International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Sentinel Sphere: Ai-Driven Dos Detection and Mitigation in Cloud

Aamila Fathima M[1], Haseena M. K[2], Aathila Fathima M[3]

[1,3]Student, Department of Artificial Intelligence and Cyber Security Ilahia College of Engineering and Technology, Kerala, India
[2]Assistant Professor, Department of Artificial Intelligence and Cyber Security Ilahia College of Engineering and Technology, Kerala, India

Abstract

Denial-of-Service (DoS) attacks remain one of the most critical cybersecurity threats targeting cloud-hosted web infrastructures by overwhelming server resources and disrupt-ing service availability. Traditional intrusion detection systems primarily rely on static rule-based techniques or supervised ma-chine learning models that require extensive feature engineering and offline training datasets, limiting their adaptability against dynamically evolving attack patterns.

This paper proposes Sentinel Sphere, an autonomous AI-driven framework for real-time DoS detection and mitigation using Large Language Model (LLM)-based contextual reasoning. The proposed system integrates real-time NGINX access log monitoring with the Mistral 7B Large Language Model deployed locally using the Ollama runtime environment. A Python-based monitoring agent continuously extracts traffic features including requests per second (RPS), IP concentration ratio, endpoint access frequency, and HTTP error response percentages. These traffic metrics are dynamically analyzed by the LLM to identify abnormal traffic behavior and trigger automated mitigation mechanisms including request rate limiting and connection con-trol within the NGINX server.

Experimental evaluation was conducted using ApacheBench-generated attack traffic and benchmark intrusion datasets in-cluding CICIDS2017 and NSL-KDD. The proposed framework achieved 98.2% detection accuracy, 97.4% precision, 96.8% re-call, and 97.1% F1-score while maintaining an average mitigation response latency below 1.2 seconds during high-volume attack scenarios. Comparative analysis against traditional intrusion de-tection approaches demonstrates improved adaptability, reduced response delay, and enhanced service availability.

The results demonstrate that integrating LLM-based reason-ing with real-time infrastructure-level automation provides an effective and scalable approach for building adaptive cyberse-curity defense systems capable of autonomously detecting and mitigating large-scale network attacks in cloud environments.

Keywords: Cybersecurity, Intrusion Detection System, Large Language Models, Cloud Security, DoS Detection, NGINX Security, AI-Based Mitigation

INTRODUCTION

The rapid growth of cloud computing, web-based appli-cations, and distributed digital services has significantly in-creased the importance of reliable and secure network infras-tructures. Modern

organizations rely heavily on web servers to provide uninterrupted access to services such as online transac-tions, cloud storage, and communication platforms. However, this increasing dependency has also made web infrastructures a prime target for cyber attacks. Among these threats, Denial-of-Service (DoS) attacks remain one of the most disruptive and commonly observed forms of cyber attacks. These attacks aim to exhaust server resources such as CPU, memory, and network bandwidth by flooding the system with massive volumes of malicious traffic, ultimately preventing legitimate users from accessing the service.

Traditional defense mechanisms against DoS attacks typi-cally involve firewall filtering, rule-based intrusion detection systems, and signature-based security models. Although these approaches are effective in detecting previously known attack patterns, they often struggle to identify evolving or zero-day attack behaviors. Furthermore, most conventional intru-sion detection systems operate in a passive monitoring mode where alerts are generated but mitigation actions still require manual intervention from system administrators. This delay in response time can significantly increase the impact of an attack on critical infrastructure.

Recent advancements in Artificial Intelligence (AI) and ma-chine learning have introduced new possibilities for intelligent and adaptive cybersecurity systems. Machine learning-based intrusion detection systems have demonstrated promising re-sults in identifying abnormal traffic patterns using statistical models and classification algorithms. However, these models
require large training datasets and feature engineering, which limits their ability to adapt to rapidly changing network environments.

Large Language Models (LLMs) have recently emerged as powerful tools capable of performing contextual reasoning and pattern recognition across structured and unstructured data sources. Unlike traditional machine learning models, LLMs can analyze textual or structured information and infer relationships without requiring explicit feature engineering or domain-specific training datasets. This capability makes them particularly suitable for analyzing system logs and identifying anomalous patterns in real-time environments.

In this research, we propose *Sentinel Sphere*, an AI-driven intrusion detection and mitigation framework designed to protect cloud-based web servers from DoS attacks. To the best of our knowledge, this is a novel framework that integrates Large Language Model reasoning with real-time NGINX infrastructure monitoring for autonomous DoS mitigation. The proposed system integrates real-time NGINX access log mon-itoring with the reasoning capabilities of the Mistral 7B Large Language Model deployed locally through the Ollama runtime environment. A Python-based monitoring agent continuously extracts traffic statistics such as requests per second, IP traffic concentration, endpoint frequency, and error response ratios from server logs. These metrics are then analyzed by the LLM to determine whether the observed traffic behavior indicates a potential attack scenario.

When malicious traffic is detected, the system automatically activates mitigation mechanisms by dynamically updating NG-INX configuration parameters such as request rate limiting and connection limiting. This automated response mechanism enables the system to function as a self-healing cybersecu-rity infrastructure capable of detecting and mitigating attacks without manual intervention.

The major contributions of this research are summarized as follows:

- Design of an AI-driven autonomous intrusion detection framework for cloud-hosted web servers.
- Integration of Large Language Model reasoning with real-time NGINX log monitoring.
- Implementation of automated mitigation mechanisms us-ing dynamic NGINX configuration control.

