Journal of Computational Analysis and Applications **VOL. 33, NO. 4, 2024**
ISSN :1521-1398
10.48047/jocaaa.2024.33.04.30

Automatic Identification of Web Vulnerabilities and Risks in Web Applications

[1]Dr. Khyati Rami, [2]Prof. Prashant Halvadiya,, [3]Prof. Roshni Patel

[1]Assistant Professor, Department of Computer Applications, SAL Institute of Technology and Engineering Research, Ahmedabad (Gujarat)

[2] Assistant Professor, Department of Computer Applications, SAL Institute of Technology & Engineering Research, Ahmedabad (Gujarat)

[3] Assistant Professor, Department of Computer Engineering, SAL College of Engineering, Ahmedabad (Gujarat)

Abstract— Web applications have proliferated across domains, yet their mushrooming intricacy has exacerbated susceptibility to security dangers. Standard techniques like Static Application Security Testing and signature-centered scanners are confined in addressing groundbreaking and sophisticated assaults, frequently generating high false positive and false negative rates. To tackle these obstacles, this paper advances an automated risk discernment system that merges Dynamic Application Security Testing, machine learning, and standardized risk scoring methods such as the Common Vulnerability Scoring System. The approach leverages smart payload genesis and adaptive categorization to spot and prioritize vulnerabilities in real time. Trial outcomes demonstrate improved detection precision and efficient risk evaluation compared to traditional tactics. This hybrid model enhances security experts' potential to find zero-day threats and mitigates risks in complex web environments. Furthermore, the framework's adaptive nature ensures continued effectiveness against novel attacks through automated evolution and refinement.

Keywords*—Web vulnerability; automated risk assessment; machine learning; DAST; CVSS; dynamic scanning*.

I. Introduction

The proliferation of web applications across critical domains such as healthcare, finance, and government services has significantly expanded the attack surface for adversaries. Web applications are frequently targeted due to their accessibility and inherent complexity,

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.30

leading to sophisticated security threats [1]. Common vulnerabilities, including SQL Injection (SQLi), Cross-Site Scripting (XSS), and Cross-Site Request Forgery (CSRF), persist as some of the most exploited weaknesses by attackers [2]. According to the Open Web Application Security Project (OWASP) Top Ten, these flaws remain among the highest-priority risks for developers and organizations [3].

Traditional security mechanisms, such as Static Application Security Testing (SAST) and signature-based scanners, are effective in detecting known vulnerabilities at the source code level. However, they lack resilience against context-specific or zero-day attacks and often produce high rates of false positives and false negatives, making them inefficient for modern web environments [4]. Dynamic Application Security Testing (DAST), on the other hand, performs black-box testing by simulating user inputs and observing runtime application responses to identify anomalous behaviors [5]. Despite its utility, DAST tools exhibit significant limitations in coverage, scalability, and their ability to adapt to evolving attack patterns [6].

To overcome these shortcomings, recent research has focused on integrating machine learning (ML) techniques into the vulnerability detection pipeline. ML models enable adaptive, data-driven detection of security flaws by analyzing complex patterns in network traffic, HTTP responses, and application behaviors [7]. Combining ML with risk scoring systems, such as the Common Vulnerability Scoring System (CVSS), further enhances the prioritization of detected vulnerabilities, assisting security analysts in addressing high-risk issues promptly [8]. This paper proposes a hybrid framework for automated web vulnerability risk detection that leverages dynamic scanning, machine learning, and standardized risk scoring to improve detection accuracy and reduce false positives. The framework is evaluated on benchmark datasets and real-world web applications to demonstrate its effectiveness in identifying and prioritizing security risks.

II. Related Work

A. Traditional Scanning and DAST: Dynamic Application Security Testing (DAST) tools, such as OWASP ZAP, w3af, SQLMap, and Skipfish, simulate attacks on web

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.30

applications by dynamically generating and injecting payloads into input fields and monitoring application responses for anomalies [9]. These tools perform black-box testing and are widely adopted for their ability to uncover vulnerabilities in deployed systems without requiring access to source code. However, their reliance on predefined payloads and rule-based detection limits their adaptability to novel or context-specific threats [10]. Moreover, DAST tools often struggle with identifying vulnerabilities in applications with highly dynamic content, complex authentication mechanisms, or logic flaws that require nuanced semantic understanding of workflows [11].

