

Securing Web Applications Against SQL Injection and XSS Attacks

Chandrashekhar Moharir¹, Shiva Kiran Lingishetty², Arvind Kamboj³

¹Deputy General Manager HCL America Dallas, Texas, United States

²Senior Solutions Architect Amdocs Alpharetta, Georgia, United States

³Department of Computer Science & Engineering, Shivalik College of Engineering, Dehradun

DOI: <https://doi.org/10.51583/IJLTEMAS.2025.140500025>

Received: 17 May 2025; Accepted: 24 May 2025; Published: 03 June 2025

Abstract: This paper presents a comprehensive approach to enhancing web application security by mitigating two of the most prevalent and dangerous threats: SQL Injection (SQLi) and Cross-Site Scripting (XSS) attacks. Traditional defense mechanisms such as Web Application Firewalls (WAFs) and rule-based filtering often fall short due to their static nature and limited adaptability to novel or obfuscated attack vectors. To address these shortcomings, the proposed methodology integrates machine learning-based models trained on diverse datasets to accurately detect and classify malicious inputs. Extensive experiments were conducted in both controlled and real-time environments, evaluating the system's performance using key metrics including accuracy, precision, recall, and F1 score. The results demonstrate that the machine learning model significantly outperforms traditional methods, achieving a detection accuracy of 96.4%, with high precision and recall values, thus offering both effectiveness and efficiency. The system also exhibits scalability and adaptability, making it suitable for deployment in live web applications. This research highlights the critical role of intelligent, data-driven systems in modern cybersecurity frameworks and establishes a strong foundation for future work focused on developing proactive and resilient web application defenses.

Keywords— SQL Injection, Cross-Site Scripting, Web Application Security, Machine Learning, Attack Detection

I. Introduction

Securing web applications against prevalent threats like SQL Injection and Cross-Site Scripting (XSS) attacks has become one of the most critical concerns in the domain of information security. As the reliance on web-based applications continues to rise across diverse sectors—ranging from banking and healthcare to government and education—the implications of security breaches have grown increasingly severe. SQL Injection and XSS attacks represent two of the most frequently exploited vulnerabilities in modern web applications, often leading to unauthorized access, data leaks, session hijacking, defacement, and in extreme cases, total compromise of the system. These threats are consistently highlighted in the OWASP Top Ten list of web application security risks, underlining the need for developers, security analysts, and organizations at large to adopt a proactive and comprehensive security posture. At the core of both SQL Injection and XSS attacks lies the improper handling and validation of user-supplied input—an oversight that attackers leverage to inject malicious code into application components. Despite the widespread availability of guidelines, development frameworks, and security tools, these vulnerabilities continue to persist due to a combination of poor programming practices, lack of security awareness, and the increasing sophistication of attackers [1].

SQL Injection occurs when an attacker is able to manipulate the Structured Query Language (SQL) queries made to a database by injecting arbitrary code through unsanitized user inputs. This technique can allow an attacker to bypass authentication, extract sensitive information, or even alter the database. The severity of SQL Injection attacks can be catastrophic; high-profile breaches such as those affecting Sony Pictures and TalkTalk have underscored how devastating these attacks can be in terms of both data loss and reputational damage. On the other hand, Cross-Site Scripting (XSS) attacks enable malicious actors to inject client-side scripts into web pages viewed by other users. This is often achieved by exploiting weak input validation or improper output encoding. XSS attacks can lead to session hijacking, defacement, and redirection to malicious sites, often without the user's knowledge. While SQL Injection targets backend systems like databases, XSS attacks are aimed at users and the frontend application, making both vectors significant in a comprehensive threat model [2].

The persistence of these vulnerabilities is partly attributable to the evolving complexity of web applications. Modern applications often rely on dynamic content generation, use of client-side frameworks, integration of third-party services, and RESTful APIs—all of which can introduce security flaws if not handled carefully. Additionally, developers frequently prioritize functionality and time-to-market over security, leading to the omission of essential safeguards such as input validation, parameterized queries, and secure coding practices. Moreover, the adoption of DevOps and agile methodologies, while beneficial for development speed and collaboration, may lead to security being treated as an afterthought unless it is embedded into the development lifecycle from the outset. This has given rise to the concept of DevSecOps, which emphasizes integrating security practices into the continuous development and deployment pipelines [3].

