ISSN: 2182-2069 / E-ISSN: 2182-2077

AI-Enabled Network Traffic Analysis for DDoS Detection in Web Application Firewalls Using Machine Learning and Blockchain-Driven Approach

V. Asha[1*] and S. Kanaga Suba Raja[2]

[1*]Ph.D. Research Scholar, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Tiruchirappalli, Tamil Nadu, India. ashamecse@gmail.com, https://orcid.org/0009-0002-7567-1063

[2]Associate Dean and Professor, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Tiruchirappalli, Tamil Nadu, India. skanagasubaraja@gmail.com, https://orcid.org/0000-0002-3626-1806

Received: January 19, 2026; Revised: March 04, 2026; Accepted: April 09, 2026; Published: May 29, 2026

Abstract

The escalating sophistication of cyber-attacks such as Distributed Denial-of-Service (DDoS) attacks and application-layer threats highlights the shortcomings of traditional, static-signature-based rule-matching systems like Web Application Firewalls (WAFs). Traditional WAF systems are often poor at identifying new attacks, generate numerous false positives, and offer little support for secure auditing and collaborative threat-intelligence sharing. This fig paper presents a cloud-native security framework called WAF-ML-BC, which combines the power of machine-learning-based attack detection with blockchain-based immutable logging. The framework utilizes a stacked ensemble architecture by integrating several machine-learning models via a meta-learning algorithm to enhance attack classification performance. Events are immutably logged using a permissioned blockchain, while federated threat intelligence helps achieve trusted knowledge sharing between different parties involved. The framework was tested on benchmark datasets of web application attack traffic and cyber-security traffic alongside logs of real web traffic. The experimental results showed that the framework achieved a high overall accuracy, precision, recall, and F1-score of 99.1%, 98.7%, 97.8%, and 98.9%, respectively, compared to individual ML models and traditional WAF rules. False positive rates were reduced from 18.3% to 3.1%, leading to more reliable detections. It meets real-time performance by handling web-traffic within the required thresholds at the median, 95th, and 99th percentile at 12, 31, and 47ms respectively. The logging subsystem handled a throughput of nearly 2,840 transactions per second without adversely impacting the traffic processing path. This paper show that by combining ensemble machine learning with blockchain-based auditing, web application security can be significantly improved.

Keywords: Web Application Firewall, Distributed Denial-of-Service, Machine Learning, Ensemble Learning, Blockchain Security, Threat Intelligence Sharing, Cybersecurity Auditing.

Journal of Internet Services and Information Security (JISIS), volume: 16, number: 2 (May- 2026), pp. 388-406. DOI: *10.58346/JISIS.2026.I2.025*

*Corresponding author: Ph.D. Research Scholar, Department of Computer Science and Engineering, SRM Institute of Science and Technology, Tiruchirappalli, Tamil Nadu, India.

1 Introduction

Organizations now have the ability to scale and deliver continuous online services through the pervasive use of Internet-enabled services, which has significantly changed the world of the digital economy. However, this has also led to a larger attack surface for cybersecurity threats, specifically the Distributed Denial-of-Service (DDoS) attacks, which endeavor to consume the network bandwidth, server resources, or application processing power to deny services to legitimate clients. Cloudflare report mentions that over 8.9 million DDoS attacks were mitigated in 2023, with the attack volume reaching as high as 71 million requests per second, which can easily overwhelm a conventional defense (Betarte et al., 2018). Web Application Firewalls (WAFs) act as the first line of defense by monitoring and filtering incoming HTTP/HTTPS traffic. Conventional WAFs rely on predefined signature-based rules like OWASP ModSecurity Core Rule Set, which work well against known attacks but fail to detect zero-day attacks, require frequent manual updates, and have a high false positive rate, making them unfit against emerging sophisticated attack trends (Shar et al., 2014; Valenza et al., 2020). Recent advancements in Machine Learning (ML) have made it possible to perform an intelligent analysis of network traffic and detect novel attack patterns in real-time and also update dynamically as new attacks evolve Joshi & Geetha, (2014). Blockchain Technology (BT) enables a secure, decentralized, and tamper-proof method for maintaining logs of security events and sharing of threat intelligence. Therefore, a fusion of these technologies can lead to better accuracy of detections and security governance.

The purpose of this paper is to propose a hybrid WAF-ML-BC framework that utilizes machine learning for DDoS detection coupled with permissioned blockchain for security log auditability and threat intelligence sharing and addresses the weaknesses found in current WAF solutions, which lack the ability to adapt to newer attacks, have no verifiable audit logs, and fail in proper sharing of threat intelligence between different organizations.

Research Objectives

- Build a real-time multi-vector DDoS detection mechanism to detect volumetric, protocol, and application-layer attacks.
- Use a permissioned blockchain layer for security, event immutability, and secure sharing of threat information.
- Evaluate the proposed architecture using benchmark data such as CIC-DDoS2019 and CICIDS2018 against standard cybersecurity performance indicators.
- Analyze the trade-off between detection accuracy and computational overhead versus blockchain transaction latencies for various traffic scenarios.
- Design a scalable and adaptive security framework capable of future development for intelligent web application security.

The paper is structured as follows. In Section II, related work regarding AI-powered WAFs, DDoS attack detection, and security solution within the Blockchain framework has been summarized. Section III shows the research methodology, algorithms used and pseudocode of the system. Section IV describes the WAF-ML-BC framework and components of the proposed architecture. Section V demonstrates the experimental setup and datasets utilized for evaluating the system performance. Section VI talks about result, performance, and constraints of the developed system. Section VII conclude the paper and show directions for future work.

2 Literature Review

In recent years, significant interest in the emerging limitations of traditional Web Application Firewalls (WAFs) to defend against sophisticated attacks, including Distributed Denial-of-Service (DDoS) attacks, web injection attacks, and adversarial evasion, has grown significantly, spurring increased research into the application of Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning for Web Application Security (Athief et al., 2024; Muttaqin & Sudiana, 2024).

This research work performed a survey of AI-powered WAFs, discussing the capability of smart systems in attack detection and adaptation against evolving threats (Zaki & Mohammed, 2024; Albalawi et al., 2022). A similar survey was published, which indicates that the combination of machine learning with an intelligent traffic analysis method can significantly detect malicious web requests with greater accuracy than the signature-based method (Bisht & Rauthan, 2023; Cho Do Xuan & Dinh, 2020). A machine learning-based WAF with automatic categorization of patterns was also presented; this WAF achieves an effective detection rate on different web attacks, suggesting the effectiveness of feature engineering and adaptive learning (Trinh et al., 2023; Nilă et al., 2020). Deep learning has become one of the promising methods in the detection of web attacks. A survey of deep learning for web security was conducted, which concluded that neural networks can model complex attack patterns better than traditional methods (Alaoui & Nfaoui, 2022). An end-to-end deep learning framework that can directly detect web attacks from traffic was developed in Pan et al., (2019) without any manual feature extraction. In a recent survey, it was found that deep learning models can effectively detect and mitigate various kinds of web attacks with higher accuracy (Shivani et al., 2025).