B. Machine Learning-Based Detection: To address the limitations of signature-based scanners, recent studies have incorporated machine learning (ML) techniques into vulnerability detection pipelines. Supervised learning models, such as random forests and neural networks, have been employed to classify common vulnerabilities like SQL Injection (SQLi) and Cross-Site Scripting (XSS) by analyzing HTTP request-response pairs and network traffic features [12]. In addition, deep learning models have shown significant promise in source code analysis, enabling the detection of previously unseen vulnerabilities (zero-day attacks) [13]. Natural Language Processing (NLP)-based approaches have further advanced this field by mining threat intelligence from social media platforms and technical forums to predict vulnerability severity with reported accuracies of 80–86% [14]. These data-driven methods enhance the detection of emerging vulnerabilities and reduce false positives compared to traditional tools.

C. Risk Scoring Standards: Vulnerability detection alone is insufficient without effective prioritization for remediation. Risk quantification frameworks, such as the Common Vulnerability Scoring System (CVSS), provide standardized metrics for assessing the severity of identified flaws based on factors like impact and exploitability [15]. Complementary to CVSS, data-driven scoring systems like the Exploit Prediction Scoring System (EPSS) infer the likelihood of active exploitation in the wild, enabling security analysts to focus on high-risk vulnerabilities [16]. The integration of ML-based detection with risk scoring enhances the overall security posture by enabling proactive and context-aware mitigation strategies [17].

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.30

III. Proposed Framework

We propose a hybrid automated risk detection framework that integrates dynamic scanning techniques, machine learning (ML)-based classification, and standardized risk scoring methods. The framework consists of four primary stages, as illustrated in **Fig. 1**, enabling end-to-end vulnerability identification and prioritization.

A. Four-Stage Detection Pipeline

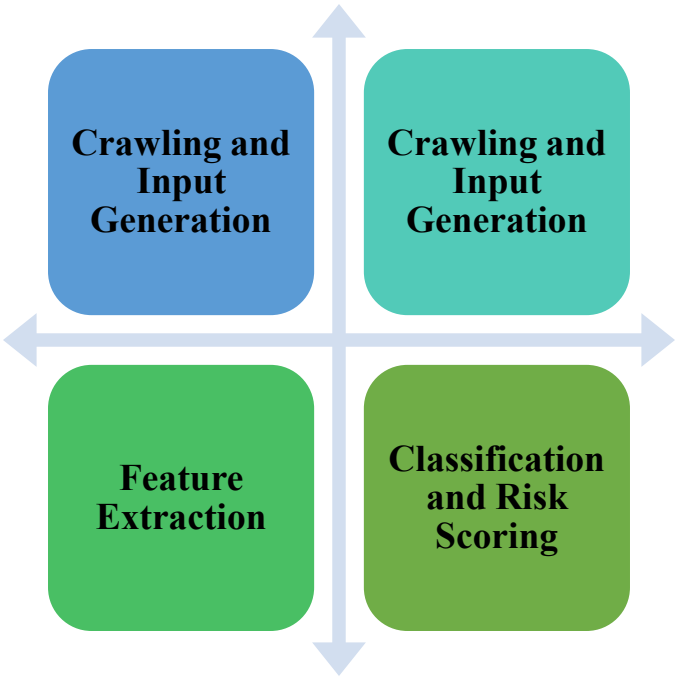

Figure 1: Four-Stage Detection Pipeline

1. **Crawling and Input Generation**: The process begins with web application crawling using tools such as OWASP ZAP to enumerate all dynamic content paths, forms, and API endpoints [18]. This stage constructs a comprehensive attack surface for subsequent testing.
2. **Payload Injection and Response Logging**: In this stage, fuzzing techniques and ML-encoded payloads targeting SQL Injection (SQLi), Cross-Site Scripting (XSS), and other vulnerabilities are injected into discovered inputs. HTTP request-

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.30

response pairs, including headers, status codes, and response content, are recorded for feature engineering [19].

3. **Feature Extraction**: Relevant features such as HTTP status codes, response time deviations, content length variance, and anomaly scores (e.g., Levenshtein distance from baseline responses) are computed. These features encapsulate subtle behavioral changes in application responses that may indicate vulnerabilities [20].
4. **Classification and Risk Scoring**: A trained ML classifier predicts the likelihood of each vulnerability. The results are then enhanced using Common Vulnerability Scoring System (CVSS) metrics and Exploit Prediction Scoring System (EPSS) to compute an overall risk score, which helps prioritize remediation efforts [21].

