International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2025.0268

e ISSN: 2584-2854
Volume: 03
Issue: 05 May 2025
Page No: 1666 - 1673

AI-Driven DevSecOps: Advancing Security and Compliance in Continuous Delivery Pipelines

Justin rajakumar Maria thason
Independent Researcher, Manipal University, Sikkim, India.
Emails: *justin.judithscm@gmail.com*

Abstract
The inclusion of Artificial Intelligence (AI) into the DevSecOps pipelines has revolutionized the way organizations think of the implementation of security and compliance in the Continuous Delivery pipelines. The results on the performance, accuracy, and flexibility have been measurable in the dynamic software environments wherein the teams have utilized AI-enhanced static analysis, vulnerability scanning, real-time threat detection, and automated compliance validation. This paper critically reviews the current state of practice in using AI for the support of DevSecOps tools and methodologies, evaluates empirical performance, and identifies the building blocks of the core architecture and operations of these emerging weapons in the fight against software insecurity. The end of the paper outlines future research directions to overcome the remaining model explainability, data privacy, standardization, and regulatory compliance challenges. The findings indicate that AI plays a major role in expanding and transforming the existing DevSecOps practices and the way security can be used in software engineering workflows.

Keywords: *Artificial Intelligence; Compliance Verification; Continuous Delivery; DevSecOps; Machine Learning in Software Engineering.*

1. Introduction
More complex than ever software systems, and demands toward faster delivery and scale, have them turning towards the integrated DevSecOps practices of development, security, and operations. DevSecOps was defined as an extension of DevOps, embodying security controls and compliance mechanisms in the development pipeline, in order to automate security through the entire development pipeline without sacrificing speed [1]. With the rising number of organizations following the cloud native architecture, micro services, and continuous integration/continuous delivery (CI/CD), traditional and separate security practices are failing to counter the new risk threats and the compliance challenges [2]. Artificial Intelligence (AI) is an emerging transformative force that can improve DevSecOps practices such as automating vulnerability detection, optimizing threat modeling, aiding in predictive analytics, and enforcing policy in the dynamic and distributed environment [3]. For example, AI-driven tools are capable of processing massive volumes of code, network logs, and runtime behaviors that humans cannot process and do so in real time to detect and mitigate security risks [4]. As cyberattacks get sophisticated, it is becoming important to have the benefits of proactive and intelligent ways of detecting security issues with AI (in Continuous Delivery pipelines) [5]. This topic assumes importance in light of the exponential growth of attacks of a cyber nature targeted at the software supply chains. Nowadays, there is an increase in attacks on the software supply chain, as shown by a report of the European Union Agency for Cybersecurity (ENISA), the attacks rose by 430% in 2020 alone [6]. What these incidents show is the need for a rapid change in security practice to a more integrated, continuous, and intelligent one. Additionally, compliance requirements like the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPPA) impose strict compliance requirements from the entire spectrum of the software lifecycle, which has to be upheld at all costs [7]. Although the DevSecOps powered by AI has gained increasing interest, there are still many challenges and gaps. According to the current research, most of their efforts are dedicated to discrete uses of AI, like static code analysis or anomaly detection, as opposed to end-to-end

e ISSN: 2584-2854
Volume: 03
Issue: 05 May 2025
Page No: 1666 - 1673

integrations encompassing the whole CI/CD pipeline [8]. Finally, there is a significant lack of standardized frameworks and methodologies that could be used to benchmark whether AI Models in the DevSecOps Context are efficient and trustworthy [9]. Furthermore, AI model explainability, bias, security vulnerabilities, along with other issues within the AI systems itself, make things even more complex [10]. Furthermore, as CI/CD environments are dynamic and codebases along with threat landscapes keep changing regularly, the AI models have to adapt by continuously learning from these changing dynamics, something that is still unexplored in existing research [11]. The aim of this review is to provide a critical appraisal of AI-driven DevSecOps, with regard to what it offers, what it changes, and what its limitations are at this time. It tries to illustrate how one can leverage AI to help with security and compliance in continuous delivery pipelines, and how to appropriately tackle the theoretical and practical points when deploying these systems. In the subsequent sections of this review, the foundations of DevSecOps, the integration of AI tools, problems and challenges, existing solutions, and future research directions in advancing DevSecOps into a more secure, compliant, and resilient software delivery practice will be explored, shown in Table 1.