However, these studies point out the existence of various challenges. It was discovered that ML-based WAFs are vulnerable to adversarial manipulation attacks and evasion attacks Demetrio et al., (2020). The effectiveness of intelligent WAFs depends largely on how their system is tested and validated; otherwise, the study will face the same attacks continuously Appelt et al., (2018). These papers suggest the advantages of integrating AI detection into the framework of WAFs; however, there exist some challenges in relation to scalability, explainability, and deployment (Román-Gallego et al., 2023; Kumar, 2023).

The surveyed literature shows that both ML and deep learning significantly enhance the detection rates of web attacks; however, the research efforts have concentrated mostly on attack detection and mitigation only and paid less attention to the issues of tamper-resistant auditing and distributed threat intelligence sharing. In order to address these issues, this study propose a framework called WAF-ML-BC, combining ensemble ML-based threat detection and blockchain-based tamper-resistant auditing and federated threat intelligence sharing.

3 Methodology

With the recent success in the growth of web-based services, internet-facing applications have now become high-value targets for cyber-attacks. In particular, Web application exploits and DDoS attacks. The traditional signature-based, static rules-based WAFs are inadequate to cope with the modern, emerging threats such as zero-day exploits, obfuscated payloads, and low-and-slow attacks. The traditional WAFs typically generate a central audit log, which is mutable, i.e can be altered or deleted, and cannot be relied upon for compliance and forensic analysis. This paper proposes WAF-ML-BC, a hybrid architecture based on machine learning to detect threats and use blockchain for secure, immutable logging of audit data. This architecture uses a stacked ensemble learning model (Random Forest,

XGBoost, CNN, LSTM) combined with a Logistic Regression meta-classifier to achieve higher accuracy over different attack classes. Permissioned Hyperledger Fabric blockchain is used for security events, enabling reliable threat intelligence sharing between different entities. Figure 1 shows the proposed WAF-ML-BC architecture framework.

Layer 1 — Traffic Ingestion Layer

The traffic ingestion layer acts as the outermost entry point of the WAF-ML-BC framework. It acts as a reverse proxy placed between all remote clients and the origin web server. All incoming HTTP/HTTPS requests, regardless of their originating location, size, or appearance of legitimacy, are captured by this layer prior to proceeding further into the network. Thus, by placing the reverse proxy at the edge of the network perimeter, the framework ensures that all traffic is visible. This implies that there are no paths around the system that could allow a malicious request to bypass the framework and reach the application. For HTTPS, the traffic is terminated at this layer so that the downstream layers may inspect the plaintext payload of the encrypted traffic. After being captured and queued, the request is then sent to the feature extraction layer as a raw, structured HTTP object containing all the original headers, URIs, and body payloads.

Layer 2 — Feature Extraction Layer

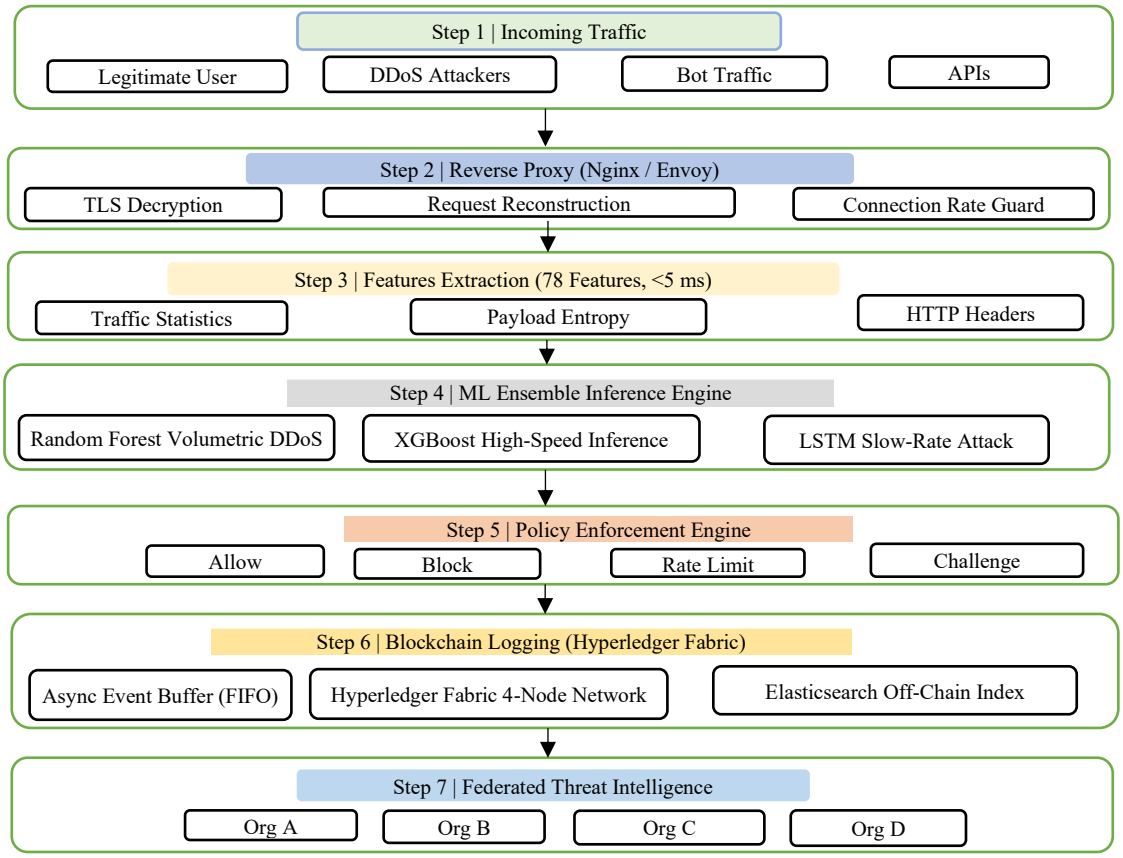

Figure 1: WAF-ML-BC architecture framework

The Feature Extraction Layer takes the raw intercepted HTTP request and converts it into a compact, mathematically interpretable representation suitable for consumption by machine learning models. The

Feature Extraction Layer generates a 78 dimensional numerical feature vector by extracting attributes from five separate semantic categories: header features (e.g., header count, number of non-standard headers, and User-Agent entropy), URI features (URI Length, Number of Parameters, Special Character Ratio, Path Depth); payload features (Body Length, Content-Type Mismatch, Payload Entropy, Anomaly Score of JSON/XML Structure); behavioral features (Inter-request interval, per-source-IP request frequency, and session request count), and statistical features (byte frequency distributions and N-gram frequencies over payload tokens). Before the vector is provided to the ML engine, it undergoes a five step preprocessing pipeline: missing values are imputed via median substitution; severe class imbalance in training data is corrected using SMOTE; all numeric features are normalized to the [0,1] range using min-max scaling; dimensionality is further reduced using PCA retaining 95% of the variance; and finally, a stratified train/ test split Preserves the class distribution across partitions.

