Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

DevSecOps for AI Systems: Security Automation, Model Protection, and Governance Frameworks

Manojkumar Reddy Surasani
Independent Researcher
Corresponding Author: Manojkumar Reddy Surasani, **E-mail**: Surasanimanoj@gmail.com

| ABSTRACT
The rapid adoption of Artificial Intelligence (AI) into the contemporary software ecosystem has augmented the necessity of strong security, governance, and compliance frameworks across the lifecycle of the system. In this paper, an in-depth overview of DevSecOps practices associated with AI systems will be provided, with a focus on automated security testing, model protection, and governance frameworks. It presents intelligent security automation with SAST, DAST, and IAST with the assistance of AIOps to provide real-time monitoring, anomaly detection, and incident resolution in CI/CD pipelines. The paper also analyzes the privacy-saving model protection methods, such as watermarking, federated learning, and encryption-based safe training. The most important security threats in Large Language Models, including prompt injection, data leakage, model extraction, and API misuse, are discussed along with the method of adversarial defence. Besides, the paper outlines deployment-time protection, real-time monitoring, and threat modelling for resilient functioning. Lastly, it discusses governance frameworks, including automation, Infrastructure as Code, regulatory compliance, and Responsible AI in a way that deploys AI across cloud-native environments in a manner that is both trustworthy, ethical, and secure and retains development agility.

| KEYWORDS

DevSecOps, Artificial Intelligence (AI) Security, Automated Security Testing, Model Protection, Large Language Models (LLMs), AI Governance

| ARTICLE INFORMATION

ACCEPTED: 15 June 2026 **PUBLISHED:** 19 August 2026 **DOI:** 10.32996/jcsts.2026.8.8.30

1. Introduction

In modern software engineering, DevOps has emerged as an effective development methodology that promotes collaboration between IT operations and development teams to enhance software quality and delivery efficiency [1][2]. With continuous integration (CI) and continuous delivery (CD), DevOps emphasizes automation across the software development lifecycle, and continuous testing, among others, so that organizations spend less time deploying software updates while ensuring system availability [3]. Nevertheless, security is commonly viewed as a distinct stage in many conventional DevOps systems and is not considered until after the development or deployment [4]. DevSecOps is a concept that advocates the idea of a shift in security left, meaning vulnerability testing, compliance testing, and vulnerability testing should be involved at the very beginning of the development cycle [5][6]. With security mechanisms embedded during CI/CD pipelines, organizations are able to constantly monitor the applications, identify vulnerabilities and address security risks throughout development as opposed to post-deployment [7].

The rapid development of artificial intelligence (AI) technologies has profoundly altered the current software systems in the healthcare, financial, cybersecurity, and intelligent manufacturing sectors. AI-based applications are based on machine learning models (ML) and big data, as well as automatic deployment infrastructures to provide intelligent decision-making abilities [8][9][10]. Nevertheless, Data poisoning is one of the many security issues associated with the expanding list of AI systems, adversarial attacks, model inversion, and unauthorized extraction of the model [11]. These weaknesses have underscored the

importance of safe development practices that safeguard AI models, data pipelines, and deployment infrastructure throughout the AI lifecycle.

DevSecOps capabilities have also been enhanced in the recent past. The emergence of AI and ML has provided a chance to apply automated security measures. In contrast to traditional rule-based approaches to security, AIs are able to handle substantial volumes of data and detect anomalies, as well as draw attention to possible vulnerabilities more efficiently. The tools are adopted to aid automated vulnerability scanning, secure code analysis, penetration testing, and threat monitoring in real-time CI/CD pipelines [12]. Therefore, both DevSecOps practices and AI systems are significant to automate security and protect the AI models and introduce governance frameworks to make AI-based applications trusted and reliable.

1.1 Organization of the paper

The remainder of this paper is structured as follows: Section II discusses DevSecOps' automated security testing for AI systems, including security automation tools and AI-driven vulnerability detection. Section III presents model protection techniques, including adversarial defences, watermarking, federated learning, and secure deployment strategies. Section IV outlines governance and compliance frameworks, while Section V reviews recent studies on AI-driven security automation in DevSecOps. Finally, Section VI concludes the study by summarizing the key conclusions and making recommendations for further research.

2. Automated security testing in DevSecOps for AI System

DevSecOps integrates the concepts of development, security, and operations into a single model in which security is treated as a shared responsibility across entire software development life cycle, rather than as an add-on applied after deployment. It emphasizes secure code techniques, automated testing, and continuous monitoring to address security issues in real time and encourages cooperation among the security, operations, and development teams [13]. The combination of AI and ML has considerably aided in its promotion, which allows automated vulnerability discovery, threat analytics, ongoing security testing, and maximized response to incidents in CI/CD pipelines [14]. AI-based solutions enhance the efficiency and accuracy of security tests, aid penetration testing, and allow real-time monitoring, which makes the software more secure and allows it to be delivered in agile environments [15]. Fig. 1 illustrates a DevSecOps lifecycle for AI systems, demonstrating ongoing security integration across the Plan, Build, Test, Deploy, Operate, and Monitor phases to guarantee safe development and implementation.

