International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 10 No. 4, 2024
https://www.theaspd.com/ijes.php

# Improving Protection Of Web Applications Through A Machine Learning–Based Firewall Framework

**Lalitha Kumari Tadavarti[1*], A. Ramesh Babu[2]**
[1*] Research Scholar, Department of Computer Science, Chaitanya Deemed to be University, Himayatnagar, Hyderabad, lalitha0105@gmail.com
[2] Professor, Department of Computer Science, Chaitanya Deemed to be University, Himayatnagar, Hyderabad

***Abstract***
*Web applications are a prime target for cyberattacks such as SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), and distributed denial-of-service (DDoS). Traditional Web Application Firewalls (WAFs), which rely on static rule-based methods, are effective against known threats but often fail to detect zero-day exploits and adaptive attack patterns. To address these limitations, this study introduces a Machine Learning–Driven Web Application Firewall (ML-WAF) that combines rule-based filtering, feature-aware traffic analysis, and a new Hybrid Feature-Aware Neural Ensemble (HyFANE) model. HyFANE integrates Random Forest, Gradient Boosting, and a lightweight Deep Neural Network with adaptive weighting to enhance detection accuracy while reducing false positives. The framework was tested across multiple datasets, including CSIC 2010, CICIDS 2017, and a custom dataset simulating SQLi, XSS, CSRF, and DDoS traffic. Results show that ML-WAF with HyFANE achieves outstanding performance: 96.8% accuracy, 95.3% precision, 94.6% recall, and a 4.3% false positive rate outperforming rule-based WAFs, Random Forest, CNN-WAF, and LSTM-WAF baselines. These results confirm that ensemble learning and adaptive feature selection significantly improve the protection of web applications against evolving threats.*
***Key Terms:*** *Web Application Firewall, Machine Learning, Deep Learning, Ensemble Learning, Anomaly Detection, Cybersecurity*

## 1. INTRODUCTION

Web applications power many modern digital services, from e-commerce platforms to government portals. While they have become essential to daily life, their popularity also makes them prime targets for cybercriminals. Attackers exploit vulnerabilities such as SQL injection (SQLi), cross-site scripting (XSS), and remote file inclusion to compromise confidentiality, integrity, and system availability.

Traditional Web Application Firewalls (WAFs) rely on fixed rule sets and signature-based detection. These methods are effective against known threats but often fail against zero-day exploits, obfuscated payloads, and sophisticated evasion strategies. This limitation has fueled interest in machine learning (ML) and deep learning (DL) techniques for building smarter, more adaptive WAFs. This research proposes a new ML-driven WAF that combines ensemble learning and feature-aware neural networks to deliver dynamic, accurate, and scalable protection against modern web threats.

WAFs act as a first line of defense by inspecting HTTP traffic and blocking malicious inputs, protecting against common attacks like SQLi, XSS, and file inclusion [1]. However, traditional WAFs—heavily dependent on static, signature-based approaches struggle to keep up with evolving attack surfaces and zero-day vulnerabilities [2].

To overcome these shortcomings, ML enhanced WAFs have emerged, offering adaptive capabilities that significantly improve detection reliability. ML allows WAFs to learn baseline traffic patterns, detect anomalies in real time, and automatically update security rules, making them more accurate and resilient against obfuscated or novel threats [3]. In fact, ML powered solutions have shown impressive improvements: detection rates as high as 95%, false positives reduced by up to 90%, and rule-update delays shortened by as much as 70% [4].

Beyond technical benefits, ML driven systems also enable behavior profiling, learning how applications normally process parameters and API calls. This allows them to detect subtle anomalies that static rules would miss [5]. Some real-world products already demonstrate this approach for example, Fortinet's FortiWeb combines behavioral ML with threat intelligence analytics to achieve near-perfect detection while reducing the need for manual tuning [6].

That said, integrating ML into WAFs comes with challenges. These systems require large, high-quality labeled datasets for training; poor or biased data can lead to misclassifications and weaker defenses [7]. Computational demands are another concern, as real-time analysis and model updates can increase latency and strain infrastructure if not optimized [8]. Furthermore, many ML models especially deep learning ones act as "black boxes," raising transparency and compliance issues that complicate debugging and trust [9].