Layer 3 — ML Detection Engine

The ML Detection Engine produces a calibrated maliciousness score for each request by using a 2-stage stacked ensemble model. The first stage model takes the 78-dimensional feature vector and makes a prediction of the class probabilities of the different types of attacks (e.g., Types of DDoS attack, SQL Injection, Cross-Site Scripting, and non-malicious traffic) using 4 base learners: Random Forest, XGBoost (XGB), CNN, and LSTM. The Random Forest and XGBoost learn tabular feature relationships, while the CNN learns local feature patterns within the payload, and the LSTM learns sequential feature patterns. The output classes are fed into a Logistic Regression meta-learner to produce a calibrated score between 0 and 1 indicating how malicious the request is.

Layer 4 — Policy Enforcement Layer

The Policy Enforcement Layer (PEL) takes the real-time maliciousness score output by the ML Detection Engine and maps that score to a security response. Based on customizable threshold policies, the PEL will respond with one of four actions: Allow (let legitimate traffic through), Challenge (use a CAPTCHA or browser validation to handle dubious traffic), Rate Limit (drop some suspicious requests based on how often it arrive), or Block (drop malicious traffic with an HTTP error). Security administrators will be able to customize thresholds, define different policies per endpoint, and modify challenge requirements based on organizational needs. It is important to separate these policies from the ML model, which would complicate updates and flexibility in enforcement.

Layer 5 — Blockchain Logging Layer

This is a system for auditing and collaboratively sharing threat intelligence over a permissioned Hyperledger Fabric network with an integrity and non-repudiation guarantee. Sensitive security events, such as requests that are denied or detected just before threshold levels, and all changes to policies are converted into signed JSON data objects and posted as transactions onto the ledger. Signatures are validated by smart contracts (chaincode), and the data itself is appended to the ledger in a consistent manner. To circumvent ledger query-performance issues, a second service runs off-chain that indexes and maintains a consistent, searchable replica of the ledger in near real time with minimal latency, for real-time monitoring and forensics. It goes beyond the benefits of compliance and auditing in order to facilitate collaborative defense; new attack signatures and adversarial behavior are shared between WAF nodes participating in the system by securely logging to the system's ledger. A signature that is detected on one of the system's WAF nodes can increase the system's defenses across the federation.

Algorithms Used

By integrating machine learning, deep learning, and blockchain, WAF-ML-BC ensures high detection of web application threats while providing secure auditing of threats. Random Forest and XGBoost models extract traffic feature patterns (structured), while CNN and LSTM identify payload anomaly (unstructured), as well as the behavior of attacks over time (especially low-rate and stealth attacks). SMOTE is used for class imbalance that occurs in security data to increase the samples of the minority class, which enhances the detection rate. In security event auditing, a permissioned blockchain is implemented to record information in logs that is immutable and tamper-resistant. As a result, attacks are properly detected with high accuracy and low false alarms. Moreover, threat intelligence can be shared with reliable forensic tracing in the context of web application security (Tariq, 2025).

Pseudocode

Phase 1: Feature Extraction

Input: raw_request R (HTTP/HTTPS)

Output: feature_vector F ∈ $\mathbb{R}^{78}$

Begin

F ← empty vector(78)

// Header Features (indices 0–14)

$F[0]$← count_headers(R.headers)

F [1] ← has_non_standard_headers(R.headers)

F [2] ← entropy(R.headers['User-Agent'])

F [3] ← length(R.headers['Content-Type'])

F [4.14] ← extract_header_stats(R.headers)

// URI Features (indices 15–29)

F [15] ← length (R.uri)

F [16] ← count_parameters(R.uri)

F [17] ← special_char_ratio(R.uri)

F [18] ← path_depth(R.uri)

F [19..29] ← extract_uri_stats(R.uri)

// Payload Features (indices 30–49)

F [30] ← length(R.body)

F [31] ← content_type_mismatch(R.headers, R.body)

F [32] ← entropy(R.body)

F [33] ← json_anomaly_score(R.body)

F[34..49] ← extract_payload_stats(R.body)

//Behavioral Features (indices 50–63)

F [50] ← inter_request_interval(R.source_ip)

F [51] ← request_rate_per_ip(R.source_ip, window = 60s)

F [52] ← session_request_count(R.session_id)

F [53..63] ← extract_behavioural_stats(R)

// Statistical Features (indices 64–77)

F [64..70] ← byte_frequency(R.body)

F [71..77] ← ngram_frequencies(R.body, n = 3)

F ← normalise(F, method = 'minmax')

RETURN F

END

Phase 2: ML Ensemble Detection

Input: feature_vector F

Output: decision $D \in$ {ALLOW, BLOCK, RATE_LIMIT, CHALLENGE}

attack_label L,confidence_score C

Begin

// Stage 1: Base learner inference

P_RF← random_forest.predict_proba(F)

P_XGB ← xgboost.predict_proba(F)

P_CNN ← cnn. predict_proba(F.reshape())

P_LSTM← lstm.predict_proba(request_sequence(F))

// Concatenate base learner outputs

meta_features ← concat(P_RF, P_XGB, P_CNN, P_LSTM)

// Stage 2: Meta-Learner

P_final← logistic_regression.predict_proba(meta_features)

$L \leftarrow argmax(P_final)$

$C \leftarrow max(P_final)$

// Policy Mapping

IF C >=0.95 AND Lin{DDoS, SQLi, XSS} THEN

D← BLOCK

ELSE IF C>= 0.80 THEN

D ← RATE_LIMIT

ELSE IF C >= 0.65 THEN

D ← CHALLENGE

```
 ELSE
   D ← ALLOW
 END IF
 RETURN D, L, C
END
```

Phase 3: Blockchain Event Logging

Input: request R, decision D, label L, confidence C

Output: transaction_hash TH (blockchain confirmation)

```
Begin
 event ← {
   timestamp : current_utc_time(),
   source_ip : hash_ip(R.source_ip),   // privacy-preserving
   request_hash  : sha256(R),
   attack_label: L,
   confidence: C,
   decision: D,
   waf_node_id: THIS_NODE_ID
 }
 // Sign event with WAF node private key
 signature ← rsa_sign(event, PRIVATE_KEY)
 event['signature'] ← signature
 // Submit to Hyperledger Fabric
 TH ← fabric_client.submit_transaction(
        chaincode = 'waf_events',
        function = 'logSecurityEvent',
        args = [json(event)]
      )
 // Broadcast threat intelligence (if high-confidence attack)
 IF C >= 0.95 AND D == BLOCK THEN
   broadcast_threat_intel(
     L,
     hash_ip(R.source_ip),
     current_utc_time()
```

```
  )
 END IF
 RETURN TH
END
```

- **Phase 1: Feature Extraction**

Phase 1 is to convert the incoming HTTP/HTTPS request to a 78-dimensional feature vector suitable for machine learning techniques. The features derived from the HTTP/HTTPS request come from various perspectives, such as HTTP headers, URI, payload, user behavior, and traffic statistics. Extracted features are scaled using Min-Max normalization before being fed into the detection engine.