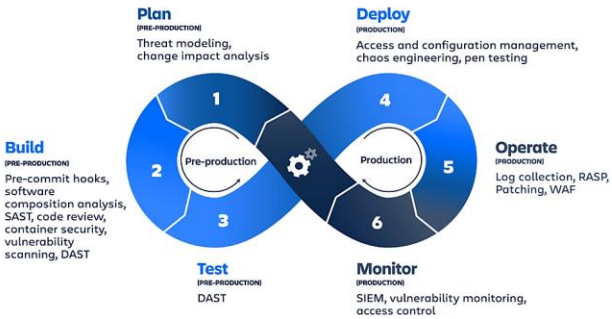

Fig. 1. The DevSecOps lifecycle for an AI system

2.1 Intelligent Security Automation and AIOps for AI Systems

AI systems need effective security automation over the lifecycle of development and operation of DevSecOps systems. Automated analysis tools, which can analyze source code, configurations, and runtime behaviour with little human effort as part of Static Application Security Testing (SAST), Dynamic Application Security Testing (DAST), and Interactive Application Security Testing (IAST), are used to find vulnerabilities in these areas [16]. Such techniques are typically combined with CI/CD processes in order to promote a secure-by-design strategy. The implementation of cloud-native systems, such as microservices and APIs, and containerized systems, has greatly enlarged the attack surface of systems based on AI, and thus, there is a need to continuously conduct automated security testing.

Artificial Intelligence in IT Operations (AIOps) is a type of software that improves security monitoring and incident response by processing large volumes of operational data (logs, network traffic, and performance metrics) using ML [17]. The features that support proactive incident management are then anomaly detection, predictive failure analysis, real-time threat detection, event correlation and automated root cause analysis. The greater implementations also offer auto-solvency, which means that proper actions are taken automatically, boosting the system's resilience. In summary, AIOps monitoring would enhance the trustworthiness and security of AI applications in a changing environment.

2.2 Key Components of Automated Security Testing

The key to automated security testing in DevSecOps environments is the use of specific security tools and methods for discovering vulnerabilities and compliance testing. These tools help to establish a secure software pipeline, helping identify threats early in the development lifecycle, meet regulatory requirements, and provide ongoing assurance.

1) Vulnerability Tools

The whole idea of automated security testing is to find and respond to vulnerabilities in code and systems, in a timely fashion. A variety of tools are deployed for this purpose, each with its own capabilities and applications.

- **Static Application Security Testing (SAST):** The SAST tools scan source code, bytecode, or even binaries without running the application. Developers may find vulnerabilities in their code while it is being written by adding SAST into the CI/CD pipeline, which expedites and reduces the cost of addressing vulnerabilities.
- **Dynamic Application Security Testing (DAST):** DAST tools work by running the software and keeping an eye on its execution patterns, but they do so in a quality assurance or staging environment rather than in production. They are adept at identifying vulnerabilities that static analysis would overlook, such SQL injection, cross-site scripting (XSS), and authentication issues.
- **Interactive Application Security Testing (IAST):** Combining these advantages allows IAST to monitor application behaviour during testing while maintaining context for the real code. IAST provides more operational results, which are sharper in real time by analyzing data streams and code traces. The use of AI-enhanced IAST tools by newer applications traces vulnerabilities through more smart approaches across many services and dependencies.

2) Compliance Validation Tools

This is accomplished by automating compliance validation processes, which simplifies checks and balances and permits continuous adherence to legal and industry requirements.

a) Regulatory Requirements (e.g., GDPR, HIPAA)

In highly regulated industries like banking and healthcare, automating adherence to regulations such GDPR and HIPAA is essential. The use of AI technologies for compliance monitoring and verification of compliance with the principles of data retention, consent and data protection standards can be applied[18]. On the contrary, present how the GDPR's requirements for tools that are semantically modelled and that meet the requirements of the GDPR might justify the informed consent[19]. These automated audits allow for greater audit preparation time savings, help enforce compliance, and stay current with regulatory updates[20].

b) Industry Standards (e.g., OWASP, NIST)

The OWASP Top Ten and NIST guidelines offer commonly accepted criteria to identify and deal with common risks. These standards can be used by compliance validation software for remedial reports, tracking progress over time, and automatically identifying gaps. Further, multimodal machine learning-based technologies can prioritize infractions based on the potential impact of the infraction and gain a better understanding of the causes of context-specific dangers. When used with automatic compliance, vulnerability scanning technologies can help ensure a DevSecOps pipeline secures the whole time.

2.3 AI Techniques in Security Automation

There are several different types of AI-powered security methodologies that can be used to improve the ability to detect security breaches accurately and efficiently. Older, rule-based security solutions are struggling to catch up with new security threats that have come with more sophisticated software applications[21]. AI has been embraced by organizations as part of their solution to the problem of identifying threats, detecting vulnerabilities, and improving the accuracy of security testing, among other ways[22][23][24]. Vulnerability detection is one application of AI that is particularly adept at automating discovery processes and allowing predictive security verification, it is a crucial part of contemporary DevSecOps pipelines. Together, these techniques allow automation platforms to adjust to new threats, minimize false-positive alarms, and optimize its resource usage. AI has a revolutionary effect on the detection, analysis, and ranking of vulnerabilities.

1) Machine Learning (ML) Algorithms

To identify unwanted patterns connected to vulnerable code, ML algorithms may be trained on a large number of known vulnerabilities, configuration bugs, or behavioural anomalies. With such elaborate models, it is possible to identify potentially vulnerable areas of the code using probabilistically aided static analysis. Simultaneously, it explains how supervised learning algorithms are incorporated into enterprise-scale AI security solutions to analyze commit histories, identify code trends that might point to vulnerabilities, and instantly identify potential threats [25]. There are also advanced scanning tools that apply ML to identify problems at various levels of an application: cloud infrastructure, third-party libraries, APIs, and source code.A system that

uses heuristics and alters the test cases to be scanned and the sorts of tests to be run generates evolutionary algorithms to optimize the effectiveness of security scanning and take use of the potential to detect previously undisclosed vulnerabilities.