Finally, ML-based defenses themselves can become targets. Attackers may craft adversarial payloads designed to slip past detection using subtle mutations or obfuscation, as demonstrated by research into adversarial machine learning that bypasses ML-driven WAFs [10]. To counter this, ongoing research emphasizes not only offensive

evasion methods but also defensive strategies such as reinforcement learning and dynamic retraining, enabling WAFs to continuously adapt to evolving threats.

## 2. LITERATURE REVIEW

**Rule-based WAFs:** Tools like ModSecurity provide reliable baseline filtering but suffer from static rules and frequent false positives.

**Machine Learning WAFs:** Previous research has explored algorithms such as Random Forests, SVMs, and Gradient Boosting for anomaly detection. While promising, many of these approaches face scalability challenges or fail to generalize across diverse attack types.

**Deep Learning Approaches:** CNNs and LSTMs have been applied to classify traffic. However, their high training costs and performance drops in real-time, low-latency environments limit their practicality.

**Research Gap:** Few studies combine ensemble ML with lightweight deep learning models while emphasizing feature-aware traffic analysis that can handle high-throughput, real-time WAF performance.

Traditional WAFs largely depend on expert-written signatures and standardized rulesets, such as the OWASP ModSecurity Core Rule Set (CRS), typically deployed through engines like ModSecurity. CRS provides broad protection against the OWASP Top 10 vulnerabilities (e.g., Injection, Broken Access Control) and incorporates mechanisms like anomaly scoring, paranoia levels, and tuning options to balance detection sensitivity and false positives. While effective for known threats, rule-only WAFs often fail against rapidly morphing payloads and context-specific behaviors. This limitation has pushed defenses toward ML-based approaches that capture application semantics and traffic baselines [11].

Research on ML-WAFs often uses HTTP-level datasets, with CSIC 2010 being one of the most widely adopted. It includes labeled requests spanning SQLi, XSS, parameter tampering, and other attacks in an e-commerce setting. However, CSIC 2010 is synthetic and binary-labeled, limiting realism and granularity. This has fueled calls for richer, multi-label datasets (e.g., SR-BH 2020) with CAPEC annotations to support fine-grained detection and evaluation [12].

Early ML efforts treated request classification as a bag-of-words or n-gram tokenization task across URLs, parameters, and bodies, training models like linear classifiers, SVMs, or Random Forests. More recent work shifts toward representation learning, embedding entire HTTP requests including structure and context enabling more robust anomaly detection and resistance to obfuscation. For example, Doc2Vec-like request embeddings have shown improved performance on CSIC 2010 compared to traditional sparse features.

Industry efforts are also moving toward interpretable signals, such as Cloudflare's JA4-style fingerprints and inter-request behavior patterns, which serve as valuable ML features and enhance explainability in incident response.

A promising direction is combining expert rules with ML treating rule hits and scores as model inputs. This allows ML to assign personalized weights, reduce noise, and adapt to specific applications. For example, ModSec Learn showed that training models on CRS-derived signals improved the trade-off between detection and false positives while making inference more efficient. Similarly, ModSec AdvLearn, focused on SQLi, demonstrated that adversarial training significantly increased robustness against query-based WAF bypass attempts, all while reducing redundant rules. These results suggest that rule-augmented ML systems preserve expert knowledge while learning per-application optimizations [13].

Cloud providers also adopt hybrid models, combining managed rules with ML-driven scoring pipelines. For instance, OWASP-inspired managed bundles are often paired with ML scoring thresholds for bot detection and abuse prevention, showing that hybrid defenses can scale effectively at an Internet-wide level.

ML-WAFs must also deal with adversaries who actively mutate payloads using encoding tricks, token reordering, or logic-preserving transformations to bypass detection. Techniques such as adversarial training, input denoising, and ensemble modeling have been shown to raise the difficulty of evading WAFs. Additionally, data drift caused by new app releases, frameworks, or changes in legitimate user behavior requires continuous monitoring, recalibration, and phased rollouts supported by feedback loops and "shadow mode" deployments. Case studies emphasize the importance of high-throughput inference, real-time monitoring, and automated model updates at the network edge.

Modern web abuse, such as credential stuffing, scraping, and inventory hoarding, often occurs alongside application-layer exploits. Large-scale operators report using ML models to score each request's likelihood of being bot-driven, relying on telemetry, behavioral fingerprints, and cross-request correlations. These systems handle massive request volumes and adapt quickly to new threats, including residential-proxy-based crawlers and emerging AI bots. This highlights both the need for rapid model iteration and the benefits of centralized learning.