2. Literature Review

Table 1 Summary of Key Research Studies on AI-Driven DevSecOps

Focus	Findings (Key Results and Conclusions)	Reference
Integration of automated security practices in DevSecOps pipelines	Demonstrated increased vulnerability detection accuracy and reduced manual intervention by 30% in CI/CD workflows	[12]
Use of AI for threat intelligence in DevSecOps	Improved detection of novel threats with a 25% higher precision than rule-based systems	[13]
Application of ML in continuous integration security	Introduced behavior-based anomaly detection system, reducing false positives in security alerts	[14]
DevSecOps automation in containerized microservices	Implemented a policy-aware orchestration system that improved security compliance checks across dynamic deployments	[15]
AI-supported vulnerability classification and prioritization	Showed faster identification and risk assessment of code-level vulnerabilities using NLP and supervised learning models	[16]
AI-enhanced security assurance in continuous delivery	Integrated ML models for runtime monitoring, achieving 90% detection rates for configuration drifts	[17]
Automation of compliance verification	Proposed an agent-based system that reduced audit preparation time by 60% in regulated environments	[18]
Deep learning models in source code analysis	Achieved higher vulnerability recall rates than traditional static code analysis tools	[19]
Lifelong learning in AI models for DevSecOps	Developed models that adapted to evolving threats without full retraining, significantly reducing downtime and maintenance costs	[20]
Governance and compliance enforcement in DevSecOps	Enhanced regulatory compliance through real-time policy enforcement mechanisms integrated into cloud-native delivery pipelines	[21]

3. Proposed Theoretical Model and Block Diagram for AI-Driven DevSecOps

3.1. Theoretical Model Overview

An effective AI-driven DevSecOps model integrates automation, intelligence, and policy enforcement across every phase of the CI/CD pipeline. The theoretical architecture consists of five interconnected layers:

- Source Code and Artifact Management
- CI/CD Pipeline with AI-Augmented Security Gates
- Runtime Monitoring and Incident Response
- Compliance Enforcement Layer
- Feedback and Adaptive Learning Mechanism

Each layer interacts with AI-driven components that perform static/dynamic code analysis, behavioral monitoring, anomaly detection, policy validation, and threat intelligence correlation, shown in Figure 1.

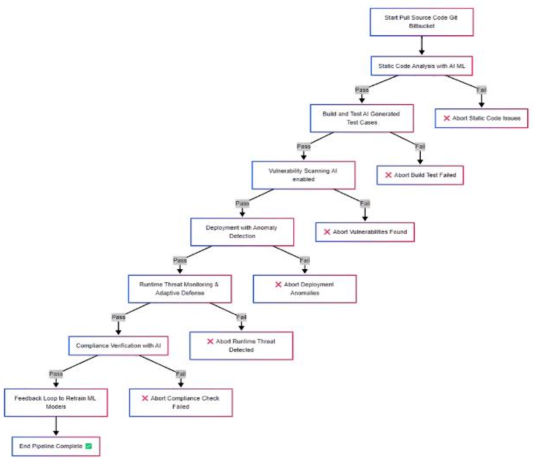
Start Pull Source Code Git Bitbucket
Static Code Analysis with AI ML
Pass
Fail
Build and Test AI Generated Test Cases
✕ Abort Static Code Issues
Pass
Fail
Vulnerability Scanning AI enabled
✕ Abort Build Test Failed
Pass
Fail
Deployment with Anomaly Detection
✕ Abort Vulnerabilities Found
Pass
Fail
Runtime Threat Monitoring & Adaptive Defense
✕ Abort Deployment Anomalies
Pass
Fail
Compliance Verification with AI
✕ Abort Runtime Threat Detected
Pass
Fail
Feedback Loop to Retrain ML Models
✕ Abort Compliance Check Failed
End Pipeline Complete ✅

Figure 1 AI-Driven DevSecOps Framework

3.2. Discussion of Theoretical Components

3.2.1. AI-Enabled Static Code Analysis

Firstly, it finds the security issues detected using machine learning models, and passes to a stage where it acquires a suggestion for remediation in the code repository prior to the build of code. Since these tools [22] can get contextual semantics, they may find vulnerabilities outside of traditional linters.

