Threat Detection and Mitigation Techniques in Web Application Security: Systematic Review

A. F. Lufna, F. S. Ahamed, M. I. Abzan Begum, M. R. M. Hanan, and A. Mohamed Aslam Sujah

Abstract The web applications have come to the forefront of digital infrastructure making it possible to conduct business, governmental and individual activities. It has contributed to the increase in cyber threats, and the most common are SQL Injection (SQLi) and the Cross-Site Scripting (XSS).This review is a systematic review that follows the standard procedures of rigor and reproducibility. An extensive search was performed on the major academic databases, such as IEEE Xplore, Scopus, ACM Digital Library, and SpringerLink between 2015-2025. Peer-reviewed journal articles, conference papers, and surveys were considered to be inclusion criteria and had to specifically target the issue of SQLi/XSS detection and mitigation in web application security. The exclusion criteria also covered publications that are not in English, duplication, and ones that do not contain empirical evidence. The search provided 500 records, and after eliminating duplicates and reviewing titles/abstracts, 124 full-text articles were evaluated, and 50 studies were eligible to the qualitative synthesis process. The PRISMA flow diagram was used to record the process of study selection. The review discovered that existing means of threat detection and mitigation are based on the interaction between static/dynamic analysis, penetration testing, AI/ML-driven detection, and secure SDLC frameworks. SQLi and XSS are also very resistant and machine learning models demonstrate excellent prospects but are not as effective in their application. Together, multi-layered security measures and automated test systems help to increase resilience, but they are still not fully integrated with common standards (e.g., OWASP). SQL Injection and Cross-Site Scripting remain the leading platforms of web-based attacks, and though the current methods of using AI-based detection, secure design, and testing in hybrid mode have prospects, there still remains a lot of room to cover in terms of scalability and practical implementation. Future studies are to be dedicated to adaptive and real-time protection systems and all-encompassing security integration across the software lifecycle.

***Index Terms*— Web Application Security, Cybersecurity, Penetration Testing, Security Vulnerabilities, Threat Mitigation**

I. Introduction

In this digital age, web applications have turned out to be the primary means of business and government usage, and the activity of the individual to provide the services and information. Consequently, it has raised the likelihood of abuse since people are dependent on such sites. There is a wide range of security issues that can affect web applications but the most common and the most devastating are Cross Site Scripting (XSS) and SQL injection (SQLi) attacks. According to the latest statistics, the number of documented breaks in the category of SQLi and XSS combined has been over 30% [1].

A.F. Lufna is undergraduate from the Department of ICT, South Eastern University of Sri Lanka, Sri Lanka.

F.S. Ahamed is undergraduate from the Department of ICT, South Eastern University of Sri Lanka, Sri Lanka.

M.I. Abzan Begum is undergraduate from the Department of ICT, South Eastern University of Sri Lanka, Sri Lanka.

M.R.M. Hanan is a graduate from the Department of ICT, South Eastern University of Sri Lanka, Sri Lanka.

A. Mohamed Aslam Sujah is a Lecturer (Prob) at the Department of ICT, South Eastern University of Sri Lanka, Sri Lanka. (Email: ameersujah@seu.ac.lk)

Weak input validation is exploited by such vulnerabilities to control or corrupt the behavior of the application and to deny or grant access to any sensitive data. SQL injections are the instances where the untrusted inputs are converted into an SQL query, which may potentially enable an attacker to access, alter and destroy important database data. However, XSS is the use of the vile addition of scripts to the web pages that the unsuspecting user will view- it may result in session hijacking, data theft, or malware distribution [2]. The two attacks pose a risk to businesses and might significantly drain them in revenues, reputation loss, and noncompliance with regulations. These attacks are quite harmful to organizations with very severe effects in terms of incurring heavy losses, damaged image, and non-adherence to regulations.

These threats develop a persistent quality since such threats are simple to carry out and since most web applications do not have stringent input validation systems. Irrespective of the technical and non-technical controls that could be used in the secure coding practice to counter this threat, most of the systems are susceptible to ignorance, lack of testing, or outdated security models. Indicatively, studies in relation to web applications found out that more than half of the sampled applications would fail under basic penetration testing [11]. That is why the urgency to work out effective detection and prevention methods is very great. The

ISSN: 2961-5992

issue that the proposed research will solve is that it will research new methods and approaches of detecting and preventing Web application attack. Raw strength of the proposed technique is integrated in dynamically and static analysis which is efficient in application security. The study determines the origins of the vulnerabilities that cause the emergence and development of the new and novel threats and develops proactive controls over the threats.

Past systematic reviews are not usually comparative in terms of new AI/ML-driven solutions and conventional ways. They rarely look at how they can be used together or successfully integrated with practical development models such as secure SDLC or DevSecOps or overcome adoption barriers such as scalability and adherence to changing security requirements.

II. METHODOLOGY

This systematic literature review focuses on analyzing already in place methods and tools designed to detect and mitigate SQL injection and Cross Site Scripting (XSS) attacks in web applications. These two vulnerabilities remain among the most common and damaging security threats, making their effective detection and prevention crucial for safeguarding sensitive data and maintaining user trust. By carefully selecting and evaluating a broad range of scholarly articles, technical reports, and case studies, this review highlights the advantages and limitations of current approaches. The methodology strictly adheres to the principles of systematic reviews, ensuring that the research process is rigorous, transparent, and reproducible. This structured approach minimizes bias and increases the findings' dependability. Ultimately, the review focus to give comprehensive perspectives on the current situations of the art techniques, identifying gaps in existing solutions, and offer guidance for future research and practical implementations to strengthen web application security against these common attack vectors.

A. Research Questions

TABLE I
RESEARCH QUESTIONS

No	Research Question
1	Right now, what are the cutting-edge methods for discovering the presence of SQLi and XSS?
2	To what extent are the current methods capable of addressing modern-day XSS and SQLi attacks?
3	What are the identified gaps and future directions in Protecting web applications from SQLi and XSS attacks?

B. Search Strategy

Such massive searches took place in the prominent academic portals include IEEE Xplore, MDPI, Scopus, ACM Digital Library, and SpringerLink. Keyword search and Boolean operations would be utilized in generating studies in these areas: Web Application Vulnerabilities, Web Application Security, Web Security and SQL injection Detection.

