WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Machine Learning in Web Application Security: A Literature Review

Mr. Shubham Patil[1], Dr. Pushpalata Aher[2], Dr. Mohd Muqeem[3], Dr. Pawan Bhaladhare[4]

[1] Research Scholar, School of Computer Science and Engineering, Sandip University, Nashik, India.
[2,3] Professor, School of Computer Science and Engineering, Sandip University, Nashik, India.
[4] HOD & Professor, School of Computer Science and Engineering, Sandip University, Nashik, India.

A B S T R A C T :

Web applications are critical targets of cyberattacks, and traditional security measures (signatures, static rules) struggle to keep pace with evolving threats. In recent years, artificial intelligence (AI) and machine learning (ML) techniques have been increasingly applied to harden web application defences. This review surveys current ML-based methods and tools used in web security – including anomaly-based intrusion detection, intelligent static/dynamic vulnerability scanning, and adaptive web filtering – drawing on recent academic studies and industry case reports. We analyse major challenges and gaps in these approaches, such as the scarcity of high-quality labelled data, adversarial manipulation of models, and the phenomenon of model drift (concept drift) in changing web environments. Finally, we outline future research directions, emphasizing the development of drift-aware, explainable, and robust ML systems. The goal is to provide a comprehensive theoretical overview of how AI/ML enhances web application security and what remains to be addressed.

Keywords: AI-Driven Cybersecurity, Automated Web Vulnerability Detection, Knowledge Gaps in Web Security, Web Application Security, Model Drift, Intrusion Detection Systems, Artificial Intelligence, Security Automation

1. Introduction

The security of web applications is of paramount importance in modern digital infrastructure. Organizations increasingly rely on web services for e-commerce, data analytics, cloud computing, and other enterprise functions, making them lucrative targets for attackers. Traditional defences (e.g. rule-based firewalls and static signature IDS) often fail to detect novel or obfuscated attacks. For example, signature-based systems have difficulty recognizing new SQL injection or cross-site scripting (XSS) payloads [1]. Recent literature highlights that the rapid evolution of attack techniques demands more proactive and adaptive security measures. In response, AI and ML have emerged as promising solutions: data-driven models can learn complex patterns of benign and malicious behaviour from large datasets, potentially identifying threats that static rules miss [2,7]. Studies have proposed ML-enabled web vulnerability detectors and intelligent intrusion detection systems, using features such as HTTP parameters, code structures, and network flows [7]. Despite this progress, the deployment of ML in web security brings new challenges. Notably, as Kirda et al. observe, the legitimate behaviour of web applications changes over time (e.g. via updates), leading to "web application concept drift" where models trained on past data may flag normal requests as malicious after an update and vice versa [3]. This review presents a structured examination of existing AI/ML methods in web application security, identifies their limitations (including concept drift and adversarial vulnerabilities), and suggests directions for future research.

2. Methodology

This review employed a structured literature survey approach to identify, analyse, and synthesize academic and industry research on the application of artificial intelligence (AI) and machine learning (ML) in web application security. The methodology followed these key steps:

2.1. Scope Definition

The review focused on peer-reviewed publications, technical reports, and white papers published between 2010 and 2024. Relevant topics included ML-based intrusion detection systems (IDS), vulnerability scanning, web application firewalls (WAFs), adversarial ML in security contexts, and concept drift in web environments.

2.2. Literature Search

Databases used:
IEEE Xplore, ACM Digital Library, SpringerLink, ScienceDirect, Google Scholar

Keywords included:

"Machine learning for web security", "AI in intrusion detection", "web application vulnerability detection", "adversarial machine learning", "concept drift web IDS", "ML-based WAF", and combinations thereof.

2.3. Inclusion and Exclusion Criteria

Included:

- Studies proposing or evaluating ML/AI methods for web application security
- Surveys and reviews addressing ML-driven cybersecurity approaches
- Papers discussing real-world deployments or datasets

Excluded:

- General cybersecurity articles with no web or ML focus
- Duplicate studies or those lacking technical detail or evaluation

2.4. Data Extraction and Analysis

Selected articles were analysed to extract:

- Targeted security task (e.g. intrusion detection, scanning)
- ML techniques used (e.g. supervised, unsupervised, DL)
- Datasets and evaluation methods
- Challenges addressed (e.g. adversarial robustness, drift)

Themes were categorized into application domains, methodological trends, and known gaps (e.g. data sparsity, model drift, explainability).