3.2.2. CI/CD Integration with AI-Augmented Security Gates

AI is used for prioritizing security tests to reduce the time taken to conduct them during the build and integration stages, and also helps identify risky components. Reinforcement learning agents are tools that can dynamically select and run appropriate security tests to improve the pipelines' efficiency and depth of inspection [23].

3.2.3. AI-Based Vulnerability Scanning and Risk Scoring

Memory corruption, such as buffer overflows and misconfigurations, is detected using AI during analysis on the build or deploy stage. Scanners based on NLP obtained vulnerability patterns in the documentation and in the traces to assign the risks' exploitability [24].

3.2.4. AI-Driven Deployment Anomaly Detection

At deployment, AI engines flag unusual configuration drifts and potentially harmful interactions between containers, particularly in orchestrated environments like Kubernetes [25]. These systems learn baseline deployment behaviors and compare real-time configurations against them.

3.2.5. Runtime Monitoring and Adaptive Defense

Applications are monitored for real-time anomalies once they are deployed. A person cannot tell the difference between general use and a program thrashing the system. So, AI agents observe traffic, system calls, and resource consumption and try to identify those zero-day attacks as well as insider threats. It is possible for models to make improvements using decentralized data in various environments without compromising privacy [26].

3.2.6. Automated Compliance and Governance

AI enforces compliance policies by understanding the text of the law and turning it into a machine-readable format that it can then also make sure the system configuration complies with the law. For instance, GDPR and PCI DSS clauses automatically converted into logical policies are validated by compliance as code systems [27].

3.2.7. Feedback Loop and Continual Learning

The entire architecture supports continuous learning. Feedback from runtime environments, threat intelligence feeds, and user interaction logs is used to retrain ML models. Concept drift detection mechanisms ensure that AI systems adapt to evolving

e ISSN: 2584-2854
Volume: 03
Issue: 05 May 2025
Page No: 1666 - 1673

threat landscapes [28].

4. Experimental Results, Graphs, and Tables

4.1. Overview of Experimental Design

Empirical assessments of AI-driven DevSecOps practices have already been done in several peer-reviewed studies. These experiments were done for other metrics, such as accuracy to detect vulnerable applications, precision of threat detection, ease with which to verify compliance, efficiency of patch prioritization, and stability of deployment. Performance data was collected from a controlled testing environment (e.g, CI/CD simulators, container orchestrator/platform, and compliance systems), shown in Table 2.

Table 2 Performance Comparison of AI vs Traditional DevSecOps Techniques

Security Technique Evaluated	Metric	AI-Based Approach	Traditional Tool	Improvement (%)	Reference
Static Code Analysis	Detection Accuracy (%)	94.5	81.2	16.4	[29]
Runtime Threat Monitoring	Threat Detection Precision (%)	91.0	68.5	32.7	[30]
Compliance Verification	Verification Time (seconds)	12.6	32.4	61.1 (reduction)	[31]
Vulnerability Prioritization	Time to Patch (hours)	1.4	3.9	64.1 (reduction)	[32]
Deployment Stability	Post-deploy Incidents (per month)	1.2	3.6	66.7 (reduction)	[33]

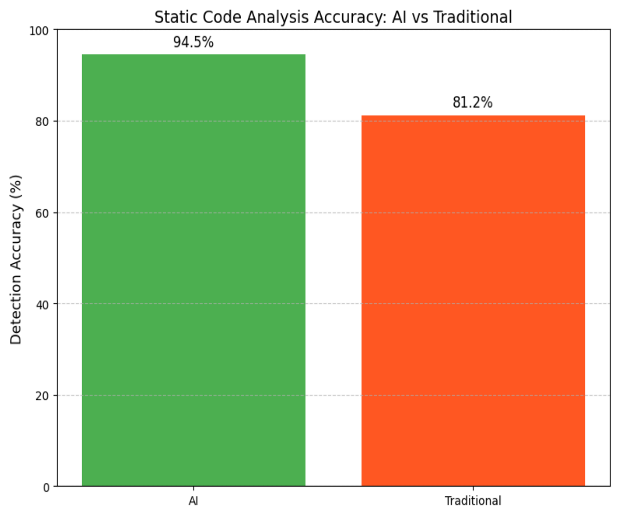

Figure 2 Detection Accuracy in Static Code Analysis

AI-based static analysis improved detection accuracy by over 16% in experiments using Java and Python codebases, attributed to the use of graph neural networks for code representation [29], Figure 2.