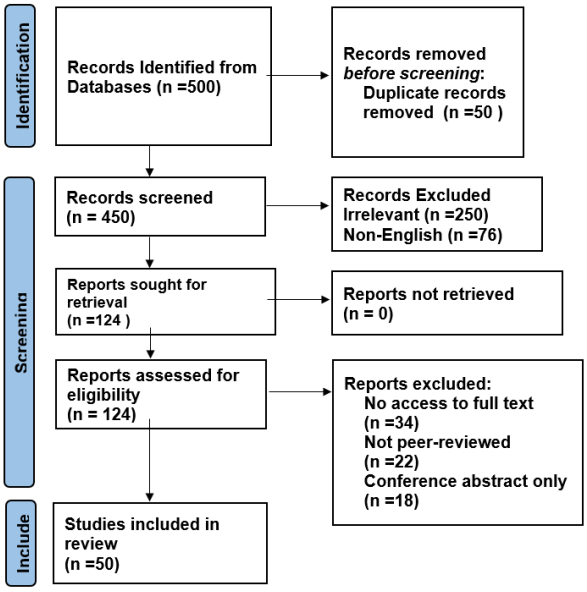
C. Inclusion and Exclusion Criteria

Inclusion Criteria: The inclusion criteria for this systematic literature review encompassed peer-reviewed journal articles, conference papers, and survey studies published within the last ten years (2015–2025). Only research specifically focusing on the detection and prevention of SQL Injection (SQLi) and Cross-Site Scripting (XSS) vulnerabilities was considered to ensure relevance to web application security.

Exclusion Criteria: Studies were excluded if they were published in languages other than English, were duplicate publications, or lacked empirical evidence supporting their findings. This approach ensured that the selected literature was both recent and directly applicable to the objectives of this review.

D. Study Selection

The preliminary appraisal was concerned with the titles and abstracts of articles for filtering out irrelevant studies. Full text reviews were then conducted to establish alignment between the articles and research questions. A PRISMA flow diagram would document the selection process. Initially, filters were placed on titles and abstracts to assess and rule out irrelevant studies. These full text articles were then reviewed to affirm the research questions. The process of selection was documented using a PRISMA flow diagram (See Fig. 1).



. Fig. 1: PRISMA Diagram

E. Data Extraction Synthesis

Data relevant for the research were gathered through a predetermined template, which included all of the following points: Names of authors and publication year, Proposed techniques and evaluation metrics as well as datasets used and Key findings and limitations methodologies.

ISSN: 2961-5992

F. Quality Assessment

A quality assessment checklist was applied to each selected study, taking into consideration the elements of methodological rigor, evaluation, completeness, and reproducibility of results.

III. Systematic Literature Review

A. Introduction

The use of web applications is now omnipresent in the contemporary society as these applications play a pivotal role in finance, health, online shopping, education, and government services. Since these applications involve sensitive user data, they are ideal targets of attackers aiming at gaining unauthorized access, stealing data or disrupting the service. The vulnerabilities typical of web applications are SQL Injection (SQLi), Cross-site Scripting (XSS), weak authentication, misconfigurations and flaws in business logic [1]-[2]. The risks associated with these vulnerabilities include loss of money, tarnished reputation as well as legal consequences. Web technologies have also developed at a high rate which has led to the rise of attack surfaces, and conventional security controls are no longer sufficient. To solve this, scholars have placed more emphasis on proactive detection, automated prevention, and artificial intelligence (AI) and machine learning (ML) to detect and thwart threats in real time [20], [27]. This is a systematic review of 50 recent studies, in which the author identifies the trends in threat detection, prevention models, and new strategies that can enhance web application security.

B. SQL Injection Detection and Prevention

SQL Injection remains among the most urgent threats of the web application as a result of the lack of proper input verification and query and query building dynamism [3], [22]. A number of studies have suggested detection frameworks based on signature-based as well as anomaly-based approaches. The predictive models are AI-based solutions that can dynamically detect attacks that were never seen before, which is better than conventional rule-based solutions [27], [49]. The combination of a static and a dynamic analysis is often mentioned as being effective in the detection of SQLi. An example of this is SDLC-based which focuses on incorporating security checks at all stages of software development life cycle to avoid vulnerabilities during the design and coding process [38].

Most recent works also include the methods of query comparison and encryption so that the malicious access to the database could be prevented [48]. In general, these contributions show the change to smart, active, and multi-layered SQLi prevention.

C. Cross-Site Scripting (XSS) and WAF Security

Another significant security threat is Cross-site Scripting attacks, where the attacker inserts malicious scripts in the web pages being accessed by other users. Studies have aimed at examining the concept of static analysis, pattern recognition and detection using AI to counter these attacks [6], [33]. Massive research demonstrates that several XSS flaws go unmitigated because of incompetence on the part of developers and insufficient automated testing [4], [39]. It has been demonstrated that AI-based systems are capable of automatically identifying XSS attempts and creating test conditions to test web application firewalls (WAFs) [20], [41]. Research focuses on the significance of integrating AI detection with best coding practices, input sanitization, output encoding and user education [29], [32]. It is agreed that the use of multi-layered defenses based on combining AI models with conventional security mechanisms can be of great help in preventing XSS.

D. Vulnerability Assessment and Standardized Frameworks

Technological frameworks, including the OWASP Top 10, have become commonly adopted to perform vulnerability assessment and security advice [1], [2], [5]. These frameworks offer a holistic advice on how to avoid the usual vulnerabilities, such as SQLi, XSS, and insecure settings. The latest studies emphasize the advantages of massive automated tests to track the vulnerabilities in a variety of platforms and technologies [4], [44]. It has become a major research trend that hybrid vulnerability assessment techniques which are primarily based on static, dynamic, and hybrid analysis have been developed [12]. The dynamic monitoring methods are also employed in the reproduction of real-life attacks to capture vulnerabilities that may not be detected by the use of the static tools [11]. Vulnerability scanning systems and automated penetration tests are used to help improve audit of security and minimize the chance of vulnerabilities that are not patented [45], [46]. Investigations comparing scanning tools with quantitative studies reveal inconsistent accuracy and efficiency, which gives importance on the integration of multi-tools [44].