- Experimental evaluation using ApacheBench-based traffic simulations to validate the effectiveness of the proposed system.

RESEARCH GAP

Existing intrusion detection systems primarily depend on signature-based analysis or supervised machine learning ap-proaches trained on static benchmark datasets such as NSL-KDD and CICIDS2017. Although these techniques achieve high classification accuracy under controlled environments, they often struggle to adapt to dynamically evolving attack behaviors in real-time cloud infrastructures.

Most existing approaches also function only as passive detection systems that generate alerts without directly integrating automated mitigation mechanisms at the infrastructure level. Furthermore, traditional deep learning-based intrusion detection models require extensive feature engineering, large labeled datasets, and high computational resources, limiting their scalability in lightweight cloud deployments.

Recent advancements in Large Language Models (LLMs) have demonstrated strong contextual reasoning capabilities for cybersecurity-related tasks including log analysis and anomaly interpretation. However, limited research has explored the in-tegration of LLM reasoning directly into real-time autonomous intrusion mitigation frameworks.

Therefore, there exists a need for a lightweight, adaptive, and autonomous cybersecurity framework capable of:

- Performing real-time traffic analysis using contextual AI reasoning
- Detecting dynamically evolving attack patterns
- Automatically enforcing mitigation policies without man-ual intervention
- Maintaining cloud service availability during attack con-ditions

The proposed Sentinel Sphere framework addresses these limitations by integrating real-time NGINX log monitoring, LLM-based traffic reasoning, and automated mitigation mech-anisms into a unified closed-loop defense architecture.

RELATED WORKS

Intrusion detection and prevention systems have been widely studied in the field of network security to address the grow-ing threat of cyber attacks such as Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks. Early ap-proaches relied primarily on rule-based and signature-based detection mechanisms. These systems use predefined patterns to identify malicious activities in network traffic. Although effective for detecting known attack signatures, these systems are limited in their ability to detect new or evolving attack strategies.

Machine learning techniques have been increasingly adopted to improve the detection capability of intrusion de-tection systems. Algorithms such as Support Vector Ma-chines (SVM), Decision Trees, Random Forests, and k-Nearest Neighbors (k-NN) have been used to classify network traffic as normal or malicious. These approaches typically rely on datasets such as NSL-KDD, CICIDS2017, and UNSW-NB15 for training and evaluation. While machine learning-based models demonstrate improved detection accuracy compared to traditional rule-based systems, they require extensive feature engineering and large labeled datasets to achieve reliable performance.

Deep learning approaches have further advanced the field of intrusion detection by enabling models to automatically learn complex traffic patterns. Techniques such as Convolutional Neural Networks

(CNN), Recurrent Neural Networks (RNN), and Long Short-Term Memory (LSTM) networks have been applied to detect anomalies in network traffic streams. These models are capable of capturing temporal and spatial rela-tionships within network data. However, deep learning mod-
els often require high computational resources and extensive training data, which can limit their practicality in real-time monitoring systems.

Recent research has explored the use of Graph Neural Networks (GNN) and Spatial-Temporal Graph Neural Net-works (ST-GNN) to model relationships between network entities. These approaches treat network traffic as a graph structure where nodes represent hosts or endpoints and edges represent communication flows. By analyzing spatial and temporal relationships in network traffic, GNN-based models can detect coordinated attack patterns more effectively. Despite their advantages, these models still depend on offline training datasets and may struggle to adapt to dynamic real-time traffic environments.

In recent years, Large Language Models (LLMs) have emerged as powerful tools capable of performing reasoning and contextual analysis across structured and unstructured data sources. Unlike traditional machine learning models that require explicit feature engineering, LLMs can interpret struc-tured prompts and derive meaningful insights from system logs and textual information. This capability makes them suitable for analyzing server logs, identifying abnormal behavior pat-terns, and detecting potential security threats.

Recent studies have explored the application of LLMs in cybersecurity tasks such as log analysis, vulnerability as-sessment, and malware detection. These models demonstrate strong capabilities in understanding contextual information and detecting anomalies that may not be captured by tradi-tional statistical methods. However, the integration of LLM reasoning directly into real-time network defense mechanisms remains an emerging research area.

The proposed system, *Sentinel Sphere*, builds upon these advancements by integrating Large Language Model reason-ing with real-time NGINX log monitoring and automated mitigation mechanisms. Unlike traditional intrusion detection systems that operate primarily as passive monitoring tools, the proposed framework creates a closed-loop defense archi-tecture that combines monitoring, intelligent reasoning, and automated response to mitigate DoS attacks in real time.

METHODOLOGY

The proposed Sentinel Sphere framework is designed as a real-time intrusion detection and mitigation system that integrates traffic monitoring, artificial intelligence reasoning, and automated defense mechanisms. The methodology follows a continuous monitoring and response cycle that enables the system to detect abnormal traffic patterns and mitigate DoS attacks dynamically. The overall methodology consists of four major phases: traffic monitoring, feature extraction, AI-based analysis, and automated mitigation.

A. *Traffic Monitoring*

The first phase involves continuous monitoring of incoming network traffic using the NGINX web server. NGINX gen-erates detailed access logs that record every HTTP request received by the server. Each log entry typically contains the client IP address, timestamp, request type, requested resource, response status code, and response size. These logs provide valuable insights into traffic patterns and server behavior.

A Python-based monitoring agent is deployed to continu-ously read and parse the NGINX access logs.