2.5. Synthesis and Reporting

Findings were synthesized into structured sections: current applications, challenges, and research directions. Key studies were highlighted to illustrate progress and open problems. Special focus was given to underexplored issues like model drift, adversarial ML, and data limitations, aligning with observed gaps in the literature.

3. Current Applications

AI/ML techniques have been applied across multiple facets of web application security. Major application areas include:

Table 1 - ML Applications in Web Security

Application Area	Description	ML Techniques	Examples/Notes
Intrusion & Anomaly Detection	**Detect suspicious web traffic or logs**	**Supervised (DT, NN), Unsupervised (Clustering, Autoencoders)**	**Deep learning for HTTP/session analysis**
Vulnerability Detection	**Detect vulnerable code and configurations**	**Static code features, DL (RNN, CNN)**	**SQL injection, XSS detection**
Web Filtering & Behavioral Analytics	**Detect obfuscated attacks, phishing, credential stuffing**	**ML-enhanced WAFs, user behavior analytics**	**Cloudflare's ML-powered WAF**
Hybrid Security Solutions	**Combine static/dynamic analysis with ML**	**Dynamic execution traces, fusion models**	**ML-guided fuzzing, hybrid scanners**

3.1. Intrusion and Anomaly Detection

ML-based intrusion detection systems (IDS) analyse web traffic and system logs to identify suspicious activity. Supervised classifiers (e.g. decision trees, neural networks) can be trained on labelled attack traffic, while unsupervised models (e.g. clustering, autoencoders) learn patterns of normal HTTP behaviour to detect anomalies. Given the increasing frequency and severity of network attacks, IDS are considered essential components of security architecture. For example, deep learning models have been explored to parse HTTP requests or session metadata, flagging deviations from learned norms. These approaches complement traditional IDS by adapting to new attack patterns and reducing reliance on manually written signatures. [2,6]

3.2. Vulnerability Detection and Scanning

ML is used to detect software vulnerabilities in web application code and configurations. Researchers have trained models on code features and attack payloads to identify issues such as SQL injection, XSS, buffer overflows, and insecure coding patterns [1]. Feature extraction methods include static code properties (e.g. abstract syntax tree patterns, control-flow graphs) and program embeddings; deep learning architectures (RNNs, CNNs) can then learn complex correlations in this data [7]. Such models are often more adaptive than rule-based scanners, automatically learning from examples of vulnerable and secure code. For instance, one survey notes that ML-driven vulnerability detectors have shown high accuracy in identifying SQL injection attacks while reducing false positives compared to static methods. [1]

3.3. Web Filtering and Behavioural Analytics

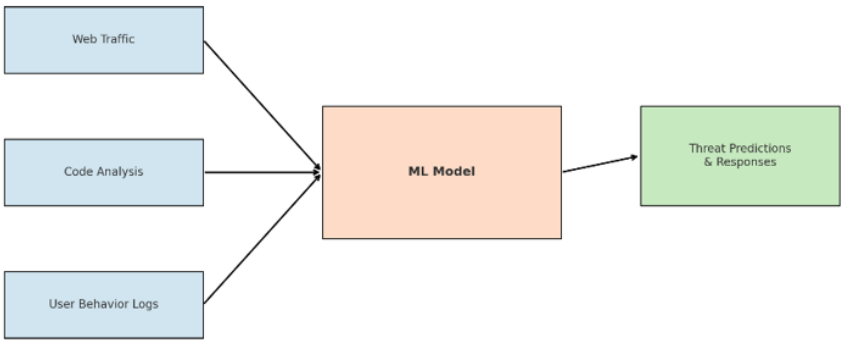
ML is incorporated into web application firewalls (WAFs) and content filters to catch obfuscated or zero-day attacks. For example, Cloudflare and other vendors use ML models to refine rule-based WAFs, learning to spot anomalous query patterns or malicious payloads in HTTP requests. ML also powers URL and email filtering to detect phishing sites or spam that target web applications. Additionally, ML-driven user behaviour analytics monitor login patterns and session data to detect credential stuffing or account hijacking. These models can process large volumes of events (e.g. from SIEM systems) to prioritize security alerts based on learned risk patterns. [2]

3.4. Hybrid Security Solutions

Some systems combine static and dynamic analysis with ML to improve coverage. For example, hybrid vulnerability scanners may use dynamic execution traces as features for ML classifiers, or fuse network- and host-based signals. Other emerging tools apply ML to orchestrate security tasks (e.g. automatically generating inputs for fuzzing guided by model predictions). In general, ML is used wherever large-scale, complex data makes manual analysis infeasible. The survey by Chughtai et al. notes that modern web security involves a mix of static, dynamic, and hybrid tools, and that AI techniques are increasingly used in intrusion detection to augment these methods. [5]

Together, these applications illustrate the broad potential of ML in web security. *Figure 1* (below) conceptually shows how ML models fit into a web security framework: they ingest data from web traffic, code analysis, or user logs, and output threat predictions that inform protective actions.