E. AI and Machine Learning Applications in Web Security

The world of web application security has changed with AI and machine learning. The machine learning models have the ability to forecast the pattern of attack, identify abnormal behavior and categorize the web traffic on the basis of security requirements [15], [18]. Fuzzy logic/ multi-criteria decision-making/ feature fusion Hybrid models have been useful in considering comprehensively the web application security [18], [41]. SQLi and XSS detection intelligent systems have proven to be more accurate than the traditional ones [20], [27], [40]. GenXSS, an AI-based framework, resorts to automated testing and adaptive learning in order to prevent attacks to WAFs proactively [20]. Misconfiguration detection, real-time monitoring, and vulnerability prioritization are also applicable to the application of AI, as security teams can distribute resources effectively [44], [45]. The overall implication of these studies is that there is a shift in paradigm with respect to smart and proactive web security.

F. Emerging Trends in Integrated Web Security Approaches

The new studies indicate the combination of various security techniques, such as AI-based detection, vulnerability testing, penetration testing, and an industry-wide standard framework [1], [2], [4], [5], [39], [44]. It is suggested that layered security plans

ISSN: 2961-5992

be implemented and that they include secure coding methods, WAFs, active monitoring, and automated fixing [45], [46], [48]. There is also a tendency towards more widespread use of large-scale approaches, in which automated systems test their vulnerability against thousands of web applications and thus allow a comparative study of security practices [4], [44]. The other new technology is combining biometric authentication and access control methods, which enhances security of web applications [10].

G.Discussion and Research Gaps

The analyzed literature provides a number of valuable findings about the latest state of the web security research. SQL injection (SQLi) and cross-site scripting (XSS) are the most common types of vulnerabilities and still plague the developers and security experts in spite of the advanced security frameworks that have been introduced [3], [6], [14], [20]. Recent reports suggest that AI-based solutions, in general, and machine learning models, in particular, have demonstrated the promising potential of improvements in detection and prevention processes. Nevertheless, scalability and the ability to integrate such models in the real-life setting continue to be a significant problem that still requires investigation [20], [27], [41]. Also, structured frameworks such as OWASP Top 10 and ISSAF are useful security guidelines, but have not been effectively combined with intelligent and adaptive detection tools, and thus their usefulness in addressing emerging and evolving threats remains limited [5], [46]. Another point brought out in the literature is the fact that to be able to detect all vulnerabilities the analysis of the code alone is not enough. Therefore, dynamic and mixed testing approaches should be adopted to model and recreate real-time attack situations to offer a more comprehensive test with regard to security [11], [12]. Moreover, the contribution of constant monitoring and automated incident response systems have not been thoroughly studied despite being important elements in ensuring a healthy operational security in the Internet applications of today [44], [45]. All in all, the current results underpin the importance of future studies, which should aim at investigating AI-based real-time web applications protection, adaptive testing techniques of web application firewalls (WAFs), large-scale automated vulnerability tests, and the ability of multi-layered security systems to integrate seamlessly. These developments will lead to the development of a more robust, smart and self-protective web security environment that can tackle the changing cyber threats.

H. Conclusion

This academic literature review is a compilation of 50 articles on web application security that were written during 2015-2025. SQLi and XSS are still common threats, whereas AI and machine learning are useful in detection and prevention. The vulnerability assessment frameworks, penetration testing, and monitoring tools also contribute to the resilience of security. Further studies should be directed on scalable AI implementation, automated response mechanisms, and real-time dynamic defense mechanisms in order to counter the new web threats. The paper has demonstrated the changing nature of web application security, and the increasing value of smart, combined, and proactive controls.

IV.RESULTS AND DISCUSSION

A. Overview of Web Application Vulnerabilities

Web applications are one of the critical elements of the modern digital infrastructure that provide online banking, e-commerce, online education, and healthcare services. Nevertheless, the blistering development of web technologies also has made them vulnerable to a wide variety of security threats that are actively used by their opponents. Among them, the two most common and most endeavored weaknesses are SQL Injection (SQLi) and Cross-site Scripting (XSS), which have contributed to a significant proportion of web-based attacks over recent years [3], [6], [14], [20]. Improper input validation, insecure query construction, and lack of output sanitization are the common causes of these vulnerabilities during web development. SQLi lets attackers exploit the backend databases by injecting malicious SQL commands in the input fields whereas XSS lets attackers run arbitrary script in browsers of the users compromising their session tokens and sensitive information. Research studies that have been conducted in the last 10 years have shown that even with the ongoing development of defensive technologies, the traditional, static analysis and filtering systems are not capable of detecting sophisticated or obfuscated attack patterns [11], [12], [29]. The growing adoption of dynamic and hybrid testing methods has improved the precision of vulnerability detection but it has problems in detecting zero-day exploits and context-specific threats [31], [41]. Scholars have thus suggested that machine learning and artificial intelligence models be incorporated into web security systems to automate the process of detection and response mechanisms [20], [27], [47]. These smart systems use the behavior incidents and the anomaly of network traffic to forecast the attack in advance, which decreases the human participation and reaction time.

Other severe threats to modern web applications are Cross-Site Request Forgery (CSRF), Remote File Inclusion (RFI), Server-Side Request Forgery (SSRF) and Broken Authentication [5], [15], [18]. The OWASP Top 10 framework remains as a reference of defects detection and prevention of such vulnerabilities and focuses on secure coding practices, authentication, and appropriate configuration operations [5], [46]. However, it is not that such guidelines are easily integrated into the software development lifecycle (SDLC). In most organizations, security is still being regarded as a secondary consideration, as opposed to the fundamental design, and as such, this feature results in the same vulnerabilities being created in production environments [23], [25]. The recent surveys also noted the increased complexity of cloud-hosted and API-based web applications that pose new attack surfaces like the JSON injection, insecure deserialization, and API key exposures [7], [19], [34]. The implementation of microservices architecture and containerization technologies also makes the traditional security models more complicated, requiring the adaptive and constant monitoring mechanisms [42], [45]. It has been reported, therefore, that researchers have shifted their efforts

