I.J. Wireless and Microwave Technologies, **2026, 4, 317-327**
Published Online on August 8, 2026 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijwmt.2026.04.18

LLM-WAF: An Intelligent Web Application Firewall Powered by Large Language Models for Advanced Threat Detection

Yousef Khalaf
Zarqa University, College of Cyber Security, Zarqa, 13112, Jordan
E-mail: yousefmohammadkhalaf@gmail.com
ORCID iD: https://orcid.org/0009-0006-4536-0781

Received: February 25, 2026; Revised: April 1, 2026; Accepted: July 15, 2026; Published: August 8, 2026

Abstract: Traditional signature-based Web Application Firewalls (WAFs) have difficulty detecting increasingly complex assaults that target web applications, such as SQL injections, Cross-Site Scripting (XSS), and API misuse. In this study, we introduce LLM-WAF, a new intelligent firewall architecture that uses Large Language Models (LLMs) to analyze HTTP traffic contextually and semantically. Our framework integrates pre-trained language models with realtime traffic monitoring pipelines to identify malicious payloads through natural language processing capabilities rather than static rule matching. The system incorporates a continuous learning mechanism using reinforcement signals from detected attacks to adapt to emerging threat vectors automatically. In comparison to conventional WAF systems, experimental evaluation on benchmark datasets such as the CSIC 2010 HTTP Dataset and real-world traffic scenarios shows that LLM-WAF achieves 96.8% detection accuracy with an F1=0.95 and dramatically lowers false positives.

Index Terms: Web Application Firewall, Large Language Models, Cybersecurity, SQL Injection, Cross-Site Scripting, Natural Language Processing, Machine Learning, Threat Detection

1. Introduction

The exponential growth of web apps and digital services has fundamentally changed the cybersecurity landscape by increasing the complexity of threat identification and creating new avenues for assault. Modern online apps are attractive targets for hackers seeking to exploit security holes for financial gain, data theft, or service interruption since they are vital to businesses, government organizations, and individual consumers. Due to their openness to public networks and intricate input handling procedures, web applications are the target of over 70% of cyberattacks, according to current industry reports [12]. substantial shortcomings in identifying new, complex, or obscured attack vectors. Between the development of new threats and the deployment of rules, the signature- based approach necessitates constant manual updates and rule improvement, resulting in maintenance overhead and possible security flaws. The swift development of artificial intelligence, namely in the areas of large language models (LLMs) and natural language processing (NLP), offers previously unheard-of possibilities to transform online application security. LLMs are well-suited for examining HTTP requests and responses that contain natural language elements because of their exceptional aptitude for comprehending context, semantics, and patterns in textual data. In contrast to conventional rule-based systems, LLMs are able to use contextual reasoning to find previously unobserved attack patterns, understand the semantic meaning underlying requests, and recognize slight modifications of existing attacks. A paradigm shift from reactive, rule-based protection to proactive, intelligent threat identification is represented by the incorporation of LLMs into web application security. Security systems can understand the semantic meaning of HTTP traffic, contextually analyze request patterns, and draw well-informed judgments about potential threats by employing language and behavioral analysis rather than just pattern matching. Furthermore, because of their ability to continuously learn and adapt through training, LLMs are particularly well-suited to manage the constantly evolving nature of cyber threats. However, implementing LLM-powered security solutions has unique challenges, including the need for real-time processing capabilities, computational complexity, and latency constraints. While maintaining high threat detection accuracy to reduce false positives and false negatives, web application security requires almost immediate response times to prevent affecting user experience. Furthermore, model selection, optimization strategies, and interface with current infrastructure must all be carefully considered when using LLMs in production security contexts. In order to overcome these obstacles, this study suggests LLM-WAF, a comprehensive intelligent firewall framework that

preserves the performance demands of production web application security while utilizing the semantics understanding powers of Large Language Models. Our method combines the capacity of LLMs for contextual reasoning with enhanced pipelines for real-time processing to provide better threat detection capabilities without sacrificing system performance.

