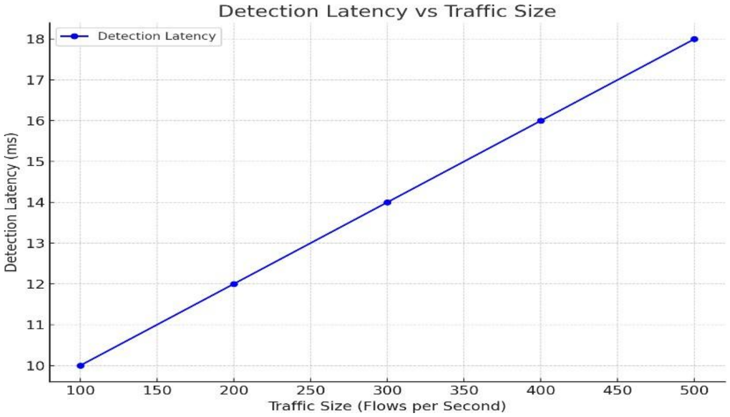

Fig. 14 Detection Latency vs Traffic Size

Mitigation layer: During the simulated SYN flood attack the system decreases malicious traffic by 92% within the first 30 seconds of detection. The average response time for applying mitigation actions, such as IP blacklisting and rate limiting is around 15 milliseconds.

Table no 3. Simulation

Attack type	Traffic before mitigation(req/s)	Traffic after mitigation (req/s)
SYN Flood	1500	120
UDP Flood	2000	180

Below table summarizes response times for different mitigation strategies:

Table no 4. Mitigation strategies

Mitigation Strategy	Average Latency
Rate Limiting	15
IP Blacklisting	12
Traffic Redirection	20

The system continuously monitors the effectiveness of actions and updates policies to handle evolving threats.

Table no 5. Dynamic policy updates

Scenario	Initial Policy	Adapted Policy	Reason for change
Persistent Attack	Threshold:100 req/s	Threshold: 80 req/s	Attack continued after initial
New IP Range Attack	Static IP List	Dynamic Range Detection Added	Evolving attack vector

Comparative Analysis: The framework outperforms baseline approaches including traditional threshold-based methods and non-explainable AI models. Threshold-based methods achieve average detection accuracy of 85% with high false positive rate of 10%. Non-explainable AI models achieve similar accuracy but did not have transparency provided by SHAP that makes it less actionable.

Table no 6. Comparison of proposed and traditional frameworks

Method	Accuracy (%)	FPR (%)	Explainability
Proposed Framework	98.5	2.1	Yes
Threshold-Based Method	85	10	No
Non-Explainable AI Models	98	3	No

IV. Conclusion

The proposed framework for distributed DDoS detection and mitigation effectively addresses the challenges of anomaly detection and traffic control in large-scale systems. The processing layer with machine learning and explainability tools like SHAP delivers an accuracy of 98.5% with minimal false positives. The mitigation layer ensures a consistent reduction of 90% in malicious traffic. By leveraging SHAP, the framework provides transparency into detection decision and finally the dynamic policy detection in the mitigation layer ensure resilience against evolving attack patterns. This research demonstrates the potential for combining machine learning with explainability and adaptive strategies to address cybersecurity challenges. This framework can also be extended to other domains such as IoT security and smart grid protection.

References

[1] N. Mohamed, "DDoS Attacks Mitigation: A Review of AI-Based Strategies and Techniques," in *IEEE Communications Surveys & Tutorials*, 2024. [Online]. Available: https://ieeexplore.ieee.org

[2] B. Bala and S. Behal, "AI techniques for IoT-based DDoS attack detection: Taxonomies, comprehensive review and research challenges," in *Journal of Network and Computer Applications*, 2024. [Online]. Available: https://www.sciencedirect.com

[3] S. Ahmadi, "AI in the Detection and Prevention of Distributed Denial of Service (DDoS) Attacks," in *International Conference on Cybersecurity Research and Innovation*, 2024. [Online]. Available: https://hal.science

[4] N. Moustafa, "A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets," in *Future Generation Computer Systems*, 2021. [Online]. Available: https://www.sciencedirect.com

[5] C. S. Kalutharage, X. Liu, C. Chrysoulas, and N. Pitropakis, "Explainable AI-based DDoS attack identification method for IoT networks," in *Computers*, 2023. [Online]. Available: https://www.mdpi.com

[6] S. Kumar, M. Dwivedi, M. Kumar, and S. S. Gill, "A comprehensive review of vulnerabilities and AI-enabled defense against DDoS attacks for securing cloud services," in *Journal of Information Security and Applications*, 2024. [Online]. Available: https://www.sciencedirect.com

[7] S. A. Varma and K. Ganesh Reddy, "An AI-based IDS framework for detecting DDoS attacks in cloud environment," in *International Journal of Computer Networks & Communications*, 2024. [Online]. Available: https://www.tandfonline.com

[8] O. Polat, S. Oyucu, M. Türkoğlu, H. Polat, and A. Aksoz, "Hybrid AI-Powered Real-Time Distributed Denial of Service Detection and Traffic Monitoring for Software-Defined-Based Vehicular Ad Hoc Networks," in *Applied Sciences*, 2024. [Online]. Available: https://www.mdpi.com