ISSN: 2961-5992

TABLE II
GROUP-WISE CLASSIFICATION OF REVIEWED WEB APPLICATION SECURITY STUDIES

Group	Ref. (IEEE)	Focus / Research Area	Methodology / Approach	Key Findings / Contributions
1. SQL Injection (SQLi) Detection and Prevention	[3], [13], [22], [24], [26], [28], [30], [35], [36], [38], [39], [40], [47], [48], [49], [50]	SQLi vulnerability detection, prevention frameworks, and hybrid solutions	Static/dynamic testing, ML models, hybrid detection, and secure SDLC frameworks	SQLi remains the most persistent vulnerability in web apps. Studies emphasize parameterized queries, encryption, hybrid detection (signature + anomaly), and AI-driven prevention frameworks. Machine learning-based models ([49]) show superior accuracy but face deployment and scalability challenges.
2. Cross-Site Scripting (XSS) Detection and Mitigation	[6], [29], [31], [32], [33], [37], [41]	Detection and prevention of XSS in modern web environments	Static/dynamic analysis, GA-based search, ML fusion models	XSS detection has evolved from static filtering ([6]) to intelligent hybrid systems ([41]). ML-based fusion techniques improve precision, while GA approaches ([31]) uncover complex injection patterns. Studies highlight content sanitization and CSP enforcement as essential.
3. Machine Learning (ML) and AI-driven Web Security	[20], [27], [41], [42], [44], [45], [47], [49]	AI-based detection and defense automation	Deep learning, adversarial generation, multi-feature fusion, automated platforms	AI-driven models (e.g., GenXSS [20]) can autonomously detect and generate attacks to strengthen WAFs. Unified frameworks like UniEmbed ([41]) combine multiple features for enhanced accuracy. However, real-world deployment remains limited due to computational cost and data imbalance.
4. Web Application Security Frameworks and SDLC Integration	[5], [18], [23], [25], [34], [38], [46]	Secure development lifecycle (SDLC) and frameworks like OWASP, ISSAF, OSSTMM	Comparative evaluation, case study, and structured testing	Framework integration ([5]) shows measurable reduction in vulnerabilities when applied early in the SDLC. ISSAF-based pentests ([46]) demonstrate value in structured assessment. Studies urge embedding security metrics during design stages for consistent results.
5. General Web Application Security Surveys and Reviews	[1], [2], [7], [8], [12], [15], [16], [19], [21], [22], [29], [34], [37], [40]	Comprehensive analyses of web vulnerabilities and trends	Systematic and comparative surveys	These surveys provide foundational insights into evolving threats such as SQLi and XSS, emphasizing human errors and misconfigurations as key enablers. They also summarize progress from static testing to AI-driven and automated methods.
6. Security Tools, Visualization, and Automation	[4], [9], [10], [11], [17], [25], [42], [43], [44], [45]	Automated scanning, visualization, and dynamic monitoring	Tool and framework development, visualization-based analysis	Tools like SCWAD ([45]) and monitoring frameworks ([11]) improve real-time detection and response. Visualization methods ([17]) aid in vulnerability prioritization. Automated scanning still misses logical flaws, stressing hybrid integration with AI.

to Web Application Firewalls (WAFs) and runtime protection platforms such as SecRASP and GenXSS to provide real-time protection and mitigation [20], [41]. These systems utilize both a combination of the static analysis and anomaly detection, as well as adaptive learning to identify and block attacks dynamically, yet the problems of deployment, including high false-positive rates and processing overhead, have not been resolved yet [41], [47].

Besides the technological vulnerabilities, human and organizational vulnerability are also critical in web application insecurity. Systems are often misconfigured, have weak access controls policies, weak password management and weak patching practices, which often lead to data breaches [12], [15], [21]. As noted in several reviews, even the most sophisticated tools cannot be efficient unless the developers are correctly informed, trained in the area of secure coding and audited on a regular basis [1], [2], [22]. Therefore, it is as important to develop a culture of security in the process of development and maintenance as it is to install technical security. Conclusively, the literature highlights the fact that vulnerabilities of web applications are a significant issue of cybersecurity even advances with ongoing research and innovation. The assault vectors remain dominated by SQLi and XSS, with new problems arising because of the growing nature of web ecosystems. The intersecting Application of AI-guided analytics, integrated-secure SDLC, and automatic shared runtime defensive represents the present boundary in web safety development fields. Nevertheless, to implement holistic resilience, the future strategies should integrate intelligent detection, proactive mitigation, and human-centered security practices into a single defense measure [20], [27], [44], [45], [46].

B. Advances in Detection and Prevention Technology

The past ten years have seen studies yield a disparate array of web-application threat detection and prevention tools which have shifted the focus away from rule-based and signature-based defense mechanisms to hybrid, behavior-aware and AI-driven tools. Missing the handling of context information and run-time behavior, often considered their weaknesses, still makes traditional actors of the static-analysis tools and rule engines core to the finding of innocent mistakes and misconfigurations in code, nonetheless clear catalysts for compensatory methods like dynamic analysis, runtime monitoring, and hybrid pipelines, which combine them into delinquent approaches [6], [11], and [12]. It has been demonstrated that dynamic instrumentation and application monitoring can replicate and expose bugs that are not detected by the static scanning system, allowing more certain determination of the exploitation paths and defective runtime assumptions [11]. Similarly, comprehensive vulnerability

ISSN: 2961-5992

TABLE III

WEB APPLICATION VULNERABILITIES AND PREVENTION TECHNIQUES

No	Vulnerability Type	Key Findings	Prevention/Detection Techniques
1	SQL Injection (SQLi)	Widely prevalent, exploits poor input validation and database interaction vulnerabilities	Input validation, prepared statements, parameterized queries, automated tools like SQL-map, and machine learning-based detection
2	Cross-Site Scripting (XSS)	Targets client-side code to execute malicious scripts, affecting user sessions and data	Content security policies, input sanitization, and AI-driven pattern recognition tools
3	Broken Access Control	Results in privilege escalation and unauthorized access due to misconfigurations	Role-based access control, regular audits, and proper session handling
4	Security Misconfigurations	Occur due to improper server and application settings	Tools like SCAAMP and MVS for auditing and automated configuration adjustments
5	Cross Site Request Forgery (CSRF)	Misuses user trust in sessions that were authenticated	Nonce-based random token validation and referrer headers

assessment practices using automated and manual vulnerability analysis enhances coverage and lays emphasis on actionable vulnerability analysis findings to fix the vulnerability [12]. Methods of detection have developed in two primary directions, namely: (1) developments in the area of program-analysis (better static, dynamic, symbolic execution, taint tracking) and (2) data-driven / learning methods. The advanced forms of the static and dynamic analyzers are able to provide program models, propagation of taints, as well as symbolic arguments detecting the advanced input-flow vulnerabilities which are missed by the simplistic pattern-checks [6], [29], [33]. Semi-automatic model-based testing and test generation Work focuses on using formalized models of application behavior to increase test coverage and discover logic level bugs [9]. Simultaneously, hybrid analysis that combines both still predictions and dynamic verification decreases the number of false positives and produces actionable alerts, which has been confirmed in a number of comparative assessments [30], [31].