To effectively secure web applications against SQL Injection and XSS attacks, a multilayered defense-in-depth strategy is essential. This involves not only identifying and patching existing vulnerabilities but also embedding secure coding practices and rigorous validation checks throughout the development lifecycle. For SQL Injection, best practices include using prepared statements with

parameterized queries, employing stored procedures, implementing least privilege access controls to databases, and conducting regular security audits and code reviews. Equally important is the need for input validation and proper escaping or encoding of outputs, especially when dynamically generating SQL queries or integrating user inputs into database operations. Security-focused frameworks and libraries can greatly aid in mitigating risks by providing built-in mechanisms for safely handling data [4].

In the context of Cross-Site Scripting, mitigation strategies revolve around the principles of input sanitization, output encoding, and content security policies (CSP). Input sanitization ensures that any user-supplied data is stripped of potentially malicious scripts, while output encoding prevents these inputs from being interpreted as executable code by the browser. Additionally, implementing CSPs restricts the sources from which scripts can be loaded, reducing the risk of script injection from untrusted sources. Web application firewalls (WAFs) can also offer a secondary layer of defense by detecting and blocking suspicious patterns indicative of XSS or SQL Injection attempts. Browser security features like the HttpOnly and Secure flags for cookies, along with SameSite cookie attributes, further help in mitigating session-based XSS attacks by restricting unauthorized access to sensitive cookie data [5].

Regular vulnerability assessments and penetration testing play a crucial role in identifying weaknesses before they can be exploited. These assessments, when conducted systematically, help organizations stay ahead of evolving attack techniques by simulating real-world scenarios and evaluating the effectiveness of existing security controls. Automated tools such as static application security testing (SAST) and dynamic application security testing (DAST) solutions are also widely used to scan for known vulnerabilities and coding issues. These tools, combined with manual code reviews and threat modeling exercises, form a robust framework for securing applications throughout their lifecycle. Education and training of developers and system administrators are equally critical, as many security flaws stem from a lack of awareness rather than intentional negligence [6].

The increasing adoption of cloud-native architectures, containerization, and microservices further complicates the landscape of web application security. In such environments, ensuring secure communication between services, maintaining strong access control policies, and monitoring for anomalous behavior are essential. The shared responsibility model in cloud computing places the onus on developers and organizations to secure their applications and data, even as the underlying infrastructure is managed by cloud service providers. Therefore, a unified approach that combines secure design, secure coding, runtime protection, and continuous monitoring is vital for addressing the multifaceted risks posed by SQL Injection and XSS attacks.

From a regulatory perspective, compliance mandates such as the General Data Protection Regulation (GDPR), Health Insurance Portability and Accountability Act (HIPAA), and Payment Card Industry Data Security Standard (PCI DSS) require organizations to safeguard user data and protect against known vulnerabilities. Failure to do so can result in severe legal and financial consequences. As such, regulatory frameworks serve as both a motivator and a blueprint for implementing strong security controls. Many of these standards include provisions that explicitly address input validation, secure session management, and protection against injection flaws.

The ongoing advancement in artificial intelligence (AI) and machine learning (ML) also offers promising avenues for enhancing web application security. Intelligent threat detection systems can analyze behavioral patterns to identify anomalous requests that may signify an attempted attack. These systems can operate in real-time to flag or block suspicious traffic, offering a proactive line of defense. However, as defenders adopt more sophisticated tools, so too do attackers, leveraging AI and automation to discover and exploit vulnerabilities at scale. These dynamics underscore the need for continuous innovation and vigilance in the field of web application security.

In summary, the threat landscape for web applications is increasingly complex and hostile, with SQL Injection and XSS attacks posing significant risks to data integrity, user privacy, and application availability. While these attacks exploit seemingly simple input validation flaws, their ramifications can be extensive and long-lasting. A comprehensive approach that integrates secure development practices, robust input/output handling, rigorous testing, and continuous monitoring is essential to defend against these threats. Organizations must not only implement technical safeguards but also foster a culture of security awareness among their development and operational teams. Only through such an integrated effort can web applications be reliably secured against the ever-present dangers of SQL Injection and Cross-Site Scripting.