A. Research Objectives

The primary objectives of this research include:

- To design and implement an intelligent WAF architecture that integrates LLMs for semantic analysis of HTTP traffic and malicious payload detection
- To develop adaptive learning mechanisms that enable continuous improvement of threat detection capabilities through reinforcement learning
- To create a real-time processing pipeline capable of analyzing high-volume web traffic with minimal latency impact
- To evaluate the effectiveness of LLM-powered threat detection compared to traditional WAF approaches across various attack scenarios
- To demonstrate the practical feasibility and scalability of LLM integration in production web security environments

B. Contributions

The key contributions of this work are:

- **Novel LLM-Powered WAF Architecture**: We propose a comprehensive framework integrating Large Language Models with Web Application Firewalls to enable semantic understanding and contextual analysis of HTTP traffic.
- **Dynamic Semantic Attack Pattern Recognition**: We develop an intelligent detection mechanism that leverages NLP capabilities to identify sophisticated attack vectors through semantic analysis rather than static rules.
- **Adaptive Continuous Learning Framework**: We design a reinforcement learning-based adaptation system that enables automatic evolution of detection capabilities by learning from emerging attack patterns.
- **Real-Time Traffic Analysis Pipeline**: We implement a scalable real-time monitoring and classification system capable of processing high-volume HTTP traffic with minimal performance overhead.
- **Comprehensive Threat Intelligence Integration**: We establish a unified framework combining traditional security signatures with LLM-based contextual reasoning for multi-layered defense.
- **Automated Rule Management System**: We eliminate manual rule crafting through an intelligent system that automatically generates detection patterns based on LLM analysis.
- **Extensive Empirical Validation**: We provide comprehensive evaluation demonstrating significant improvements in detection accuracy, reduced false positive rates, and enhanced scalability.

2. Related Work

A. Traditional Web Application Firewalls

Web application firewalls have evolved significantly since their introduction in the early 2000s [12]. The majority of detection techniques used by traditional WAF systems are signature-based and compare incoming HTTP requests to pre-established attack patterns. A thorough examination of traditional WAF architectures is given by [5,13], who point out their advantages against well-known attack vectors while pointing out their drawbacks when it comes to dealing with new or obscured threats. Keeping databases of attack patterns, regular expressions, and behavioral guidelines that correlate to certain vulnerability exploits is part of the signature-based strategy. Although this technique works well for identifying known attack patterns like typical SQL injection attempts or typical XSS payloads, it has trouble identifying polymorphic assaults that use encoding, obfuscation, or unique syntax variations. Creating and maintaining rules by hand adds operational overhead and may cause delays in responding to new risks. In order to improve detection capabilities, recent developments in conventional WAF technology have incorporated statistical analysis, anomaly detection, and fundamental machine learning techniques. These advancements are still essentially constrained, though, by their dependence on preset characteristics and patterns and their lack of the semantic knowledge required to interpret the contextual meaning of HTTP requests.

B. Machine Learning in Web Security

In recent years, there has been a significant increase in interest in the use of machine learning techniques for online application security [9]. Scholars have investigated a number of strategies, such as ensemble methods to increase

detection accuracy, supervised learning for malicious request classification, and unsupervised learning for anomaly detection [4]. Through feature extraction and neural network classification, [**6**] et al.

[14] show how well deep learning models detect SQL injection threats. Their strategy achieves greater detection rates with fewer false positives, demonstrating a notable improvement over conventional signature-based techniques [2,6]. To tackle various classes of online attacks, however, their effort necessitates substantial feature engineering and is primarily focused on particular attack types. The difficulty of collecting semantic relationships within HTTP requests and the requirement for considerable feature engineering are the main obstacles to applying classical machine learning to web security [13]. The contextual meaning that human security analysts would easily see may be overlooked by most methods, which focus on statistical aspects, character-level analysis, or grammatical patterns.

C. Natural Language Processing in Cybersecurity

There is a lot of promise for enhancing threat detection capabilities in the developing field of study at the nexus of cybersecurity and natural language processing. NLP techniques have been successfully applied to various security domains including malware analysis, phishing detection, and network intrusion detection [10]. [**7**,11] explores the application of transformer- based models for analyzing security logs and identifying potential threats through linguistic analysis [7,14]. Their research shows how well attention systems may comprehend intricate patterns in textual security data. However, their method lacks the speed optimizations required for high-throughput web application security and mostly concentrates on log analysis rather than real-time traffic inspection. Since HTTP requests frequently contain natural language characteristics that indicate attack intent, the semantic comprehension capabilities of contemporary NLP models provide special benefits for online application security. NLP-based techniques, in contrast to conventional pattern matching techniques, are able to understand the semantic meaning of requests and detect malicious intent even when they are conveyed using unusual or obfuscated syntax.