Machine learning (ML) and artificial intelligence (AI) have been widely used in detection and prevention. Request classification, anomalous query shape detection, and patterns that signify SQL injection (SQLi), cross-site scripting (XSS) and other types of attacks are classified using supervised and unsupervised models [27], [41]. Multi-feature fusion techniques, which are the combination of lexical features of inputs, structural features of queries, and contextual runtime cues, have demonstrated a higher detection in both SQLi and XSS and perform better than single-feature detectors in a variety of benchmarked environments [41]. Adversarial testing has also been made possible by AI: architectures that automatically produce a variety of novel attack inputs (such as adversarial SQLi/XSS samples) test Web Application Firewalls (WAFs) and clarify its vulnerabilities that cannot be predicted by static signature sets [14], [20]. These generative and adversarial methods have demonstrated the delicate edges of most currently deployed WAFs and hence encouraged generates constant adaptation defenses. In the prevention category, progress can be made in three mutually complementary approaches: (a) secure development practices as a part of SDLC, (b) runtime mitigations (WAFs, RASP), and (c) proactive hardening (parameterization of queries, encoding of outputs, encrypting them). Security checks and threat modeling embedded in the SDLC has the benefit of minimizing the number of vulnerabilities that are introduced early in the development process and enables automated testing gates to reject insecure changes prior to release [28], [38]. In-situ protection, which is offered by Runtime Application Self-Protection (RASP) and the development of WAFs, especially with the use of ML, operates on request inspection, preventing bad traffic, and performing corrective transformations; nonetheless, these systems need to strike a balance between detection sensitivity and latency and false-positive risks [20], [41], [47]. The algorithms used to prevent exploitation have been proposed to include query-comparison, canonicalization and selective encryption of sensitive query elements, which will make exploitation substantially more difficult in the presence of injection primitives [48].

It has been stated in a number of works that detection tooling should be assessed and measured on a continuous basis. The quantitative measurements of scanner and pen test tools efficiency state that neither single tool nor the tool ensemble can identify all categories of vulnerabilities and the tool ensembles with complementary capabilities provide the most practical value [44], [45]. End-to-end discovery and reduced manual effort Automated pen testing platforms that coordinate multiple scanners, exploit modules and validation steps are better than

ISSN: 2961-5992

manual orchestration logic but need strong orchestration logic to eliminate false positives and safely verify findings [45]. Multi-project dashboards and visualization can be used to triage and monitor remediation activities through the surfacing of high-impact findings and time series in vulnerability discovery [17]. Although AI has potential, there are still issues with real-world implementation. ML models need representative, labeled data; they are sensitive to concept drift as behavior changes; and they can be easily evaded by adversarial examples unless adversarial training and constant re-training are used [14], [20]. Additionally, computation and latency cannot scale the deployment of more complex models especially when high throughput visions are needed in production settings. Some studies thus recommend having hybrid designs to leverage the lightweight and fast rule components to first filter the cases before escalating suspicious cases to more demanding ML models or the use of dynamic analyzers to confirm the suspicious cases [30], [36]. This multi-level strategy conserves performance and has the advantage of the accuracy of sophisticated methods of detection. Overall, today's state of the art in detection and prevention can be summarized as (i) the use of hybrid pipelines, where static, dynamic, and machine learning signals are combined to enhance the accuracy of the detection; (ii) generative/adversarial testing, where WAFs are hardened and evasions are tested; (ii) SDLC and runtime integration, to prevent and reduce vulnerabilities throughout the software lifecycle; and (iv) orchestration and measurement, to test tools under realistic and large-scale conditions [6], [9], [11], [14 Further research must explore ways to make adaptive detectors deployable at scale, develop the most continuous protection against adversarial inputs, and have common reference point based on the sophistication of the modern web ecosystem.

C. Challenges and Future Direction

Although web application security has developed greatly, there are still a number of challenges in dealing with the vulnerabilities like SQL Injection (SQLi), Cross-Site Scripting (XSS), and ineffective authentication tools [3], [6], [20], [27]. A number of solutions available today are very much dependent on statical or signature detection which in most cases is incapable of detecting zero-day exploits and obfuscated attack payloads [11], [12]. The scalability of AI-driven models is also an issue because real-world implementation demands constant learning and adaptation to the changing attack patterns [20], [41]. Moreover, the combination of smart security systems and industry-based standards, including OWASP and ISSAF remains limited, which leads to the fragmented nature of defense mechanisms [5], [46]. The issue of runtime threats, which is not covered by the static analysis, has to be addressed with the help of dynamic and hybrid testing techniques [12], [18]. Also, operational resilience to large-scale attacks becomes weak due to the absence of automated incident response and real-time monitoring [44], [45]. The need of future studies should be to create adaptive AI-assisted Web Application Firewalls (WAFs) that can dynamically respond, analyze vulnerabilities on a large scale, and form self-learning systems. Besides, more interdisciplinary systems that incorporate human knowledge and machine learning may result in sustained improvement of web security and a more resilient defense ecosystem.