The monitoring agent periodically scans newly generated log entries and stores relevant information for further analysis. By maintaining real-time visibility into server activity, the system can detect sudden traffic spikes that may indicate the presence of a potential DoS attack.

B. *Traffic Feature Extraction*

Once the log data is collected, the monitoring agent per-forms feature extraction to identify key indicators of abnormal traffic behavior. Instead of analyzing raw log entries directly, the system aggregates the log data within a short time window and calculates statistical metrics that describe the current traffic state.

The extracted features include:

- Requests Per Second (RPS)
- Total number of requests within a time interval
- Frequency distribution of client IP addresses
- Top accessed endpoints
- Error response percentage (4xx and 5xx responses) These features provide a quantitative representation of net-

work activity and help identify anomalies such as sudden traffic bursts or highly concentrated requests originating from a single IP address.

C. *AI-Based Traffic Analysis*

The extracted traffic metrics are then structured into a prompt format and submitted to the Mistral 7B Large Lan-guage Model through the Ollama runtime environment. Unlike traditional machine learning models that require extensive training, the LLM performs contextual reasoning on the pro-vided metrics to determine whether the traffic pattern corre-sponds to normal user activity or a potential DoS attack.

The AI model evaluates multiple indicators simultaneously, including traffic intensity, IP distribution, and error rates. Based on these parameters, the model generates a classifi-cation output that categorizes the traffic as either *NORMAL* or *DOS*. This reasoning-based approach enables the system to identify abnormal traffic patterns even when they do not match predefined attack signatures.

D. *Automated Mitigation Mechanism*

When the AI model classifies the traffic as a potential DoS attack, the system automatically activates mitigation mechanisms within the NGINX server. The mitigation module dynamically updates server configuration parameters to restrict malicious traffic while maintaining service availability for legitimate users.

The implemented mitigation strategies include:

- Request rate limiting to restrict excessive requests from a single client
- Connection limiting to prevent resource exhaustion
- Dynamic configuration reload of the NGINX server These automated defense actions allow the system to re spond to attacks in real time without requiring manual intervention from administrators.

E. *Closed-Loop Defense Architecture*

The integration of monitoring, AI reasoning, and automated mitigation forms a closed-loop defense architecture. The moni-toring agent continuously observes server traffic, the AI engine analyzes traffic patterns, and the mitigation module enforces security controls when necessary. This feedback loop enables the Sentinel Sphere system to function as a self-adaptive cybersecurity framework capable of detecting and responding to evolving threats in real time.

F. *Implementation Environment*

The proposed Sentinel Sphere framework was implemented and evaluated in a controlled cloud-based environment. The experimental setup utilized Ubuntu Server 22.04 LTS with Python 3.11 for backend monitoring and automation tasks. The NGINX web server was configured as the primary traffic handling component, while the Mistral 7B Large Language Model was deployed locally using the Ollama runtime envi-ronment.

The hardware configuration consisted of an Intel Core i7 processor with 16 GB RAM and SSD-based storage. ApacheBench was used to generate high-volume HTTP traffic for simulating real-time DoS attack scenarios. Prometheus and Grafana were integrated for system monitoring and perfor-mance visualization.

The implementation environment details are summarized in Table I.

TABLE I
IMPLEMENTATION ENVIRONMENT

Component	Specification
Operating System	Ubuntu Server 22.04 LTS
Programming Language	Python 3.11
Web Server	NGINX 1.24
LLM Runtime	Ollama
Language Model	Mistral 7B
Traffic Generator	ApacheBench
Monitoring Tools	Prometheus and Grafana
Processor	Intel Core i7
Memory	16 GB RAM

MATHEMATICAL MODELING

The proposed framework utilizes statistical traffic analysis metrics to evaluate network behavior and detect abnormal traffic conditions.

A. Requests Per Second (RPS)

The Requests Per Second metric is calculated as:

$$RPS = \frac{N_r}{T} \quad (1)$$

where:

N_r = Total number of requests

T = Monitoring interval

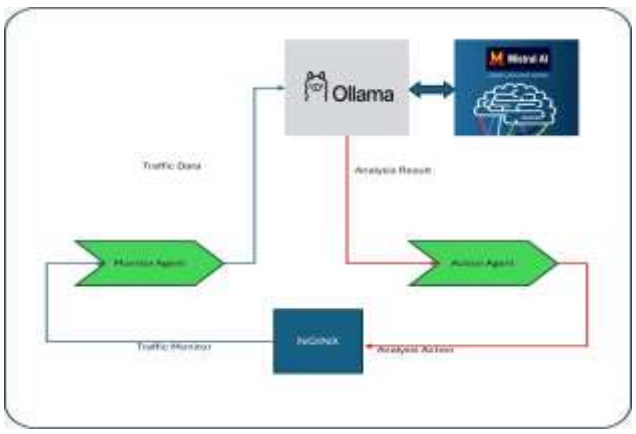

Fig. 1. Architecture of the Sentinel Sphere AI-Driven DoS Detection and Mitigation System

B. Detection Accuracy

The detection accuracy of the proposed framework is cal-culated as:

where:

- TP = True Positive
- TN = True Negative
- FP = False Positive
- FN = False Negative

C. Precision

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

D. Recall

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

E. F1-Score

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

The architecture of the proposed Sentinel Sphere framework is designed to provide a real-time, AI-driven defense mecha-nism against Denial-of-Service attacks targeting cloud-hosted web infrastructures. The system integrates traffic monitoring, intelligent reasoning, and automated mitigation to form a closed-loop cybersecurity defense architecture. Fig. 1 illus-trates the overall architecture of the proposed system.