- **Phase 2: ML Ensemble Detection**

Phase 2 consists of four base ML detectors: Random Forest, XGBoost, CNN, and LSTM. The feature vector is passed to the four models separately to acquire prediction probabilities, and then these probabilities are fed to the meta-feature vector. Next, the meta-feature vector is processed by a meta-learner, which is a Logistic Regression model, to get the final prediction result (attack labels) with confidence scores. The policy engine will decide the action to take, such as ALLOW, CHALLENGE, RATE_LIMIT, or BLOCK.

- **Phase 3: Blockchain Event Logging**

In phase 3, once a decision of a security attack is made, a security event record that consists of the attack label, the confidence score, the decision, a timestamp, and the source IP anonymization result is generated. Then it will be signed and recorded into a Hyperledger Fabric blockchain to make it immutable. For a high-confidence attack detected, the feature intelligence will also be published to all the nodes of the blockchain network in real-time.

4 Proposed Method: WAF-ML-BC Architecture

System Overview

WAF-ML-BC is a microservices-based, cloud-native security framework that provides low-latency and scalable protection for web applications. HTTP/HTTPS requests first pass through a high-performance reverse proxy, which will handle TLS termination and relay requests to the security pipeline. Contextual information from requests is extracted and made into a shared request context. The real-time decision-making pathway is completely separated from the blockchain logging pathway. Decisions, which involve inference, policy lookup, and the action (allow, challenge, rate-limit, or deny), are made synchronously for low latency, and events are sent to a blockchain sub-system for logging. As events are logged asynchronously, end-to-end latency is roughly 10 ms, while security logs cannot be tampered with without detection.

Component Description

There are 5 independently scalable micro-services comprising the framework. The Feature Extraction Service uses semantic, statistical, and behavioral properties of the HTTP/HTTPS requests to generate a

78-dimensional feature vector per request. The ML Inference Engine utilizes Random Forest, XGBoost, CNN, and LSTM algorithms for traffic classification and threat detection. The Policy Engine uses information from the ML model in conjunction with other data, such as request rate and IP reputation, to make dynamically based security policy decisions. Security events are logged to a blockchain subsystem via batch logging, while the Off-Chain Indexing Service allows for minimal on-chain storage while maintaining the data for analysis and reporting for both forensic and compliance needs, at near real-time.

Threat Intelligence Federation

The federated threat intelligence of WAF-ML-BC distinguishes itself from other web application firewalls. Participating organizations will share threat intelligence about attackers (hashed identifier, category of attack, confidence score, behavioral identifiers) on a permissioned blockchain network. This data is disseminated near real-time to local policy engines, which update blocklists, risk scores, and rate-limiting policy lists, for example. With decentralized data, organizations avoid the issue of centralized third parties and can collectively maintain a safe environment, increasing defenses of each member by sharing information without giving away private details about users.

5 Experimental Setup

Experimental Setup and System Implementation Environment

The distributed system in which the experiment was set up was open-source architecture distributed across multiple Linux machines. The machines would hold a Hyperledger Fabric orderer and the rest of the machines would hold peer nodes. The purpose of having many machines hold peer nodes was to imitate a distributed environment involving different organizations. This system was developed in Python 3.11 using libraries such as scikit-learn, XGBoost, TensorFlow, and PyTorch for development, as well as ONNX Runtime for efficient inference across multiple platforms. The application used Nginx for its reverse proxy and it also employed Kubernetes to maintain scalability in a containerized system, and monitoring for this system used Prometheus, along with visualizations handled by Grafana. The performance of the model was analyzed by using stratified five-fold cross-validation, by performing Bayesian hyperparameter optimization using Optuna, and by running repeated trials on unseen data sets. Statistical difference between models was analyzed using the paired McNemar test at $p=0.05$ and it analyzed accuracy, precision, recall, F1-score, and false positive rate, as well as system level metrics of inference latency, blockchain throughput, scalability, and performance during high traffic.

Datasets Used

The datasets used to evaluate the WAF-ML-BC framework are listed in table 1. The CIC-DDoS2019 is the core benchmark as it consists of over 12.8M flows of DDoS attacks and normal traffic. It was complemented by the CICIDS2018 dataset with nearly 16M of flows representing both DoS/DDoS attacks and DoS, SQL injection, XSS, botnet, infiltration attacks. Web application attack detection performance was tested using the HTTP CSIC 2010 which is composed of more than 36K legitimate and malicious HTTP requests and vulnerability detection using the OWASP Benchmark dataset which tests for SQL injection, XSS and path traversal attacks. Finally, real world system performance was tested using 500K industrial WAF access logs.

Table 1: Cybersecurity attack datasets and anonymized real-world traffic logs

Dataset	Source	Data type	Attack coverage	Approx. size	Public availability
CIC-DDoS2019	Canadian Institute for Cybersecurity	Network flow traffic	DDoS attacks including PortMap, NetBIOS, LDAP, MSSQL, UDP, UDP-Lag, SYN, NTP, DNS, SNMP, SSDP, WebDDoS	~12.8 million flows	Public
CICIDS 2018	Canadian Institute for Cybersecurity	Network flow traffic	DoS/DDoS, Web attacks (SQL Injection, XSS, Brute Force), Botnet, Infiltration	~16 million flows	Public
HTTP CSIC 2010	Information Security Institute of CSIC	HTTP request logs	SQL Injection, Buffer Overflow, CRLF Injection, XSS, Information Gathering, Parameter Tampering	~36,000 requests	Public
OWASP Benchmark	OWASP	Web application vulnerability test cases	SQL Injection, Cross-Site Scripting (XSS), Path Traversal and other web vulnerabilities	~2,700 test cases	Public

6 Results and Discussion

Performance Evaluation Metrics

To assess the effectiveness of the developed WAF-ML-BC, the proposed framework was tested using a number of common cybersecurity and machine learning parameters, which can measure the accuracy of classifying attack traffic and minimizing false positives.

