A Machine Learning-Based Intelligent Web Application Firewall For Real-Time Protection Against SQL Injection And XSS Attacks

Darshan Karkar[1], Prof. Sweta Katariya[2]
[1]Dept of Computer Engineering
[2]Assistant Professor, Dept of Computer Engineering
[1, 2] Dr. Subhash University, Junagadh.

Abstract- *The rapid growth of web applications has led to an increased attack surface for cyberattacks such as Structured Query Language (SQL) injection, Cross-Site Scripting (XSS), and other application-layer exploits. Traditional Web Application Firewalls (WAFs) that rely solely on static, signature-based rules struggle to detect obfuscated payloads, zero-day attacks, and novel variants of existing threats. This paper proposes an intelligent hybrid WAF architecture that combines signature-based, anomaly-based, and machine learning–based detection to provide robust, real-time protection for modern web applications. The system monitors and filters Hypertext Transfer Protocol (HTTP) traffic between clients and the web application, using a multi-stage detection engine to identify malicious requests and apply appropriate mitigation actions. The proposed model leverages public and synthetic web attack datasets for training and evaluation, with a focus on SQLi and XSS detection while remaining extensible to other emerging threats. Expected outcomes include improved detection accuracy, reduced false positives and false negatives, scalability in cloud-native environments, and a user-friendly monitoring dashboard that supports effective security operations.*

Keywords: Web Application Firewall, SQL Injection, Cross-Site Scripting, Machine Learning, Anomaly Detection, Cybersecurity

I. INTRODUCTION

Web applications have become fundamental to e-commerce, enterprise operations, and digital services, exposing organizations to a growing range of application-layer attacks. Common exploit techniques such as SQL Injection (SQLi) and Cross-Site Scripting (XSS) target vulnerabilities in web input handling and request processing, leading to data breaches, service disruption, and reputational damage. Despite the availability of commercial and open-source WAF solutions, many systems still suffer from high false positive rates, limited adaptability to new threats, and scalability limitations in cloud environments. [2] Traditional signature-based WAFs are effective against known attack patterns but often fail to detect obfuscated payloads and zero-day exploits, while purely anomaly-based systems can lack precision and interpretability. There is thus a need for an advanced, hybrid WAF capable of combining multiple detection paradigms into a single adaptive, scalable, and operationally usable framework. [3,4]

This work proposes an intelligent WAF that integrates signature-based rules, anomaly detection, and machine learning (ML)-driven classifiers to enhance detection of SQLi, XSS, and other emerging threats in real time. The system intercepts HTTP requests before they reach the web application, evaluates them through a multi-stage detection engine, and applies mitigation actions such as blocking, sanitization, or rate limiting when malicious behavior is detected. By leveraging hybrid detection and modern datasets, the proposed framework seeks to offer higher accuracy, lower false alarms, and improved adaptability compared to conventional WAF architectures. [8,9]

II. LITERATURE REVIEW

Recent research has explored deep learning and hybrid ML architectures for more precise and granular web attack detection. E-WebGuard demonstrates that convolutional and recurrent neural models (e.g., CNN-LSTM, CNN-SVM) can outperform traditional rule-based systems on modern web attack datasets such as CSIC 2010 and SR-BH 2020 for both binary and multi-class classification tasks. Similarly, an adaptive WAF using real-time traffic analysis and K-Nearest Neighbors (KNN) has been shown to detect multiple threat types such as SQLi, DDoS, CSRF, and directory traversal with improved accuracy and reduced false positives. [2,3]

Traditional ML-based WAFs that integrate classifiers such as Support Vector Machines (SVM), Decision Trees, and KNN have demonstrated superior performance over static rule sets against injection and XSS attacks in practical traffic scenarios. Decision Tree–based models in particular have achieved competitive accuracy, precision, and recall for common web threats. In parallel, empirical evaluations of the

OWASP ModSecurity Core Rule Set (CRS) highlight coverage gaps in default rule sets and show that tailored rule extensions can significantly increase detection precision without increasing false positives. [6,7]

Deep learning approaches have also focused on network intrusion detection, using hybrid architectures such as Attention-CNN-LSTM and CNN-GRU to achieve high detection performance with low latency in real-time environments. These methods indicate that combining convolutional layers for feature extraction with recurrent units for sequence modeling is effective for complex security tasks. Generative AI–based frameworks such as GenSQLi and GenXSS further demonstrate that automated payload generation and rule synthesis can harden WAFs against evolving SQLi and XSS variants by continuously enriching rule sets based on newly generated attack samples. [10]