V. CHALLENGES AND OPEN ISSUES

Web application security has advanced considerably, yet persistent challenges and open issues continue to undermine effective defense. SQL Injection (SQLi) and Cross-Site Scripting (XSS) remain the most frequently exploited vulnerabilities due to their adaptability and the prevalence of legacy systems [3], [6], [14], [20]. Static and signature-based detection methods are insufficient for identifying zero-day exploits, polymorphic attacks, and obfuscated payloads, highlighting the limitations of traditional security approaches [11], [12].AI-driven detection models offer significant promise, enabling automated identification of complex attack patterns; however, real-world deployment faces scalability challenges, model drift, and high resource requirements [20], [27], [41]. Integration with standard frameworks such as OWASP Top 10, ISSAF, and other security guidelines is often partial, resulting in fragmented protection and inconsistent security policies [5], [46]. The development of web application security has gone significant yet the issues and unresolved challenges are still hanging on ensuring successful defense. The most common vulnerabilities to use are SQL Injections (SQLi) and Cross-site Scripting (XSS) because of their flexibility and the number of outdated systems [3], [6], [14], [20]. Existing security measures prove only to be problematic in detecting zero-day exploits, polymorphic attacks, and obfuscated payloads as the concept of static and implausible signature creation cannot detect advanced attacks, which have clearly shown their abilities as detection models, but extensive resources are still needed for bigger operations in practice [11], [12]. Detection models derived using AI have proven a great deal in detecting complex attack patterns and models but there are serious shortfalls in scalability and overwhelming resource requirements that limit deployment opportunities in the real world [20], [27], [41]. It is frequently only integrated with more general frameworks like OWASP Top 10, ISSAF, and other security frameworks in a partial manner, leading to fragmentary protection and ad hoc security policies [5], [46]. Also, dynamic testing and hybrid analysis methods, though useful are not popularly employed resulting in susceptibility to runtime attacks going undetected [12], [18]. These are continuous monitoring, automated response, and real-time adaptation, which are underutilized domains that are significant to operational resilience [44], [45]. There is also additional complication of web applications and microservices, as well as integration of third-party apps that enhance these issues. New areas of open research are AI-enabled Web Application Firewalls (WAFs) with adaptive learning, scalable vulnerability assessment systems, and collaborative systems that integrate security knowledge with automated vulnerability identification and analysis. These concerns will be a critical metric on which the way to robust, scalable and proactive web application security posture is reached.

ISSN: 2961-5992

VI. CONCLUSION

The literature review methodology scouts deep in the complex and multifaceted problems of securing wicked web applications; an aspect that continues rapidly changing together with technological advancements alongside professional cyber-attacks. This is the primitive vulnerability in SQLi, cross Scripting (XSS) and other misconfiguration associated challenges that are still there even in the decades following the research and development of security solutions across decades. These have always been ranked at the top of the list of the most threatening endangering threats but they are applied by the accidents of the perpetrators of the international environment. The minimization of such issues has had a significant push over the years, and one of the most pronounced trends towards it has been the process of automated and AI-driven threat and prevention, and its response. Such tools as SQL Map and SCAAMP can be listed as examples of such movement that have proved to be rather helpful in various environments and circumstances. Nonetheless, the tools to possess shared vulnerabilities, especially in regard to the scalability in terms of large scale and complicated systems, the simplicity with which one can incorporate them in a wide range of development environments and the general trial against their performance on diverse situations. The results of this review are applicable to both academicians and in the practitioners of the industry. The information provided in any case by the academia is a roadmap in the systematic application of the gaps in the available study especially gaps in quality of datasets, rigor of utilized methods, and utility of the applied methods. Devoting investigation efforts to the areas of such kind, the academic community might offer much to streamline the vulnerability detection processes, preventing them and making them more stable. The implications of the same are also useful to the industry stakeholders. The vulnerability of web applications is incredibly dynamic, therefore, the ability to add security in dynamic manner which is both proactive is required. It is also accompanied by the appearance of new trends, including DevSecops, that consider the security questions at all stages of software creation and the introduction of new solutions, which can be easily modified in case of a new threat. Industry players are expected to be flexible to enable them integrate the emerging papers on research and security technologies so that the critical applications mitigate the ever-changing threats. Based on this overview, it must be noted that there are some areas of research that would also prove to be productive in future. Some of the suggestions include creation of quality standard datasets and benchmarks which would be utilized in comparison and estimation of performance of different vulnerability detectors. It would allow more objective and consistent assessment in not only the academic community, but also in the industry. The other notable one is the research of resilience to adversaries in the context of AI-assisted detection models in that way to make sure that the latter abide by intentional action of either avoiding or abusing it. Moreover, greater attention should be paid to the inclusion of security controls at much earlier stages of software development thus reducing the vulnerability prior to an entrenchment will occur in the applications. Further research to monitor and eradicate attacks to supply chain is also quite crucial as it becomes more challenging to align the software ecosystem, and implementation of third-party software. Finally, the ability to research the scope beyond the current has already included the emerging web technologies, i.e. web assembly and progressive web applications, the relevance and efficiency of security measures can be assured in the future since the web continues to evolve. By these guidelines which have been adhered to, the academic community and also the professional community can have a better and safer future of web application development and deployment.

In order to inform this systematic review, we developed three research questions that are presented in Table I. This paper is further divided into sections to answer each question, with the first section III outlining the innovative detection and mitigation methods (RQ1), the second section IV outlining their efficiency in the current setting (RQ2), and the last section V outlining the main gaps and the future research directions (RQ3).

REFERENCES

[1] A. C. Aladi, "Web Application Security: A Pragmatic Exposé," Commun. ACM, vol. 67, no. 7, pp. 64–73, Jul. 2024. doi: https://doi.org/10.1145/3644394

[2] A. Mohammed, J. Alkhathami, H. Alsuwat, and E. Alsuwat, "Security of Web Applications: Threats, Vulnerabilities, and Protection Methods," IJCSNS Int. J. Comput. Sci. Netw. Secur., vol. 21, no. 8, p. 167, Aug. 2021. doi: https://doi.org/10.22937/IJCSNS.2021.21.8.22

[3] D. Alam, M. Alamgir Kabir, T. Bhuiyan, and T. Farah, "A Case Study of SQL Injection Vulnerabilities Assessment of .bd Domain Web Applications," Proc. CyberSec 2015, 2015, pp. 73–77. https://doi.org/10.1109/CyberSec.2015.23

[4] P. Mutchler, A. Doupé, J. Mitchell, C. Kruegel, and G. Vigna, "A Large-Scale Study of Mobile Web App Security," Proc. MoST (Mobile Security Technologies) Workshop, 2015. https://www.ieee-security.org/TC/SPW2015/MoST/papers/s2p3.pdf

[5] N. Nedeljković, N. Vugdelija, and N. Kojić, "Use of 'OWASP Top 10' in Web Application Security," ITEMA 2020 Conference Proc., pp. 25–30, 2020. doi: https://doi.org/10.31410/ITEMA.2020.25

[6] A. W. Marashdih, Z. F. Zaaba, K. Suwais, and N. A. Mohd, "Web application security: An investigation on static analysis with other algorithms to detect cross site scripting," Procedia Comput. Sci., vol. 163, pp. 1173–1181, 2019. doi: https://doi.org/10.1016/j.procs.2019.11.230

[7] X. Li and Y. Xue, "A Survey on Web Application Security." https://www.isis.vanderbilt.edu/sites/isis.vanderbilt.edu/files/bibcite_files/main_0_0.pdf

[8] M. Shema and J. Blanco Alcover, Hacking Web Apps: Detecting and Preventing Web Application Security Problems. (book).