Figure 1: Conceptual ML Integration in Web Application Security



4. Challenges and Knowledge Gaps

Despite promising results, AI/ML solutions for web security face several critical challenges and gaps:

Table 3 - Key Challenges in ML for Web Security

Challenge	Description	Impact / Concern	Reference / Notes
Data Quality & Labeling	**Scarce, imbalanced, outdated, and sensitive datasets**	**Poor training, overfitting**	**Agoro et al., data sparsity**
Adversarial Attacks	**Model evasion, poisoning, reverse engineering**	**Security breaches, reduced trust**	**Paracha et al., adversarial ML risks**
Concept & Feature Drift	**Changing web app behavior causes model degradation**	**False positives, reduced accuracy**	**Kirda et al., model retraining need**
Explainability & Trust	**Black-box nature of complex ML models**	**Hard to debug, lack of accountability**	**Vourganas et al., interpretability gap**
Scalability & Real-Time	**High volume and latency requirements**	**Trade-off between speed and accuracy**	**Resource constraints**
Integration with Tools	**Compatibility with existing security systems**	**Adoption challenges**	**Workflow integration**

4.1. Data Quality and Labelling:

ML models require large, high-quality datasets. In security, obtaining labelled examples of novel attacks is difficult and costly. Many security datasets are imbalanced (very few attacks vs. abundant normal instances) or become outdated quickly. This data sparsity makes supervised training challenging. Agoro et al. point out that ML for vulnerability detection suffers from a "need for high-quality labelled datasets" and is prone to overfitting if data are limited [7]. Moreover, organizations may be reluctant to share sensitive security logs, further reducing data availability. The lack of standardized, up-to-date benchmark datasets for web attacks means that models are often tested on outdated or synthetic data, raising concerns about real-world effectiveness.

4.2. Adversarial Attacks on Models:

Security systems must assume adversarial conditions, but ML models themselves can be targeted. Adversaries may probe a publicly deployed model or reverse engineer its behaviour to craft inputs that evade detection. They may also attempt data poisoning (injecting malicious training samples) or backdoor attacks. Paracha et al. (2024) review this landscape and note that in critical systems (smart grids, networks, IDS, etc.), "mal-actors can reverse-engineer publicly available models, gaining insight into the algorithms" [8]. A single successful evasion or poisoning attack on a model could cascade into privacy leaks or integrity breaches. Paracha et al. further warn that common mitigations (data sanitization, adversarial training, differential privacy) have trade-offs: cleaning data may underfit the model, and differential privacy often comes at the cost of reduced performance [8]. In summary, ML-based security tools introduce a new attack surface of adversarial ML, which current defences only partially address.

4.3. Concept and Feature Drift:

Web application environments are highly dynamic. Code updates, configuration changes, evolving user behaviour, and new technologies cause the underlying data distribution to shift over time. This "concept drift" means that a model trained on yesterday's traffic or code may misclassify today's normal behaviour. Kirda et al. coined the term web application concept drift to describe this effect: they found that legitimate changes to a web app's functionality often produce new input patterns, and without retraining the anomaly detector, these benign requests look like attacks, causing false alarms [3]. Other authors confirm that most existing IDS and ML models do not handle drift well: one survey observes a "glaring omission" in literature, where concept drift and feature drift are often studied separately from IDS applications, leading to "fragmented understanding" of web security needs [9]. In practice, models without drift adaptation may suffer serious performance degradation when the threat landscape evolves. Kuppa and Le-Khac (2022) highlight this as a core concern: because labelled data may not be available at deployment, and attackers continually evolve tactics, ML-based detectors must include on-line drift detection and model update mechanisms [4]. In summary, handling model drift – detecting it and enabling continuous or incremental learning – remains a significant knowledge gap in ML-driven web security.

4.4. Explainability and Trust:

Many ML models (especially deep learning) are "black boxes," making it hard for security analysts to understand why a request was flagged. This lack of transparency undermines trust and makes debugging difficult. The review by Vourganas et al. stresses that explainability, bias assessment, and ethical considerations are under-emphasized in current research [10]. For example, if an IDS triggers an alert, operators may need an explanation of the key features behind the decision to act on it. Without model interpretability, it is also hard to ensure the model is not making biased or unsafe predictions. This gap means that many ML solutions are not yet deployable in high-assurance environments where accountability is required.