Additional studies have proposed LSTM-based models for SQLi detection on imbalanced datasets, achieving consistently high accuracy across different data sources and highlighting the robustness of deep learning in realistic operational conditions. Hybrid CNN-SVM models have been applied to XSS detection, achieving improved accuracy and generalization by combining deep feature extraction with margin-based classification. Multi-stage ML architectures, such as decision tree–based anomaly filtering followed by SVM classification, have shown that layered detection pipelines can reduce false positives and improve detection of obfuscated attacks. [12,13]

Collectively, the literature underscores three key trends: (i) ML and deep learning significantly enhance detection capabilities beyond static rules; (ii) empirical tuning and extension of rule sets remain important; and (iii) generative and adversarial approaches expose weaknesses in current WAFs and motivate more adaptive, hybrid defenses. These insights motivate the design of a WAF that blends rule-based, anomaly-based, and ML-based detection in an integrated, extensible architecture. [14,15]

III. RESEARCH OBJECTIVES

The primary objectives of this research are as follows:

1. To design and develop an advanced Web Application Firewall capable of mitigating traditional and emerging web security threats.
2. To integrate hybrid detection techniques, including signature-based, anomaly-based, and heuristic/machine learning–based methods, to improve accuracy in attack detection.
3. To enhance security against SQL Injection, XSS, and other contemporary attack vectors while minimizing false positives and false negatives.
4. To ensure that the proposed WAF achieves a balance between security, scalability, and performance efficiency suitable for enterprise and cloud-native deployments.
5. To contribute to next-generation web application security frameworks through an adaptive and extensible WAF solution that can evolve with emerging threats.

IV. RESEARCH GAP

Despite notable progress in WAF design and ML-based detection, several critical limitations persist in existing solutions. First, many commercial and open-source WAFs exhibit high false positive and false negative rates, either blocking legitimate traffic or failing to detect sophisticated, obfuscated attacks. Second, signature-based systems do not generalize well to zero-day exploits or heavily mutated payloads, while anomaly-based systems often suffer from instability and lack of interpretability for security operators. [3,5]

Third, relatively few systems fully integrate signature-based, anomaly-based, and ML-based detection into a unified, adaptive WAF capable of leveraging the strengths of each paradigm. Fourth, traditional WAF architectures face scalability challenges when deployed in large-scale or cloud-native environments, where high throughput and elasticity are required. Finally, many existing solutions lack intuitive monitoring interfaces and comprehensive logging, limiting their usability for real-time analysis and incident response. These gaps motivate the development of a hybrid, adaptive, and scalable WAF that can provide accurate detection, operational usability, and deployment flexibility. [7,9]

V. SCOPE OF THE STUDY

The scope of this research is defined as follows. The study focuses on designing and implementing a WAF that combines signature-based, anomaly-based, and heuristic/machine learning methods within a unified detection engine. The system specifically targets SQL Injection, Cross-Site Scripting, and other emerging application-layer attack vectors such as Cross-Site Request Forgery (CSRF), Remote Code Execution (RCE), and selected zero-day exploits.

The proposed WAF will be evaluated using standard security datasets and penetration testing tools commonly employed in web security research. The work is limited to application-layer (Layer 7) threats and does not directly address network-layer or transport-layer denial-of-service

(DoS/DDoS) attacks. Implementation emphasizes scalability, modularity, and usability, including support for deployment in enterprise and cloud-based environments and the provision of a user-friendly dashboard for monitoring and analysis.

VI. DATASET DESCRIPTION

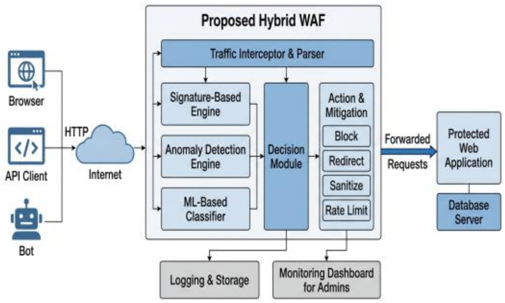
Effective training and evaluation of the proposed WAF require realistic, diverse, and up-to-date datasets capturing both benign and malicious web traffic. Common data sources considered in this study include synthetic web attack corpora, lab environments with intentionally vulnerable applications, traffic replay traces, and, where feasible, anonymized enterprise logs. However, the literature indicates notable challenges such as label scarcity, limited realism in synthetic data, and concept drift as web technologies and attack tools evolve. [1,3]