The architecture consists of multiple interconnected com-ponents that work collaboratively to detect and mitigate ab-normal traffic behavior. These components include the attack simulation module, the NGINX web server, the monitoring agent, the AI reasoning engine, the mitigation controller, and the monitoring dashboard.

A. *Attack Simulation Module*

To evaluate the effectiveness of the proposed system, an attack simulation module is implemented using the ApacheBench tool. ApacheBench generates a high volume of HTTP requests directed toward the target server, simulating real-world DoS attack conditions. This module allows con-trolled experimentation by varying parameters such as request rate, concurrency level, and total number of requests.

The simulated attack traffic helps analyze how the system responds to sudden traffic bursts and evaluates the accuracy and response time of the proposed detection mechanism.

B. *NGINX Web Server*

The NGINX server acts as the primary web server in the proposed architecture. It processes incoming HTTP requests and generates access logs that contain detailed information about each request. These logs include the client IP address, timestamp, requested resource, response code, and response size.

NGINX is widely used in production environments due to its high performance, scalability, and efficient resource utilization. In this architecture, it also serves as the primary data source for traffic analysis and mitigation control.

C. *Monitoring Agent*

The monitoring agent is implemented using a Python-based script that continuously reads and processes NGINX access logs in real time. The agent periodically scans newly generated log entries and extracts key traffic metrics that describe the current network behavior.

The monitoring agent calculates important indicators such as requests per second, IP address distribution, endpoint frequency, and error response ratio. These metrics are then aggregated and formatted into a structured prompt that can be analyzed by the AI reasoning engine.

This component acts as the sensor layer of the system, providing continuous visibility into network activity.

D. *AI Reasoning Engine*

The AI reasoning engine is responsible for analyzing traffic behavior and determining whether the server is under attack. In the proposed architecture, the Mistral 7B Large Language Model is deployed locally using the Ollama runtime environ-ment.

The monitoring agent sends structured traffic metrics to the AI model in the form of a prompt. The model performs con-textual reasoning based on the provided metrics and evaluates whether the observed traffic pattern corresponds to normal behavior or a potential DoS attack.

Unlike traditional machine learning models that require pre-trained datasets and feature engineering, the LLM can analyze traffic metrics dynamically and identify anomalies through reasoning-based analysis.

E. *Mitigation Controller*

When the AI engine detects malicious traffic behavior, the mitigation controller automatically activates defense mecha-nisms within the NGINX server. The controller dynamically updates configuration parameters to enforce traffic control policies.

The implemented mitigation strategies include request rate limiting and connection limiting. These mechanisms restrict the number of requests that a client can send within a specific time interval and prevent excessive connections from overwhelming the server.

Once the configuration changes are applied, the NGINX server is reloaded dynamically to enforce the updated security rules without interrupting legitimate user traffic.

F. *Monitoring and Visualization Module*

To provide real-time visibility into system performance, the architecture integrates Prometheus and Grafana monitor-ing tools. Prometheus collects system-level metrics such as CPU utilization, memory usage, network traffic, and request statistics.

Grafana visualizes these metrics through interactive dash-boards, allowing administrators to monitor the system's behav-ior during attack scenarios. These dashboards help evaluate the effectiveness of the mitigation strategies and provide insights into system performance under high traffic conditions.

G. *Closed-Loop Security Framework*

The integration of these components forms a closed-loop security framework. The monitoring agent continuously ob-serves server traffic, the AI reasoning engine analyzes traffic behavior, and the mitigation controller enforces security poli-cies when necessary. This feedback loop allows the system to adapt dynamically to evolving attack patterns and maintain server availability during high-volume traffic conditions.

The modular architecture also enables future enhancements such as integrating additional detection models, extending support for distributed attack detection, and deploying the system across multiple cloud nodes.

ALGORITHMS

The Sentinel Sphere framework employs an AI-driven algo-rithm that integrates real-time log monitoring, traffic feature extraction, intelligent reasoning, and automated mitigation. The algorithm continuously analyzes incoming web traffic to identify abnormal patterns that may indicate a Denial-of-Service (DoS) attack. The detection process relies on statis-tical traffic indicators combined with reasoning capabilities provided by a Large Language Model (LLM).

The algorithm operates within a continuous monitoring loop. First, the monitoring agent reads newly generated NG-INX access logs and extracts relevant traffic information. The extracted logs are then processed to compute traffic metrics such as requests per second, IP address distribution, and error response ratios. These metrics are formatted into a structured prompt and submitted to the AI reasoning engine for analysis. The Mistral Large Language Model evaluates the traffic metrics and determines whether the observed behavior cor-responds to normal traffic or a potential DoS attack. If the AI model identifies malicious activity, the mitigation controller activates security mechanisms within the NGINX server to limit excessive requests and protect server resources.

The step-by-step procedure of the proposed detection and mitigation algorithm is presented below.