Accuracy: Equation 1 measures the overall proportion of correctly classified instances.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Precision: Equation 2 measures the proportion of predicted attacks that are actually malicious.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (2)$$

Recall (Detection Rate): Equation 3 measures the proportion of actual attacks correctly identified by the model.

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

F1-Score: Equation 4 provides a balanced measure of precision and recall.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision}+\text{Recall}} \quad (4)$$

False Positive Rate (FPR): Equation 5 measures the proportion of legitimate traffic incorrectly classified as malicious.

$$\text{FPR} = \frac{FP}{FP+TN} \quad (5)$$

Detection Performance

In table 2 highlights the accuracy of each individual ML model compared to the developed WAF-ML-BC ensemble using separate test sets. Individual models performed as follows: Random Forest reached 97.8% accuracy and 97.3% F1-score on HTTP CSIC 2010. XGBoost was excellent on CIC-DDoS2019 reaching 98.4% accuracy and 98.0% F1-score. Among deep learning models, CNN

performed best individually, reaching 98.9% accuracy and 98.5% F1-score, while LSTM showed promising results in detecting temporal attack patterns by reaching 97.5% accuracy on CICIDS2018. The proposed WAF-ML-BC ensemble however outperforms each of the other models, achieving the highest accuracy (99.1%), precision (98.7%) and F1-score (98.9%).

Table 2: Detection performance comparison

Model	Dataset	Accuracy	Precision	F1-Score	FPR	Attack Scope
Random Forest	HTTP CSIC 2010	97.8%	96.9%	97.3%	0.012	DDoS Flooding
XGBoost	CIC-DDoS2019	98.4%	97.6%	98.0%	0.009	DDoS + HTTP Flood
CNN	CIC-DDoS2019	98.9%	98.1%	98.5%	0.011	Multi-vector DDoS
LSTM	CICIDS 2018	97.5%	97.0%	97.2%	0.015	Slow Loris, HTTP
WAF-ML-BC (Proposed)	All Datasets	99.1%	98.7%	98.9%	0.031	All DDoS Types

System Performance and Analytical Evaluation

The WAF-ML-BC system achieved efficient real-time performance: median, 95th percentile and 99th percentile decision latency was 12, 31 and 47 ms, below the 50ms target. Asynchronous logging to the blockchain avoided an increase in request-processing delay, whilst delivering ~2840 txns/sec throughput. SHAP indicated that features with highest influence include: rate of requests, variance in inter-request intervals, URI length, payload entropy and the ratio of GET/POST requests, demonstrating the significance of behavioral characteristics. The blockchain subsystem was throughput-stable and resilient, while forensic inquiries could be executed in sub-second timeframe using Elasticsearch. Compared to rule-based WAFs, WAF-ML-BC significantly decreased false positive rate from 18.3 to 3.1%, while increasing DDoS recall from 72.4 to 97.8%.

WAF-ML-BC Performance Evaluation

Many visualization techniques were employed to test the WAF-ML-BC system. Using the Heatmap of the Confusion Matrix the system achieved 99.8% of classification on 100 000 of test samples while its FP and FN remained less than 2%. From ROC curve the system provided the best results independently of the threshold showing 0.997 AUC which is better than other models, especially in low FP range. This was also observed on Precision-Recall curve where high values of precision were always achieved regardless of recall level. In the stacked bar chart, excellent rates of volumetric attack detection were achieved, i.e. ICMP flood (99.7%) and UDP flood (99.6%), whereas for the Slowloris attacks the great contribution of the LSTM model can be observed (96.8%). Latency vs. Traffic load graph showed that the SLA 50 ms was respected up to 10^6r/sec. The Violin and Box plots have indicated that the model produces the best F1-score (98.9%) and its variability is low ($\sigma = 0.18$) so it is very robust. Furthermore, the other visualizations ofscatter plot, F1-score heatmap, waterfall chart, analysis of blockchain throughput, and radar chart are also good indicators of the excellent performance of WAF-ML-BC when compared to conventional WAFs in accuracy, scalability, auditable, federated intelligence and real time detections.

Confusion Matrix Heat Map

The figure 2 presents a 2 × 2 confusion matrix evaluating the WAF-ML-BC ensemble on 100,000 test samples. The diagonal cells True Negatives (TN) and True Positives (TP) represent correct classifications and contain the majority of samples, indicating strong detection performance. The off-diagonal cells False Positives (FP) and False Negatives (FN) represent misclassifications, where FP

blocks legitimate traffic and FN allows attacks to pass undetected. The color gradient highlights the concentration of predictions along the correct diagonal. Both error types remain below 2% of total samples, demonstrating a balanced classification capability. Overall, the framework achieves a raw classification accuracy of 97.8%.

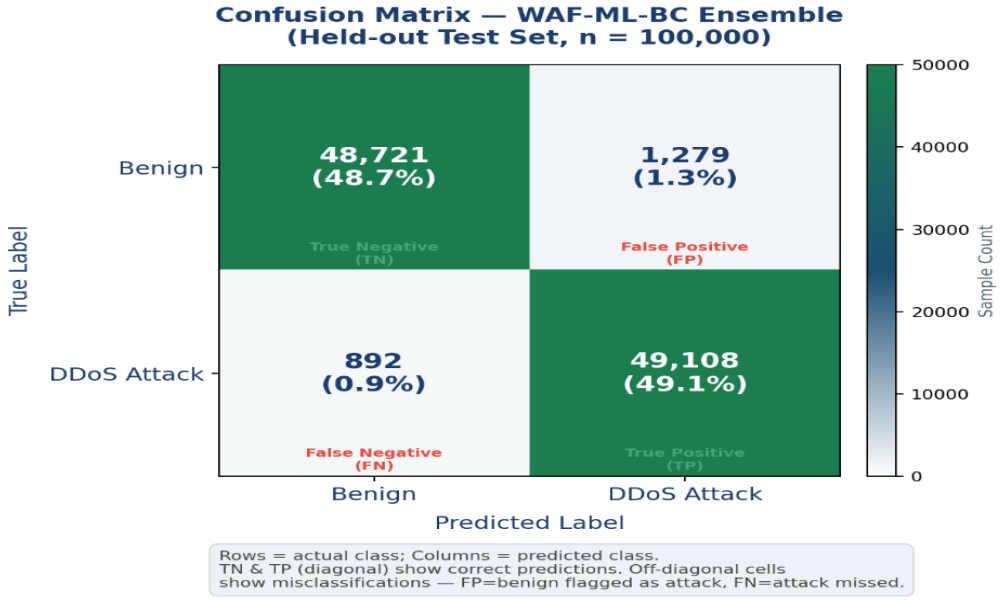

Figure 2: Confusion matrix heat map

ROC Curve (Multiple Models)