2) Natural Language Processing (NLP) for Threat Intelligence

Threat intelligence mining now has a new outlet thanks to NLP's ability to automatically evaluate unstructured data, including CVEs, hacker forums, and dark web activity. According to it, NLP-driven tools are able to offer information about compromise, attack strategy, and information enrichment database vulnerability, and add to them knowledge based on context. NLP is used to proactively glean intelligence out of vulnerability reports, security advisories, and open-source feeds that enable tools to point out the emergence of new threats prior to them hitting systems [26]. NLP can also aid in the technique of deduplicating vulnerability reports, thus, leading to cleaner triage by the security team and fewer noises.

2.4 Best Practices for Implementing Automated Security Testing in DevSecOps

Best practices should be followed to guarantee that automation results in the desired outcomes without being a challenge or hazard. The development and operations teams take on a shared responsibility for security.

1) Integrating Tools Seamlessly within CI/CD Pipelines

Security testing solutions need to be easy to integrate within the CI/CD pipeline in order to be able to be automated and run regularly throughout the Software Delivery Lifecycle. DAST and IAST tools normally measure application behaviour during staging or deployment, whereas SAST tools generally analyze code at the time of commitment. Version control, issue tracking and integration with container registries enable coordinated vulnerability management and a holistic security workflow[27].

2) Training and Upskilling on New Technologies

Automated security testing tools require a lot of training for the development team as well as the security team. The staff should have the potential to set up tools, analyze vulnerability reports, and implement secure coding measures [28]. Cross-functional training as well as role-based learning enhance collaboration and competence, whereas explainable AI aspects ensure that users are aware of identified problems, enhancing trust and making remediation quicker.

3) Regularly Updating and Maintaining Security Tools

Automated security tools require updates and configuration management as a matter of their effectiveness. Inconsistent tools might not be able to detect new threats or provide false positives as a result of outdated rules. Thus, periodic vulnerability database updates, security plugins, and machine learning models need to be updated and real-time threat intelligence feeds need to be incorporated to deal with emerging risks.

3. Model Protection Techniques in DevSecOps

Securing AI models is a vital part of DevSecOps to make sure that AI systems are safe, confidential, and dependable throughout their life cycle. The model protection methods aim at ensuring that trained models are not exposed to threats like model theft, adversarial attacks, data leakage, and unauthorized access both during training, deployment and operational phases[29].

3.1 AI Model and Privacy-Preserving Watermarking

AI watermarking is a process that secures intellectual property associated with the machine learning model by carrying invisible and verifiable marks into the model. These watermarks ensure that developers can confirm they own a model and help identify cases of illegal copying and redistribution of models [30]. Approaches to watermarking could be broadly divided into white-box schemes, involving the addition of parameter patterns in a model or in internal activations, and black-box schemes, which base their ownership verification on particular input-output behaviour.

Privacy-preserving techniques are usually used along with watermarking tools in order to further increase security and privacy [31]. Such privacy-sensitive watermarking methods are of particular concern in applications where watermarking is widely used, such as healthcare, financial systems, and telemedicine, where the integrity of both the model and user information is paramount.

3.2 Federated Learning and Encryption for Secure AI Training

Federated Learning (FL) is a distributed machine learning technique that enhances data security and privacy by enabling several devices or organizations to collaboratively train AI models without revealing raw data.[32]. Such a decentralized approach to learning limits the risk of sensitive data exposure to a significant degree and is especially applicable to areas including healthcare, finance, and Industrial Internet of Things (IIoT) environments where data confidentiality is highly sensitive. Federated learning enables organizations to construct precise global models, as well as, retain tight control over their datasets because data remains local [33]. Federated learning systems are enhanced with encryption methods to improve their security and privacy

[34]. The techniques of homomorphic encryption, secure multiparty computation and secure aggregation ensure that the updates to the model are encrypted among the participants and not revealed to other users.

3.3 Security Challenges in LLM Systems

Securing LLM deployment is essential to incorporate security controls, governance, and risk management throughout the model lifecycle, including integration, inference, and continuous monitoring.

1) Prompt Injection and Jailbreaking Attacks

Adversarial prompts can be used to manipulate the actions of models, bypass safety filters, or to trick models into disclosing private information. The jailbreaking methods don't consider weaknesses in the alignment mechanisms and thus facilitate the creation of damaging or policy-infringing content.

2) Data Leakage and Privacy Risks

Using generated answers, LLLMs can accidentally reveal sensitive information such as memorized training samples, user-confidential data, or proprietary information, which is indeed a serious privacy and data compliance issue.

3) Model Inversion and Extraction Attacks

Viewing the outputs enables the attacker to reproduce part of the training data or be able to imitate model functionality, which may cause intellectual property theft or sensitive dataset leakage.

4) API Misuse and Insecure Inference Endpoints

Most LLMs, as they are accessed via APIs, can be abused using automated systems, they can also be subjected to unauthorized queries, denial-of-service, and data scraping when they are not protected by authentication, rate limiting and monitoring services.

5) RAG Pipeline and External Data Vulnerabilities

The system is based on the Retrieval-Augmented Generation and relies on external sources of knowledge that can be poisoned or manipulated, which means that malicious content can affect the model outputs and can be used to compromise the system integrity.

3.4 Secure Model Deployment and Monitoring in DevSecOps

Once the AI models have been deployed, there are a number of operational risks that are encountered based on the continuous update of data, system interaction, and the constantly changing cyber threats. DevSecOps practices can involve the seamless integration of security measures into the deployment and operational phases to maintain security and enable timely vulnerability detection [35]. Threat modelling is significant for revealing potential risks, including unauthorized access to the model, model extraction, data poisoning, and the corrupt use of input data [36]. These risks have been addressed through the use of security measures like tight access control policies, encrypted channels and secure model interfaces.

