technologies

MDPI

Systematic Review

Evolution of DevSecOps and Its Influence on Application Security: A Systematic Literature Review

Khwaja Imran Mohammed [1,*], Bharanidharan Shanmugam [2,*] and Jamal El-Den [1]

[1] Faculty of Science and Technology, Charles Darwin University, Sydney, NSW 2000, Australia; jamal.el-den@cdu.edu.au
[2] Faculty of Science and Technology, Charles Darwin University, Darwin, NT 0810, Australia
* Correspondence: khwajaimran.mohammed@cdu.edu.au (K.I.M.); bharanidharan.shanmugam@cdu.edu.au (B.S.)

Abstract

This systematic literature review (SLR) investigates the current landscape of DevSecOps by analysing peer-reviewed studies published between 2012 and 2025. The review identifies key themes including automation, continuous security integration, toolchain orchestration, and cultural transformation. At the same time, it underscores the persistent challenges of embedding security within fast-paced development cycles, such as the absence of standardised practices, difficulties in achieving tool interoperability, and resistance at the organisational level. By synthesising findings across 36 selected studies, this paper rightly identified the gaps including seamless integration of security tools, lack of comprehensive frameworks for effective deployment, scalability in cloud-native technologies for effective DevSecOps adoption. In response, the study outlines future research directions aimed at bridging these gaps, particularly in areas of empirical validation and scalability. The study contributes to both academia and industry by offering a comprehensive understanding of DevSecOps evolution, its practical implications, and the critical factors influencing its success.

Keywords: DevSecOps; shift-left security; artificial intelligence (AI); machine learning (ML); predictive analytics; application security

check for updates

Academic Editor: Arash Habibi Lashkari

Received: 6 October 2025
Revised: 14 November 2025
Accepted: 20 November 2025
Published: 25 November 2025

Citation: Mohammed, K.I.; Shanmugam, B.; El-Den, J. Evolution of DevSecOps and Its Influence on Application Security: A Systematic Literature Review. *Technologies* **2025**, *13*, 548. https://doi.org/10.3390/technologies13120548

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

In the past decade, the global banking and financial sector has experienced an unprecedented surge in both the frequency and complexity of cyberattacks. From denial-of-service disruptions in the United Kingdom to sophisticated ransomware attacks in Latin America and gigantic data breaches in Asia and the Middle East, financial institutions have repeatedly found themselves at the frontlines of an evolving threat landscape. Each incident not only inflicted operational and reputational damage but also exposed the persistent gaps in how banks address application and infrastructure security. The diversity and severity of these events ranging from the HSBC distributed denial-of-service (DDoS) attack of January 2016 [1] that crippled digital access on a key payday, to the Sepah Bank breach of March 2025 [2] where 42 million customer records were exfiltrated and held for ransom underscore that traditional, reactive security practices are no longer sufficient to defend critical financial systems.

A common thread running through these high-profile incidents is the exploitation of systemic vulnerabilities: weak monitoring and detection, ineffective response procedures,

reliance on outdated or unsegmented architectures, and a lack of security integration in both application development and IT operations. The Bangladesh Bank cyber heist of February 2016 [3] and the October 2016 data breach in major Indian banks which involved State Bank of India, HDFC Bank, ICICI Bank, YES Bank, and Axis Bank [4] both highlighted how cybercriminals leveraged poorly monitored systems, under-protected third-party infrastructure, and undetected malware to orchestrate multilayered attacks. In the aftermath, impacted institutions were forced into costly, large-scale containment efforts, such as wholesale card replacements and multinational investigations, remedies that highlight a fundamentally reactive, rather than proactive, approach to application security and risk management.

The shifting balance of power in favour of sophisticated threats, whether the cloud misconfigurations that enabled the Capital One data breach of July 2019 [5], or the phishing vectors and lateral movement uncovered in the BancoEstado ransomware attack of September 2020 [6] and Interbank cyberattack of October 2024 [7], demands a new security paradigm. No longer can banks afford siloed or "bolt-on" security solutions that operate in isolation from development and operational lifecycles. Instead, there is a clear imperative for continuous security integration: a methodology where security is embedded at every phase of application development and infrastructure management, powered by automation, real-time monitoring, and cross-functional collaboration.

This is where the principles and practices of DevSecOps become central. As shown in Figure 1, DevSecOps represents a transformative evolution of the DevOps approach, advocating for security as a shared and automated responsibility throughout the software delivery pipeline. Through practices such as automated vulnerability scanning, secure configuration management, routine code reviews, and rapid incident response, DevSecOps fosters resilience against both internal and external threats. It also ensures that security posture keeps pace with the rapid development cycles and cloud-native architectures that now dominate modern banking and fintech ecosystems [8,9].

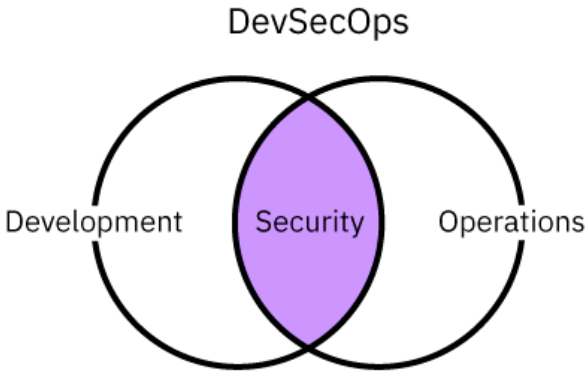


Figure 1. DevSecOps [8,9].