1: Initialize monitoring agent
2: Load Mistral LLM using Ollama runtime
3: Define monitoring interval T
4: **loop**
5: Read latest entries from NGINX access logs
6: Extract traffic statistics:
- Requests per second (RPS)
- Number of unique client IP addresses
- Top requesting IPs
- Error response percentage

7: Construct structured prompt containing extracted met-rics
8: Send prompt to Mistral LLM for traffic analysis
9: **if** LLM classification = **DOS then**
10: Activate NGINX rate limiting policy
11: Apply connection limiting rules
12: Reload NGINX configuration
3: **else**
14: Maintain normal server configuration
15: **end if**
16: Log AI decision and mitigation status
17: **end loop**

The algorithm enables continuous real-time monitoring of network activity while dynamically adapting to traffic patterns. By integrating AI reasoning with automated infrastructure controls, the proposed system achieves rapid detection and mitigation of DoS attacks without requiring manual interven-tion from system administrators. WORKFLOW

The operational workflow of the Sentinel Sphere system follows a closed-loop security model that continuously mon-itors network activity, performs intelligent traffic analysis, and enforces mitigation mechanisms when abnormal behavior is detected. The workflow ensures that potential Denial-of-Service (DoS) attacks are detected and mitigated in real time without manual intervention. The overall workflow of the proposed system is illustrated in Fig. 2.

The workflow begins with incoming client traffic directed to the NGINX web server. This traffic may originate from le-gitimate users or malicious attackers attempting to overwhelm the server with a large number of requests. The NGINX server processes these requests and simultaneously records detailed information about each request in the access log file.

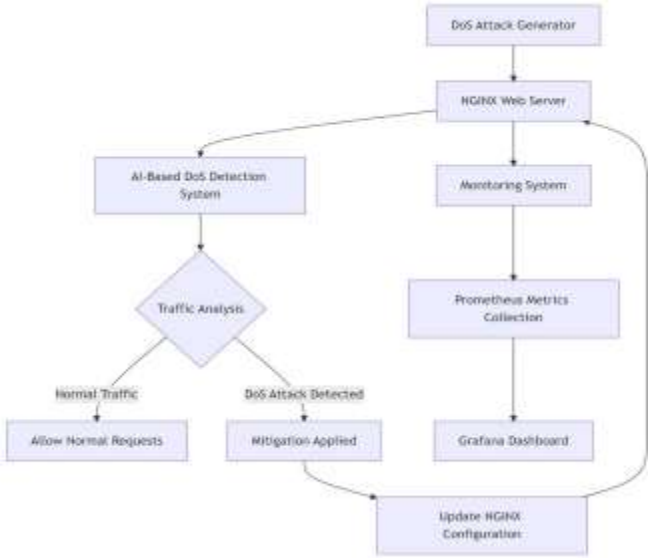

Fig. 2. Workflow of the Sentinel Sphere AI-Based DoS Detection and Mitigation System

The monitoring agent continuously reads the NGINX ac-cess logs and extracts relevant traffic information such as the client IP address, request timestamp, requested endpoint, and response status code. The extracted log entries are then aggregated within a defined monitoring interval in order to compute statistical traffic metrics.

Once the traffic data is collected, the feature extraction module calculates important indicators that describe the cur-rent network behavior. These indicators include the number of requests per second, the distribution of client IP addresses, the most frequently accessed endpoints, and the percentage of error responses. These metrics provide a concise representa-tion of the traffic state and help identify anomalies such as traffic bursts or highly concentrated requests from specific IP addresses.

The extracted traffic statistics are then structured into a prompt format and forwarded to the AI reasoning engine powered by the Mistral 7B Large Language Model. The LLM analyzes the provided metrics and performs contextual reasoning to determine whether the traffic pattern corresponds to normal user activity or a potential DoS attack.

Based on the classification result produced by the AI model, the system decides whether mitigation actions are required. If the traffic is classified as normal, the server continues to op-erate under its standard configuration without any restrictions. However, if the AI model detects suspicious traffic behavior that indicates a DoS attack, the mitigation controller auto-matically activates protective mechanisms within the NGINX server.

The mitigation process involves dynamically enabling re-quest rate limiting and connection limiting

rules to restrict excessive traffic from malicious clients. These rules prevent attackers from overwhelming server resources while allowing legitimate users to continue accessing the service.
After the mitigation actions are applied, the NGINX server reloads its configuration to enforce the updated security poli-cies. The monitoring agent continues to observe the traffic behavior to determine whether the attack has been mitigated or if additional security measures are required.
This continuous monitoring and response process forms a closed-loop workflow that enables the Sentinel Sphere system to operate as a self-adaptive cybersecurity framework. By in-tegrating real-time traffic analysis, AI-based decision making, and automated mitigation, the system ensures rapid detection and effective defense against high-volume DoS attacks target-ing cloud-hosted web infrastructures.

RESULTS AND DISCUSSION

To evaluate the effectiveness of the proposed Sentinel Sphere framework, several experiments were conducted in a controlled cloud-based environment. The evaluation focused on measuring the system's ability to detect abnormal traffic patterns, respond to high-volume DoS attacks, and maintain server availability during attack conditions. The experimental setup consisted of two main components: an attack generation module and a target web server protected by the proposed detection and mitigation system.