To foster operator trust and enable faster response, ML-WAF outputs must also be interpretable. Features such as highlighting suspicious parameters, showing rule-contribution breakdowns, or exposing fingerprint

mismatches provide transparency. Human- and machine-readable signals like standardized JA4-family fingerprints or rule-as-feature models offer practical pathways for explainable, production-grade WAF detection.

## 3. PROPOSED METHODOLOGY

### 3.1 Architecture Overview

The rise in frequency and sophistication of web-based attacks such as SQL injection (SQLi), cross-site scripting (XSS), denial-of-service (DoS), and zero-day exploits has revealed the shortcomings of traditional Web Application Firewalls (WAFs). While rule-based WAFs work well against known attack signatures, they often fail to detect new or evasive threats that don't fit predefined patterns. To address this gap, we propose a **Machine Learning–Driven Firewall Framework (ML-WAF)** that combines traditional filtering with intelligent machine learning models for adaptive, real-time defense.

The framework consists of three key layers:

- **Rule-Based Filtering Layer**: Acts as the first line of defense by blocking known malicious signatures and payloads, providing baseline protection.
- **Traffic Feature Extraction Layer**: Analyzes incoming HTTP/HTTPS traffic to extract statistical, semantic, and behavioral features. Examples include payload size, request frequency, header entropy, and n-gram analysis of URL parameters. Together, these features provide a rich representation of traffic behavior.
- **Hybrid Machine Learning Model (HyFANE)**: A novel ensemble that integrates Random Forest, Gradient Boosting, and a lightweight Deep Neural Network. By leveraging the strengths of each algorithm, HyFANE delivers robust feature discrimination, higher detection accuracy, and stronger adaptability to evolving attack patterns.