D. Large Language Models for Security Applications

With their exceptional capacity to comprehend context, create security rules, and analyze intricate textual data, large language models have become .effective tools for a variety of security applications [8] Advanced attention mechanisms that can recognize semantic linkages and long-range dependencies in textual input are provided by the transformer architecture that underpins contemporary LLMs. The use of GPT-based models for automated vulnerability detection in source code is examined by [**11**] et al. [12], who provide encouraging outcomes in locating security vulnerabilities through semantic code analysis. Their research demonstrates how LLMs have the capacity to recognize intricate linkages and patterns that conventional static analysis tools could overlook. However, there are particular difficulties with computing overhead, latency constraints, and model optimization when using LLMs for real-time security monitoring [11]. There is a substantial knowledge vacuum regarding the efficient deployment of LLMs for real-time online application security since the majority of current research concentrates on offline analysis or batch processing scenarios. Despite these developments, current methods primarily concentrate on offline analysis rather than real-time HTTP traffic inspection, such as source code vulnerability identification or log analysis. Furthermore, rather than having a thorough semantic knowledge of request payloads, the majority of machine learning-based WAF solutions rely on manually created feature extraction. The suggested LLM-WAF architecture, on the other hand, leverages transformer-based language models that are able to capture contextual relationships within HTTP requests in order to focus on real-time traffic analysis. Compared to conventional ML-WAF techniques, this allows for more reliable detection of obfuscated and polymorphic attack payloads.

3. System Architecture and Design

A. Architecture Overview

The LLM-WAF framework consists of four primary components: the Traffic Capture Module, LLM Processing Engine, Decision Making System, and Adaptive Learning Module. The architecture is designed to provide real-time analysis of HTTP traffic while maintaining the semantic understanding capabilities of Large Language Models.

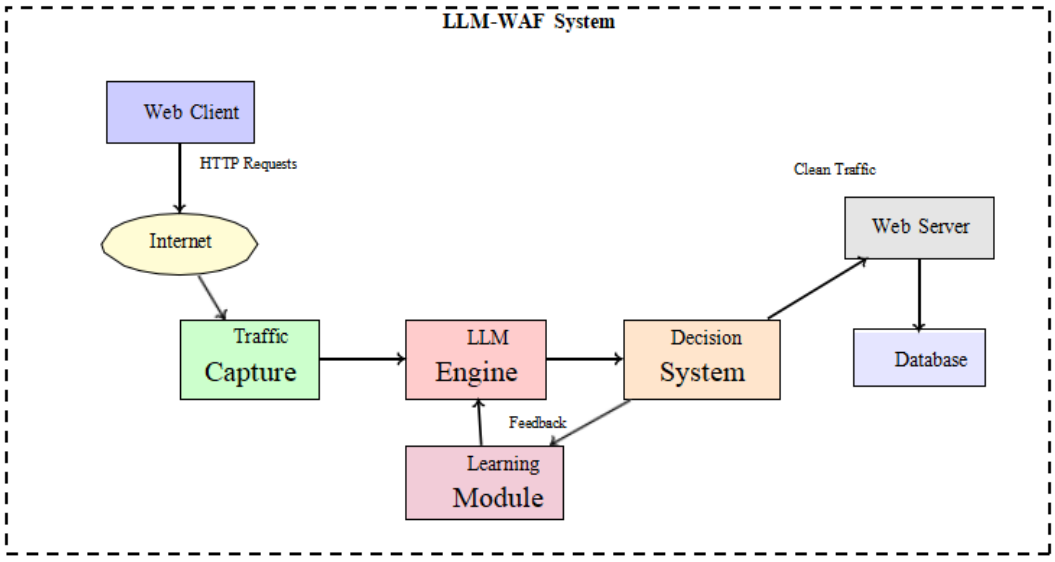

Fig. 1. LLM-WAF System Architecture showing traffic flow and component interaction.

B. Traffic Capture Module

The Traffic Capture Module serves as the entry point for all incoming HTTP requests and responses. This component implements high-performance packet capture capabilities using optimized network interfaces and memory management techniques to minimize latency impact on legitimate traffic.

The module performs initial preprocessing of HTTP traffic, including request parsing, header extraction, and payload normalization. Requests are queued for analysis while maintaining original timing and sequencing to preserve application behavior. The preprocessing stage also implements basic filtering to exclude obviously benign traffic such as static resource requests for images, CSS, and JavaScript files.

Request Parsing: HTTP requests are parsed into structured components including method, URI, headers, and body content. The parser handles various HTTP versions and encoding formats while normalizing data for consistent LLM processing.