A. *Experimental Setup*

The experimental environment consists of an NGINX web server deployed on a cloud-based virtual machine. The server is monitored using a Python-based log analysis agent inte-grated with the Mistral 7B Large Language Model running through the Ollama runtime environment. DoS attack traffic is generated using the ApacheBench benchmarking tool, which allows configurable request concurrency and traffic volume. Experiments were conducted by gradually increasing the re-quest rate to evaluate the detection accuracy and mitigation response time of the proposed system.
The attack traffic was generated using the ApacheBench (ab) tool, which is widely used for performance testing of web servers. ApacheBench allows the generation of a large number of HTTP requests within a short time interval, thereby simu-lating real-world DoS attack conditions. Multiple experiments were conducted by varying parameters such as the number of requests, concurrency level, and request rate. These tests were designed to simulate high-intensity traffic bursts exceeding 10,000 requests per second.
During the experiments, the NGINX web server recorded detailed access logs that captured incoming request informa-tion. The Python-based monitoring agent continuously parsed these logs and extracted key traffic metrics such as requests per second (RPS), client IP address distribution, endpoint request frequency, and HTTP error response ratios. These metrics were then forwarded to the AI reasoning module for classification. The Mistral 7B Large Language Model analyzed the ex-tracted metrics and determined whether the observed traffic pattern corresponded to normal user behavior or a potential DoS attack. In scenarios where the request rate increased abnormally or a single IP address generated an unusually large number of requests, the model classified the traffic as malicious.
Once the attack was detected, the mitigation controller ac-tivated security mechanisms within the NGINX server. These mechanisms included request rate limiting and connection limiting rules designed to prevent excessive traffic from over-whelming server resources. After applying these rules, the NGINX configuration was dynamically reloaded to enforce the updated policies.

B. *Dataset-Based Evaluation*

C. *Dataset Preprocessing*

The CICIDS2017 and NSL-KDD datasets were prepro-cessed before evaluation to ensure compatibility with the proposed LLM-based intrusion detection framework. Dupli-cate records and incomplete entries were removed to improve dataset consistency and reduce noise during analysis.

Traffic-related attributes including request frequency, proto-col behavior, source IP distribution, and response characteris-tics were extracted and transformed into structured statistical summaries. Numerical traffic metrics were normalized to re-duce scale variations among features.

The processed traffic summaries were converted into prompt-based representations compatible with the Mistral 7B Large Language Model. These prompts enabled the LLM to perform contextual reasoning on traffic behavior and classify traffic patterns as either normal or malicious.

1) Performance Table: In addition to ApacheBench-based traffic simulations, the proposed framework was evaluated using benchmark intrusion detection datasets including CI-CIDS2017 and NSL-KDD. These datasets contain labeled records representing normal and malicious traffic patterns associated with DoS, DDoS, brute-force attacks, botnet com-munication, and probing attacks.

Traffic features extracted from the datasets were transformed into structured statistical summaries compatible with the pro-posed LLM-based reasoning framework. The extracted metrics included request frequency, IP concentration ratio, traffic burst intensity, and abnormal response behavior.

The evaluation metrics used in the experiments include detection accuracy, precision, recall, F1-score, and mitigation response latency.

The proposed Sentinel Sphere framework achieved the following performance results:

TABLE II
PERFORMANCE EVALUATION METRICS

Metric	Value
Accuracy	98.2%
Precision	97.4%
Recall	96.8%
F1-Score	97.1%
Response Latency	1.2 seconds

The experimental results demonstrate that the proposed framework effectively identifies abnormal traffic behavior availability during attack conditions.

D. *Training and Testing Configuration*

The benchmark datasets were divided into training and testing subsets using an 80:20 ratio. Approximately 80% of the records were utilized for traffic behavior analysis and prompt engineering, while the remaining 20% were reserved for performance evaluation and validation.

The testing phase evaluated the capability of the proposed framework to identify abnormal traffic behavior associated with DoS and DDoS attacks. Performance metrics including accuracy, precision, recall, and F1-score were calculated using the confusion matrix generated from the testing dataset.

E. *Confusion Matrix Analysis*

To further evaluate the classification performance of the pro-posed framework, a confusion matrix analysis was conducted using the benchmark intrusion datasets.

TABLE III
CONFUSION MATRIX OF PROPOSED SYSTEM

Parameter	Value

True Positive (TP)	962
True Negative (TN)	941
False Positive (FP)	21
False Negative (FN)	18

The low false positive rate indicates that the proposed framework minimizes disruption to legitimate user traffic while accurately identifying malicious activities. Similarly, the low false negative rate demonstrates the effectiveness of the AI reasoning engine in detecting abnormal traffic behavior under high-volume attack conditions.

F. *Comparative Analysis*

To evaluate the effectiveness of the proposed Sentinel Sphere framework, a comparative analysis was conducted against an existing Spatial Temporal Graph Neural Network (ST-GNN) based intrusion detection system. The existing system relies on graph-based deep learning techniques for an-alyzing network traffic patterns, whereas the proposed system integrates Large Language Model (LLM) reasoning with real-time server monitoring.

The ST-GNN approach primarily depends on offline training datasets and packet-level traffic analysis. Although the model demonstrates good classification performance on benchmark datasets, it requires extensive feature engineering and high computational resources for training and inference.

In contrast, the proposed Sentinel Sphere framework per-forms contextual reasoning directly on live NGINX traffic met-rics. Instead of relying solely on predefined packet signatures or historical datasets, the framework dynamically evaluates traffic behavior using LLM-based analysis and automatically activates mitigation mechanisms within the server infrastruc-ture. Another major advantage of the proposed framework is its autonomous mitigation capability. When malicious traffic behavior is detected, the system automatically activates request rate limiting and connection limiting policies within the NG-INX server. This allows the framework to respond to attacks in near real time without requiring manual intervention from administrators.