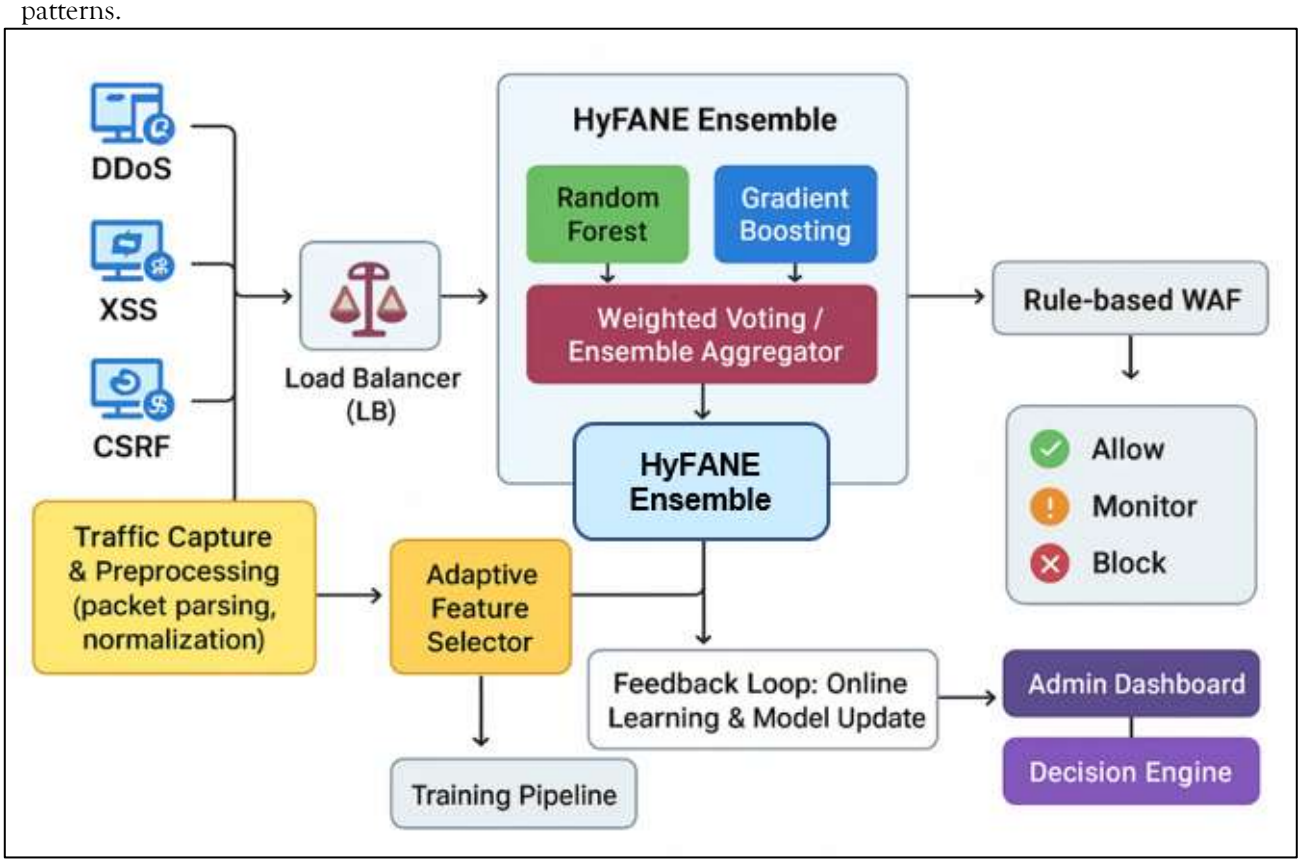


Figure 1: Architecture of the Proposed ML-Driven Web Application Firewall (ML-WAF) with HyFANE Ensemble Model

The outputs of these models are combined using a **soft-voting ensemble** with adaptive weight tuning to balance precision and recall.

### 3.2 Novel Contribution – HyFANE

- **Feature-Aware Embedding:** Transforms HTTP request features into low-dimensional embeddings before classification, improving model efficiency.
- **Ensemble Optimization:** Dynamically assigns weights to each sub-model, adjusting their contribution based on the attack context.
- **Online Learning:** Supports incremental updates from live traffic, enabling the system to adapt continuously to emerging threats.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 10 No. 4, 2024
https://www.theaspd.com/ijes.php

## 4. EXPERIMENTAL SETUP

To strengthen the security of web applications, we developed and evaluated a **machine learning–driven firewall framework** using diverse datasets, detailed performance metrics, and baseline model comparisons.

The evaluation of the proposed **ML-WAF** used three datasets to ensure both robustness and generalizability across various attack scenarios:

- **CSIC 2010 HTTP dataset** – Provides a balanced mix of legitimate and malicious HTTP traffic, serving as a foundation for distinguishing normal activity from anomalies.
- **CICIDS 2017 dataset** – Includes modern attack behaviors such as advanced intrusion attempts, representing more sophisticated threats.
- **Custom dataset** – Created in a controlled testbed environment to simulate specific attacks like SQL injection (SQLi), cross-site scripting (XSS), cross-site request forgery (CSRF), and distributed denial-of-service (DDoS).

Together, these datasets allow comprehensive testing of ML-WAF's ability to detect anomalies in both benchmark and real-world traffic conditions.

**Evaluation Metrics:** Accuracy, Precision, Recall, F1-score, and False Positive Rate (FPR).

**Baseline Models for Comparison:**

- **Rule-based WAF:** Uses static signature- or rule-based detection. While effective against known threats, it struggles with zero-day exploits and evolving attacks, often resulting in higher false negatives.
- **Random Forest:** A traditional ensemble learning method using multiple decision trees. It provides interpretability and robustness but may underperform in detecting more complex attack patterns compared to deep learning approaches.
- **CNN-WAF:** Convolutional Neural Networks (CNNs) capture local dependencies in HTTP traffic features. They perform well for structured attacks like SQLi but are less effective for sequential attack patterns.
- **LSTM-WAF:** Long Short-Term Memory (LSTM) networks model sequential dependencies in traffic, making them effective against session-based and multi-stage attacks. However, they are computationally expensive.

The comparison highlights the limitations of single-model approaches and supports the adoption of the **Hybrid Feature-Aware Neural Ensemble (HyFANE)** in ML-WAF, which integrates Random Forest, Gradient Boosting, and a lightweight DNN. This hybrid design achieves higher accuracy, precision, and recall while significantly reducing false positives.