When scrutinised through the lens of recent cyber incidents, the influence of DevSecOps on application security becomes increasingly apparent. For example, real-time monitoring and automated credential management espoused by DevSecOps could have drastically reduced dwell time and data exfiltration risks in the Interbank and Sepah Bank breaches. Automated security testing and continuous integration checks stand as robust defences against unpatched vulnerabilities and misconfigurations, the very weaknesses exploited in the Capital One and Indian bank breaches. Regular table-top exercises, least privilege enforcement, and network segmentation, all core to DevSecOps, directly mitigate the risks of lateral movement and privilege escalation observed in ransomware and targeted attacks.

Therefore, as this systematic literature review unfolds, it is both timely and essential to critically examine how DevSecOps has emerged as an influential force in advancing application security in the banking sector. By mapping the evolution of attacks with the principles and real-world impact of DevSecOps adoption, this review aims to provide a holistic understanding of why, how, and with what effect modern financial institutions are rearchitecting their security strategies for the digital era.

In recent years, the software development landscape has undergone significant transformation, driven by the adoption of agile methodologies and automation technologies that prioritise speed and efficiency. The Development + Operations (DevOps) framework has become widely adopted for enabling rapid software delivery and fostering improved collaboration between development and operations teams [10]. However, the increasing sophistication and frequency of cyber threats have exposed critical vulnerabilities in traditional DevOps practices, where security considerations were often secondary. This has catalysed the emergence of Development + Security + Operations (DevSecOps), which integrates security as a seamless and continuous element throughout the software lifecycle [11,12].

As shown in Figure 2, the Gartner report illustrates the evolution of new and updated services through an iterative DevSecOps cycle, emphasising the integration of security as a shared responsibility throughout the DevOps process [13]. DevSecOps marks a fundamental shift in organisational approaches to security, embedding shared responsibility for security from the outset of development. By addressing evolving cybersecurity threats proactively, DevSecOps ensures that applications remain secure without sacrificing the speed advantages of modern development.

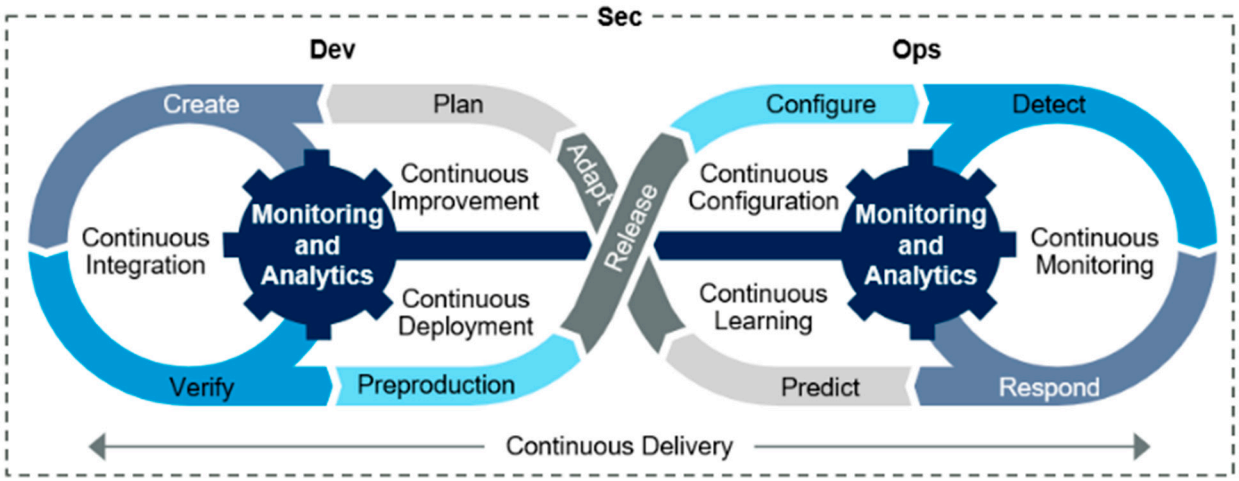


Figure 2. DevSecOps as per the Gartner report [13].

As shown in Figure 3, core principles such as "Shift-Left Security," automation, and "Security as Code" exemplify this proactive stance, reducing security risks while supporting the demands of fast-paced software delivery [14–16].

Despite its increasing relevance, the adoption of DevSecOps is not without challenges. Organisations often encounter resistance to change, skill shortages, and the technical complexity of integrating security tools into Continuous Integration/Continuous Deployment (CI/CD) pipelines. Highly regulated sectors such as healthcare, finance, and government face additional hurdles due to stringent compliance requirements, further complicating DevSecOps implementation. These challenges underscore the need for comprehensive, accessible models of DevSecOps principles, frameworks, and technologies [17–20].