4.5. Scalability and Real-Time Constraints

Web applications operate at high scale and often under strict latency requirements. ML models must process millions of requests or code scans per day, which can be computationally intensive. Ensuring that ML-based detectors run in real time (e.g. for HTTP requests) or efficiently (e.g. in CI pipelines) while maintaining accuracy is non-trivial. Efficiency and resource constraints must be addressed alongside accuracy. In many cases, complex models may not be feasible, forcing trade-offs between speed and detection performance

4.6. Integration with Existing Tools:

Another practical gap is how to integrate ML into established security workflows. Many organizations use mature tools (WAFs, SIEMs, static analysers) and may be reluctant to adopt experimental ML components. Ensuring compatibility and providing proper feedback loops (e.g. ML suggestions integrated with rule-based engines) is an open challenge.

In summary, while ML offers powerful new capabilities for web security, current systems face gaps in data availability, adversarial robustness, adaptability to drift, explainability, and operational integration. Overcoming these gaps is crucial for the practical deployment of ML-based security defences.

5. Future Research Scope

Addressing the above challenges requires several key research directions:

Table 3 – Future Research Directions

Research Focus	Description	Proposed Solutions/Methods
Drift-Resilient Models	Adapt to concept drift in real-time	Online learning, drift detection, ensemble adaptation
Adversarially Robust & Explainable ML	Defend against attacks and improve transparency	Adversarial training, interpretable AI methods, human-in-the-loop
Scalability & Integration	Efficient real-time models and seamless workflow fit	Lightweight models, hybrid ML-rule systems
Data Availability & Sharing	Access to quality datasets and collaborative learning	Federated learning, synthetic data generation

5.1. Drift-Resilient Models:

A core need is to develop ML models and frameworks that can detect and adapt to concept drift in real time. Recent work proposes drift-aware IDS architectures: for example, Alzubaidi et al. suggest integrating dynamic feature selection and continuous retraining into IDS design [9]. Research should focus on lightweight drift detectors that flag when a model's accuracy degrades, coupled with incremental or online learning methods that update the model without full retraining. As Vourganas et al. identify (RQ7), adopting modern incremental learning approaches could enable IDS models to learn continuously from new benign and attack data streams while avoiding catastrophic forgetting [10]. Exploring methods like ensemble adaptation, concept-adaptive random forests, or neural networks with internal memory may be fruitful. Benchmarking these techniques on realistic, evolving web application datasets will be important to validate their effectiveness.

5.2. Adversarially Robust and Explainable ML:

Future systems must be designed with adversarial attacks in mind. Research on robust training (e.g. adversarial examples specific to web inputs, poisoning-resilient algorithms) should be expanded. In parallel, adding explainability will build trust: techniques from interpretable AI (feature attribution,

rule extraction, saliency maps) can help explain why a request or code snippet is flagged. Those parameters like explainability and robustness "must undertake a larger role" in cybersecurity ML research. For instance, integrating human-in-the-loop learning (where analysts can label or correct predictions) could improve transparency. Federated learning is another promising area: security teams could collaboratively train models on distributed logs without sharing raw data, enhancing generalizability while preserving privacy. Vourganas et al. specifically mention the use of federated learning and human-in-the-loop approaches to meet ethical and robustness requirements. [10]

5.3. Improved Data and Feature Engineering:

To overcome data scarcity, researchers should devise methods for generating realistic synthetic attack data and improving feature representations. Techniques like data augmentation for code (e.g. mutating code to inject vulnerabilities) or adversarial example generation could enrich training sets. Investigating which features are most informative yet privacy-preserving (in line with Vourganas's RQ1) is also needed [10]. High dimensional feature spaces should be reduced via automatic feature selection or representation learning. Furthermore, creating and sharing open benchmark datasets of web application logs and code samples would help standardize evaluation.

5.4. Cross-Domain and Multi-Modal Learning:

Future work could explore cross-layer ML approaches that combine network, application, and user-level data for richer context. For example, graph neural networks might model relationships between users, requests, and IPs to detect coordinated attacks. Transfer learning between different applications or domains could also be studied: a model trained on one web framework might be fine-tuned on another.