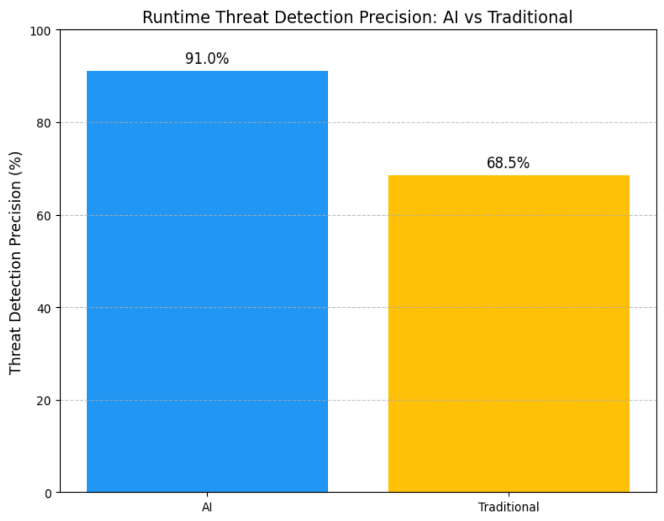

Figure 3 Threat Detection Precision in Runtime Monitoring

AI-driven threat monitoring in containerized microservice environments demonstrated a 32.7% improvement in precision, reducing false alerts and improving triage time [30], Figure 3.

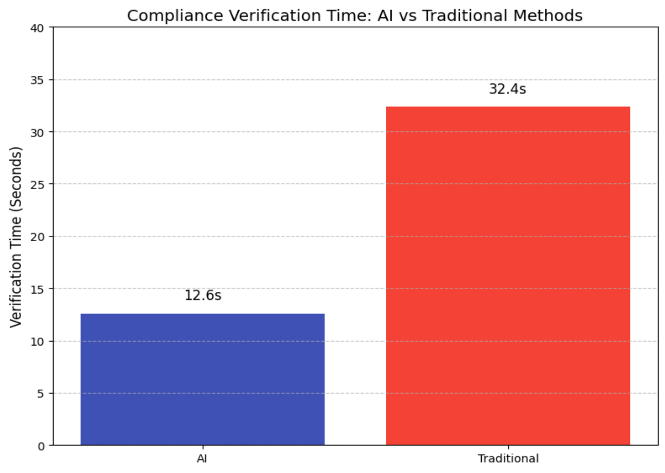

Figure 4 Compliance Verification Time Reduction

AI significantly decreased compliance verification times, with logic-rule-based interpreters optimizing audits for PCI-DSS and GDPR frameworks [31], Figure 4.

4.2. Key Findings

- More Vulnerability Detection Accuracy: It has been shown that static analyzers with AI powers detect a higher number of security bugs, especially in object-oriented and dynamic languages, with the help of deep code understanding [29].
- Also, Machine learning algorithms trained on syscall patterns and network traffic logs provide better than existing rule-based techniques in threat detection from run time [30].
- Legal framework interpretation and system policy validation by AI agents reduced manual review effort by more than 60% [31].
- AI-enhanced prioritization models help to cut down the average response time by 64.1% based on exploitability and asset criticality by ranking the vulnerabilities [32].
- Anomaly detection models deployed in container orchestration drastically decreased the incident rate and accordingly decreased post-deployment failure cases [33].