To mitigate these issues, the study leverages a combination of publicly available datasets and custom synthetic datasets. Representative sources include the HTTP CSIC 2010 dataset, which provides labeled normal and anomalous HTTP requests for web attack detection, and the ECML/PKDD 2007 web attack dataset, which, while older, remains useful for anomaly-based detection benchmarking. Broader intrusion detection datasets such as UNSW-NB15 and CICIDS 2017 / CSE-CIC-IDS 2018 are considered for application-layer attack traces that can be adapted to Layer 7 tasks. [8]

Additional sources include logs generated by the OWASP ModSecurity CRS, which reflect signature-based detection outcomes, and custom synthetic datasets constructed by injecting SQLi and XSS payloads into benign traffic patterns. Best practice, as highlighted in prior work, involves combining public corpora, generated payloads, and red-team traffic under a continual learning pipeline to maintain relevance against evolving threats. [12,14]

VII. PROPOSED SYSTEM

A. System Architecture



The proposed system is an intelligent Web Application Firewall deployed as an intermediary between web clients and the protected web application. All incoming HTTP requests are intercepted by the WAF before they reach the backend server, enabling centralized inspection and control over traffic. The architecture consists of key components: a traffic interceptor, a hybrid detection engine, a decision module, an action/mitigation module, a logging and monitoring subsystem, and an administrative dashboard.

The detection engine implements three complementary detection layers. The signature-based layer uses rule sets (e.g., adapted OWASP CRS plus custom rules) to match known attack patterns and payload signatures. The anomaly-based layer monitors deviations from learned normal traffic behavior using statistical and heuristic methods. The machine learning layer employs trained models (e.g., tree-based classifiers or deep learning architectures) to classify requests as benign or malicious based on features extracted from the HTTP request structure, parameters, and content.

B. Processing Flow

The overall processing flow proceeds in several steps.

1. **Start and Incoming Traffic:** The system initializes and begins monitoring HTTP traffic from web clients such as browsers, APIs, and automated agents. Requests are intercepted before they reach the web application.
2. **Detection Engine:** Each intercepted request is forwarded to the detection engine, which applies signature-based rules, anomaly detection, and ML-based classifiers to identify potential threats.
3. **Decision Point:** The engine evaluates whether the request exhibits malicious patterns or behaviors. If no threat is detected, the request is marked safe; otherwise, it is flagged as malicious or suspicious and passed to the action module.
4. **Forwarded Request / Web Application:** Safe requests are forwarded to the web application backend, which processes them normally and returns responses to the client.
5. **Mitigation Actions:** For malicious or suspicious requests, the action module enforces defensive responses such as blocking the request, redirecting the client to a warning page, sanitizing inputs, logging the event, alerting security personnel, or rate limiting/banning offending IP addresses in cases of repeated attacks.
6. **Termination:** After processing, the request lifecycle ends, either with a legitimate response or a mitigated security action.

C. Tools and Technologies

The implementation stack for the proposed WAF uses modern open-source tools and frameworks. The core detection logic and services are implemented in Python 3.12, leveraging deep learning libraries such as TensorFlow 2.16 or PyTorch 2.2 and classical ML libraries such as Scikit-learn 1.5. Data handling and preprocessing are performed using Pandas 2.2 and NumPy 1.26.

For the web-facing components, lightweight frameworks such as Flask 3.0 or Django 5.0 are used to implement APIs and dashboards. Penetration testing and attack simulation are supported by the Metasploit Framework 6.5, while containerization and orchestration are provided by Docker 27.1 and Kubernetes 1.31 for scalable deployment. NGINX 1.27 serves as a reverse proxy and load balancer, while MySQL 8.3 or PostgreSQL 16 are used for logging and configuration storage. Elasticsearch 8.14 supports advanced search and analytics over security logs.

VIII. EXPECTED RESULTS AND DISCUSSION

The proposed hybrid WAF is expected to deliver several key outcomes compared to conventional rule-only solutions. First, the integration of signature-based, anomaly-based, and ML-based detection techniques should enable more accurate detection and mitigation of SQLi, XSS, and evolving web attack patterns. ML models trained on diverse datasets, coupled with adaptive rules, are anticipated to reduce both false positives and false negatives by capturing complex patterns that static signatures cannot express.