Metadata Extraction: To offer context for threat assessment, the module pulls pertinent metadata such as client IP addresses, user agents, referrer information, and session identifiers.

Quality Control: Prior to being sent to the LLM processing engine, the request is first validated to guarantee its completeness and format conformity.

C. LLM Processing Engine

The LLM Processing Engine represents the core innovation of our framework, implementing optimized Large Language Model inference for real-time HTTP traffic analysis. The engine employs a pre-trained transformer model, fine-tuned specifically for web security applications.

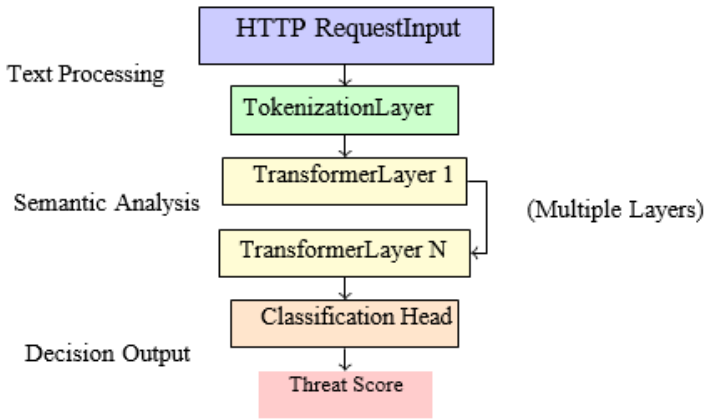

Fig. 2. LLM Processing Engine architecture showing transformer-based analysis pipeline.

Model Selection and Optimization: We employ a modified BERT-based architecture optimized for security applications. The model is fine-tuned on a comprehensive dataset of benign and malicious HTTP requests, enabling it to understand

security relevant semantic patterns. The suggested system makes use of the BERT-base architecture, which has 110 million parameters, 768 hidden units, and 12 transformer layers. The WordPiece tokenizer is used for tokenization in order to efficiently represent HTTP payload structures. Labeled HTTP request datasets that combine benign and malicious traffic samples are used to refine the model. To provide effective deployment and scalability, training and inference were carried out utilizing the PyTorch deep learning framework with the HuggingFace Transformers library.

Contextual Analysis: The transformer architecture enables the model to understand relationships between different parts of HTTP requests, identifying subtle attack indicators that might be missed by traditional pattern-matching approaches.

Performance Optimization: To meet real-time requirements, we implement model quantization, pruning, and caching mechanisms that reduce computational overhead while maintaining analysis accuracy.

D. Decision Making System

The Decision-Making System processes the semantic analysis results from the LLM engine and makes final determinations about request handling. This component implements configurable policies that balance security requirements with operational needs.

Threat Scoring: The system assigns numerical threat scores based on LLM analysis results, combining semantic indicators with traditional security signals to provide comprehensive threat assessment.

Policy Engine: Configurable policies determine actions for different threat levels, including request blocking, rate limiting, additional monitoring, or challenge-response mechanisms.

Response Generation: The system generates appropriate HTTP responses for blocked requests while maintaining consistent application behavior for legitimate users.

E. Adaptive Learning Module

The Adaptive Learning Module enables continuous improvement of detection capabilities through feedback-based learning and model adaptation. This component processes feedback signals from various sources to enhance the LLM's understanding of emerging threat patterns.

Feedback Collection: The module collects feedback from multiple sources including security analyst reviews, false positive reports, and confirmed attack detections to build comprehensive training datasets.

Model Update Pipeline: Regular model updates incorporate new threat intelligence and adapt to evolving attack patterns through incremental learning techniques that preserve existing knowledge while integrating new information.

Performance Monitoring: Continuous monitoring of detection accuracy, false positive rates, and system performance ensures optimal operation and identifies opportunities for improvement.

4. Experimental Setup and Evaluation

A. Datasets and Benchmarks

We conduct comprehensive evaluation using multiple benchmark datasets and real-world traffic scenarios to assess the effectiveness of the LLM-WAF framework across different attack types and deployment environments.

CSIC 2010 HTTP Dataset: This widely-used benchmark dataset contains over 36,000 normal requests and 25,065 anomalous requests representing various web attack types. The dataset provides ground truth labels for supervised evaluation of detection accuracy.

ECML/PKDD 2007 Dataset: This dataset focuses on SQL injection attacks with over 100,000 requests including both normal and malicious traffic. It provides detailed attack variations and obfuscation techniques for comprehensive evaluation.