Monitors are used to track the performance and behaviour of models, as well as system activities, with the aim of identifying irregularities or possible security breaches. Automated incident response and rollback capabilities provide a way to return to a stable version of a past model in case of failure or attack. Continuous monitoring, security considerations, and quick response mechanisms enable DevSecOps to ensure affordability, security, and resilience for AI models in dynamic operational landscapes.
Table I gives a comparative overview of different adversarial defence mechanisms applied to increase robustness and security of AI models. Provides an overview of essential details, advantages, and limitations of several approaches to protecting AI systems from adversarial attacks.

Table 1: Summary of Adversarial Defence Mechanisms for AI Models

Defense Technique	Description	Advantages	Limitations	Application Areas
Adversarial Training	Training models using both normal and adversarially perturbed data to improve robustness.	Improves model resistance to adversarial attacks and enhances generalization.	Computationally expensive and may reduce model accuracy on clean data.	Image classification, autonomous driving, healthcare AI
Input Preprocessing	Applies transformations such as filtering, normalization, or feature squeezing to remove adversarial	Simple to implement and can reduce attack effectiveness.	May degrade useful features and does not defend against all	Image recognition, speech

	noise from input data.		attacks.	recognition
Defensive Distillation	Uses knowledge distillation to train models with softened probability outputs to reduce sensitivity to small input perturbations.	Improves model stability and reduces gradient-based attacks.	Less effective against stronger adaptive attacks.	Image classification, deep learning models
Gradient Masking	Hides or obfuscates gradient information to prevent attackers from generating adversarial examples.	Makes gradient-based attacks more difficult.	Can be bypassed by advanced attack methods.	Neural network security
Adversarial Detection	Detects adversarial inputs using anomaly detection or statistical analysis before feeding them to the model.	Prevents malicious inputs from affecting predictions.	Detection models may generate false positives.	Cybersecurity, fraud detection
Robust Optimization	Incorporates robustness constraints during model training to minimize worst-case adversarial perturbations.	Improves theoretical robustness guarantees.	Requires high computational resources.	Safety-critical AI systems

3.5 Security Considerations Across Different AI Model Types

Artificial intelligence systems have many different model architectures, each with its own security vulnerabilities and threat surfaces. These are some differences to understand when developing effective protection strategies in the context of DevSecOps.

1) ML and DL Models

Traditional ML models, such as DT, SVM, and LR, are especially susceptible to data poisoning attacks, in which unclean inputs added to the training dataset can alter model behaviour. Such models can also be prone to model inversion and membership inference attacks, which disclose sensitive information about the training data. Convolutional and recurrent deep neural networks are extremely vulnerable to adversarial examples, as are well-constructed inputs that are meant to trick model predictions without apparent alteration to human participants. Their excessive complexity and the size of the parameter spaces also expose them to overfitting, training backdoor attacks, and query-based model theft.

2) Transformer and Large Language Models

Transformer-based models are especially prone to causing specific risks because they are generative and exposed to natural languages. Such systems can be readily subjected to prompt injection, jailbreak, leakage of data in generated responses and model abuse using publicly available APIs. They can also be exposed to risks presented by third-party knowledge sources, e.g. in Retrieval-Augmented Generation systems, and thus present misinformation or dangerous outputs.

3) Reinforcement Learning Models

The working principle of reinforcement learning (RL) systems is their ability to adapt to changing environments in a continuous manner, and hence, they are prone to reward manipulation, environment poisoning, and policy exploitation. The adaptive behaviour of RL models also makes it difficult to detect malicious interference because malicious behaviour can develop over time.

4. Governance and Compliance Frameworks

The regulations and governance systems play a crucial role in ensuring the reliable, ethical, and secure implementation of Artificial Intelligence (AI) systems in key sectors such as healthcare, finance, and autonomous systems [37]. In the new high-stakes decision-making system, an organization needs to establish a formal governance framework to tackle the risks, meet the regulatory requirements, protect privacy and be accountable in the AI life cycle.

Cloud-native DevSecOps has emerged as an important facilitator of AI governance in the modern environment to enable scalable AI pipelines of training, validation, deployment, and real-time inference. Both container orchestration and serverless technologies are more flexible, but also come with their own challenges, including multi-tenant risks, misconfigured resources, expanding attack surface, and cross-border data management issues. Sensitive data and the model assets are thus protected by good Identity and Access Management (IAM), encryption, compliance enforcement, and ongoing monitoring. Additionally, Secure MLOps provides a more robust governance framework, allowing for controlled lifecycle management, traceability, auditing, and policy enforcement, ensuring that AI deployments in cloud-native environments remain compliant, resilient, and trustworthy.

4.1 Role of Automation, CI/CD, and IaC in Regulatory Adherence

Infrastructure as Code (IaC) and pipelines for Continuous Integration and Deployment (CI/CD) are key technologies that help to enforce governance policies in AI development environments. Compliance verification and security validation are usually done manually in traditional systems, and it is time-consuming and may be erroneous.[38]. Automation enables companies to instil governance checks in the AI development process and make sure that the results of the AI models are routinely evaluated based on their security and compliance with the relevant regulations. The pipelines of CI/CD enable testing AI models, security checks, and compliance checks with different regulations prior to their use [39].

Infrastructure as Code (IaC) is another tool that improves the nature of governance because it enables organizations to programmatically describe infrastructure configurations. IaC assists in instating security measures, encryption policies and access management policies in every environment, development, testing and production. Having infrastructure and security policies codified helps organizations to minimize configuration errors, standardized environments and ensure regulatory compliance. Fig. 2 demonstrates the major security and compliance advantages of CI/CD, such as mitigation of risks, automated compliance tests, and better response to incidents.