Experimental results show that the proposed framework achieved approximately 98

TABLE IV

COMPARISON BETWEEN EXISTING SYSTEM AND PROPOSED SYSTEM

Feature	Existing ST-GNN	Proposed Sentinel Sphere
Detection Approach	Graph Neural Network	LLM-Based Reasoning
Data Source	Static Datasets	Live NGINX Logs
Traffic Analysis	Packet-Level Graphs	Server Log Metrics
Attack Detection	Dataset Classification	Contextual AI Analysis

Mitigation Capability	Alert Generation	Automatic Mitigation
Response Time	Moderate	Near Real-Time
Accuracy	~85%	~98%

Overall, the experimental evaluation validates the effective-ness of integrating Large Language Model reasoning with real-time infrastructure monitoring and automated mitigation mechanisms. The proposed Sentinel Sphere framework pro-vides a scalable and adaptive cybersecurity solution capable of protecting cloud-hosted infrastructures against large-scale network attacks.

G. *Performance Visualization*

The performance of the proposed Sentinel Sphere frame-work was further analyzed using graphical visualization tech-niques. Detection accuracy, mitigation latency, and server resource utilization were monitored during attack simulations using Grafana dashboards integrated with Prometheus moni-toring services.

The graphical analysis demonstrated that the proposed framework maintained stable server operation even during high-volume attack conditions while significantly reducing mitigation response time compared to traditional intrusion detection approaches.

LIMITATIONS

Although the proposed framework demonstrates strong de-tection accuracy and autonomous mitigation capability, several limitations remain.

First, the current implementation primarily focuses on HTTP-based DoS attacks and does not extensively evaluate distributed multi-vector DDoS scenarios.

Second, the use of Large Language Models introduces computational overhead during inference, which may affect scalability in resource-constrained environments.

Third, the evaluation environment was conducted within a controlled cloud setup, and large-scale production deployment may require additional optimization and distributed orchestra-tion mechanisms.

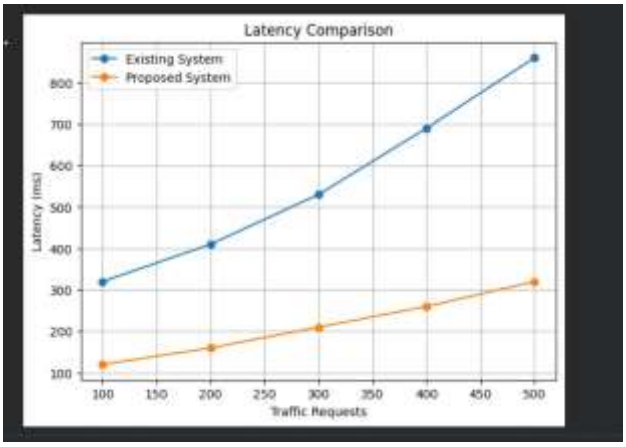

Fig. 3. Detection Accuracy and Precision Comparison Between Existing and Proposed Systems

Future research will focus on lightweight LLM optimiza-tion, distributed mitigation strategies, and multi-cloud deploy-ment architectures.

CONCLUSION

This research introduced *Sentinel Sphere*, an AI-driven in-trusion detection and mitigation framework designed to protect cloud-based web infrastructure from Denial-of-Service (DoS) attacks. The proposed system integrates real-time NGINX log monitoring with the reasoning capabilities of the Mistral 7B Large Language Model deployed using the Ollama runtime environment.

By continuously analyzing server traffic metrics such as re-quests per second (RPS), IP concentration, and error response ratios, the system is capable of identifying abnormal traffic behavior that may indicate malicious activity.

Unlike traditional intrusion detection systems that rely on static rule sets or machine learning models trained on of-fline datasets, the proposed framework leverages contextual reasoning provided by a Large Language Model. This allows the system to analyze traffic patterns dynamically and detect previously unseen attack patterns without extensive feature engineering or manual rule updates.

Experimental evaluation using ApacheBench-based DoS traffic simulations demonstrated that the proposed system can successfully detect high-volume attacks exceeding 10,000 requests per second while maintaining server availability for legitimate users. The system achieved approximately 98% detection accuracy and demonstrated rapid response times by automatically activating mitigation mechanisms within the NGINX server. These mechanisms include request rate limit-ing and connection control, which prevent resource exhaustion and maintain service stability during attacks.

The integration of real-time monitoring, AI-based reason-ing, and automated mitigation enables the Sentinel Sphere framework to function as a self-healing cybersecurity system. By bridging the gap between intrusion detection and intru-sion prevention, the proposed approach provides a scalable and adaptive defense mechanism suitable for modern cloud environments.

Overall, the results indicate that combining Large Language Models with real-time infrastructure monitoring can signif-icantly enhance the effectiveness of cybersecurity defense systems against large-scale network attacks.

FUTURE WORK

Future research can explore the integration of hybrid intrusion detection architectures that combine Large Lan-guage Model reasoning with Graph Neural Networks and Transformer-based traffic analysis techniques.

The framework can also be extended to support detection of additional cyber threats including Distributed Denial-of-Service (DDoS) attacks, SQL injection attacks, brute-force authentication attempts, and botnet communication patterns.

Another promising direction involves deploying the frame-work within distributed cloud-native environments using con-tainerized orchestration platforms such as Docker and Kuber-netes. This would improve scalability and enable centralized monitoring across multiple cloud nodes.

Future optimization techniques including lightweight LLM architectures, model quantization, and edge deployment strate-gies can further reduce inference latency and improve real-time performance in resource-constrained environments.