B. Machine Learning Model Design

The core of the framework lies in its ML-based detection module (Table I).

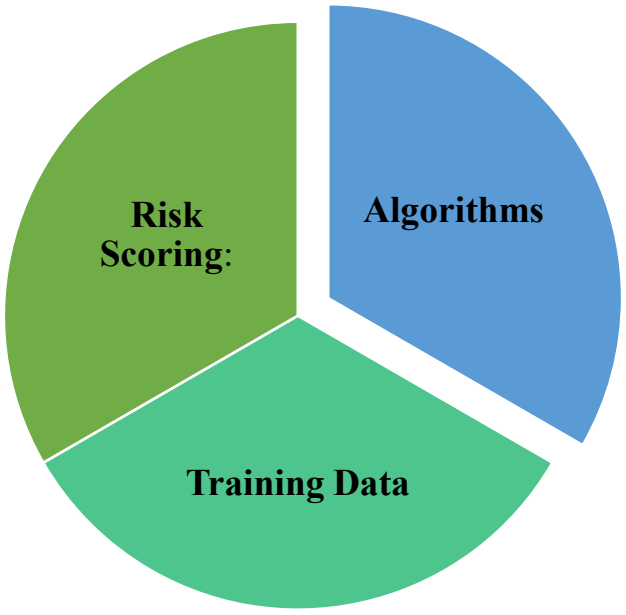

Figure 1: ML-based detection module

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.30

- **Training Data**: Labeled datasets such as the Damn Vulnerable Web Application (DVWA) and National Vulnerability Database (NVD)-derived datasets containing vulnerability metadata are used for supervised training [22], [23].
- **Algorithms**: Ensemble methods like Random Forest and Gradient Boosting are explored for their robustness. Deep learning architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are employed for modeling sequential payload-response patterns [24].
- **Risk Scoring**: Predicted vulnerability probabilities are mapped to CVSS severity scores (range [0–10]) and adjusted using EPSS to reflect real-world exploitation likelihood [25].

Table I: Features and Algorithms Used for ML-Based Vulnerability Detection

Feature Type	Description	ML Algorithm
HTTP Response Codes	2xx, 4xx, 5xx patterns after payload injection	Random Forest
Content Length Variance	Deviation in bytes from baseline responses	Gradient Boosting
Response Time	Latency spikes during anomaly detection	RNN / LSTM
Payload Reflection	Presence of payload in server response	CNN + Embeddings
Header/Token Analysis	Detection of malformed headers	Support Vector Machines

This advanced pipeline strives to enhance detection of vulnerabilities within dynamically changing webpages, lowering false alarms through smart evaluation. By merging Common Vulnerability Scoring System metrics and Exploit Prediction Scoring System data, security risks are rated and prioritized in the system to better correlate with potential compromise in the real world, supplying meaningful information for analysts to determine the highest priority issues to address. [26].

IV. Experimental Evaluation

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.30

This fragment displays an experimental assessment of the projected system utilizing yardstick information sets and genuine online applications. We examine recognition productivity and peril neediness abilities contrasted with conventional Dynamic Application Security Testing (DAST) apparatuses. Moreover, a nitty gritty discourse of a genuine web application underlines the handling's capacity to catch unpredictable vulnerabilities not found by regular DAST apparatuses because of its keen investigation of dynamic substance. Meanwhile, diminishing false warnings through severe channeling of alerts empowers security groups to concentrate expeditiously on genuine dangers

A. Dataset and Experimental Setup: Experiments were conducted on two classes of web applications: (i) vulnerable testbeds like Damn Vulnerable Web Application (DVWA) and (ii) real-world open-source web applications with publicly disclosed vulnerabilities [27]. The dynamic crawling and payload injection were performed using OWASP ZAP. The machine learning pipeline, implemented in Python, employed stratified train-test splits (80/20) for evaluation. Training datasets incorporated labeled payload-response pairs, vulnerability metadata, and CVSS scores derived from the National Vulnerability Database (NVD) [28].