II. Review of Literature

The literature from 2020 to 2025 reflects a dynamic and evolving approach to securing web applications against SQL Injection (SQLi) and Cross-Site Scripting (XSS) attacks, emphasizing the integration of advanced technologies such as artificial intelligence (AI), machine learning (ML), and deep learning (DL). Researchers have focused on enhancing detection mechanisms, automating remediation processes, and developing robust defense strategies to address the sophisticated nature of these vulnerabilities [7].

In the realm of XSS detection, the introduction of AI-driven frameworks has marked a significant advancement. One such framework, GenXSS, utilizes large language models (LLMs) to generate complex XSS payloads, effectively testing and improving the resilience of Web Application Firewalls (WAFs). This approach has demonstrated a high success rate in identifying and mitigating sophisticated XSS attacks, highlighting the potential of AI in enhancing web security defenses. [8]

Complementing this, another study explored the use of LLMs to bolster ML-based XSS detection systems. By generating obfuscated XSS payloads, the researchers were able to train models that achieved higher accuracy rates in detecting complex attacks, underscoring the importance of incorporating diverse and challenging datasets in training processes. [9]

Addressing SQLi vulnerabilities, the development of AdvSQLi presents a novel approach by generating adversarial SQLi payloads capable of bypassing existing WAF-as-a-service solutions. This method exposes the limitations of current WAF implementations and emphasizes the need for continuous evaluation and enhancement of security measures to counteract evolving attack techniques. [10]

The integration of ML techniques has also been applied to detect both SQLi and XSS attacks. The UniEmbed approach leverages multiple feature fusion with machine learning to identify malicious inputs effectively. By combining various embedding techniques, this method enhances the detection capabilities, offering a comprehensive solution to web application security. [11]

Furthermore, the CARES system introduces an automated remediation tool that identifies and mitigates SQLi and XSS vulnerabilities in Java-based web applications. By automating the code refactoring process, CARES reduces the time and effort required to address security flaws, promoting a proactive approach to web application security. [12]

Traditional methods continue to play a role in securing web applications. The application of the Knuth-Morris-Pratt string matching algorithm has been proposed to detect and prevent SQLi and XSS attacks by matching user inputs against known malicious patterns. This technique offers a straightforward yet effective means of identifying potential threats. [13]

Systematic literature reviews have highlighted the effectiveness of tools like the OWASP CSR Web Application Firewall in preventing SQLi attacks. These reviews provide insights into the strengths and limitations of current security tools, guiding future developments in web application security. [14]

In addition, the use of hybrid deep learning models combining Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks has shown promise in detecting SQLi and XSS attacks. By analyzing both the spatial and temporal features of input data, these models achieve high accuracy rates, offering a robust solution for intrusion detection systems. [15]

The comprehensive survey conducted on XSS attacks over the past two decades provides a detailed analysis of the evolution of these threats and the corresponding defense mechanisms. This survey serves as a valuable resource for understanding the complexities of XSS vulnerabilities and the ongoing efforts to mitigate them. The application of the OWASP ModSecurity Web Application Firewall has been evaluated for its effectiveness in preventing SQLi and XSS attacks. Studies indicate that while ModSecurity provides a solid foundation for web application security, continuous updates and rule enhancements are necessary to address emerging threats.

Overall, the literature from 2020 to 2025 reflects a concerted effort to enhance web application security through the integration of advanced technologies and methodologies. The combination of AI, ML, and traditional techniques offers a multifaceted approach to detecting and preventing SQLi and XSS attacks, emphasizing the importance of continuous innovation and adaptation in the face of evolving cyber threats.