[9] M. Büchler, J. Oudinet, and A. Pretschner, "Semi-Automatic Security Testing of Web Applications from a Secure Model." https://doi.org/10.1109/SERE.2012.38

[10] W. Kusuma Herdanu, R. Alayham, A. Helmi, and M. Syamsudin, "Integration biometrics in web application: Security for web apps," Int. J. (vol. 3, no. 2), p. 103, 2023. https://doi.org/10.5281/zenodo.8139747

[11] D. Wang, M. Galster, and M. Morales-Trujillo, "Application Monitoring for Bug Reproduction in Web-Based Applications," Journal of Systems and Software, vol. 207, Jan. 2024, Art. no. 111834. doi: https://doi.org/10.1016/j.jss.2023.111834

[12] P. S. S. K. Gandikota, D. Valluri, S. B. Mundru, G. K. Yanala, and S. Sushaini, "Web Application Security through Comprehensive Vulnerability Assessment," Procedia Comput. Sci., vol. 230, 2023, pp. 168–182. doi: https://doi.org/10.1016/j.procs.2023.12.072

[13] J. Li, "Analysis and Prevention of SQL Injection based on Web," J. Theory Pract. Eng. Sci., vol. 4, no. 09, pp. 35–38, Oct. 2024. https://doi.org/10.53469/jtpes.2024.04(09).05

[14] Z. Qu, X. Ling, T. Wang, X. Chen, S. Ji, and C. Wu, "AdvSQLi: Generating Adversarial SQL Injections against Real-world WAF-as-a-

ISSN: 2961-5992

Service," arXiv preprint, Jan. 2024. : https://doi.org/10.48550/arXiv.2401.02615

[15] J. Shahid, M. K. Hameed, I. T. Javed, K. N. Qureshi, M. Ali, and N. Crespi, "A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions," Applied Sciences, vol. 12, no. 8, Apr. 2022. https://doi.org/10.3390/app12084077

[16] A. K. Pandey and F. Alsolami, "Malware Analysis in Web Application Security: An Investigation and Suggestion." (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 11, No. 7, 2020

[17] F. O. Sonmez and B. G. Kilic, "Holistic Web Application Security Visualization for Multi-Project and Multi-Phase Dynamic Application Security Test Results," IEEE Access, vol. 9, pp. 25858–25884, 2021. doi: https://doi.org/10.1109/ACCESS.2021.3057044

[18] M. Alenezi, A. Agrawal, R. Kumar, and R. A. Khan, "Evaluating Performance of Web Application Security through a Fuzzy Based Hybrid Multi-Criteria Decision-Making Approach: Design Tactics Perspective," IEEE Access, vol. 8, pp. 25543–25556, 2020. doi: https://doi.org/10.1109/ACCESS.2020.2970784

[19] B. Eshete and A. Villafiorita, "Early Detection of Security Misconfiguration Vulnerabilities in Web Applications." https://doi.org/10.1109/ARES.2011.31

[20] V. Babaey and A. Ravindran, "GenXSS: An AI-Driven Framework for Automated Detection of XSS Attacks in WAFs," arXiv preprint arXiv:2504.08176, Apr. 2025. https://doi.org/10.48550/arXiv.2504.08176

[21] K. Curran and S. McNally, "Web Application Vulnerabilities." https://doi.org/10.1109/ISCSIC64297.2024.00081

[22] D. Abdoulaye Kindy and A.-S. Khan Pathan, "A Survey on SQL Injection: Vulnerabilities, Attacks, and Prevention Techniques." https://doi.org/10.1109/ISCE.2011.5973873

[23] D. Mitropoulos, P. Louridas, M. Polychronakis, and A. D. Keromytis, "Defending Against Web Application Attacks: Approaches, Challenges and Implications," IEEE Trans. Dependable Secure Comput., vol. 16, no. 2, pp. 188–203, Mar. 2019. doi: https://doi.org/10.1109/TDSC.2017.2665620

[24] Yusuf Bukar Maina, "A Critical Evaluation of Security Approaches for Detection and Prevention of SQL Injection Attacks in Web-Based Applications," FUDMA Journal of Sciences, 2024. doi: https://doi.org/10.33003/fjs-2024-0802-2308

[25] A. Wainakh, A. Wabbi, and B. Alkhatib, "Design & Develop Misconfiguration Vulnerabilities Scanner for Web Applications." International Review on Computers and Software (I.RE.CO.S.), Vol. 9, n. 10

[26] N. Bedeković, L. Havaš, T. Horvat, and D. Crčić, "The Importance of Developing Preventive Techniques for SQL Injection Attacks," Tehnički Glasnik, vol. 16, no. 4, pp. 523–529, Sep. 2022. doi: https://doi.org/10.31803/tg-20211203090618

[27] Krishna Chaitanya Chaganti, "AI-Driven SQL Injection Prevention: Strengthening Database Security," International Journal of Science And Engineering, vol. 11, no. 1, 2025. doi: https://doi.org/10.53555/ephijse.v11i1.285

[28] A. S. Kumar and P. R. Sharma, "Enhancing SQL Injection Attack Prevention: A Framework for Detection, Secure Development, and Intelligent Techniques," Journal of Informatics and Communication Technology (JICT), vol. 5, no. 2, pp. 138–148, Dec. 2023, https://doi.org/10.52661/j_ict.v5i2.233

[29] G. Rodríguez, J. Torres, and P. Flores, "Cross-Site Scripting (XSS) Attacks and Mitigation: A Survey." https://doi.org/10.1016/j.comnet.2019.106960