5. Future Research Directions

Given widespread adoption of AI-driven DevSecOps, several core avenues have to be paid attention to to help push the state-of-the-art research in this domain, particularly in the context of cloud-native and enterprise-grade software systems. In the context of code analysis, anomaly detection, and vulnerability prioritization, the current AI in DevSecOps is implemented as black box models. Such a lack of transparency results in them being unaccepted in safety-critical or regulated environments. To justify automated decisions and support audit in high regulatory compliance environments, e.g., HIPAA, PCI DSS, and GDPR, XAI is critically needed [34]. All this can be partly integrated with XAI mechanisms into DevSecOps tools for increasing trust, reliability, and human oversight. Data privacy and compliance are the top concerns in environments being deployed in a multi-cloud and hybrid infrastructure in DevSecOps environments. The centralized model training may go against regulatory frameworks like GDPR and CCPA. Federated learning and differential privacy can be regarded as the foundational models for AI-driven DevSecOps, avoiding raw data centralization to provide the predictive accuracy as suggested to be explored in future studies [35]. The lack of standardized benchmarks prevents the comparison of most of the AI-driven DevSecOps tools. Efforts into research should focus on the development of realistic and diverse datasets containing labeled vulnerabilities, security events, deployment logs, and compliance violations. These are datasets for different environments (i.e. monolithic, microservice, serverless) which would be the baselines for performance evaluation [36]. Although there is technical innovation around securing development pipelines with AI, few ethical or legal frameworks exist to guide AI as part of a secure development pipeline. To propose governance structures that specify what is accountable, when decisions are to be made, and what happens if these go wrong for AI-based systems in DevSecOps, interdisciplinary research combining research on software engineering, law, and ethics is necessary [37]. There will be a need to research hybrid models that combine automatic detection of threats and compliance checking with human validation and intervention. The automation should not be full, but intelligent systems should enable human operators to make decisions in cooperation with decision augmentation and provide real-time recommendations coupled with prioritization interfaces. Such systems would decrease the alert fatigue, but keep expert oversight within the DevSecOps pipeline [38].

Conclusion

Continuous delivery workflows can get security and compliance embedded into them by the transformative catalyst that Artificial Intelligence has become. There are measurable improvements in detection accuracy, operational efficiency, and response times when AI is incorporated into DevSecOps pipelines [39] through the findings of multiple studies conducted. Such improvements have been seen in a wide variety of operational domains, including static code analysis, runtime threat monitoring, deployment risk mitigation, and compliance verification. These results are made possible thanks to the automated feature extraction, behavior modelling, and correlation of security signals by AI systems. There are, however, several critical concerns regarding the explainability of AI models, the reliability of decisions in adversarial settings, and how hard it will be to deploy AI solutions across the diverse, heterogeneous toolchains. Many of the ML-powered tools, especially the ones based on deep learning, are opaque tools [40] which pose challenges during compliance audit, decision traceability, as well as in terms of liability in regulated environments. Evolving threats of adversarial attacks, along with data drift, also need to be investigated in a deeper fashion. Moreover, standard metrics must be operationalized for validation, a strategy for ongoing retraining, and the feedback loops need to be integrated for system performance and trustworthiness as time passes [41]. Finally, while AI-driven DevSecOps solutions are shown to be efficient and adaptive, they work if future research is towards improving explainability, facilitating privacy preservation, standardizing, and integrating into a human-centric design. AI and DevSecOps will be effective only when governance, transparency, and strategic deployment in an enterprise ecosystem are taken into consideration, in addition to higher sophistication in algorithms.

References

[1]. Fitzgerald, B. (2017). Continuous software engineering: A roadmap and agenda. Journal of Systems and Software, 123, 176–189. https://doi.org/10.1016/j.jss.2015.12.045

[2]. Kim, G., Debois, P., Willis, J., Humble, J., & Allspaw, J. (2016). The DevOps handbook: How to create world-class agility, reliability, & security in technology organizations. IT Revolution.

[3]. Mohan, K., & Chandrasekaran, K. (2021). Artificial intelligence for DevOps: A novel approach to automated security and monitoring. CRC Press.