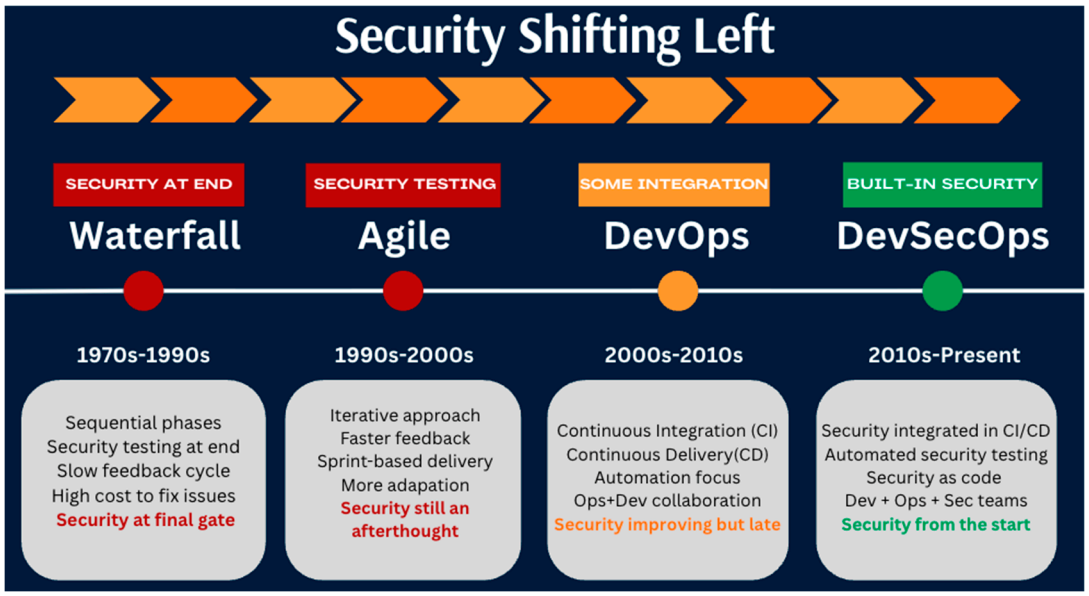


Figure 3. The evolution of development methodology "Shift Left"— shifting security to the left [14–16].

This systematic literature review (SLR) aims to map the chronological evolution of DevSecOps and its implications for application security. The study is guided by key research questions that explore the progression, adoption barriers, and best practices associated with DevSecOps in contemporary software engineering.

This study seeks to address the following research questions, which explore the core principles and frameworks that form the foundation of DevSecOps:

- RQ1: What core principles and what frameworks define DevSecOps?
- RQ2: How is DevSecOps impacting application security practices across various sectors?
- RQ3: What are technology, organisational and cultural challenges from the adoption perspective of DevSecOps?
- RQ4: What are the advantages of DevSecOps implementation using new technologies such as AI and IaC?
- RQ5: What are the research gaps and opportunities for advancing DevSecOps practices and frameworks?

1.1. Context & Motivation

This systematic literature review synthesises academic and industry research published between 2012 and 2025, with the aim of identifying prevailing trends, research gaps, and future directions for the advancement of DevSecOps. The review systematically examines how the rapid acceleration of software development and the demand for higher-quality, faster-delivered applications have driven the evolution from traditional development and DevOps towards practices that prioritise efficiency and collaboration. While DevOps catalysed a foundational cultural shift emphasising automation, continuous integration, and continuous deployment [21] the literature reveals that existing surveys and case studies frequently focus on large organisations and may not fully capture the diversity of implementation experiences across different industries and scales.

A major theme in the reviewed works is the emergence of security as a core, shared responsibility throughout the software lifecycle, marking a transformation from legacy security practices typically positioned at the end of development. However, the rapid evolution of DevOps has also exposed significant security vulnerabilities. Traditional security practices, often relegated to the final stages of the development lifecycle, have struggled to keep pace with the iterative and dynamic nature of modern software delivery. This lag in

integrating security measures has led to increased vulnerability exposure and has resulted in financial and reputational risks for organisations due to security breaches [22,23]. In response, DevSecOps stands out for its proactive, "Shift-Left" approach, incorporating early and continuous security testing and validation to pre-empt emerging vulnerabilities [24]. Nevertheless, the review identifies that despite strong theoretical support for Shift-Left and automation, there remains a lack of extensive empirical validation particularly regarding the long-term effectiveness, scalability, and organisational challenges of DevSecOps adoption. Limited longitudinal studies and the underreporting of practical obstacles further constrain the generalisability of many findings.

Compliance and risk management appear as additional focal points, with compliance automation recognised as underexplored in both research and industry practice [25]. While DevSecOps aims to institutionalise security and compliance within automated pipelines, few studies rigorously assess the integration of regulatory requirements or the practicalities of ensuring continuous assurance. In tracing foundational principles, frameworks, and key implementation challenges, this review provides a comprehensive synthesis for both scholars and practitioners. However, limitations remain due to the heterogeneity and variable quality of included studies, potential publication bias, and the ongoing evolution of technologies, which may render some insights rapidly outdated and highlight the need for continued empirical research and validation in diverse organisational contexts [26,27].

1.2. Problem Statement

Despite the transformational impact of DevOps in accelerating software delivery and fostering cross-functional collaboration, its initial implementations frequently deprioritised security, resulting in persistent vulnerabilities amid an increasingly complex threat landscape. The emergence of DevSecOps seeks to resolve these critical gaps by embedding security at every phase of the development lifecycle. However, the adoption of DevSecOps remains inconsistent and fraught with multifaceted obstacles including fragmented and immature frameworks, technical integration challenges, pronounced cultural and organisational resistance, and a lack of tailored, industry-specific methodologies. The deployment of emerging innovations such as artificial intelligence and Infrastructure as Code in DevSecOps remains nascent and largely unvalidated in practice. Furthermore, comprehensive and longitudinal evaluations that assess the real-world effectiveness of DevSecOps practices on application security are scarce across diverse organisational contexts.