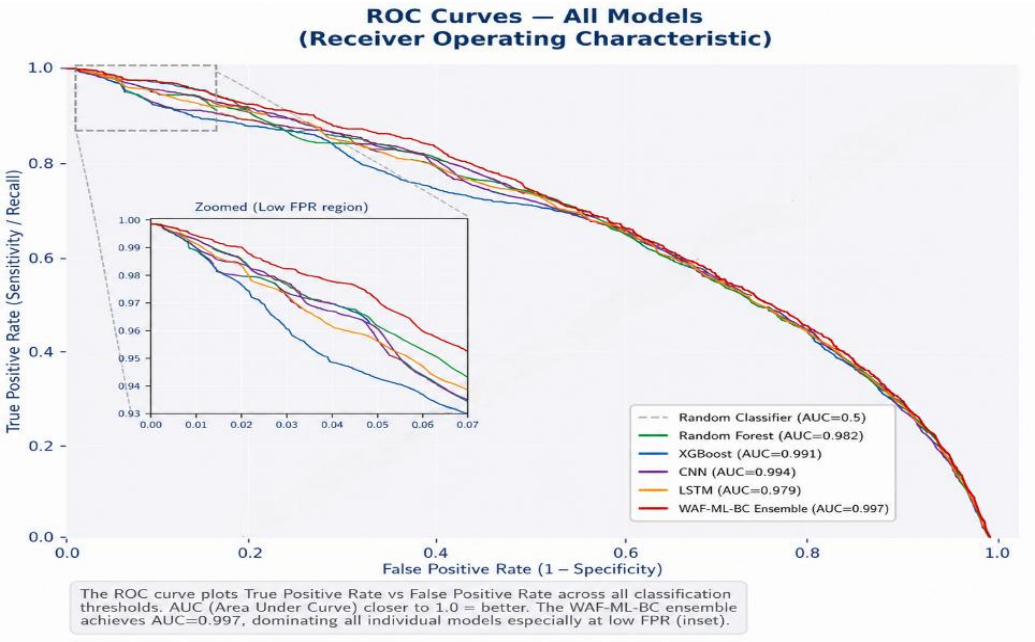

Figure 3: ROC curve

The ROC curves for each of the selected models can be seen in figure 3, each graph shows the balance between true positive and false positive rate over the whole range of decisions threshold. The

performance obtained by proposed model, WAF-ML-BC ensemble, is the best as shown by the AUC value of 0.997 compared to the Random Forest (0.982), XGBoost (0.991), CNN (0.994), and LSTM (0.979) models. The baseline random classifier is illustrated (AUC = 0.5). The small box on the graph indicates the small region of low false positive rates (0 to 0.07) most relevant for the WAF usage; in this region the classification is again the best.

Detection Performance Across Different Attack Types

In figure 4 displays a stacked bar chart of the detection performance of WAF-ML-BC over different types of DDoS attacks. Each bar represents the percentage of successfully detected attacks and the false negative percentage over a particular type of attack. Volumetric attacks like ICMP Flood (99.7%) and UDP Flood (99.6%) achieved high accuracy rates owing to their particular characteristics of traffic. Other attacks like HTTP Flood (98.9%), DNS Amplification (98.1%) and SYN Flood (99.3%) also performed well with 90% above accuracies. Slowloris was the one with the lowest detection rate (96.8%) and its slow rate behavior mimicked that of normal traffic and thus it required time sequence analysis with the LSTM module.

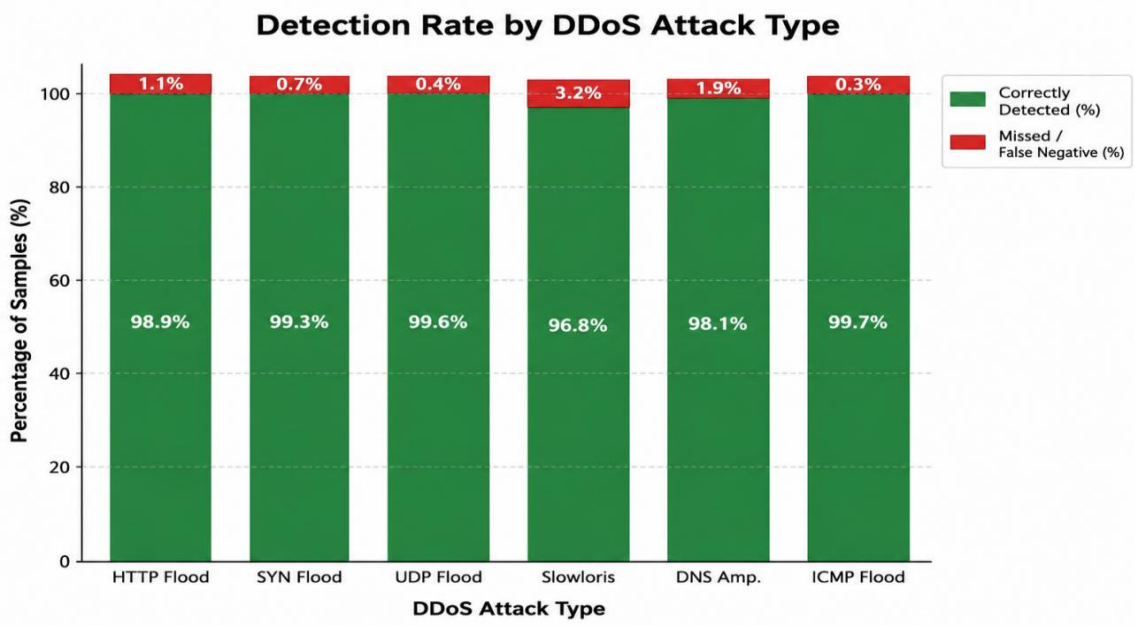

Figure 4: Detection performance across different attack types

Latency vs Traffic Load

In figure 5 analyze how well the WAF-ML-BC framework scales. End to end inference latency values is plotted when traffic ranges from 10,000 to 2,000,000 request per second. P50, p95 and p99 are all shown to give an overall sense of the distribution. A production SLA threshold is set at 50ms. From moderate to high traffic volumes the latency stays relatively stable with p50 and p95 are very close to the SLA value until approximately one million requests per second, then the latencies grow drastically, requiring scaling the cluster horizontally with more inference nodes.