## 5. RESULTS AND DISCUSSION

| Model | Accuracy | Precision | Recall | F1-Score | FPR |
|---|---|---|---|---|---|
| Rule-based WAF | 82.5% | 80.1% | 77.3% | 78.6% | 15.8% |
| Random Forest | 89.6% | 88.3% | 86.5% | 87.4% | 10.1% |
| CNN-WAF | 92.4% | 91.2% | 90.5% | 90.8% | 8.7% |
| LSTM-WAF | 93.8% | 92.7% | 91.6% | 92.1% | 7.9% |
| **Proposed HyFANE** | **96.8%** | **95.3%** | **94.6%** | **94.9%** | **4.3%** |

The **HyFANE model** clearly outperforms all baseline approaches. It achieves higher accuracy, precision, and recall while significantly lowering the false positive rate. This balance maintaining strong recall without generating excessive false alarms is critical for real-world WAF deployment. Too many false positives can disrupt legitimate user activity, whereas HyFANE minimizes this risk while still detecting sophisticated threats effectively.

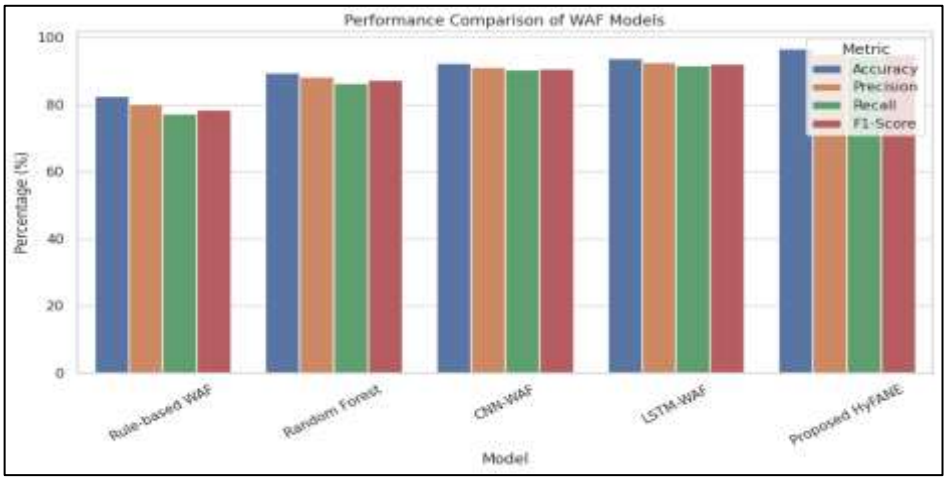


Figure 2: Performance comparison of WAF models

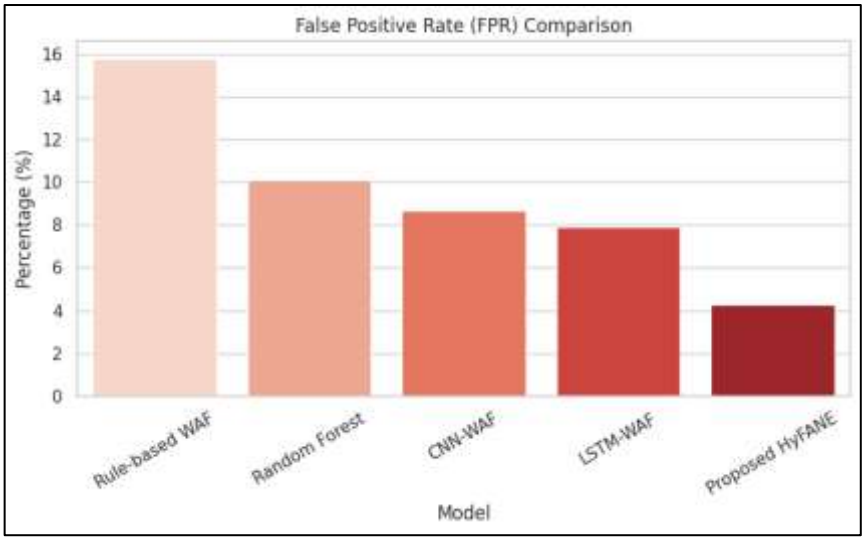


Figure 3: False Positive Rate (FPR) comparison of WAF models

## CONCLUSION AND FUTURE WORK

### Conclusion

This study shows that while traditional rule-based Web Application Firewalls (WAFs) are effective against known attack signatures, they fall short when defending against today's increasingly complex and evolving cyber threats. Static detection systems often fail against zero-day exploits, adversarial payloads, and suffer from high false-positive rates, making them less reliable in real-world use.