If these challenges are not systematically addressed, organisations face significant negative impacts: the continued exposure to preventable security breaches, increased financial and reputational risk, regulatory non-compliance, and the erosion of stakeholder trust. The failure to effectively implement and evaluate DevSecOps can also stifle innovation, hinder organisational agility, and exacerbate the gap between security and development objectives, ultimately undermining the resilience of technology ecosystems. In light of these concerns, this systematic literature review addresses the urgent need for a critical synthesis of existing research, rigorously identifies unresolved gaps, and formulates evidence-based recommendations to support robust, scalable, and secure DevSecOps implementation.

1.3. Scope

This study explores the DevSecOps paradigm and its influence on application security from 2012 to 2025. The review draws upon academic literature sourced from Scopus and relevant grey literature identified via Google Search. The analysis is structured around four key thematic areas:

- Principles & Frameworks: Examining the foundational concepts and structured methodologies that underpin DevSecOps.

- Technological Enablers: Assessing the role of emerging technologies, such as artificial intelligence (AI) and Infrastructure as Code (IaC), in facilitating DevSecOps practices.

1.4. Contributions

This systematic literature review offers a thorough and critical examination of DevSecOps, charting its progression from traditional DevOps to its present-day focus on security. It explores fundamental principles such as Shift-Left security and continuous security integration, while also evaluating the influence of emerging technologies like artificial intelligence, machine learning, and Infrastructure as Code. The review uncovers various organisational, technical, and cultural challenges that act as barriers to the broad adoption of DevSecOps practices, providing valuable insights that cater to both industry practitioners and academic researchers.

Furthermore, this study identifies enduring gaps in the current research landscape of DevSecOps and underscores the importance of advancing AI-driven security solutions along with the development of standardised metrics to measure their effectiveness. By laying a robust groundwork, it promotes DevSecOps as a critical enabler for achieving resilient and secure software development in today's digital environment, thus setting a clear agenda for future research and practical application in the field.

1.5. Comparison of Existing Survey Papers

The surveyed literature consistently positions DevSecOps as the principal theme underpinning advancements in secure software development, highlighting its significance in aligning security, development, and operations. Across these works, DevSecOps is consistently portrayed as a transformative practice that dissolves traditional barriers between different disciplines by embedding security practices throughout every phase of the software development lifecycle. Whether presented as systematic reviews, framework proposals, or migration models, these studies uniformly regard DevSecOps as fundamental for enabling modern, resilient, and responsive technology landscapes. Nevertheless, a limitation of the current literature is the tendency to focus on conceptual models and best practices, with relatively little empirical evidence from large-scale or longitudinal studies, thereby potentially limiting the generalisability of these findings across different organisational contexts and industries.

Application security emerges as both a primary aim and an ongoing process within DevSecOps implementations. Particularly in the works of Leite et al. [21], Rajapakse et al. [28], and Akbar et al. [29], the papers stress that security controls should be integrated at every stage of development. There is a notable consensus that security should not serve merely as a final check but must be woven intrinsically into all workflows and automated pipelines. However, the surveyed literature often lacks detailed accounts of the practical challenges and organisational hurdles encountered during such integration and generally provides limited insight into the effectiveness of these approaches over extended periods or within smaller or resource-constrained organisations.

A recurring pattern across the literature is the adoption of the Shift-Left approach, which advances security assessments and enforcement to earlier stages of the development process. Many of the surveyed papers advocate this methodology, documenting its efficacy in identifying and mitigating vulnerabilities before they propagate to later development stages. Recommendations for Shift-Left frequently include coupling it with automated pipeline integrations and proactive risk management, reinforcing its value in contemporary DevSecOps strategies. Despite these advantages, few studies offer quantitative analyses or robust comparative data on the long-term impact and scalability of Shift-Left initiatives, particularly in complex or distributed development environments.

The role of IaC features prominently in these comparisons, particularly in the works of Ibrahim et al. [20] and Alonso et al. [30]. The literature notes that while IaC enables greater consistency and operational efficiency, it also poses potential systemic risks if security is not inherently built into provisioning scripts and configurations. The frameworks under review emphasise mandatory security validation and checks within IaC routines to support secure infrastructure management practices. Nonetheless, most frameworks presented are yet to be validated through broad empirical deployment, and the literature typically does not address the sector-specific limitations or the readiness of different organisational contexts to adopt such rigorous practices.

Continuous Integration and Continuous Deployment (CI/CD) practices are universally acknowledged as accelerators for software delivery, but the papers point out the critical necessity of robust, automated security within these rapidly evolving pipelines. From Leite et al. [21] through Nisha & Khandebharad [31], there is strong agreement that continuous security validation and integration are essential to close potential security gaps as code progresses through the development pipeline. A key limitation here is the relative scarcity of case studies addressing legacy integration challenges, organisational resistance, or resource constraints that can impede the automation and optimisation of security within CI/CD workflows.