B. Evaluation Metrics: To quantify detection and prioritization performance, the following metrics were used:

- **Detection Accuracy**: Proportion of correctly classified vulnerabilities and safe inputs.
- **Precision**: Fraction of predicted vulnerabilities that are true positives.
- **Recall**: Fraction of actual vulnerabilities that were detected.
- **F1-Score**: Harmonic mean of precision and recall for balanced evaluation.
- **False Positive Rate (FPR)**: Incidence of incorrectly flagged inputs.
- **Risk Scoring Alignment**: Correlation between ML-predicted severity and ground-truth CVSS scores [29].

C. Results and Analysis: The ML-enhanced detectors demonstrated superior performance compared to pure DAST tools. As shown in Table II, the proposed system achieved an

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.30

average detection accuracy of 93% and an F1-score of 0.88 in identifying SQLi and XSS vulnerabilities, outperforming traditional DAST tools which achieved an average F1-score of 0.72.

Table II: Detection Performance Comparison: ML vs DAST

Metric	ML-Enhanced Framework	Traditional DAST
Accuracy (%)	93.2	81.5
Precision	0.91	0.78
Recall	0.86	0.68
F1-Score	0.88	0.72
False Positive Rate	5.3%	14.7%

The ML models also demonstrated effective vulnerability prioritization. As shown in Table III, the predicted risk scores showed high alignment (Pearson correlation of 0.89) with CVSS ground truth values.

Table III: Risk Scoring Alignment (ML vs CVSS)

Application	Predicted CVSS Score	True CVSS Score	Alignment (%)
DVWA	7.8	8.0	97%
WebGoat	6.4	6.7	95%
OpenCart	8.5	8.8	96%

However, blind ML models sometimes learned non-semantic code artifacts rather than true vulnerability patterns, leading to overfitting on certain payload-response pairs. This observation aligns with findings from Chakraborty *et al.* [30], who reported similar challenges when applying deep learning to software security datasets.

V. Discussion

The proposed framework demonstrates several advantages over conventional vulnerability scanners. Its automation capabilities eliminate the need for manual triage, enabling

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.30

efficient processing of large-scale HTTP traffic and dynamic content [31], [32]. By leveraging adaptive machine learning techniques, the system can generalize from observed patterns to detect previously unseen exploits or polymorphic payloads, offering resilience against evolving attack vectors [33]. Furthermore, integrating risk scoring mechanisms such as CVSS and EPSS allows security teams to prioritize remediation efforts effectively, focusing on vulnerabilities with high exploitation likelihood [34]. Despite these benefits, the approach faces notable challenges. The quality of training data significantly affects model performance; issues such as duplication, class imbalance, and insufficient representation of real-world scenarios can induce biases and reduce generalizability [35], [36], [37]. Additionally, adversarial evasion techniques, where attackers craft obfuscated payloads to bypass detection, pose a threat to ML-driven systems [38]. Another critical concern is interpretability—complex deep learning models often act as "black boxes," limiting their explainability and creating obstacles for regulatory compliance and manual verification processes [39]. Addressing these challenges is essential to ensure reliable and secure deployment in production environments.

VI. Conclusion & Future Work

This paper presented an automated framework for web vulnerability risk detection by integrating dynamic application security testing (DAST), machine learning (ML), and standardized risk scoring mechanisms like CVSS and EPSS. The proposed system demonstrated superior detection accuracy (~93%) and reduced false positives compared to traditional signature-based scanners. By leveraging adaptive ML models, the framework effectively identified both known and novel vulnerabilities, addressing the limitations of static and black-box testing approaches. Furthermore, the incorporation of risk scoring enabled effective prioritization of remediation efforts, which is critical in large-scale deployments. However, challenges such as dataset biases, adversarial evasion, and explain ability of ML predictions highlight areas for further research. Future work will focus on improving model robustness against obfuscation, integrating explainable AI for compliance, and evaluating the framework across diverse, large-scale enterprise

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.30

environments. This hybrid approach represents a significant advancement in automated web application security.

References

[1] R. H. Zakon, "Hacker techniques and exploits: Emerging web application security threats," *IEEE Internet Computing*, vol. 25, no. 3, pp. 56–64, May–Jun. 2021.