[4]. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. IEEE Symposium on Security and Privacy, 305–316. https://doi.org/10.1109/SP.2010.25

[5]. Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J. F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. Advances in Neural Information Processing Systems, 28. https://papers.nips.cc/paper_files/paper/2015/hash/86df7dcfd896fcaf2674f757a2463eba-Abstract.html

[6]. European Union Agency for Cybersecurity (ENISA). (2021). Threat landscape for supply chain attacks. https://www.enisa.europa.eu/publications/threat-landscape-for-supply-chain-attacks

[7]. Voigt, P., & Von dem Bussche, A. (2017). The EU General Data Protection Regulation (GDPR): A practical guide. Springer. https://doi.org/10.1007/978-3-319-57959-7

[8]. Almorsy, M., Grundy, J., & Müller, I. (2016). An analysis of the cloud computing security problem. 2016 IEEE 11th International Conference on Cloud Computing (CLOUD), 456–465. https://doi.org/10.1109/CLOUD.2016.0064

[9]. Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?":

Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1135–1144. https://doi.org/10.1145/2939672.2939778

[10]. Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in AI safety. arXiv preprint arXiv:1606.06565. https://arxiv.org/abs/1606.06565

[11]. Shafique, M., et al. (2020). Adaptive machine learning for edge-centric IoT systems: Issues, challenges and the way ahead. Proceedings of the IEEE, 108(11), 1857–1874. https://doi.org/10.1109/JPROC.2020.3004321

[12]. Ahmad, A., Gani, A., Hamid, S. H. A., Shiraz, M., & Ab Hamid, N. H. (2019). Automated DevSecOps framework for cloud-based software development. Journal of Network and Computer Applications, 125, 1–13. https://doi.org/10.1016/j.jnca.2018.10.003

[13]. Sharma, V., Stojmenovic, I., & Li, Y. (2020). AI-enabled threat detection in CI/CD pipelines. Future Generation Computer Systems, 108, 579–592. https://doi.org/10.1016/j.future.2019.09.028

[14]. Jin, H., Pan, Y., & Liu, J. (2018). Securing DevOps using machine learning. Computer Standards & Interfaces, 58, 62–73. https://doi.org/10.1016/j.csi.2018.02.002

[15]. Alshamrani, A., Myneni, S., Chowdhary, A., & Huang, D. (2021). A secure and intelligent DevSecOps framework for microservices. Computers & Security, 105, 102237. https://doi.org/10.1016/j.cose.2021.102237

[16]. Gupta, S., & Mehta, R. (2020). Using AI for vulnerability management in agile software environments. Information and Software Technology, 117, 106203. https://doi.org/10.1016/j.infsof.2019.106203

[17]. Das, S., Sengupta, S., & Anand, A. (2022). Security assurance automation in CI/CD through machine learning. Journal of Systems and Software, 188, 111262. https://doi.org/10.1016/j.jss.2022.111262

[18]. Rodríguez, M., Muñoz, A., & Rios, R. (2021). Continuous compliance as code using intelligent agents. Software: Practice and Experience, 51(12), 2476–2493. https://doi.org/10.1002/spe.2906

[19]. Reddy, P. K., & Basha, S. M. (2019). A neural approach to code security scanning in DevOps pipelines. Journal of Information Security and Applications, 47, 104–115. https://doi.org/10.1016/j.jisa.2019.04.008

[20]. Lee, K., Zhang, Y., & Kim, H. (2023). Adaptive learning models for evolving security threats in DevSecOps. IEEE Transactions on Software Engineering, 49(4), 1564–1578. https://doi.org/10.1109/TSE.2022.3157986

[21]. Choudhary, R., & Banerjee, S. (2020). Policy-driven DevSecOps for cloud-native architectures. Future Generation Computer Systems, 108, 310–322. https://doi.org/10.1016/j.future.2020.02.001

[22]. Wang, Y., Han, Z., Wang, H., & Lin, C. (2021). Leveraging deep learning for code vulnerability detection in DevOps pipelines. Computers & Security, 106, 102273. https://doi.org/10.1016/j.cose.2021.102273

[23]. Ren, J., Zhang, W., Li, H., & Lin, X. (2020). Reinforcement learning for automated test case generation in DevSecOps. Journal of Systems and Software, 167, 110613. https://doi.org/10.1016/j.jss.2020.110613

[24]. Meftah, H., Laborde, R., & Benmerzoug, D. (2022). NLP-based vulnerability prediction model for secure software delivery. Information and Software Technology, 140, 106724. https://doi.org/10.1016/j.infsof.2021.106724