A structured approach to DevSecOps adoption is evident, as all surveyed papers either propose or critically examine various frameworks. These approaches encompass decision-making tools, migration roadmaps, maturity models, and reference architectures, which together help organisations evaluate their current status, address site-specific needs, and benchmark their progress against industry standards. This focus on methodical frameworks fosters a more reliable and systematic transition to secure software delivery pipelines. However, many proposed frameworks have not undergone rigorous testing or validation outside of limited pilot projects or particular domains, and few studies have assessed their adaptability to rapidly changing technological landscapes.

Practical adoption challenges and the broader industry context are also rigorously assessed. Ramaj et al. [25] and Rajapakse et al. [28] document the technical, cultural, and compliance-related barriers organisations face when adopting DevSecOps, while other works provide prescriptive strategies for change management and migration. Such frameworks consider improvements across technical processes as well as organisational culture and resources. Still, most studies rely on high-level or theoretical discussions rather than in-depth empirical evaluations, and individual organisational or regional contexts are often underexplored, reducing the actionable precision of their recommendations.

There is marked variation in sectoral and thematic breadth among the surveyed literature. Some works, like Al-Garadi et al. [32] branch into areas such as IoT security with a machine learning lens, while others address regulatory and compliance challenges. This diversity underscores the adaptability and flexibility that DevSecOps must offer across different domains, emphasising the importance of context-sensitive methodologies. Nonetheless, the breadth of coverage can dilute focus and result in a lack of detailed, sector-specific insights, impeding the development of universally accepted best practices.

A distinctive theme that is beginning to emerge concerns the integration of AI/ML in DevSecOps. Al-Garadi et al. [32] and Pakalapati et al. [33] provide strategic assessments of using AI/ML for threat detection and automated decision-making within DevSecOps pipelines, signalling the start of a broader movement toward intelligent, data-driven security automation. Although most of the surveyed papers only mention AI/ML briefly, these contributions lay the groundwork for future enhancements and research directions. However, at present, practical implementation and longitudinal outcomes of AI/ML in-

tegration in DevSecOps remain sparsely documented, leaving significant gaps for future empirical study.

Bahaa et al. [34] provide a systematic literature review centring on monitoring real-time security threats in IoT systems via DevSecOps principles, highlighting innovations in security integration but focusing primarily on application and infrastructure dimensions. Sinan et al. [35] expand this analysis, investigating the integration of security controls within DevSecOps, and emphasising the "shift left" paradigm. Their work details current challenges, solutions, and maps out future research directions for embedding security earlier within CI/CD pipelines, offering a comprehensive framework perspective.

On the automation and technology front, Pranav et al. [36] turn their attention to advances in cybersecurity automation, aligning DevSecOps practices with emerging AI/ML solutions. Similarly, Rangaraju et al. [37] critically evaluate how AI-driven strategies are vital for securing cloud environments under DevSecOps regimes, addressing practical implications for infrastructure as code and adaptive frameworks. Zhou et al. [38] approach DevSecOps from an industry-academic lens, providing an evidence-based evaluation of real-world implementations, while Lombardi & Fanton [39] advocate for an extreme shift-left CyberDevOps architecture. Bedoya et al. [40] investigate the merger of Large Language Models (LLMs) with Security Chaos Engineering as a novel automation-enhanced experiment which demonstrates how LLMs can automate the creation of attack–defence scenarios, enabling more robust and proactive defence mechanisms throughout the software lifecycle, especially when combined with Chaos Engineering's resilience testing. Their vision brings security considerations deep into software lifecycles, moving beyond mere adoption to fundamental architectural change.

Together, these findings reflect an evolutionary shift in DevSecOps research and implementation, marked by its diverse adoption, the increased utilisation of automation, and the essential development of adaptable frameworks for effective software security management. The evolution of the field is distinguished by greater methodological rigour, stronger integration between automated security tools and overarching frameworks, and a rising focus on advanced AI/ML technologies that emphasise the value of intelligent systems within secure development pipelines.

The evidence suggests that future trajectories in DevSecOps will involve the seamless, automated, and early integration of security driven by well-established frameworks and flexible practices. Ongoing research is poised to prioritise empirically grounded evaluations, further innovation in automated solutions, and continuous enhancement of the relationship between human expertise and intelligent, adaptive security systems. Table 1 shows a comparison of the survey papers.

Table 1. Review of survey papers.

Survey Paper	DevSecOps	Application Security	Shift Left	Infrastructure As Code (IaC)	CI/CD	Framework	AI/ML
[20]	✓	✓	✓	✓	✓	✓	✗
[21]	✓	✓	✓	✓	✓	✓	✗
[25]	✓	✓	✓	✓	✓	✓	✗
[28]	✓	✓	✓	✓	✓	✓	✗
[29]	✓	✓	✓	✓	✓	✓	✗
[30]	✓	✓	✓	✓	✓	✓	✗
[31]	✓	✓	✓	✓	✓	✓	✗
[32]	✓	✓	✗	✗	✗	✓	✓
[33]	✓	✓	✓	✓	✓	✓	✓

Table 1. *Cont.*

Survey Paper	DevSecOps	Application Security	Shift Left	Infrastructure As Code (IaC)	CI/CD	Framework	AI/ML
[41]	✓	✓	✓	✓	✓	✓	✗
[34]	✓	✓	✗	✓	✓	✗	✗
[35]	✓	✓	✓	✓	✓	✓	✗
[36]	✓	✓	✗	✗	✓	✓	✓
[37]	✓	✗	✓	✓	✗	✓	✓
[38]	✓	✓	✓	✗	✓	✓	✗
[39]	✓	✓	✓	✗	✓	✓	✗
[40]	✓	✓	✓	✗	✓	✓	✓
This Research	✓	✓	✓	✓	✓	✓	✓

2. Methodology

This research adopts a systematic literature review methodology to provide a comprehensive and integrated perspective on recent advancements in DevSecOps and their implications for application security. A structured and replicable search strategy was employed to ensure thorough coverage of both academic and grey literature. Academic sources were primarily retrieved from the Scopus database, while grey literature including white papers and industry reports was identified through targeted Google searches.