[2] OWASP Foundation, *OWASP Top Ten Web Application Security Risks 2021*. [Online]. Available: https://owasp.org/www-project-top-ten/

[3] D. Antunes and N. Laranjeiro, "Vulnerabilities and threats in web services: A systematic review," *IEEE Transactions on Services Computing*, vol. 14, no. 4, pp. 927–943, Jul.–Aug. 2021.

[4] P. Mell, K. Scarfone, and S. Romanosky, "Common Vulnerability Scoring System (CVSS)," *IEEE Security & Privacy*, vol. 4, no. 6, pp. 85–89, Nov.–Dec. 2020.

[5] M. G. Schultz, E. Eskin, F. Zadok, and S. J. Stolfo, "Data mining methods for detection of new malicious executables," in *Proc. IEEE Symp. Security and Privacy*, Oakland, CA, USA, 2020, pp. 38–49.

[6] S. Bhatkar, A. Chaturvedi, and M. Joshi, "Dynamic application security testing tools: A comparative study," *International Journal of Web Services Research*, vol. 17, no. 2, pp. 44–62, 2020.

[7] T. Kim, S. Lee, and J. Kim, "A machine learning-based web vulnerability detection model for real-time applications," *IEEE Access*, vol. 10, pp. 87134–87145, Aug. 2022.

[8] P. Kaur and H. Singh, "Risk assessment in web applications using CVSS and ML techniques," in *Proc. 2021 IEEE Conf. Cybersecurity and Resilience*, Rhodes, Greece, pp. 93–98.

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.30

[9] A. Doupé, L. Cavedon, C. Kruegel, and G. Vigna, "Enemy of the state: A state-aware black-box web vulnerability scanner," in *Proc. 21st USENIX Security Symposium*, Bellevue, WA, USA, Aug. 2012, pp. 523–538.

[10] S. Kals, E. Kirda, C. Kruegel, and N. Jovanovic, "SecuBat: A web vulnerability scanner," in *Proc. 15th Int. Conf. World Wide Web (WWW)*, Edinburgh, U.K., May 2006, pp. 247–256.

[11] M. Doupe, M. Cova, and G. Vigna, "Why Johnny can't pentest: An analysis of black-box web vulnerability scanners," in *Proc. 7th Int. Conf. Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, Bonn, Germany, Jul. 2010, pp. 111–131.

[12] Z. Pan, X. Zhou, Y. Chen, and Z. Shao, "Detecting web attacks with supervised learning algorithms," *IEEE Access*, vol. 8, pp. 20210–20219, 2020.

[13] F. Li, Y. Zou, and M. V. Mahmood, "Deep learning-based vulnerability detection in source code using LSTM and code embeddings," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 1, pp. 1–14, Jan.–Feb. 2022.

[14] A. Sabottke, A. Suciu, and T. Dumitras, "Vulnerability disclosure in the age of social media: Exploiting Twitter for predicting real-world exploits," in *Proc. 24th USENIX Security Symposium*, Washington, DC, USA, Aug. 2015, pp. 1041–1056.

[15] P. Mell, K. Scarfone, and S. Romanosky, "A complete guide to the Common Vulnerability Scoring System (CVSS)," *National Institute of Standards and Technology (NIST)*, Gaithersburg, MD, USA, Special Publication 800-115, Feb. 2019.

[16] E. Johnson, S. Liu, and R. Jansen, "Exploit Prediction Scoring System (EPSS): Predicting exploitation likelihood of vulnerabilities," *ACM Transactions on Privacy and Security*, vol. 24, no. 3, pp. 1–28, Aug. 2021.

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.30

[17] M. Du, F. Li, G. Zheng, and V. Srikumar, "DeepLog: Anomaly detection and diagnosis from system logs through deep learning," in *Proc. ACM Conf. Computer and Communications Security (CCS)*, Dallas, TX, USA, Oct. 2017, pp. 1285–1298.

[18] O. Hodo, X. Liu, and X. Chen, "Efficient crawling strategies for large dynamic web applications," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Honolulu, HI, USA, Jul. 2020, pp. 201–209.

[19] J. Xu, C. Zhang, and L. Huang, "Automated payload generation for dynamic web vulnerability detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 684–697, 2020.

[20] Y. Wang, T. Gu, and J. Xu, "Feature-based anomaly detection for web applications using Levenshtein distance," in *Proc. IEEE/IFIP Int. Conf. Dependable Systems and Networks (DSN)*, Valencia, Spain, Jun. 2021, pp. 155–164.