[30] N. Joshi, T. Sheth, V. Shah, J. Gupta, and S. Mujawar, "A Detailed Evaluation of SQL Injection Attacks, Detection and Prevention Techniques," in 5th IEEE Int. Conf. Advances in Science and Technology (ICAST 2022), 2022, pp. 352–357. doi: https://doi.org/10.1109/ICAST55766.2022.10039662

[31] Pooja Panadiya, Prof. Manish Kumar Singhal, "Advanced Detection and Prevention of SQL Injection Attacks Using Machine Learning Techniques for Enhanced Web Security", December 2024International Journal of Scientific Research in Science and Technology 11(6):554-564 https://doi.org/10.32628/IJSRST241161101

[32] O. Okusi, "Cyber Security Techniques for Detecting and Preventing Cross-Site Scripting Attacks," World Journal of Innovation and Modern Technology, 2024. https://doi.org/10.56201/wjimt.v8.no2.2024.pg71.89

[33] M. Liu, B. Zhang, W. Chen, and X. Zhang, "A Survey of Exploitation and Detection Methods of XSS Vulnerabilities," IEEE Access, vol. 7, pp. 182004–182016, 2019. doi: https://doi.org/10.1109/ACCESS.2019.2960449

[34] Y. Sadqi and Y. Maleh, "A Systematic Review and Taxonomy of Web Applications Threats," Taylor & Francis, 2022. doi: https://doi.org/10.1080/19393555.2020.1853855

[35] I. Iacob and M. Pirnau, "SQL Injection Attacks and Vulnerabilities," JOURNAL OF INFORMATION SYSTEMS & OPERATIONS MANAGEMENT, Vol. 14.1, May 2020

[36] Sarajaldeen Akram Bahjat Arif, "The Implications for a Hybrid Detection Technique Against Malicious SQL Attacks on Web Applications", April 2025Journal of Information Systems Engineering & Management 10(35s):1101-1109, https://doi.org/10.52783/jisem.v10i35s.6219

[37] U. Sarmah, D. K. Bhattacharyya, and J. K. Kalita, "A Survey of Detection Methods for XSS Attacks," Journal of Network and Computer Applications, Sep. 15, 2018. doi: https://doi.org/10.1016/j.jnca.2018.06.004

[38] Muhammad Naufal Hafizh, Muhammad Naufal Hafizh and Nuril Anwar, "Security Development Life Cycle (SDLC)-Based Approach for Designing Intrusion Detection and Prevention Systems to Counter SQL Injection Attacks at MAN 2 Magetan", March 2025 Mobile and Forensics 7(1):55-68, https://doi.org/10.12928/mf.v7i1.9365

[39] Sayed Mansoor Rahimy, Sayed Hassan Adelyar & Said Rahim Manandoy, "Vulnerabilities That Threaten Web Applications in Afghanistan," Springer Proceedings in Complexity, Feb. 2024. doi: https://doi.org/10.1007/978-981-99-6974-6_12

[40] Nilima D. Bobade, Vinit A. Sinha, and Swati S Sherekar, "A diligent survey of SQL injection attacks, detection and evaluation of mitigation techniques," IEEE Proceedings Article, 2024. doi: https://doi.org/10.1109/sceecs61402.2024.10481914

[41] R. Bakır, et al., "UniEmbed: A Novel Approach to Detect XSS and SQL Injection Attacks Leveraging Multiple Feature Fusion with Machine Learning Techniques," Arabian Journal for Science and Engineering, 2025. doi: https://doi.org/10.1007/s13369-024-09916-4

[42] D. Moreira, J. P. Seara, J. P. Pavia, and C. Serrão, "Intelligent Platform for Automating Vulnerability Detection in Web Applications," Electronics, vol. 14, no. 1, Article 79, 2025. doi: https://doi.org/10.3390/electronics14010079

[43] A. E. Hafez and M. M. Almustafa, "Detecting Security Vulnerabilities in Web Applications: A Proposed System," International Journal of Safety and Security Engineering, vol. 14, no. 6, pp. 1933-1940, Dec. 2024. doi: https://doi.org/10.18280/ijsse.140627

[44] G. Zogaj, et al., "Statistical Analysis of Unique Web Application Vulnerabilities: A Quantitative Assessment of Scanning Tool Efficiency," SEEU Review, vol. 20, issue 1, Jun. 2025, pp. 136-152. doi: https://doi.org/10.2478/seeur-2025-0021

[45] N. Talon, V. T. Tong, G. Guette, Y. Han, and Y. Laarouchi, "SCWAD: Automated Pentesting of Web Applications," in Proc. 21st Int. Conf. Security and Cryptography (SECRYPT 2024), 2024, pp. 424-433. doi: https://doi.org/10.5220/0012721000003767

[46] I. Gusti Agung Surya Pramana Wijaya, Gusti Made Arya Sasmita, I Putu Agus Eka Pratama, "Web Application Penetration Testing on Udayana University's OASE E-learning Platform Using ISSAF and OSSTMM," International Journal of Information Technology and Computer Science (IJITCS), vol. 16, no. 2, pp. 45-56, Apr. 2024. doi: https://doi.org/10.5815/ijitcs.2024.02.04

[47] A. Arvind Kamboj, Chandrashekhar Moharir, Shiva Kiran Lingishetty, "Securing Web Applications Against SQL Injection and XSS Attacks," International Journal of Latest Technology in Engineering, Management & Applied Science, 2025. doi: https://doi.org/10.51583/ijltemas.2025.140500025

[48] M. M. Hossain, M. A. Hasnat, and M. S. Islam, "Enhancing Web Security: A Comprehensive Approach to Detect and Prevent SQL Injection Attacks through Innovative Query Comparison and Encryption Algorithms," Int. J. Sci. Res. Modern Technol., vol. 4, no. 1, pp. 123-133, Feb. 2025. doi: https://doi.org/10.5281/zenodo.14960117

[49] Krishna Chaitanya Chaganti, "AI-Driven SQL Injection Prevention: Strengthening Database Security," International Journal of Science And

ISSN: 2961-5992

Engineering, vol. 11, no. 1, 2025. doi: https://doi.org/10.53555/ephijse.v11i1.285

[50] Shahbaaz Mohammed Hayat Chaki, Mazura Mat Din and Maheyzah Md Siraj, "Integration of SQL Injection Prevention Methods" , November 2019International Journal of Innovative Computing 9(2), https://doi.org/10.11113/ijic.v9n2.232

ISSN: 2961-5992