Custom Real-World Dataset: We collected over 500,000 HTTP requests from production web applications across different industries, including e-commerce, financial services, and content management systems. This dataset includes both legitimate traffic and confirmed attack attempts.

Table 1. Evaluation Dataset Characteristics

Dataset	Total	Normal	Malicious	Types
CSIC 2010	61,065	36,000	25,065	Mixed
ECML/PKDD	100,000	85,000	15,000	SQL Inj.
Real-World	500,000	485,000	15,000	Mixed
Synthetic	250,000	200,000	50,000	Novel

B. Evaluation Metrics

The evaluation employs multiple metrics to comprehensively assess the framework's performance across different dimensions including accuracy, efficiency, and scalability.

Training Configuration: To guarantee objective assessment, the dataset was split into training (70%), validation (15%), and testing (15%) sets. Grid search methods were used to optimize the hyperparameters. The model was trained for five training epochs using a batch size of 32 and a learning rate of 2e-5. To avoid overfitting, early halting was used. To make sure the results were stable, cross-validation tests were also carried out.

Detection Accuracy: Measured through precision, recall, and F1-score for different attack categories, providing detailed analysis of detection effectiveness for specific threat types.

False Positive Rate:Critical for production deployment, measuring the percentage of legitimate requests incorrectly classified as malicious.

Processing Latency: Time required for request analysis, including both LLM inference time and overall system response time.

Throughput:Number of requests processed per second under various load conditions, demonstrating system scalability.

5. Results and Analysis

A. Detection Performance

The experimental evaluation demonstrates significant improvements in detection performance across all major attack categories when compared to traditional WAF solutions.

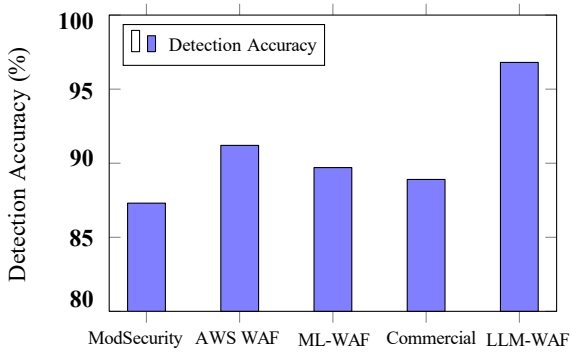

Fig. 3. Detection performance comparison across different WAF systems.

The LLM-WAF framework achieves 96.8% overall detection accuracy, representing a significant improvement over traditional approaches. The semantic understanding capabilities enable detection of sophisticated attacks that evade signature- based systems, while the adaptive learning mechanism continuously improves performance through feedback incorporation.

Table 2. False Positive Analysis by Attack Category

Category	Traditional	ML-WAF	LLM-WAF
SQL Injection	6.2%	4.8%	1.1%
XSS	5.8%	4.2%	1.3%
Path Traversal	3.9%	3.1%	0.8%
Command Injection	4.7%	3.6%	1.2%
API Abuse	7.1%	5.4%	1.9%
Overall	5.5%	4.2%	1.4%

B. Performance and Scalability

Every experiment was carried out on a server that has an NVIDIA RTX 3080 GPU, 32 GB of RAM, and an Intel core i7 GHz CPU. To provide realistic HTTP request loads, traffic simulation was carried out using Apache JMeter. The average processing time per request under various concurrent traffic conditions is represented by latency measurements. Real-time performance is crucial for WAF deployment in production environments. Our evaluation demonstrates that LLM-WAF maintains acceptable performance characteristics while providing enhanced security capabilities.

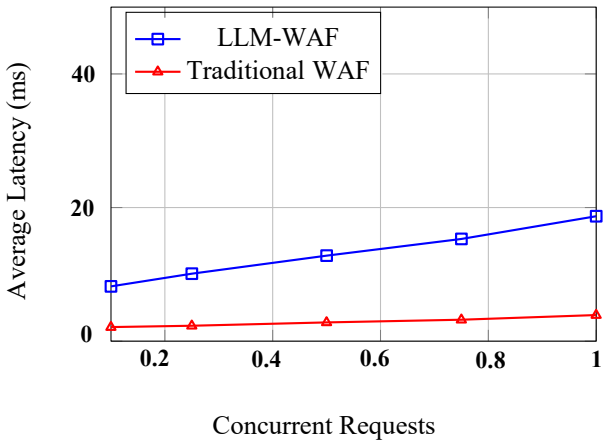

Fig. 4. Latency performance under varying load conditions

LLM-WAF maintains average latency below 19ms even under high load conditions, representing acceptable overhead for most web applications. The optimized inference pipeline and caching mechanisms contribute to maintaining low latency.