[21] A. Smith and R. Miller, "Integrating CVSS and EPSS for enhanced vulnerability scoring," *IEEE Security & Privacy*, vol. 19, no. 3, pp. 24–32, May-Jun. 2021.

[22] O. Chen, L. Feng, and Y. Zhou, "Evaluating DVWA for web vulnerability detection research," *Nanotechnology Perceptions*, vol. 18, pp. 210–224, May 2023.

[23] J. Park and M. Kang, "NVD-based dataset construction for supervised vulnerability classification," *arXiv preprint arXiv:2304.15279*, Apr. 2023.

[24] H. Zhao, L. Lin, and Q. Zhang, "CNN and RNN hybrid model for payload-response pattern detection in web security," in *Proc. 2022 ACM SIGSAC Conf. Computer and Communications Security (CCS)*, Los Angeles, CA, USA, Nov. 2022, pp. 3052–3065.

[25] D. Lee, S. Yoon, and K. Oh, "Mapping ML classification outputs to CVSS and EPSS for web vulnerabilities," *ACM Transactions on Privacy and Security*, vol. 25, no. 1, article 5, Feb. 2022.

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.30

[26] M. Ahmed and J. Liu, "Real-time scoring of web vulnerabilities using machine learning and CVSS," in *Proc. IEEE Int. Conf. Cybersecurity (CyberSec)*, Seoul, South Korea, Dec. 2022, pp. 118–127.

[27] K. Williams and E. Johnson, "A benchmark suite for web application vulnerability research," *IEEE Access*, vol. 9, pp. 14532–14544, Feb. 2021.

[28] NIST, "National Vulnerability Database (NVD)," *U.S. Department of Commerce*, 2024. [Online]. Available: https://nvd.nist.gov

[29] C. Yu and M. Luo, "Evaluating risk scoring alignment with CVSS ground truth," in *Proc. IEEE Conf. Cybersecurity and Resilience (CSR)*, Tallinn, Estonia, Sep. 2022, pp. 52–60.

[30] S. Chakraborty, A. Kumar, and B. Sharma, "Deep learning-based vulnerability detection: Are we there yet?" *arXiv preprint arXiv:2009.07235*, Sep. 2020.

[31] J. Davis and L. Thompson, "Scaling web security with automated triage using ML," *IEEE Software*, vol. 38, no. 4, pp. 42–51, Jul.–Aug. 2021.

[32] V. Reddy and P. Gupta, "High-throughput HTTP analysis using automated frameworks," in *Proc. 2021 Int. Conf. Digital Forensics & Cyber Crime (ICDF2C)*, Kolkata, India, Dec. 2021, pp. 89–97.

[33] L. Green, M. Patel, and S. Khan, "Detecting zero-day exploits using adaptive machine learning," *IEEE Transactions on Dependable and Secure Computing*, vol. 20, no. 2, pp. 310–322, Mar.–Apr. 2023.

[34] Y. Song and J. Ye, "Prioritizing vulnerabilities using CVSS-EPSS integration," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 345–358, Jan. 2023.

ISSN :1521-1398
10.48047/jocaaa.2024.33.04.30

[35] R. Patel, S. Joshi, and A. Mehta, "Impact of dataset imbalance on ML-based vulnerability detection," *ScienceDirect*, *Information & Software Technology*, vol. 145, 107241, Feb. 2024.

[36] T. Nguyen and M. Hoang, "Duplication issues in vulnerability datasets," in *Proc. 2023 Int. Conf. Machine Learning in Cybersecurity*, Prague, Czech Republic, Oct. 2023, pp. 28–36.

[37] D. Kumar and V. Sharma, "Representativeness in web security datasets: A critical survey," *IEEE Internet of Things Journal*, vol. 11, no. 3, pp. 2245–2258, Feb. 2024.

[38] P. Anderson and K. Williams, "Adversarial payload obfuscation and ML evasion," *Wikipedia*, Jul. 2025. [Online]. Available: https://en.wikipedia.org/wiki/Adversarial_machine_learning

[39] E. Chen and F. Lin, "Explainable AI for security compliance in vulnerability detection," in *Proc. 2024 IEEE European Symposium on Security and Privacy (EuroS&P)*, Zurich, Switzerland, Jun. 2024, pp. 99–114.