5.5. Ethics, Accountability and Standards:

As ML decisions increasingly influence security outcomes, future research must address the legal and ethical dimensions. This includes defining standards for evaluating ML security tools, certifying their robustness, and understanding the liability of automated security decisions. Vourganas et al. raise open questions about the legal and accountability implications of ML-based IDS actions [10]. Establishing guidelines and certification (e.g. adapted from NIST's AI risk management) will be important for real-world adoption.

5.6. Interdisciplinary Collaboration:

Finally, meeting these challenges will require combining expertise from machine learning, security, software engineering, and human factors. For instance, designing user-friendly interfaces for explaining alerts, or coordinating between developers and security teams to label new threats, requires cross-domain knowledge. Only through such collaboration can ML models be effectively integrated into the dynamic process of web application development and protection.

In essence, the future of AI/ML in web security lies in adaptive, transparent, and resilient models. Overcoming model drift, defending against adversarial tampering, and ensuring ethical deployment are central to this agenda. Addressing these research directions will help realize the promise of AI for robust, next-generation web application security.

6. Conclusion

This survey has reviewed the state-of-the-art of AI and ML in web application security. Current solutions from ML-driven intrusion detection systems to code-vulnerability scanners – demonstrate improved detection capabilities beyond traditional rule-based tools [2,7]. However, significant challenges remain. Key gaps include the scarcity of labelled security data, the vulnerability of ML models to evasion or poisoning attacks, and the difficulty of handling non-stationary web environments. In particular, concept drift in web applications remains a core concern: models must detect when "normal" behaviour has shifted and adapt accordingly to avoid false positives. Looking forward, researchers emphasize the need for incremental learning IDS, stronger adversarial robustness, and explainable models. Future work must prioritize improved data, model transparency, and robustness. By addressing these directions, the community can develop AI-enhanced security systems that are both powerful and trustworthy, ready to protect web applications in the ever-evolving threat landscape.

REFERENCES :

1. Oudah, M. A. M., & Marhusin, M. F. (2024). SQL Injection Detection using Machine Learning: A Review. *Malaysian Journal of Science Health & Technology*, *10*(1), 39–49. https://doi.org/10.33102/mjosht.v10i1.368
2. Rathore, D., & Pareta, C. (2024). Machine learning for web vulnerability detection. *Nanotechnology Perceptions*, *20*(7), 2123–2138.
3. Maggi, F., Robertson, W., Kruegel, C., & Vigna, G. (2009). Protecting a moving target: Addressing web application concept drift. In *Proceedings of the 12th International Symposium on Recent Advances in Intrusion Detection (RAID)* (pp. 21–40). Springer. https://doi.org/10.1007/978-3-642-04342-0_2
4. Kuppa, A., & Le-Khac, N.-A. (2022). Learn to adapt: Robust drift detection in security domain. *arXiv*. https://doi.org/10.48550/arXiv.2206.07581
5. Chughtai, M. S., Bibi, I., Karim, S., Shah, S. W. A., Laghari, A. A., & Khan, A. A. (2024). Deep learning trends and future perspectives of web security and vulnerabilities. *Journal of High Speed Networks*, *30*(12), 1–32. https://doi.org/10.3233/JHS-230037
6. Momand, A., Jan, S. U., & Ramzan, N. (2023). A systematic and comprehensive survey of recent advances in intrusion detection systems using machine learning: Deep learning, datasets, and attack taxonomy. *Journal of Sensors*, *2023*, Article 6048087. https://doi.org/10.1155/2023/6048087

7. Agoro, H., Emma, O., & Doe, J. (2024). Security vulnerability detection using machine learning. *ResearchGate*. https://doi.org/10.13140/RG.2.2.36547.18722
8. Paracha, A., Arshad, J., Ben Farah, M., Ismail, K., & others. (2024). Machine learning security and privacy: A review of threats and countermeasures. *EURASIP Journal on Information Security*, *2024*(10). https://doi.org/10.1186/s13635-024-00158-3
9. Shyaa, M. A., Ibrahim, N. F., Zainol, Z., Abdullah, R., Anbar, M., & Alzubaidi, L. (2024). Evolving cybersecurity frontiers: A comprehensive survey on concept drift and feature dynamics aware machine and deep learning in intrusion detection systems. *Engineering Applications of Artificial Intelligence, 137*, Article 109143. https://doi.org/10.1016/j.engappai.2024.109143
10. Vourganas, I. J., & Michala, A. L. (2024). Applications of machine learning in cyber security: A review. *Journal of Cybersecurity and Privacy*, *4*(4), 972–992. https://doi.org/10.3390/jcp4040045