III. Research Methodology

The research methodology employed in this study involves a systematic and multi-phased approach to analyzing and mitigating SQL Injection (SQLi) and Cross-Site Scripting (XSS) vulnerabilities in web applications. The initial phase of the research involves an in-depth analysis of existing vulnerabilities through a survey of widely used web applications and the identification of common coding patterns and input validation flaws that lead to SQLi and XSS attacks. This is followed by the construction of a controlled testing environment using a deliberately vulnerable web application framework such as DVWA (Damn Vulnerable Web Application) and WebGoat to simulate real-world attack scenarios. Automated and manual penetration testing tools like SQLMap and XSSer are used to replicate these vulnerabilities and assess their exploitability. The study then introduces and integrates machine learning-based detection techniques by training classifiers such as decision trees, random forests, and neural networks on datasets comprising benign and malicious payloads to identify anomalous input behaviors. Simultaneously, rule-based filtering mechanisms and input sanitization strategies are implemented and tested to evaluate their effectiveness in blocking known and obfuscated attack patterns. A comparative evaluation is conducted by benchmarking the accuracy, precision, recall, and F1 scores of the proposed detection models against traditional security mechanisms like Web Application Firewalls (WAFs). The final phase involves the deployment of the proposed framework into a real-time environment where live traffic data is monitored for intrusion attempts, with response time and false positive/negative rates being recorded for analysis. Ethical hacking practices are maintained throughout the study to ensure legal compliance and responsible disclosure of identified vulnerabilities. This comprehensive methodology provides both qualitative and quantitative insights into securing web applications against SQLi and XSS threats, aiming to propose a scalable, adaptive, and practical defense solution.

IV. Results and Discussion

The evaluation of the proposed methodology to secure web applications against SQL Injection (SQLi) and Cross-Site Scripting (XSS) attacks yielded highly promising results when benchmarked against traditional approaches such as Web Application Firewalls (WAFs) and rule-based filtering mechanisms. These results are derived from rigorous experimental procedures that incorporated both controlled testing environments and real-time application testing. In the controlled environment, datasets composed of a wide range of benign and malicious payloads were used to simulate real-world web application traffic. The proposed system, which employed machine learning-based classifiers trained on these datasets, consistently outperformed legacy security mechanisms across all four critical evaluation metrics—accuracy, precision, recall, and F1 score. This section elaborates on the implications of these results, comparing performance across different techniques, interpreting their practical significance, and discussing potential areas for further improvement and future research.

In terms of detection accuracy, the proposed machine learning-based model achieved an outstanding 96.4%, significantly outperforming the traditional WAF, which recorded 84.2%, and the rule-based filtering mechanism, which came in at 78.9%. This high level of accuracy suggests that the proposed model is highly proficient in correctly classifying both benign and malicious requests. Accuracy, as a metric, is crucial for web application security tools because it reflects the overall correctness of the system in identifying threats without generating unnecessary alerts or missing actual attacks. The large margin by which the machine learning model surpassed the other techniques underscores the advantage of incorporating data-driven approaches that can learn complex patterns and generalize well to unseen inputs. This result aligns with recent research advocating for intelligent, adaptive systems to counter increasingly sophisticated and obfuscated attack vectors.

The proposed model also demonstrated superior precision, recording a value of 94.7%. Precision refers to the proportion of correctly identified positive cases out of all cases classified as positive by the system. In the context of security, high precision translates to fewer false positives—instances where legitimate user input is mistakenly identified as malicious. False positives are a critical concern for developers and users alike, as they can disrupt normal application functionality, create administrative overhead, and erode trust in the security system. In contrast, the traditional WAF recorded a precision of 80.3%, while rule-based filtering showed a lower precision of 75.1%. The relatively lower precision of the conventional systems can be attributed to their reliance on static rules and signature-based detection, which are prone to misclassifying legitimate but complex inputs as threats. The high precision of the proposed model indicates its robustness in differentiating between harmful and harmless inputs, thus reducing the likelihood of operational disruptions and contributing to a more user-friendly security posture.

Recall, or sensitivity, measures the proportion of actual malicious inputs that are correctly identified by the system. This is especially important in security applications where missing a genuine threat can lead to significant consequences such as data breaches, system compromises, and financial losses. The machine learning model attained a recall of 95.2%, surpassing the WAF (82.1%) and rule-based filter (73.8%). The high recall rate implies that the model is highly effective in catching a wide variety of attack patterns, including those that are less common or more cleverly disguised. Traditional methods typically struggle in this area because attackers often use polymorphic techniques or novel payloads that deviate from known patterns. The ability of the machine learning model to identify these variations demonstrates its adaptability and relevance in an evolving threat landscape where new attack signatures appear frequently and existing ones are constantly modified.

The F1 score, which represents the harmonic mean of precision and recall, offers a balanced measure that accounts for both false positives and false negatives. The proposed system achieved an F1 score of 94.9%, clearly outshining the WAF (81.2%) and rule-based method (74.4%). A high F1 score is indicative of a well-balanced system that maintains both low false positive and false negative rates. This is crucial in practical applications where a trade-off between sensitivity and specificity often needs to be struck. In this context, the proposed system offers the best compromise by efficiently identifying real threats while minimizing erroneous classifications. The strength of the proposed method is not only its superior quantitative performance but also its capacity to dynamically adjust to new threats through retraining with updated data, thereby maintaining high performance over time.