2.1. Search Strategy

A structured and replicable search strategy was implemented to ensure comprehensive coverage of relevant literature for this systematic literature review. Academic articles were sourced from the Scopus database, while grey literature including white papers and industry reports was identified through targeted Google searches.

To align with the research focus and objectives, specific keywords were selected: "DevSecOps", "SecDevOps", "Continuous Security", "Application Security", "Shift-Left", "Infrastructure as Code (IaC)" and "Security as Code". Boolean operators (AND, OR, NOT) and various keyword combinations were employed to refine search results and maintain the focus within the defined scope of the review.

- ("DevSecOps" OR "SecDevOps") AND ("Application Security" OR "Shift-Left").
- ("DevSecOps" OR "Shift-Left Security") AND ("Infrastructure as Code" OR "Security as Code").
- ("AI" OR "Machine Learning") AND ("DevSecOps" OR "Application Security")
- ("DevSecOps Framework" OR "Security Practices") AND ("Continuous Integration" OR "Continuous Deployment")

To ensure the quality and relevance of the literature, only studies published between 2012 and 2025 were considered. Inclusion criteria required that sources be peer-reviewed and published in English.

Table 2 outlines the search strategy employed for DevSecOps, focusing on key areas such as foundational concepts, security practices, technological enablers, and relevant frameworks. Keywords including "Shift-Left", "AI", "Infrastructure as Code" and "Continuous Deployment" were used to capture a comprehensive range of literature. This keyword-driven approach ensures that the search effectively covers theoretical, practical, and technological aspects of security integration within software development.

Table 2. Search strategies for the study.

Search Focus	Search Strategy
Core Conceptual Search	("DevSecOps" OR "SecDevOps") AND ("Application Security" OR "Shift-Left")
Focus on Security Practices	("DevSecOps" OR "Shift-Left Security") AND ("Infrastructure as Code" OR "Security as Code")
Technological Enablers	("AI" OR "Machine Learning") AND ("DevSecOps" OR "Application Security")
Frameworks and Implementation	("DevSecOps Framework" OR "Security Practices") AND ("Continuous Integration" OR "Continuous Deployment")

2.2. Inclusion/Exclusion Criteria

Clear inclusion and exclusion criteria were established to ensure alignment with the research objectives and scope. Studies published between 2012 and 2025 were selected to capture recent developments in DevSecOps and its impact on application security. Only English-language articles were included to maintain consistency and facilitate interpretation. Eligible studies were required to address topics related to DevSecOps, SecDevOps, continuous security, or closely associated concepts such as "Shift-Left," "Infrastructure as Code," or "Security as Code." Studies focusing solely on traditional DevOps or those unrelated to security were excluded from the review.

2.3. Data Extraction

A systematic process was employed to extract and categorise data from the selected studies. Predefined coding schemes were developed to ensure that key themes such as research objectives, methodologies, findings, and limitations were consistently classified. For each study, relevant information was collected regarding its design, context, application domain, and specific contributions to DevSecOps practices. The data extraction process was facilitated using Covidence software (https://www.covidence.org/ accessed on 10 November 2025), which enhanced accuracy, ensured consistency, and minimised manual errors.

2.4. Analysis Method

This study employs the PRISMA framework [42] to systematically guide the screening and organisation of selected literature. Titles, abstracts, and full texts are rigorously reviewed to assess relevance and quality. Subsequently, thematic analysis is conducted to identify recurring trends and patterns related to the evolution of DevSecOps principles, technological enablers, challenges, and implementation frameworks. The synthesised findings highlight significant gaps, emerging opportunities, and provide a comprehensive perspective on DevSecOps contributions to strengthening application security.

Figure 4 presents the PRISMA Flow Diagram [42], providing a detailed overview of the systematic review process undertaken in this study to examine DevSecOps and its impact on application security.

From the initial search, a total of 368 references were imported for screening. In the deduplication process, 1 duplicate was manually identified and removed, while Covidence software identified 20 additional duplicates. This left 347 unique studies to be assessed further.