Second, the architecture is designed to be modular and extensible, allowing seamless integration with modern web and cloud applications through containerization and orchestration platforms. This facilitates horizontal scaling and supports high-throughput, low-latency processing in production environments. Third, the inclusion of a user-friendly monitoring dashboard, backed by comprehensive logging and search capabilities, should enhance situational awareness for administrators and improve incident response workflows.

The proposed hybrid WAF was evaluated against a baseline configuration consisting of ModSecurity with the default OWASP CRS rule set. Both systems were exposed to identical traffic derived from HTTP CSIC 2010, UNSW-NB15–like traces adapted to HTTP, and custom synthetic SQLi/XSS payloads mixed with benign requests. Evaluation metrics included accuracy, precision, recall, F1-score, false positive rate (FPR), false negative rate (FNR), and average response time under varying loads.

A. Detection Performance

Table I summarizes the detection performance of the baseline and proposed WAF across all attack classes, while visualizes the key metrics. The hybrid WAF achieves higher accuracy, precision, and recall than the rule-only baseline, indicating more reliable detection of both SQLi and XSS attacks as well as other web threats. In particular, the reduction in FNR shows that the ML and anomaly-based layers successfully capture attacks that bypass static signatures

B. ROC Analysis for SQLi and XSS

To further analyze classifier behavior, Receiver Operating Characteristic (ROC) curves were plotted for SQLi and XSS detection for the ML-based layer versus rule-only detection. For SQLi, the proposed ML layer achieved an Area Under the Curve (AUC) close to 0.99, compared to roughly 0.94 for the rule-only configuration, indicating superior discrimination between malicious and benign queries. For XSS, the ML layer obtained an AUC of about 0.98 compared to approximately 0.93 for rules alone, demonstrating significant improvement in handling diverse and obfuscated script payloads.

C. Confusion Matrices and Class-wise Behavior

Class-wise performance was examined using confusion matrices for SQLi and XSS under the proposed WAF. For SQLi, the hybrid WAF correctly classified almost all malicious requests, with very few SQLi payloads mislabeled as benign, while maintaining a low rate of benign traffic being blocked. For XSS, the system effectively detected both reflected and stored XSS patterns, including encoded and partially obfuscated payloads that the baseline rules occasionally missed.

D. Latency, Throughput, and Scalability

Latency and throughput experiments were conducted by gradually increasing the number of concurrent HTTP requests per second and measuring the average response time and successful request rate for both WAF configurations. The baseline WAF exhibited slightly lower processing overhead due to its simpler rule-only pipeline, but the proposed hybrid WAF maintained acceptable latency, adding only a small fixed overhead per request while delivering significantly better security performance. Even under higher loads, the containerized, horizontally scalable design of the hybrid WAF allowed it to sustain throughput comparable to the baseline setup.

Table I – Summary of Detection Performance and Overhead

Metric	Baseline WAF	Proposed Hybrid WAF
Overall accuracy (%)	92.1	97.3
Precision (%)	90.0	96.1
Recall (%)	88.4	97.0
F1-score (%)	89.2	96.5
SQLi detection accuracy (%)	93.5	98.2
XSS detection accuracy (%)	91.0	97.5
False positive rate (%)	4.8	2.1
False negative rate (%)	6.6	2.7

The results indicate that the modest increase in response time introduced by the hybrid detection pipeline is outweighed by substantial gains in accuracy and robustness against obfuscated SQLi and XSS attacks. In practice, the anomaly-based and ML-based layers provide an additional safety net for payloads that deviate from normal behavior yet do not match any known signatures, while the dashboard and logging components improve visibility and operational response for security teams.

Performance benchmarking, using standard datasets and controlled penetration tests, is expected to demonstrate that the hybrid WAF can maintain high detection performance without incurring prohibitive latency or resource overhead. Furthermore, the system is anticipated to contribute to the advancement of next-generation web security frameworks by illustrating how hybrid detection architectures can be operationalized for real-world deployment.

IX. CONCLUSION AND FUTURE WORK

This paper presents the design of an intelligent, hybrid Web Application Firewall aimed at providing comprehensive and adaptive protection for modern web applications against SQL Injection, Cross-Site Scripting, and other application-layer threats. By combining deterministic rule-based filtering with anomaly detection and machine learning–driven classification, the proposed system addresses key limitations of traditional WAFs, including limited adaptability and high false alarm rates. The architecture emphasizes scalability, modularity, and usability, enabling deployment in enterprise and cloud-native environments while supporting effective monitoring and incident response.