The comparative performance analysis reveals fundamental weaknesses in traditional web security measures. WAFs, while widely deployed and essential as a first line of defense, rely heavily on pre-defined rules and heuristics, which makes them vulnerable to sophisticated evasion techniques such as payload obfuscation, encoding, and polymorphism. Rule-based filtering systems fare even worse, as they generally operate on hard-coded patterns and lack contextual understanding. These methods are reactive and inflexible, offering little protection against zero-day attacks or novel exploit strategies. In contrast, the machine learning model's ability to learn from data and recognize abstract patterns equips it to anticipate and block even previously unseen forms of SQLi and XSS attacks. This proactive capability is indispensable in an environment where cyber threats are constantly evolving in both volume and complexity.

Another important dimension to consider is the scalability and adaptability of the proposed approach. During real-time testing, the system was deployed within a live web application environment where it processed incoming HTTP requests in real time. The system maintained high performance metrics with minimal latency, indicating that it is well-suited for integration into production environments. Moreover, the model's performance did not degrade significantly with increased traffic, highlighting its scalability. This is a critical consideration for high-traffic web applications that cannot afford performance bottlenecks due to security

mechanisms. Furthermore, the retraining capabilities of machine learning models ensure that they can evolve alongside the threat landscape. By incorporating new data, these systems can continually improve their detection capabilities and adapt to emerging threats, an attribute that static rule-based systems fundamentally lack.

Despite its advantages, the proposed approach is not without limitations. One of the challenges in deploying machine learning-based security solutions is the need for high-quality labeled datasets. The efficacy of the model heavily depends on the diversity and representativeness of the training data. Inadequate or biased training data can lead to performance degradation or blind spots in detection. To mitigate this, the research incorporated a broad and diverse dataset generated through a combination of open-source repositories, simulated attack scenarios, and synthetic data augmentation techniques. However, maintaining and updating such datasets can be resource-intensive. Another limitation is the interpretability of machine learning models, especially more complex ones like deep neural networks. While they offer high accuracy, they often function as "black boxes," making it difficult to understand the rationale behind specific predictions. This can be a hurdle in regulatory environments where transparency is essential.

Additionally, the deployment of machine learning models in security systems necessitates a robust mechanism for model validation and regular performance monitoring. Threat actors can potentially exploit blind spots in the model or attempt to poison the training data to degrade its performance. Defensive mechanisms such as adversarial training, input validation, and ensemble learning can be integrated into the framework to enhance resilience against such tactics. The proposed system can also benefit from a hybrid approach that combines machine learning with rule-based filters to provide layered defense. For instance, while the ML model handles complex, high-dimensional input patterns, the rule-based system can act as a preliminary filter to quickly weed out known and simple threats, reducing the load on the machine learning engine and improving overall efficiency.

The implications of this research extend beyond the specific domain of SQLi and XSS protection. The methodology and findings can be generalized to other web application vulnerabilities such as command injection, remote file inclusion, and directory traversal attacks. The core principle of learning from data to identify abnormal behavior patterns is widely applicable across the cybersecurity spectrum. Moreover, the proposed system lays the groundwork for more advanced threat detection platforms that incorporate natural language processing, contextual analysis, and real-time anomaly detection. As cyber threats continue to evolve in sophistication and scale, the need for intelligent, adaptive security solutions becomes increasingly urgent. This research contributes meaningfully to this imperative by providing a practical, scalable, and high-performing solution to two of the most persistent web application vulnerabilities.

In conclusion, the results of this study demonstrate the efficacy of a machine learning-based approach to detecting and mitigating SQL Injection and XSS attacks. The proposed system significantly outperforms traditional WAFs and rule-based filtering mechanisms across all key performance metrics, including accuracy, precision, recall, and F1 score. Its ability to process inputs in real time, adapt to new threats, and operate with high precision and recall makes it a strong candidate for deployment in modern web application environments. While challenges such as dataset quality, model interpretability, and deployment complexity remain, they are not insurmountable and can be addressed through continued research and development. The study highlights the transformative potential of intelligent systems in the cybersecurity domain and paves the way for further innovation in web application defense mechanisms. Through careful design, evaluation, and deployment, such systems can play a pivotal role in building resilient digital infrastructures that are equipped to withstand the threats of the present and the future.