Fig. 2. Implementation of a secure CI/CD pipeline

4.2 Automation and Compliance Integration in AI Governance

Automation is the facilitating element of successful governance and compliance of AI systems. Delays, inconsistency and increasing operational risks are known to be caused by manual compliance procedures. By applying automated governance controls to AI development procedures, organizations able to make sure that the regulatory and security standards are maintained[40]. Among the high-assembling ones, policy-as-code, where rules of governance and compliance are given in programmatic form and consumed in AI pipelines, exists.

The compliance monitoring tools are also automated in order to provide real-time information regarding governance compliance, and hence the organizations can be in a position to identify risks and maintain sufficient audit trails. In addition, practices such as AI governance implementation involve constant security scanning, vulnerability scanning and access control which strengthen AI governance frameworks. These machine learning controls help ensure the security of the AI systems in the lifecycle of the AI systems and are also agile in their development. With the adoption of automation in governance, organizations can control these risks, enhance transparency, and ensure responsible use of AI.

4.3 Ethical, Regulatory, and Global Governance for Responsible AI Framework

The moral control of safe and reliable AI systems, at a time when AI is making major decisions in areas like health care, finance, and government[41]. The principles of fairness, transparency, accountability, explainability, reliability, and privacy protection are encouraged in responsible AI models to deliver socially acceptable, bias-free results. Algorithmic bias is also significant and, in some instances, could be the result of biased training data or flawed model development, resulting in biased decisions.

The organizations are also guided by regulatory standards to safe and legal application of AI. The policies recommended to reduce risks in automated decision-making, misuse of data and accountability have been suggested by governments and other international organizations [42]. Guidelines on how sensitive data is protected in AI systems are provided by regulations like the General Data Protection Regulation (GDPR) on data privacy, consent, and transparency as well as information security standards like ISO/IEC 27001[43]. Sensitive data utilized in AI-based applications is protected by industry-specific regulations, including health-related policies like HIPAA. All these ethical principles and regulatory frameworks establish a platform of trustworthy, globally responsible AI systems.

5. Literature Review

The analyzed studies indicate that AI-based approaches were used for threat detection, security monitoring, and automating compliance in DevSecOps environments, and describe problems, including integration complexity, high implementation costs, and a shortage of professional skills. A summary of research on using AI in DevSecOps and CI/CD pipelines to enhance security automation and compliance with requirements is provided in Table II.

A. El Mazhari, A. Bahnasse, M. Mohy-Eddine, and M. Zegrari (2026) provide practical advice on successfully incorporating AI into DevOps to increase productivity, security, and trust. Discuss the ways to adopt AI into CI/CD pipelines, mitigate essential issues, and explore the future of cybersecurity in DevOps with AI enhancements. DevOps pipelines can be enhanced by employing AI to improve pipeline security and to identify and mitigate threats. DevOps is currently being used to deploy AI and automate routine processes, minimize human intervention, and augment CI/CD. Nevertheless, companies find it difficult to streamline AI-based security, and their efforts are still divided, which cannot be considered sufficient in the current threat environment [44].

S. V. Bezzateev and A. V. Blinov (2026) concentrates on the DevSecOps framework's methods for automating security in DevOps pipelines, the way tools, processes, and cultural changes can be integrated to improve software product security. The following tasks were established in the research: the analysis of the modern DevSecOps methodology and tools; evaluating the opportunities for the application of AI and ML to automate the information security tasks; defining the key issues and obstacles to the introduction of DevSecOps into CI/CD processes; defining the perspectives of automation development in the sphere of security. This paper employs the method of comparative analytical review, including scientific literature analysis, industrial practice, and documentation of the current DevSecOps tools, the so-called Shift-Left Security and the so-called Security as Code strategies. They used open sources, documentation of CI/CD platforms, and data concerning the application of AI in information security [45]. A. O. Adebayo (2025) The article discusses how to incorporate AI for DevSecOps pipeline security compliance automation. With the growing adoption of DevSecOps in organizations to increase security across AI-based technologies, AI is presently being utilized to automate regulatory compliance, vulnerability detection, and policy enforcement across the software development lifecycle. The study uses a questionnaire to assess the knowledge of 100 people in the DevSecOps industry, employing a descriptive research approach. The results demonstrate that issues like a lack of skilled workers, high installation costs, and problems with data quality persist despite the fact that AI technologies have grown in popularity and demonstrated great efficacy in enhancing the efficacy of security audits and resolving security risks [46].

A. Jancy, R. Lokeshwaran, and T. J. Naseem (2025) discuss how AI may enhance a DevSecOps pipeline to provide robust security, compliance, and automation of the whole SDLC. Combining development, security, and operations, DevSecOps creates a framework that helps to prevent dangers before they arise without slowing down development. The objective is to embed security checks in the development and deployment lifecycle and leverage the power of AI and automation to prevent security issues from surfacing and being addressed in the early stages. To ensure safe CI/CD procedures, important tools such as GitHub Actions, SonarQube, Trivy, OWASP ZAP, Semantic Kernel, Kubernetes and Docker are used[47].

D. P. Guda (2024) illustrates a real-time threat detection model driven by AI that is integrated into DevSecOps pipelines tailored to insurance applications. The framework can proactively detect threats using ML and DL models and anomaly detection models while remaining deployment-agile. In clinical studies, increased compliance, quicker detection times and higher accuracy have been shown. New visual analytics and quantitative benchmarks support the system's strength. In accordance with operational speed and regulatory requirements, the research proposal provides a scalable and intelligent threat management solution for the insurance industry's continuous integration and delivery pipeline [48].