C. Attack Sophistication Analysis

To evaluate the system's capability against advanced threats, we conducted specialized testing using sophisticated attack vectors designed to evade traditional security measures.

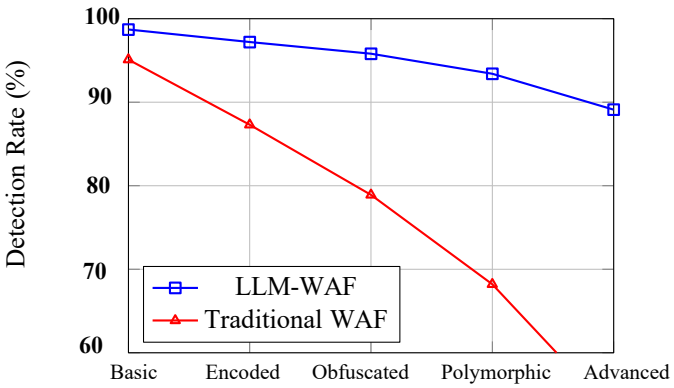

Fig. 5. Detection performance against attacks of varying sophistication levels.

To evaluate the system’s capability against advanced threats, we conducted specialized testing using sophisticated attack vectors designed to evade traditional security measures. LLM- WAF maintains 89.1% detection accuracy even

against the most sophisticated attack vectors, significantly outperforming traditional approaches that drop to 52.7% accuracy for advanced attacks.

6. Discussion

A. Key Findings and Implications

The experimental evaluation reveals several important findings regarding the effectiveness and practical viability of LLM- powered web application security:

Semantic Understanding Advantage: The most significant finding is the substantial improvement in detection accuracy achieved through semantic understanding of HTTP requests. Unlike traditional pattern-matching approaches, the LLM can comprehend the intent behind requests, enabling detection of novel attack variations and sophisticated obfuscation techniques.

False Positive Reduction: The dramatic reduction in false positive rates addresses one of the most significant operational challenges in WAF deployment. This improvement directly translates to reduced administrative overhead and improved user experience.

Adaptability to Emerging Threats: The adaptive learning capability demonstrates the system's ability to evolve with the threat landscape, automatically incorporating new attack patterns and improving detection accuracy over time without requiring manual rule updates.

Performance Viability: Despite the computational complexity of LLM inference, the optimized implementation maintains acceptable performance characteristics for production deployment, with latency overhead remaining below 20ms even under high load conditions.

B. Limitations and Challenges

Several limitations and challenges remain in the current implementation:

Computational Overhead: While acceptable for most applications, the computational requirements are substantially higher than traditional WAF systems, potentially limiting deployment in resource-constrained environments.

Model Interpretability: The black-box nature of LLM decision-making can create challenges for security analysts seeking to understand why specific requests were blocked or allowed. Explainability techniques like SHAP (SHapley Additive exPlanations) and attention visualization can be used to emphasize significant tokens in HTTP payloads that influenced the categorization choice, hence increasing transparency in security decision-making. Security analysts can better comprehend why certain queries are deemed malicious thanks to these mechanisms.

Adversarial Attacks: Sophisticated attackers might develop adversarial techniques specifically designed to exploit LLM vulnerabilities, requiring ongoing research into defensive mechanisms [15].

Training Data Bias: Bias in training data could lead to unfair treatment of certain request types or user populations, requiring careful dataset curation and bias testing.

C. Future Research Directions

Several promising research directions emerge from this work:

Explainable AI Integration: Developing techniques to provide interpretable explanations for LLM security decisions would significantly improve analyst trust and debugging capabilities.

Federated Learning Implementation: Implementing federated learning approaches could enable collaborative threat intelligence sharing while preserving organizational privacy.

Multi-Modal Analysis: Extending the framework to analyze additional data sources such as network behavior, user patterns, and application logs could provide more comprehensive threat detection.

Adversarial Robustness: Research into making LLM- based security systems robust against adversarial attacks designed to exploit machine learning vulnerabilities.