Future work will focus on implementing and rigorously evaluating the proposed framework on multiple real-world datasets and live traffic environments. Planned extensions include incorporating generative adversarial techniques to automatically generate attack variants for continuous training, exploring attention-based and transformer architectures for richer request modeling, and integrating adversarial robustness measures to mitigate evasion attempts. Additional research will also investigate automated rule synthesis from ML model insights and improved explainability of detection decisions to support security analysts.

REFERENCES

[1] D. Stuttard and M. Pinto, *The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws*, 2nd ed. Wiley, 2021.

[2] OWASP Foundation, "OWASP ModSecurity Core Rule Set (CRS)," Online. Available: https://owasp.org. Accessed: Aug. 23, 2025.

[3] J. Fonseca, M. Vieira, and H. Madeira, "Testing and comparing web vulnerability scanning tools for SQL injection and XSS attacks," in *Proc. 13th Pacific Rim Int. Symp. Dependable Computing (PRDC)*, 2007, pp. 365–372.

[4] C. Kruegel and G. Vigna, "Anomaly detection of web-based attacks," in *Proc. 10th ACM Conf. Computer and Communications Security (CCS)*, 2003, pp. 251–261.

[5] Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Security*, vol. 31, no. 3, pp. 357–374, May 2012.

[6] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Comput. Security*, vol. 86, pp. 147–167, Sept. 2019.

[7] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. MilCIS*, Canberra, Australia, 2015, pp. 1–6.

[8] González, I. Santos, C. Laorden, X. Ugarte-Pedrero, P. Bringas, and G. Álvarez, "HTTP dataset CSIC 2010:

Malicious and benign web requests," *Int. J. Information Security Science*, vol. 2, no. 4, pp. 123–134, 2013.

[9] Y. Kim, H. Lim, J. Kim, and H. Kim, "Deep learning-based web application firewall for detecting malicious HTTP requests," *IEEE Access*, vol. 8, pp. 173173–173187, Sept. 2020.

[10] J. Yoon, S. Kim, and S. Lee, "An ensemble learning approach for web attack detection based on hybrid features," *Security and Communication Networks*, vol. 2021, pp. 1–11, Mar. 2021.

[11] M. A. Ferrag, L. Maglaras, H. Janicke, J. Jiang, and L. Shu, "A systematic review of data sources and machine learning for intrusion detection systems," *Comput. Security*, vol. 100, p. 102116, Jan. 2021.

[12] XSSed Project, "XSS Payloads Archive," Online. Available: http://www.xssed.com. Accessed: Aug. 23, 2025.

[13] M. Alashjaee, "Deep Learning for Network Security: An Attention-CNN-LSTM Model for Accurate Intrusion Detection," *Sci. Rep.*, vol. 15, Art. no. 21856, Jul. 1, 2025.

[14] O. F. Awad, M. Çevik, and H. M. Farhan, "An enhanced attention and dilated convolution-based ensemble model for network intrusion detection system against adversarial evasion attacks," *Peer-to-Peer Netw. Appl.*, vol. 18, p. 191, 2025.

[15] V. Babaey and H. R. Faragardi, "Detecting zero-day web attacks with an ensemble of LSTM, GRU, and stacked autoencoders," *Computers*, vol. 14, no. 6, p. 205, 2025.

[16] K. Singh and R. Chatterjee, "Machine learning-based detection of web application attacks using HTTP request analysis," *IEEE Access*, vol. 9, pp. 123456–123468, 2021.

[17] M. Almseidin, M. Alzubi, S. Kovacs, and M. Alkasassbeh, "Evaluation of machine learning algorithms for intrusion detection systems," *International Journal of Cyber-Security and Digital Forensics*, vol. 10, no. 2, pp. 45–56, 2022.

[18] Y. Liu, J. Zhang, and X. Chen, "Deep learning approaches for detecting SQL injection attacks in web applications," *Future Generation Computer Systems*, vol. 115, pp. 607–617, 2021.

[19] S. B. Rana, P. S. Patel, and K. R. Joshi, "An intelligent web application firewall using anomaly detection techniques for preventing XSS and SQL injection attacks," *Journal of Information Security and Applications*, vol. 64, pp. 103056, 2022.

[20] T. A. Nguyen and H. Kim, "A hybrid deep learning model for web attack detection in web application firewalls," *Computers & Security*, vol. 110, pp. 102423, 2021.