S. Tatineni (2023) analyses the difficulties with compliance and auditing in DevOps from a security standpoint. It highlights how crucial it is to include security into DevOps procedures and how crucial compliance is in dynamic environments. This is due to the fact that the essay also discusses the primary security factors, including threat detection, secure code, and automated compliance checks. Additionally, DevOps organizations must navigate compliance structures. Data security, identity and access control, and vulnerability management are a few prevalent issues. Automation, integrating compliance standards into the DevOps pipeline, and a compliance culture are some of the strategies used to address such issues. Security and adherence to the DevOps environment audit [49].

Table 11: Summary of Existing Studies on AI Integration for Security Automation in DevSecOps

Authors (Year)	Study On	Methods	Key Findings	Challenges	Limitations
El Mazhari et al. (2026)	Integration of AI into DevOps pipelines for enhanced cybersecurity	Analytical review of AI integration in CI/CD pipelines and security mechanisms	AI improves threat detection, automates DevOps processes, and enhances trust and efficiency in the DevOps lifecycle	Difficulty in integrating AI-driven security within existing DevOps environments	Fragmented implementation approaches and limited maturity of AI-based security integration
Bezzateev and Blinov (2026)	Automation of security in DevOps pipelines within the DevSecOps framework	Comparative analytical review of DevSecOps tools, literature analysis, and examination of "Shift-Left Security" and "Security as Code" approaches	AI and ML can automate security tasks and improve continuous security monitoring in CI/CD processes	Barriers in integrating DevSecOps practices, lack of standardized tools, and cultural resistance	Limited empirical validation and dependence on existing tool capabilities
Adebayo (2025)	AI-based automation of security compliance in DevSecOps pipelines	Descriptive research using a questionnaire survey of 100 DevSecOps professionals	AI-driven tools improve security audits, vulnerability detection, and regulatory compliance	Lack of qualified personnel, expensive implementation, and problems with data quality	Study relies on survey responses rather than real-time experimental implementation
Jancy et al. (2025)	AI-enhanced DevSecOps pipeline for secure software development lifecycle	Practical implementation using CI/CD tools such as GitHub Actions, SonarQube, Trivy, OWASP ZAP, Docker, and Kubernetes	AI integration enables proactive threat detection, automated security checks, and improved compliance	Managing tool integration and maintaining security without slowing development processes	Limited evaluation across diverse industrial environments
Guda (2024)	AI-powered real-time threat identification in DevSecOps pipelines for insurance applications	Models for anomaly detection, machine learning, and deep learning that combine visual analytics and quantitative benchmarking	Improved accuracy of threat detection, faster response time, and better compliance with regulatory requirements	Ensuring scalability and adapting models to evolving threats	Application limited to the insurance sector and specific deployment environments
Tatineni (2023)	Security compliance and auditing challenges in DevOps environments	Conceptual and analytical discussion of compliance frameworks and DevOps security practices	Integration of security practices and automated compliance checks improves secure DevOps operations	Data security issues, vulnerability management, and identity and access management	Lacks empirical validation and focuses mainly on conceptual security strategies

6. Conclusion and Future Work

The security, protection and governance models of Artificial Intelligence systems in a DevSecOps model. It has highlighted the significance of automated security testing at every stage of development, the use of SAST, DAST, IAST, AIOps-based monitoring that will enable the detection of threats in advance and respond quickly to an incident. Other model protection mechanisms that include watermarking, federated learning, encryption and adversarial defences were also mentioned as a way of safeguarding intellectual property, data privacy and model integrity. The threats of AI security in the contemporary context and the large language models, in particular, were reviewed, and the emphasis was put on prompt injection, data leakage, model extraction, and insecure APIs. Further, governance and compliance frameworks comprising CI/CD, Infrastructure as Code, policy automation and Responsible AI were cited. These measures, as a whole, can help build credible, robust and secure AI systems that can be safely deployed in increasingly cloud-based environments.