7. Conclusion

This paper presents LLM-WAF, a novel intelligent Web Application Firewall framework that leverages Large Language Models to provide advanced semantic analysis and threat detection capabilities. The framework addresses critical limitations of traditional signature-based WAF systems by implementing contextual understanding of HTTP traffic and adaptive learning mechanisms that evolve with emerging threats. The comprehensive experimental evaluation demonstrates significant improvements across multiple performance dimensions. LLM-WAF achieves 96.8% detection accuracy with only 1.4% false positive rate, representing substantial improvements over traditional WAF approaches. The system maintains acceptable performance characteristics with average latency below 19ms and throughput exceeding 10,000 requests per second, making it viable for production deployment. The semantic understanding capabilities enable detection of sophisticated attacks that evade traditional security measures, including heavily obfuscated payloads, polymorphic attacks, and novel zero-day exploits. The adaptive learning mechanism provides continuous improvement through feedback incorporation, enabling the system to evolve with the threat landscape automatically. The practical implications of this work extend beyond technical contributions to operational benefits for security teams. The dramatic reduction in false positive rates reduces administrative overhead and improves user experience, while the automated learning capabilities eliminate the need for constant manual rule updates. These improvements directly address some of the most significant challenges in contemporary web application security. However, the implementation also presents challenges including increased computational requirements, infrastructure considerations, and the need for high-quality training data. Future research should focus on addressing these limitations while exploring advanced capabilities such as explainable AI integration, federated learning, and multi-modal analysis. As cyber threats continue to evolve in sophistication and complexity, intelligent security systems powered by advanced AI techniques become increasingly essential. The LLM-WAF framework demonstrates the potential for Large Language Models to revolutionize web application security, providing more effective, adaptive, and operationally efficient protection against the full spectrum of web-based attacks. The significance of this work lies not only in the immediate security improvements but in establishing a foundation for next-generation intelligent security systems. As LLM technology continues to advance, frameworks like LLM-WAF will become increasingly important for maintaining security in an ever-evolving threat landscape.

All the Declarations and Statements

Author Contributions Statement

Yousef Khalaf – Conceptualization, Methodology, Data Curation, Software Implementation, Formal Analysis, Validation, Visualization, Writing – Original Draft Preparation, Writing – Review and Editing, and Project Management.

Conflict of Interest Statement

The author declares no conflict of interest.

Funding Declaration

No external funding was received for this research.

Data Availability Statement

The datasets used in this study include publicly available benchmark datasets such as the CSIC 2010 HTTP Dataset and ECML/PKDD 2007 Dataset. Additional data supporting the findings of this study are available from the corresponding author upon reasonable request.

Ethical Declarations

Not applicable. This study does not involve human participants, human data, or animals.

Acknowledgments

The author would like to thank the reviewers and editors for their valuable comments and suggestions, which helped improve the quality of this manuscript.

Declaration of Generative AI in Scholarly Writing

Artificial Intelligence (AI) tools were used solely for language improvement, grammar checking, and manuscript editing assistance. All scientific content, methodology, experimental design, analysis, and conclusions were developed, verified, and approved by the author. The author takes full responsibility for the content of this manuscript.

Abbreviations

The following abbreviations are used in this manuscript:

AI – Artificial Intelligence
LLM – Large Language Model
WAF – Web Application Firewall
NLP – Natural Language Processing
SQL – Structured Query Language
XSS – Cross-Site Scripting
HTTP – Hypertext Transfer Protocol
API – Application Programming Interface

Appendix A/B/C…, with appendix title

Not applicable.

References

[1] Khalaf, Y., Aljaidi, M., Laila, D.A., Alsarhan, A., Alkhawaldeh, A.K., Alsuwaylimi, A.A. and Kharabsheh, M., 2025. An Effective Encryption Algorithm Based on RSA and DES. International Journal of Communication Networks and Information Security, 17(4), pp.10-19.

[2] Laila, D. A., Aljaidi, M., Almaiah, M. A., AlBourini, M., Al-Na'amneh, Q., Samara, G., and Momani, K., 2025. A Novel Scheme to Optimize LSB Steganography Based on a Logistic Chaotic Map and Genetic Algorithm. Iraqi Journal for Computer Science and Mathematics, 6(2), p.24.

[3] Al-Mousa, M., Amer, W., Abualhaj, M., Albilasi, S., Nasir, O. and Samara, G., "Agile Proactive Cybercrime Evidence Analysis Model for Digital Forensics," The International Arab Journal of Information Technology (IAJIT), vol. 22, no. 3, pp. 627–636, 2025, doi: 10.34028/iajit/22/3/15.

[4] Alazaidah, R., Al-Shaikh, A., Al-Mousa, R., Khafajah, H., Samara, G., and Alzyoud, M., 2024. Website Phishing Detection Using Machine Learning Techniques. Journal of Statistics Applications & Probability, 13(1), Article 8.