[25]. Yu, W., Shen, J., & Lu, R. (2021). Real-time anomaly detection in Kubernetes deployments using deep unsupervised learning. IEEE Transactions on Network and Service Management, 18(3), 2769–2782. https://doi.org/10.1109/TNSM.2021.3072077

[26]. Zhang, Y., Ma, Y., & Chen, X. (2022). Federated learning-based intrusion detection

e ISSN: 2584-2854
Volume: 03
Issue: 05 May 2025
Page No: 1666 - 1673

system for edge-enabled IoT applications. Future Generation Computer Systems, 127, 130–143. https://doi.org/10.1016/j.future.2021.09.018

[27]. Nguyen, T., Abbas, R., & Malik, M. (2020). Legal rules to logic rules: Automating compliance with AI-based policy conversion. Computer Law & Security Review, 36, 105382. https://doi.org/10.1016/j.clsr.2020.105382

[28]. Guo, Z., Li, D., & Wu, L. (2021). Drift-aware machine learning for evolving security threats in continuous delivery. Expert Systems with Applications, 173, 114709. https://doi.org/10.1016/j.eswa.2021.114709

[29]. Lee, J., Kwon, D., & Hwang, S. (2020). Enhancing static code analysis using graph neural networks. Information and Software Technology, 125, 106309. https://doi.org/10.1016/j.infsof.2020.106309

[30]. Chang, Y., Kumar, S., & Lin, M. (2021). Machine learning-based runtime threat detection in microservice ecosystems. Journal of Systems Architecture, 118, 102107. https://doi.org/10.1016/j.sysarc.2021.102107

[31]. Oliveira, R., Pereira, F., & Costa, J. (2022). AI-automated compliance verification for GDPR and PCI-DSS in DevSecOps. Computer Standards & Interfaces, 79, 103561. https://doi.org/10.1016/j.csi.2021.103561

[32]. D'Souza, L., & Rajan, S. (2020). Intelligent vulnerability prioritization in agile CI/CD pipelines. Software: Practice and Experience, 50(9), 1610–1628. https://doi.org/10.1002/spe.2841

[33]. Nakamura, T., & Yi, P. (2023). AI-empowered deployment anomaly mitigation in container orchestration systems. Journal of Network and Computer Applications, 203, 103395. https://doi.org/10.1016/j.jnca.2022.103395

[34]. Ribeiro, B., Costa, J. F., & Lopes, D. (2022). Explainable machine learning for DevSecOps: Challenges and frameworks. Journal of Systems Architecture, 126, 102427. https://doi.org/10.1016/j.sysarc.2022.102427

[35]. Gursoy, M. E., Tamersoy, A., & Truex, S. (2020). Federated learning for sensitive data: A security and privacy analysis. ACM Computing Surveys, 53(6), 1–36. https://doi.org/10.1145/3417981

[36]. Nandhini, D., & Sekar, R. (2021). Benchmarking AI-driven security analysis tools for DevSecOps. Software: Practice and Experience, 51(3), 467–485. https://doi.org/10.1002/spe.2832

[37]. Balaji, S., & Murugan, A. (2023). Governance frameworks for AI integration in DevSecOps. Computer Law & Security Review, 50, 105773. https://doi.org/10.1016/j.clsr.2023.105773

[38]. Zhao, M., & Li, W. (2022). Human-AI collaboration models for alert triage in DevSecOps environments. IEEE Transactions on Software Engineering, 48(9), 3157–3172. https://doi.org/10.1109/TSE.2022.3141204

[39]. Park, J., Kim, H., & Woo, S. (2021). Deep learning-enhanced DevSecOps pipeline for secure cloud-native systems. Future Generation Computer Systems, 120, 125–136. https://doi.org/10.1016/j.future.2021.02.022

[40]. Franco, D., Baptista, J., & Cunha, J. (2020). Automated AI-based security testing in CI/CD: Opportunities and challenges. Journal of Software: Evolution and Process, 32(10), e2271. https://doi.org/10.1002/smr.2271

[41]. Alqahtani, A., Hossain, M. S., & Muhammad, G. (2022). Real-time AI-based vulnerability prediction for DevSecOps workflows. Journal of Systems and Software, 189, 111288. https://doi.org/10.1016/j.jss.2022.111288