Further studies are required in the development of standard security benchmarks, automated risk assessment software and understandable defences against new AI architectures. Safe implementation of multimodal and autonomous systems, as well as the techniques to make them robust against adaptive systems and balancing global regulating systems will be crucial in the development of scalable, reliable, trustworthy and human-centric AI systems.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.
Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Abdiukov T., (2024) Automated security testing in DevSecOps pipelines: Integrating AI-based vulnerability discovery and compliance validation, *World J. Adv. Res. Rev.*, vol. 22, no. 1, pp. 2083–2093, Apr. 2024, doi: 10.30574/wjarr.2024.22.1.1083.
[2] Alugunuri N., (2024) AI for Continuous Security in DevOps (DevSecOps): Integrating Machine Learning into CI/CD Pipelines, *Int. J. Intell. Syst. Appl. Eng.*, vol. 12, no. 23s, pp. 3048–3062, 2024.
[3] Azmeera R. and Hyatt J., (2026) Software Errors, Product Functionality, and Organizational Cascades: Evidence from Developer-Lived Experience , Mar. 2026. doi: 10.2139/ssrn.6924439.
[4] Azmeera R., (2025) The Effects of Increased Software Errors on Software Product Functionality, University of the Cumberlands, 2025.
[5] Belidhe S., (2022) AI-Driven Governance for DevOps Compliance, *Int. J. Sci. Res. Sci. Eng. Technol.*, vol. 4099, pp. 527–532, Jul. 2022, doi: 10.32628/IJSRSET221654.
[6] Binbeshr F. and Imam M., (2025) Comparative Analysis of AI-Driven Security Approaches in DevSecOps: Challenges, Solutions, and Future Directions, in *Proceedings of the 2025 29th International Conference on Evaluation and Assessment in Software Engineering Companion*, New York, NY, USA: ACM, Jun. 2025, pp. 142–151. doi: 10.1145/3727967.3756841.
[7] Binbeshr F. and Imam M., (2025) Comparative Analysis of AI-Driven Security Approaches in DevSecOps: Challenges, Solutions, and Future Directions, in *Proceedings of the 2025 29th International Conference on Evaluation and Assessment in Software Engineering Companion*, New York, NY, USA: ACM, Jun. 2025, pp. 142–151. doi: 10.1145/3727967.3756841.
[8] Chandrashekar P. and Kari M., (2024) A Study on Artificial Intelligence in Software Engineering with Methodologies , Applications , and Effects on SDLC, *TIJER Int. Res. J.*, vol. 11, no. 12, pp. 932–937, 2024.
[9] Chandrashekar P., (2023) A Survey of Tools, Techniques, and Best Practices: CI/CD Integration in DevOps Workflows, *Int. J. Adv. Res. Sci. Commun. Technol.*, vol. 3, no. 3, pp. 1366–1376, Jul. 2023, doi: 10.48175/IJARSCT-11978V.
[10] Chatterjee S., (2023) A Data Governance Framework for Big Data Pipelines: Integrating Privacy, Security, and Quality in Multitenant Cloud Environments, *Tech. Int. J. Eng. Res.*, vol. 10, no. 5, 2023, doi: 10.56975/tijer.v10i5.158181.
[11] Chaturvedi V., (2023) Modern Software Development with Java , Spring Boot , and Python : A Survey of Frameworks and Best Practices, *ESP J. Eng. Technol. Adv.*, vol. 3, no. 4, pp. 188–197, 2023, doi: 10.56472/25832646/JETA-V3I8P121.
[12] Chintagunta S. K. and Amrale S., (2023) AI in Code , Testing , and Deployment : A Survey on Productivity Enhancement in Modern Software Engineering, *Int. J. Curr. Eng. Technol.*, vol. 13, no. 6, pp. 627–634, 2023, doi: 10.14741/ijcet/v.13.6.16.
[13] Chintagunta S. K., (2023) Survey of Containerization , Orchestration , and CI / CD Integration on DevOps in Modern Software Development, *Int. J. Curr. Eng. Technol.*, vol. 13, no. 6, pp. 610–618, 2023, doi: 10.14741/ijcet/v.13.6.14.
[14] Cyril H. P. and Kumara S., (2026) DevSecOps-Driven Security Integration in the Software Development Lifecycle Using CI/CD Pipelines, in *2026 IEEE 5th International Conference on AI in Cybersecurity (ICAIC)*, IEEE, Feb. 2026, pp. 1–6. doi: 10.1109/ICAIC67076.2026.11395737.
[15] Davuluri S. K., Challagulla V., Mudapaka V., and Konka U., (2025) AI-Driven DevOps in Telecommunications: Bridging Predictive Analytics with Continuous Delivery for Network Agility, in *2025 IEEE International Conference and Expo on Real Time Communications at IIT (RTC)*, Oct. 2025, pp. 1–4. doi: 10.1109/RTC66985.2025.11211551.
[16] Deep K S. H., Bhandar M., and Altounjy R., (2025) Enhancing DevOps Security with LLMs for Automation and Compliance, in *2025 7th International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, IEEE, Oct. 2025, pp. 1514–1519. doi: 10.1109/ICIDCA66325.2025.11280531.
[17] Deva M. R. R., (2025) DevOps and Continuous Delivery Adoption: Trends, Challenges, and Best Practices in Modern Software Development Life Cycle, *Int. J. Adv. Res. Comput. Sci.*, vol. 16, no. 4, pp. 118–124, Aug. 2025, doi: 10.26483/ijarcs.v16i4.7306.
[18] Fu M., Pasuksmit J., and Tantithamthavorn C., (2024) AI for DevSecOps: A Landscape and Future Opportunities, *ACM Trans. Softw. Eng. Methodol.*, vol. 1, no. 1, pp. 1–61, Sep. 2024, doi: 10.1145/3712190.
[19] Kanneganti R. K., (2024) Security and Compliance Issues in Cloud-Based Deployments of Content and Workflow Management Systems, *Eastasouth J. Inf. Syst. Comput. Sci.*, vol. 2, no. 01, pp. 131–138, Aug. 2024, doi: 10.58812/esiscs.v2i01.1122.
[20] Katangoori A., (2026) The Role of Big Data in Advancing Artificial Intelligence: Methods and Case Studies, *Int. J. Artif. Intell. Mach. Learn.*, vol. 6, no. 1, pp. 37–54, Jan. 2026, doi: 10.51483/IJAIML.6.1.2026.37-54.
[21] Kofi J. E., (2025) Monitoring Cloud Performance Metrics Utilizing AI to Estimate the Efficiency of Cloud Operations, in *2025 7th International Symposium on Advanced Electrical and Communication Technologies (ISAECT)*, IEEE, Dec. 2025, pp. 1–6. doi: 10.1109/ISAECT68904.2025.11318827
[22] Konev A., Shelupanov A., Kataev M., Ageeva V., and Nabieva A., (2022) A Survey on Threat-Modeling Techniques: Protected Objects and Classification of Threats, *Symmetry (Basel).*, vol. 14, no. 3, p. 549, Mar. 2022, doi: 10.3390/sym14030549.
[23] Koppireddy V., Parasad D. G., Patel D., Somula S. R., Kamaraj V., and Meher A., (2026) AI Security Governance for Trustworthy Intelligent Systems: A Critical Analysis of Cyber Risks, Privacy Threats, and Responsible Deployment Frameworks, *Int. J. Res. Trends Innov.*, vol. 11, no. 07, pp. a407–a435, July, 2026, doi: 10.56975/ijrti.v11i7.214029.