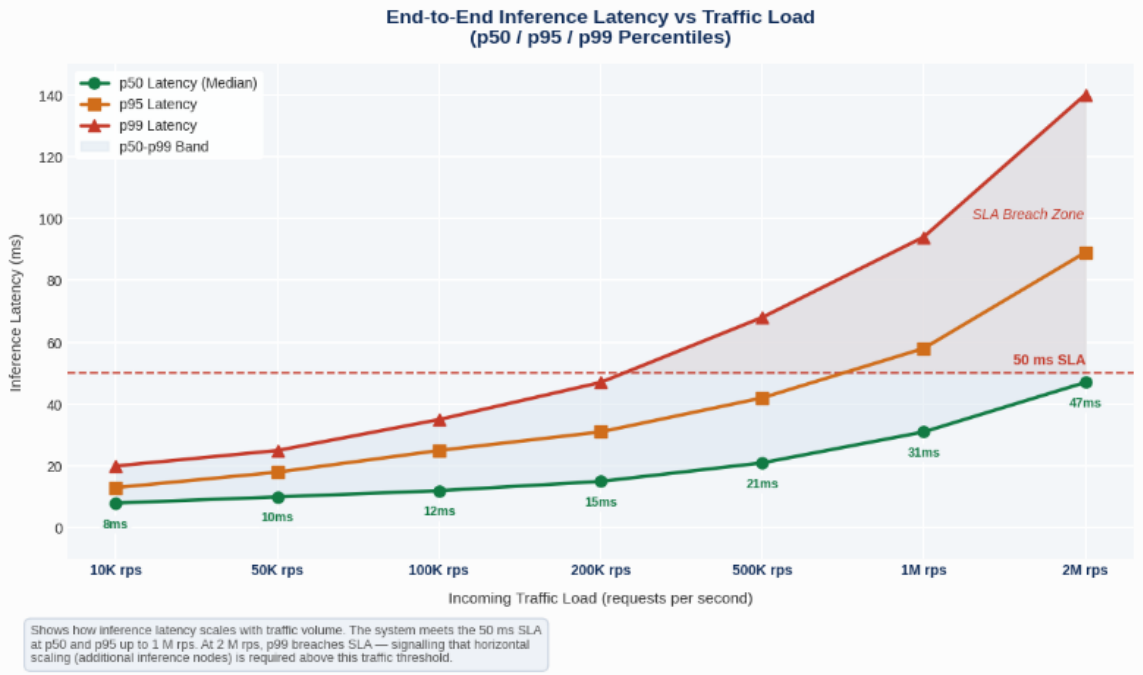

Figure 5: Latency vs Traffic load

Limitations

In spite of the promising performance, WAF-ML-BC exhibits some weakness: Blockchain throughput might pose a problem when dealing with large volume DDoS attack; Evaluation datasets might not be comprehensive in representing new types of attack thus limit the generalizability; Blockchain still faces adversarial attack, possible privacy implication with immutable logging, difficulties in large scale multi-tenancy deployment.

7 Conclusion

Proposed WAF-ML-BC (Web Application Firewall using Machine learning with Blockchain enabled immutable audit logs and federated threat intelligence) is a cloud native WAF framework designed with a view to overcome certain limitations found in current state-of-art rule-based WAF systems, specifically: reliance on pre-defined static signature rules for attack detection, lack of dynamic and adaptable protection mechanisms, and the absence of verifiable and tamper-proof security audit trail. WAF-ML-BC has employed an ensemble of Random Forest, XGBoost, CNN and LSTM models along with Logistic Regression as a meta-learner to provide efficient detection against volume, protocol based and application layer attacks. In addition to this, immutable audit logging and distributed threat intelligence sharing has been facilitated using blockchain technology. This paper evaluated the performance of the WAF-ML-BC system using different benchmark datasets (CIC-DDoS2019, CICIDS2018, HTTP CSIC 2010, OWASP Benchmark) and anonymized real world WAF logs. The performance of the proposed WAF-ML-BC in terms of overall accuracy (99.1%), precision (98.7%), recall (97.8%) and F1-score (98.9%) significantly outperforms other individual machine learning models and traditional rule based WAFs. The proposed solution achieves a substantial reduction in False Positive Rate from 18.3% to 3.1% which indicates an improved level of detection accuracy and reduced impact on legitimate users. Furthermore, it maintains its operation under real time traffic constraints

withmedian, 95th percentile and 99th percentile latencies of 12ms, 31ms and 47ms respectively which are within the defined target service level of 50ms. The blockchain subsystem (Hyperledger Fabric) sustains an average of 2840 transactions/ sec to support scalable and tamper-proof transaction logging and this does not have any detrimental impact on the request processing performance. Although the proposed system outperforms the state-of-the-art, there are few open challenges which needs to be addressed. These are: scalability issues of blockchain in case of excessive attack volume, dataset aging and its impact, robustness against adversarial machine learning attacks, and privacy concerns raised due to immutable logs. This study envisions future research in federated learning along with differential privacy, adversarially robust machine learning, privacy preserving blockchain systems such as zero knowledge proofs, light weight edge deployment, zero trust architecture-based integration and post-quantum cryptography.

Declaration

Funding

No financial support was received by the authors for this research.

Competing Interests

The authors declare that they have no competing financial or non-financial interests.

Ethical Approval

This study did not involve humans or animals; therefore, ethical approval was required.

Consent for Publication

The authors give consent for their publication.

Data Availability

The manuscript contains all data. However, more data are available upon request from the authors.

Materials Availability

Materials used in this research are available with corresponding author and given on request.

Code Availability

The code is available from the corresponding author and can be provided upon request.

Author Contribution

Asha V drafted and conceived this systematic review. The research process was supervised by Kanaga Suba Raja S, who assisted with data analysis and interpretation and provided a critical review of the manuscript. All the novelists studied the outcomes and polished the final version of the manuscript.

Acknowledgements

I would like to express my sincere gratitude to my research supervisor, Kanaga Suba Raja S, for his assistance with this research project. My husband and son stood behind me every time; therefore, I am grateful to them. I also thank my parents for their help and love that brought me this far. We appreciate the input from the SRM Institute of Science and Technology, Tiruchirappalli, for providing all the facilities and atmosphere necessary for conducting research.

Conflicts of Interest

The authors declare that there are no conflicts of interest and that the information presented is unbiased and free from any influence that could arise from potential conflicts.

Disclosure of AI Usage

In this study, the following AI tools were used: ChatGPT, Paperpal, SciSpace, Perplexity, Quiltbot, Canva, and MS Excel. Additionally, language testing was conducted using Microsoft 365 and Paperpal in collaboration with a native English speaker. The authors have reviewed and edited the AI-generated content and take full responsibility for the accuracy of the manuscript.