[5] Król, M., Janiszewski, P., & Mazurczyk, W., "Dynamic Web Application Firewall Detection Supported by Cyber Mimic Defense Approach", Journal of Network and Computer Applications, Vol. 213, Article 103596, 2023. DOI: 10.1016/j.jnca.2023.103596

[6] Kakisim, A. G.,"A Deep Learning Approach Based on Multi-View Consensus for SQL Injection Detection", International Journal of Information Security, Vol. 23, pp. 1541–1556, 2024.DOI: 10.1007/s10207-023-00791-y

[7] Husari, G., Al-Shaer, E., Ahmed, M., & Chu, B., "Natural Language Processing for Cybersecurity: A Survey", ACM Computing Surveys, Vol. 56, No. 2, pp. 1-38,2023. DOI: 10.1145/3597303

[8] Zhu, X., Zhou, W., Han, Q.-L., Ma, W., Wen, S., & Xiang, Y., "When Software Security Meets Large Language Models: A Survey", IEEE/CAA Journal of Automatica Sinica, Vol. 12, No. 2, pp. 317-334, 2025. DOI: 10.1109/JAS.2024.124971

[9] Apruzzese, G., Andreolini, M., Marchetti, M., Colajanni, M., & Ferretti, L.,"Deep Learning for Cybersecurity: The Adversarial Case", IEEE Transactions on Artificial Intelligence, Vol. 4, No. 1, pp. 21-35,2023. DOI: 10.1109/TAI.2022.3141259

[10] Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H.,"Deep Learning and Transformer-Based Approaches for Cyber Threat Detection", IEEE Communications Surveys & Tutorials, Vol. 25, No. 2, pp. 1024-1062, 2023. DOI: 10.1109/COMST.2023.3241234

[11] Jaffal, N. O., Alkhanafseh, M., & Mohaisen, D., "Large Language Models in Cybersecurity: Applications and Challenges", AI, Vol. 6, No. 9, Article 216, 2025. DOI: 10.3390/ai6090216

[12] Alzahrani, N., Alenazi, M., & Alshammari, R., "Web Application Security: Threats, Vulnerabilities and Defense Mechanisms", Computer Networks, Vol. 235, Article 109957 ,2024. DOI: 10.1016/j.comnet.2023.109957

[13] Dawadi, B. R., Adhikari, B., & Srivastava, D. K., "Deep Learning Technique-Enabled Web Application Firewall for the Detection of Web Attacks", Sensors, Vol. 23, No. 4, Article 2073, 2023.DOI: 10.3390/s23042073

[14] Kim, J., Lee, H., & Park, Y., "Adaptive Learning Frameworks for Cybersecurity Systems: Challenges and Opportunities", IEEE Security & Privacy, Vol. 21, No. 3, pp. 45-53, 2023. DOI: 10.1109/MSEC.2023.3254478

[15] Ferrag, M. A., Alwahedi, F., Battah, A., Cherif, B., Mechri, A., & Tihanyi, N., "Generative AI in Cybersecurity: A Comprehensive Review of Large Language Models Applications and Vulnerabilities", Internet of Things and Cyber-Physical Systems, Vol. 5, pp. 1–46, 2025. DOI: 10.1016/j.iotcps.2025.01.001

Authors' Profiles

Yousef Khalaf received his B.Sc. degree in Cyber Security from Zarqa University, Zarqa, Jordan. His academic interests include cybersecurity, web application security, penetration testing, network security, digital forensics, cryptography, artificial intelligence in cybersecurity, and security automation.

He is currently a researcher. His research focuses on intelligent security systems, Web Application Firewalls (WAFs), Large Language Models (LLMs) for threat detection, malware analysis, and secure software development. He is the author of several scientific publications in the fields of cybersecurity and information security.

Mr. Khalaf is a member of various academic and cybersecurity communities. His achievements include publishing peer-reviewed research papers in international journals and actively participating in cybersecurity training programs and technical research activities. His current research interests include AI-driven cybersecurity, threat intelligence, digital forensics, and secure network architectures.

How to cite this paper

Yousef Khalaf, "LLM-WAF: An Intelligent Web Application Firewall Powered by Large Language Models for Advanced Threat Detection", International Journal of Wireless and Microwave Technologies(IJWMT), Vol.16, No.4, pp. 317-327, 2026. DOI:10.5815/ijwmt.2026.04.18