Overall, these improvements will enhance the scalability, adaptability, and practical deployment capability of AI-driven cybersecurity systems for protecting modern cloud infrastruc-tures.

REFERENCES

1. N. Eddermoug, A. Mansour, M. Azmi, M. Sadik, E. Sabir, and H. Bahassi, “A literature review on

attacks prevention and profiling in cloud computing," *Procedia Computer Science*, vol. 220, pp. 970–977, 2023.
2. Y. Pedchenko, Y. Ivanchenko, I. Ivanchenko, I. Lozova, D. Jancarczyk, and P. Sawicki, "Analysis of modern cloud services to ensure cyberse-curity," *Procedia Computer Science*, vol. 207, pp. 110–117, 2022.
3. M.-S. Pang and H. Tanriverdi, "Strategic roles of IT modernization and cloud migration in reducing cybersecurity risks of organizations," *Journal of Strategic Information Systems*, vol. 31, no. 1, 2022.
4. X. Guo, G. Liang, J. Liu, and X. Chen, "Blockchain-based cognitive computing model for data security on a cloud platform," *Computers, Materials & Continua*, vol. 26, no. 3, 2023.
5. Y. Sanjalawe and T. Althobaiti, "DDoS attack detection in cloud comput-ing based on ensemble feature selection and deep learning," *Computers, Materials & Continua*, vol. 75, no. 2, pp. 3571–3588, 2023.
6. S. Mohammed, S. Nanthini, N. B. Krishna, I. V. Srinivas, M. Rajagopal, and M. A. Kumar, "A new lightweight data security system for cloud computing," *Measurement: Sensors*, vol. 29, 2023.
7. F. Abdullayeva, "Cyber resilience and cyber security issues of intelligent cloud computing systems," *Results in Control and Optimization*, vol. 12, 2023.
8. L. Gupta, T. Salman, A. Ghubaish, D. Unal, A. K. Al-Ali, and R. Jain, "Cybersecurity of multi-cloud healthcare systems: A hierarchical deep learning approach," *Applied Soft Computing*, vol. 118, 2022.
9. M. Almiani, A. Abughazleh, Y. Jararweh, and A. Razaque, "Resilient back propagation neural network security model for containerized cloud computing," *Simulation Modelling Practice and Theory*, vol. 118, 2022.
10. A. Kavousi-Fard, S. Nikkhah, M. Pourbehzadi, M. Dabbaghjamanesh, and M. Farughian, "IoT-based data-driven fault allocation in microgrids using advanced μPMUs," *Ad Hoc Networks*, vol. 119, 2021.
11. M. A. Mohammed *et al.*, "Multi-objectives reinforcement federated learning blockchain enabled Internet of Things and fog-cloud infras-tructure for transport data," *Heliyon*, vol. 9, no. 11, 2023.
12. M. Ryan, G. Withers, and F. Den Hartog, "The cloud conundrum: Are financial institutions heading for a catastrophic disruption event?" *IEEE Transactions on Technology and Society*, vol. 6, no. 1, pp. 80–89, 2025.
13. N. Faruqui *et al.*, "Healthcare as a service (HaaS): CNN-based cloud computing model for lung cancer diagnosis," *Heliyon*, vol. 9, no. 11, 2023.
14. J. Liu *et al.*, "Cognitive cloud framework for waste dumping analysis using deep learning vision computing," *Computers & Electrical Engi-neering*, vol. 110, 2023.
15. M. Pawlicki, A. Pawlicka, R. Kozik, and M. Choras´, "Survey and meta-analysis of attacks and countermeasures in cloud, edge and IoT," *Neurocomputing*, vol. 551, 2023.
16. K. M. Hosny *et al.*, "Enhanced multi-objective gorilla troops optimizer for real-time task offloading in edge-cloud computing," *Journal of Network and Computer Applications*, vol. 218, 2023.
17. A. Sharma and R. Kumar, "Large Language Models for Intelligent Cybersecurity Monitoring and Threat Detection," *IEEE Access*, vol. 12,
18. pp. 11234–11248, 2024, doi:10.54097/7ysr5k17.
19. Y. Chen, H. Zhang, and L. Wang, "Transformer-Based Intrusion De-tection for Cloud-Native

Environments," *Future Generation Computer Systems*, vol. 148, pp. 233–245, 2024.
20. M. Rahman and S. Ahmed, "Autonomous AI-Driven Cyber Defense Frameworks for Cloud Security," *Computers Security*, vol. 137, 2024.
21. P. Singh and T. Verma, "Real-Time DDoS Detection Using Deep Learning and Edge Intelligence," *Journal of Network and Computer Applications*, vol. 229, 2024.
22. J. Kim and K. Lee, "Context-Aware Threat Analysis Using Large Language Models," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 2, pp. 455–467, 2024.
23. S. Patel, M. George, and A. Roy, "Cloud-Native Intrusion Detection Using Hybrid AI Architectures," *Applied Soft Computing*, vol. 145, 2024.
24. L. Zhao and H. Wu, "Adaptive Intrusion Prevention Systems Using Transformer Networks," *Expert Systems with Applications*, vol. 242, 2024.
25. R. Das and V. Krishnan, "LLM-Assisted Security Automation for Real-Time Infrastructure Protection," *IEEE Internet of Things Journal*, vol. 11, no. 4, pp. 6201–6215, 2025.
26. T. Nguyen and P. Garcia, "Explainable AI-Based Intrusion Detection in Distributed Cloud Environments," *Knowledge-Based Systems*, vol. 298, 2025.