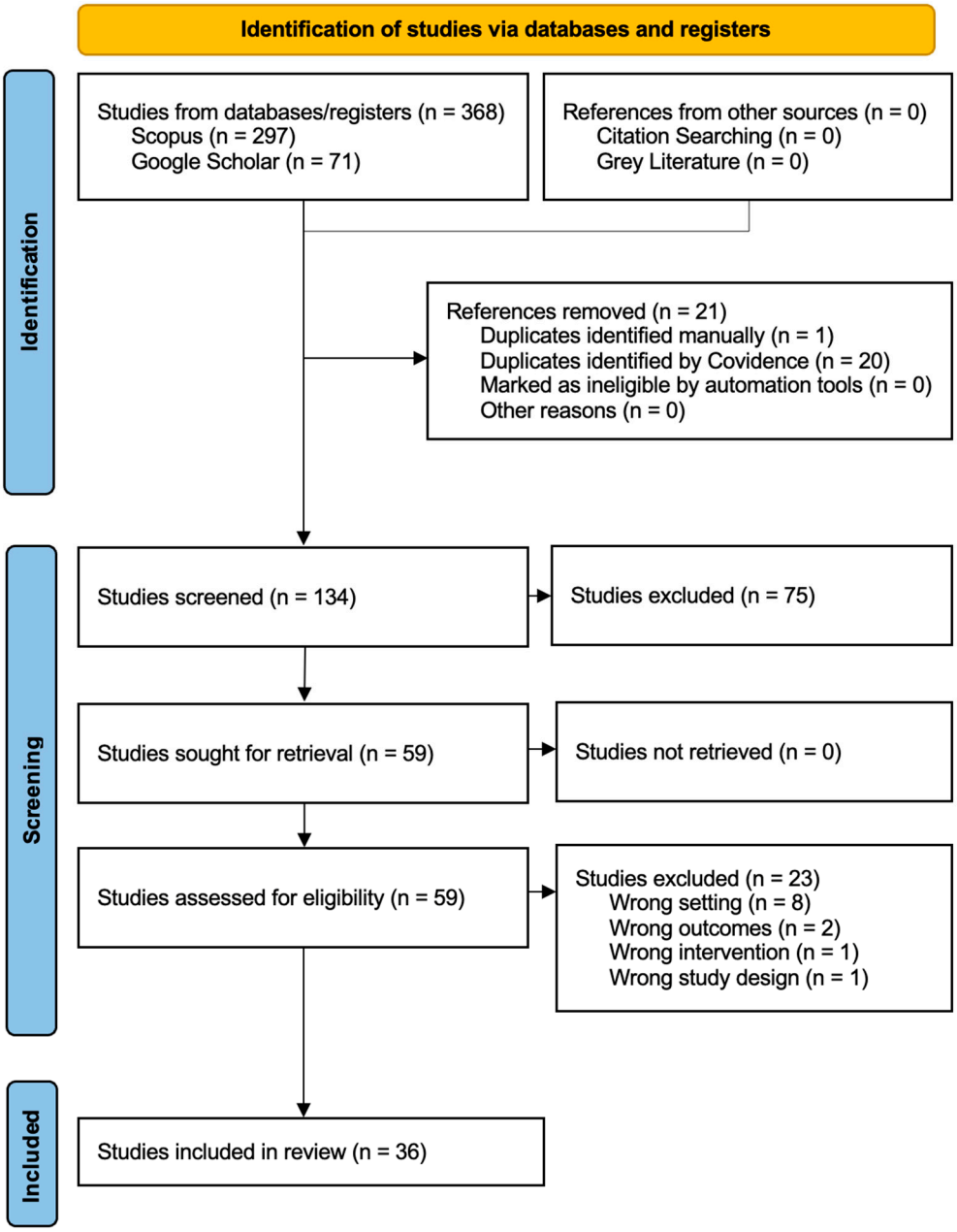


Figure 4. PRISMA Flow diagram [42].

The first level of screening included a review of the titles and abstracts of the 347 studies. From this, 75 studies were excluded because they did not meet the inclusion criteria. Of the remaining 134 studies, full texts were reviewed to assess their eligibility. 59 studies were assessed in the full-text assessment stage. Of them, 23 were excluded as follows: 12 were rejected based on inappropriate study designs, 8 based on irrelevant settings, 2 were excluded due to a non-related outcome report of a study, and 1 was excluded based on the fact that the intervention would not be included in this review. In the end, a total of 36 studies qualified and were included in the systematic review. None fell under ongoing or pending classification, which means that the review is on a completed set of good-quality relevant research. Hence, the process thoroughly addressed the systematic nature of answers to the asked questions.

3. Literature Review

DevSecOps represents a transformative approach to embedding security throughout every phase of the software development lifecycle (SDLC), addressing the limitations of traditional DevOps practices. This literature review synthesises current research to examine the evolution, core principles, frameworks, and challenges associated with DevSecOps, with particular emphasis on key innovations such as shift-left security and automation. The findings highlight the growing importance of integrating security early in development pipelines, enabled by advanced practices like Infrastructure as Code (IaC) and the adoption of AI/ML technologies. Additionally, the review identifies persistent implementation challenges including cultural adaptation, tool integration, and scalability that shape the adoption of DevSecOps. These insights provide a foundation for understanding DevSecOps as a pivotal paradigm in strengthening modern application security.

3.1. DevSecOps Evolution

The evolution from DevOps to DevSecOps marks a pivotal transformation in software engineering, shifting the focus from mere collaboration between development and operations teams to the holistic integration of security within the entire software development lifecycle (SDLC). While DevOps has been instrumental in accelerating software delivery and fostering automation, the increasing frequency and sophistication of cyber threats have exposed the limitations of traditional approaches that treat security as an afterthought. DevSecOps emerges as a response to these challenges, embedding security practices into every phase of the SDLC and promoting a culture of shared responsibility and proactive risk management.