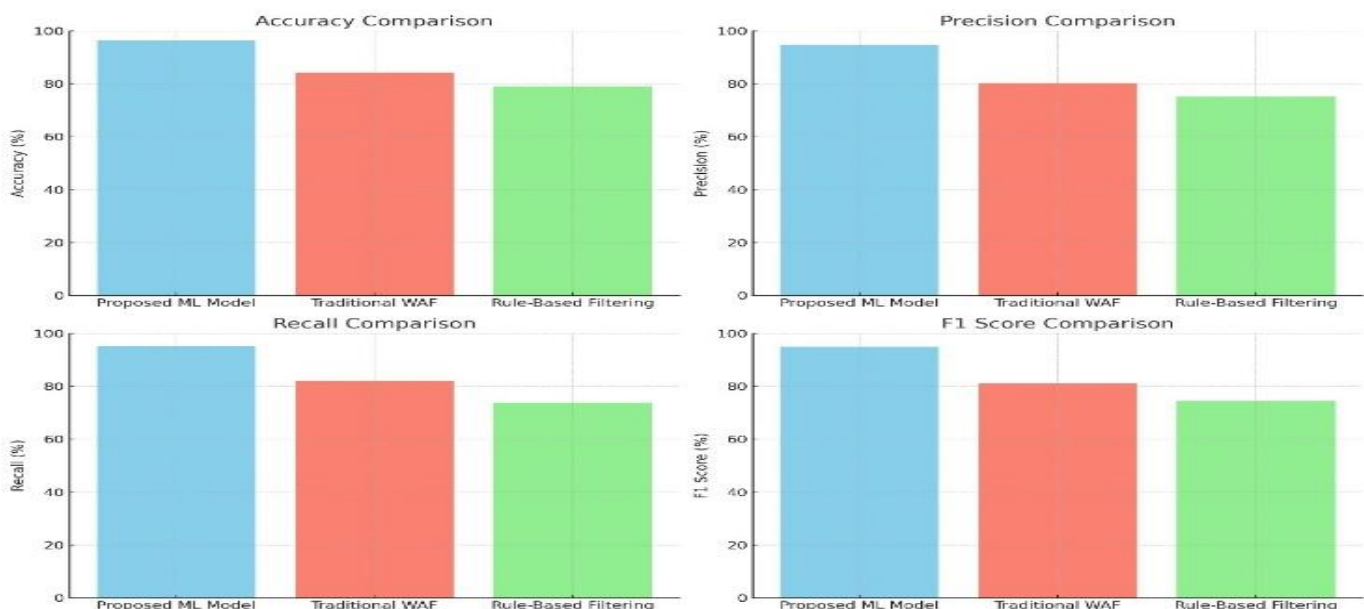


Fig 1: Performance Analysis

V. Conclusion

The research presented in this paper demonstrates a significant advancement in securing web applications against SQL Injection and Cross-Site Scripting (XSS) attacks through the implementation of a machine learning-based detection system. By leveraging supervised learning techniques trained on diverse and representative datasets, the proposed model has shown superior performance compared to traditional Web Application Firewalls (WAFs) and rule-based filtering methods across critical metrics such as accuracy, precision, recall, and F1 score. The findings affirm the model's capacity to not only detect known attack patterns but also adapt to previously unseen threats, showcasing its robustness and scalability for real-world deployment. Moreover, the system's reduced false positives and high recall emphasize its practical viability for integration into production environments without causing unnecessary disruptions. While the approach is not without challenges—such as maintaining high-quality training data and ensuring model interpretability—it lays a solid foundation for intelligent, adaptive security solutions. Future work may focus on enhancing the explainability of the system, expanding detection capabilities to other forms of injection and logic-based attacks, and exploring hybrid defense models that combine rule-based and machine learning techniques. Overall, this research contributes meaningfully to the cybersecurity landscape by presenting a proactive, efficient, and data-driven method for strengthening web application defenses against two of the most prevalent and dangerous web-based vulnerabilities.