[24] Malali N., (2022) The Role of DevSecOps in Financial AI Models: Integrating Security at Every Stage Of Ai/Ml Model Development in Banking and Insurance, *IJETRM*, vol. 6, no. 11, p. 218, 2022, doi: 10.5281/zenodo.15239176.
[25] Mittal A., (2025) AI-Augmented DevSecOps Pipelines for Secure and Scalable Service-Oriented Architectures in Cloud-Native Systems, in *2025 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, IEEE, Jul. 2025, pp. 79–84. doi: 10.1109/SOSE67019.2025.00014.
[26] Mohammed K. K., (2025) The Future is Cloud : Modernising Big Data for the Cloud Era, *Int. J. Sci. Res. Eng. Trends*, vol. 11, no. 5, pp. 1–5, 2025, doi: 10.5281/zenodo.17339856.
[27] Otunlape O., (2026) Governance Frameworks for Enterprise AI Systems operating in Regulated Environments, *Int. J. Comput. Appl.*, vol. 187, no. 74, pp. 13–21, Jan. 2026, doi: 10.5120/ijca2026926244.
[28] Pakalapati N., Konidena B. K., and Mohamed I. A., (2023) Unlocking the Power of AI/ML in DevSecOps: Strategies and Best Practices, *J. Knowl. Learn. Sci. Technol. ISSN 2959-6386*, vol. 2, no. 2, pp. 176–188, Jul. 2023, doi: 10.60087/jklst.vol2.n2.p188.
[29] Papagiannidis E., Mikalef P., and Conboy K., (2025) Responsible artificial intelligence governance: A review and research framework, *J. Strateg. Inf. Syst.*, vol. 34, no. 2, p. 101885, Jun. 2025, doi: 10.1016/j.jsis.2024.101885.
[30] Phalke S., Athave Y. D., and Ilag B. N., (2025) A Multi-Layered Approach to IT Infrastructure Governance and Compliance- Security, Hardening, and Audit Readiness V3, *Int. J. Comput. Appl.*, vol. 187, no. 12, pp. 29–33, Jun. 2025, doi: 10.5120/ijca2025925133.
[31] Prajapati N., (2025) The Role of Machine Learning in Big Data Analytics: Tools, Techniques, and Applications, *ESP J. Eng. Technol. Adv.*, vol. 5, no. 2, 2025, doi: 10.56472/25832646/JETA-V5I2P103.
[32] Rodríguez E., Otero B., and Canal R., (2023) A Survey of Machine and Deep Learning Methods for Privacy Protection in the Internet of Things, *Sensors*, vol. 23, no. 3, p. 1252, Jan. 2023, doi: 10.3390/s23031252.
[33] Sarker I. H., Janicke H., Mohsin A., Gill A., and Maglaras L., (2024) Explainable AI for cybersecurity automation, intelligence and trustworthiness in digital twin: Methods, taxonomy, challenges and prospects, *ICT Express*, vol. 10, no. 4, pp. 935–958, Aug. 2024, doi: 10.1016/j.icte.2024.05.007.
[34] Sarraf G. and Pal V., (2022) Privacy-Preserving Data Processing in Cloud : From Homomorphic Encryption to Federated Analytics, *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, vol. 8, no. 2, pp. 735–706, 2022.
[35] Sivashanmugam S. P., Kumara S., Mohile A., and Suram D. R., (2026) Improving Detection Accuracy of Phishing Attacks with Feature Selection and Machine Learning Techniques in Cybersecurity, in *2026 1st International Conference on Emerging Trends in Advancements and Applications of Computational Intelligence Techniques (ETAACT)*, Bhubaneswar, India: IEEE, Apr. 2026, pp. 1–6. doi: 10.1109/ETAACT69135.2026.11542138.
[36] Soni A. A., Parikh M., Dhenia R. N. K., Soni J. A., Jha A. R., and Shah S. M., (2026) Reinforcement Learning for Dynamic Workflow Optimization in CI/CD Pipelines, *arXiv Prepr. arXiv2601.11647*, 2026, doi: 2601.11647v1.
[37] Verma V., (2022) Big Data and Cloud Databases Revolutionizing Business Intelligence, *TIJER*, vol. 9, no. 1, pp. 48–58, 2022.
[38] Vinod K T, (2025) Adaptive Edge Intelligence: Integrating AI and DevSecOps for Real-Time Threat Mitigation, *Int. J. Comput. Exp. Sci. Eng.*, vol. 11, no. 4, pp. 10212–10219, Mar. 2025, doi: 10.22399/ijcesen.4799.
[39] Vishvakarma R. and Jacob S., (2026) Risks and Challenges of using Agentic AI in Enterprise Software Systems, *Int. J. Comput. Appl.*, vol. 975, no. 128, pp. 48–57, July, 2026.
[40] Vishvakarma R., (2026) A Governance Model For Agentic AI Use In Banking Systems, *Int. Res. J. Mod. Eng. Technol. Sci.*, vol. 8, no. 6, pp. 2917–2934, june, Jul. 2026, doi: 10.56726/IRJMETS100553.
[41] Warrier A., (2020) Federated Learning Architectures for Multi-Hospital Research Data Collaboration, *Int. J. Emerg. Res. Eng. Technol.*, vol. 1, no. 1, 2020, doi: 10.63282/3050-922X.IJERET-V1I1P110.
[42] Wei L., Xue W., Huang J., and Gao R., (2025) Data Security Governance: A Review of Theoretical Framework, Evolving Challenges, and Response Approaches, in *Proceedings of the 10th International Conference on Cyber Security and Information Engineering*, New York, NY, USA: ACM, Jul. 2025, pp. 138–146. doi: 10.1145/3759179.3759267.
[43] Zhao X., Clear T., and Lal R., (2024) Identifying the primary dimensions of DevSecOps: A multi-vocal literature review," *J. Syst. Softw.*, vol. 214, p. 112063, Aug. 2024, doi: 10.1016/j.jss.2024.112063.