References

[1] Alaoui, R. L., & Nfaoui, E. H. (2022). Deep learning for vulnerability and attack detection on web applications: A systematic literature review. *Future Internet*, *14*(4), 118. https://doi.org/10.3390/fi14040118

[2] Albalawi, M., Aloufi, R., Alamrani, N., Albalawi, N., Aljaedi, A., & Alharbi, A. R. (2022). Website defacement detection and monitoring methods: A review. *Electronics*, *11*(21), 3573. https://doi.org/10.3390/electronics11213573

[3] Appelt, D., Nguyen, C. D., Panichella, A., & Briand, L. C. (2018). A machine-learning-driven evolutionary approach for testing web application firewalls. *IEEE Transactions on Reliability*, *67*(3), 733-757. https://doi.org/10.1109/TR.2018.2805763

[4] Athief, R., Kishore, N., & Paranthaman, R. N. (2024, May). Web application firewall using machine learning. In *2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)* (pp. 1-7). IEEE. https://doi.org/10.1109/ACCAI61061.2024.10602105

[5] Betarte, G., Pardo, Á., & Martínez, R. (2018, December). Web application attacks detection using machine learning techniques. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 1065-1072). IEEE. https://doi.org/10.1109/ICMLA.2018.00174

[6] Bisht, P., & Rauthan, M. S. (2023). Machine Learning and Natural Language Processing Based Web Application Firewall for Mitigating Cyber Attacks in Cloud. *International Journal of Scientific Research in Computer Science and Engineering*, *11*(3), 1-15. https://doi.org/10.26438/ijsrcse/v11i3.115

[7] Cho Do Xuan, N. N., & Dinh, H. N. (2020). An adaptive anomaly request detection framework based on dynamic web application profiles. *International Journal of Electrical and Computer Engineering (IJECE)*, *10*(5), 5335-5346. https://doi.org/10.11591/ijece.v10i5.pp5335-5346

[8] Demetrio, L., Valenza, A., Costa, G., & Lagorio, G. (2020, March). Waf-a-mole: evading web application firewalls through adversarial machine learning. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing* (pp. 1745-1752). https://doi.org/10.1145/3341105.3373962

[9] Joshi, A., & Geetha, V. (2014, July). SQL Injection detection using machine learning. In *2014 international conference on control, instrumentation, communication and computational technologies (ICCICCT)* (pp. 1111-1115). IEEE. https://doi.org/10.1109/ICCICCT.2014.6993127

[10] Kumar, H. (2023, October). Securing web application using web application firewall (waf) and machine learning. In *2023 First International Conference on Advances in Electrical, Electronics and Computational Intelligence (ICAEECI)* (pp. 1-8). IEEE. https://doi.org/10.1109/ICAEECI58247.2023.10370872

[11] Muttaqin, R. Z., & Sudiana, D. (2024). Design of Realtime Web Application Firewall on Deep Learning-Based to Improve Web Application Security. *Jurnal Penelitian Pendidikan IPA*, *10*(12), 11121-11129. https://doi.org/10.29303/jppipa.v10i12.8346

[12] Nilă, C., Apostol, I., & Patriciu, V. (2020, June). Machine learning approach to quick incident response. In *2020 13th International Conference on Communications (COMM)* (pp. 291-296). IEEE. https://doi.org/10.1109/COMM48946.2020.9141989

[13] Pan, Y., Sun, F., Teng, Z., White, J., Schmidt, D. C., Staples, J., & Krause, L. (2019). Detecting web attacks with end-to-end deep learning. *Journal of Internet Services and Applications*, *10*(1), 1-22. https://doi.org/10.1186/s13174-019-0115-x

[14] Román-Gallego, J. Á., Pérez-Delgado, M. L., & Viñuela, M. L. (2023, June). Development of Web Application Firewall Based on Artificial Intelligence. In *International Conference on Disruptive Technologies, Tech Ethics and Artificial Intelligence* (pp. 18-27). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-38344-1_3

[15] Shar, L. K., Briand, L. C., & Tan, H. B. K. (2014). Web application vulnerability prediction using hybrid program analysis and machine learning. *IEEE Transactions on dependable and secure computing*, *12*(6), 688-707. https://doi.org/10.1109/TDSC.2014.2373377

[16] Shivani, A., Sreelakshmi, R., Bhavani, A., Ramdas, J., & Jithender, A. (2025). Detection and Mitigation of Web Attacks with End-To-End Deep Learning. *Journal of Computer Allied Intelligence (JCAI, ISSN: 2584-2676)*, *3*(3), 56-80. https://doi.org/10.69996/jcai.2025017

[17] Tariq, M. U. (2025). Harnessing Generative AI for Enhanced Web Application Security. In *Generative AI for Web Engineering Models* (pp. 161-192). IGI Global Scientific Publishing. https://doi.org/10.4018/979-8-3693-3703-5.ch008

[18] Trinh, C. V., Le, T. T., Le-Nguyen, M. K., Le, D. T., Nguyen, V. H., & Nguyen-An, K. (2023, November). An Efficient Machine Learning-Based Web Application Firewall with Deep Automated Pattern Categorization. In *International Conference on Future Data and Security Engineering* (pp. 212-225). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-99-8296-7_15

[19] Valenza, A., Demetrio, L., Costa, G., & Lagorio, G. (2020). WAF-A-MoLE: An adversarial tool for assessing ML-based WAFs. *SoftwareX*, *11*, 100367. https://doi.org/10.1145/3341105.3373962

[20] Zaki, A., & Mohammed, S. (2024). Artificial Intelligence for Web Application Firewall (WAF): A Comprehensive Review. *International Research Journal of Innovations in Engineering and Technology*, *8*(11), 219-224. https://doi.org/10.47001/IRJIET/2024.811027

Authors Biography

V. Asha is a Ph.D. Research scholar in the Department of Computer Science and Engineering at SRM Institute of Science and Technology, Tiruchirappalli, Tamil Nadu, India. She received the B.Tech. Degree in Information Technology and the M.E. degree in Computer Science and Engineering from Anna University, Chennai, in 2012 and 2015, respectively. She has over 10 years of teaching experience and has authored 10 research articles in international journals, 5 papers in international conferences and 4-book chapter. Her current research focuses on computer networks, network security, and intelligent firewall systems. Her broader research interests include network security, image and data processing, biometrics, medical image analysis, and pattern recognition.

S. Kanaga Suba Raja is an Associate Dean and Professor at the School of Computing, SRM Institute of Science and Technology, Tiruchirappalli, with 20 years of teaching and research experience. He earned his Ph.D. in Computer Science and Engineering in 2013 and has successfully guided numerous graduate and undergraduate students. Under his supervision, nine scholars have completed their Ph.D. degrees, and six are currently pursuing their research at Anna University, Chennai. He has secured research funding of INR 17,00,000 from the MSME Idea Hackathon 1.0 and has published 91 papers in international journals and conferences, including 20 SCI or WoS –indexed articles and 78 Scopus-indexed documents, amassing 453 Scopus citations (h-index: 13) and 1013 Google Scholar citations (h-index: 16). He has organized multiple international and national conferences, participated in 40 national workshops, and contributed to innovation through two granted patents, five published patents, six book chapters, and one online book. His research interests include Wireless Body Area Networks, Data Science, and Machine Learning, and he possesses expertise in computer science, administration, leadership, and research methodologies.