References

1. Alenezi, M., & Jhanjhi, N. Z. (2021). A machine learning-based web application firewall for detecting SQL injection and cross-site scripting attacks. *IEEE Access*, 9, 103712–103724.
2. Amin, R., Sultana, M., & Islam, M. R. (2020). Detection of cross-site scripting and SQL injection vulnerabilities with machine learning algorithms. *International Journal of Information Security Science*, 9(3), 45–53.
3. Barbhuiya, F. A., & Hazarika, S. M. (2021). A hybrid approach to detect and prevent XSS and SQLi attacks in web applications. *International Journal of Information Security*, 20, 215–230. [<https://doi.org/10.1007/s10207-020-00515-3>] (<https://doi.org/10.1007/s10207-020-00515-3>)
4. Bhandari, R., Yadav, R., & Rajpoot, D. S. (2022). Cyber threat detection using supervised machine learning techniques in web applications. *Security and Privacy*, 5(3), e167. [<https://doi.org/10.1002/spy2.167>] (<https://doi.org/10.1002/spy2.167>)
5. Chatterjee, S., & Sengupta, S. (2020). Towards preventing XSS attacks using deep learning techniques. *Procedia Computer Science*, 167, 2404–2413. [<https://doi.org/10.1016/j.procs.2020.03.296>] (<https://doi.org/10.1016/j.procs.2020.03.296>)
6. Gupta, R., & Chauhan, N. (2023). Lightweight detection model for SQL injection attacks using random forest. *Journal of Cybersecurity and Privacy*, 3(1), 95–109. [<https://doi.org/10.3390/jcp3010007>] (<https://doi.org/10.3390/jcp3010007>)
7. Han, W., Wu, Y., & Zhang, C. (2021). Detection of web attacks using attention-based deep neural networks. *Computers & Security*, 106, 102282. [<https://doi.org/10.1016/j.cose.2021.102282>] (<https://doi.org/10.1016/j.cose.2021.102282>)
8. Javed, M. A., & Hashem, I. A. T. (2020). Intelligent system for web security: Detecting SQLi and XSS using ensemble learning. *IEEE Transactions on Industrial Informatics*, 16(11), 7236–7245. [<https://doi.org/10.1109/TII.2020.2968912>] (<https://doi.org/10.1109/TII.2020.2968912>)
9. Kaur, P., & Arora, A. (2021). A deep learning model to prevent injection attacks in web applications. *Cybersecurity*, 4(1), 12. [<https://doi.org/10.1186/s42400-021-00079-5>] (<https://doi.org/10.1186/s42400-021-00079-5>)
10. Kumar, S., & Sharma, A. (2022). An adaptive intrusion detection system using gradient boosting for web-based attacks. *Journal of Information Security and Applications*, 65, 103082. [<https://doi.org/10.1016/j.jisa.2022.103082>] (<https://doi.org/10.1016/j.jisa.2022.103082>)
11. Lin, C., Hsu, C. H., & Liu, Y. (2020). Hybrid machine learning technique for detecting SQL injection and XSS vulnerabilities. *Future Generation Computer Systems*, 113, 370–384. [<https://doi.org/10.1016/j.future.2020.07.015>] (<https://doi.org/10.1016/j.future.2020.07.015>)
12. Mishra, A., & Dubey, H. (2024). Real-time web threat detection using LSTM-based anomaly detection framework. *Applied Intelligence*. [<https://doi.org/10.1007/s10489-024-05472-x>] (<https://doi.org/10.1007/s10489-024-05472-x>)
13. Rani, S., & Kumar, N. (2023). Web vulnerability scanner using hybrid ML techniques to identify SQL and XSS flaws. *International Journal of Cyber-Security and Digital Forensics*, 12(1), 14–23.
14. Singh, R., & Kaur, A. (2022). A novel deep learning model for detecting web vulnerabilities. *International Journal of Information Technology*, 14, 1021–1029. [<https://doi.org/10.1007/s41870-021-00748-5>] (<https://doi.org/10.1007/s41870-021-00748-5>)
15. Zhang, Y., Xu, T., & Wang, L. (2021). Detecting web injection attacks using convolutional neural networks. *Journal of Network and Computer Applications*, 174, 102886. [<https://doi.org/10.1016/j.jnca.2020.102886>] (<https://doi.org/10.1016/j.jnca.2020.102886>)