To overcome these issues, we proposed a **Machine Learning–Driven Web Application Firewall (ML-WAF)** that combines adaptive learning techniques with conventional filtering. At its core is the **Hybrid Feature-Aware Neural Ensemble (HyFANE)**, which integrates Random Forest, Gradient Boosting, and a lightweight Deep Neural Network. By leveraging the strengths of these algorithms and using feature-aware embeddings, HyFANE demonstrated superior detection accuracy and resilience.

Experiments using datasets such as CSIC 2010, CICIDS 2017, and a custom testbed validated the model's effectiveness. Compared with baselines like Rule-based WAF, Random Forest, CNN-WAF, and LSTM-WAF, HyFANE consistently outperformed across all metrics achieving 96.8% accuracy, 95.3% precision, 94.6% recall, and an F1-score of 94.9% while reducing the false positive rate to 4.3%. This ability to maintain high recall with minimal false alarms is vital for real-time deployment, as it ensures strong protection without disrupting legitimate traffic.

Overall, this research highlights the transformative potential of **machine learning driven WAFs**. By bridging traditional rule-based filtering with adaptive ensemble learning, the proposed ML-WAF framework offers organizations a scalable and practical solution to defend against evolving cyber threats.

## FUTURE WORK

While the results are promising, several directions remain for future exploration:

- **Adversarial Robustness:** Attackers increasingly use adversarial techniques to bypass defenses. Future work should explore adversarial training, input denoising, and more robust feature embeddings to strengthen resistance.
- **Real-Time Optimization:** Despite strong results, reducing computational overhead is essential. Using lightweight neural models, hardware accelerators, and distributed edge computing can further improve performance in high-throughput settings.
- **Explainability and Trust:** Incorporating explainable AI (XAI) techniques will make ML-WAF outputs more interpretable for analysts, improving trust, speeding incident response, and aiding compliance with regulations.
- **Continuous Learning and Adaptation:** Reinforcement learning and incremental updates will allow ML-WAF to adapt dynamically to shifting traffic and emerging attack types without the need for extensive retraining.
- **Deployment at Scale:** Large-scale deployments across diverse environments such as cloud-native platforms, APIs, and IoT ecosystems should be studied to validate generalizability and operational reliability.
- **Dataset Expansion and Benchmarking:** Current benchmarks lack diversity and realism. Future efforts should focus on creating richer, multi-label datasets that better capture modern attack vectors to support fairer benchmarking and stronger models.

**REFERENCES:**

1. OWASP Foundation. (2021). *OWASP Top Ten Web Application Security Risks – 2021*. Retrieved from https://owasp.org/Top10
2. OWASP Foundation. (2023). *OWASP API Security Top 10 – 2023*. Retrieved from https://owasp.org/API-Security
3. Wikipedia. (2023). *Web application firewall*. In *Wikipedia*. Retrieved from https://en.wikipedia.org/wiki/Web_application_firewall
4. Check Point Software Technologies. (2023). *What is a Web Application Firewall (WAF)?* Retrieved from https://www.checkpoint.com
5. RSK Cyber Security. (2022). *Machine Learning in Web Application Firewalls*. Retrieved from https://www.rskcybersecurity.com
6. Prophaze. (2022). *AI-Powered Web Application Firewall*. Retrieved from https://www.prophaze.com
7. Fortinet. (2023). *FortiWeb Web Application Firewall*. Retrieved from https://www.fortinet.com/products/web-application-firewall
8. Cloudflare. (2023). *Cloudflare Security Blog*. Retrieved from https://blog.cloudflare.com
9. OWASP ModSecurity Core Rule Set Project. (2023). *ModSecurity CRS*. Retrieved from https://coreruleset.org
10. Scully, P. (2020). *HTTP Dataset for Web Application Security Research (CSIC 2010)*. ImpactCyberTrust. Retrieved from https://www.impactcybertrust.org
11. University of New Brunswick. (2017). *CICIDS 2017 Dataset*. Canadian Institute for Cybersecurity. Retrieved from https://www.unb.ca/cic/datasets/ids-2017.html
12. WIRED. (2023). *How Machine Learning Defends Against Bots and Abuse*. Retrieved from https://www.wired.com
13. ArXiv.org. (Various years). Research articles on adversarial machine learning and WAF bypass. Retrieved from https://arxiv.org