Akbar et al. [29] provide a comprehensive analysis of the barriers to DevSecOps implementation by systematically identifying 18 distinct challenges, which are organised into 10 core categories. The study employs Interpretive Structural Modelling (ISM) to determine the hierarchical influence of these challenges, revealing that security standards exert the greatest impact. Furthermore, the use of Fuzzy TOPSIS prioritises the lack of secure coding standards, deficiencies in automated security testing tools, and insufficient knowledge of static security testing as the most pressing obstacles. A significant limitation of this study is its reliance on expert opinion for both the identification and prioritisation of challenges. This dependence introduces potential subjectivity, as the findings may reflect the biases or experiences of a specific group of experts. Consequently, the generalisability of the results across diverse organisational environments and industries is limited.

Alghawli & Radivilova [43] explore methods for enhancing security throughout all stages of software development, with a particular emphasis on automation tools such as Terraform and Jenkins. The study compares different infrastructures and cloud service providers, and notably applies the FAIR (Factor Analysis of Information Risk) methodology to enable real-time, quantitative risk assessment in cloud environments. This approach helps organisations optimise information security investments and supports the mitigation of vulnerabilities in dynamic, cloud-based systems. The primary limitation of this research is its strong focus on cloud-native and cloud-based environments. As a result, the proposed methods and findings may not be fully applicable to organisations operating with on-premises or hybrid infrastructures. This restricts the broader applicability of the study, especially for enterprises that have not fully transitioned to the cloud.

Caniglia et al. [44] investigate the challenges of implementing DevSecOps by proposing a set of metrics aimed at evaluating project performance, culminating in the introduction of the Framework of Business Index Concerning Security (FOBICS). This framework is designed to assess both security and testing practices while factoring in project duration and financial outcomes, thereby enabling organisations to monitor and align security initiatives

with business objectives. The authors validate FOBICS through its application in two real-world projects, demonstrating a positive correlation between FOBICS-derived metrics and the effectiveness of implemented security strategies. By integrating both qualitative and quantitative metrics, FOBICS provides an interpretable and actionable overview that supports secure software development and highlights the potential of DevSecOps for early vulnerability detection without incurring additional time or cost. However, the study presents several limitations. Firstly, the validation of FOBICS is restricted to only two case studies, which may not capture the full spectrum of organisational contexts or project complexities, thereby limiting the generalisability of the findings. Secondly, while FOBICS offers a comprehensive set of metrics, the framework primarily focuses on the measurement of security and business outcomes, potentially overlooking other critical dimensions such as team collaboration, cultural adoption, and the impact of evolving threat landscapes. Additionally, the framework's applicability to different software development methodologies or industries remains unaddressed.

A notable gap in FOBICS is its limited guidance on the operationalisation and continuous improvement of security practices over time. The framework does not sufficiently address how organisations can adapt the metrics in response to emerging security challenges, changes in technology stacks, or shifts in regulatory requirements. Furthermore, FOBICS lacks explicit mechanisms for integrating feedback from security incidents or post-mortem analyses, which are essential for iterative enhancement of security posture in dynamic development environments. While Caniglia et al. [44] make a significant contribution by introducing a metrics-driven approach to DevSecOps performance evaluation, future research should focus on expanding the empirical validation of FOBICS, incorporating additional dimensions such as cultural and process factors, and providing more robust mechanisms for continuous adaptation and learning within the framework.

Kumar & Goyal [45] introduce the ADOC model, a conceptual framework designed to automate security controls and embed continuous assurance into DevSecOps workflows. The model integrates a variety of tools, metrics, and use cases, aiming to support the delivery of secure, agile, and cost-effective software solutions. By transforming compliance-based activities into ongoing assurance processes, ADOC seeks to maintain development velocity without compromising on security. Despite its innovative approach, the ADOC model remains largely theoretical. The framework has not been empirically validated in real-world organisational settings, which raises questions about its practical effectiveness and scalability. Without robust case studies or empirical data, it is difficult to assess how the model performs in diverse and complex environments.

Zhang et al. [46] conduct a Multi-vocal Literature Review (MLR) encompassing 80 sources to map the rapidly evolving DevOps landscape, including 38 XOps terms and 13 main variants such as AIOps, DevSecOps, and MLOps. The study provides a comprehensive taxonomy, identifies prevailing trends and challenges, and offers a reference point for both practitioners and researchers navigating the expanding ecosystem. The study's breadth-over-depth approach, while valuable for providing a high-level overview, may overlook the nuanced challenges and unique solutions associated with specific XOps variants. The synthesis of a wide range of sources can dilute the depth of analysis for individual practices, potentially leaving gaps in understanding the detailed implementation issues of each variant.

Grande et al. [47] examine the adoption of DevOps in global software development (GSD) contexts, especially in light of the COVID-19 pandemic and the shift to remote work. By analysing 27 papers, the study maps the challenges of DevOps implementation to risks inherent in distributed teams and highlights benefits such as faster time-to-market, cost savings, and improved global collaboration. The conclusions drawn by this study are

constrained by the limited number of empirical investigations in distributed and global settings. This scarcity of real-world data may affect the robustness and reliability of the identified best practices and persistent challenges, making it difficult to generalise the findings to all GSD environments.

Bahaa et al. [34] systematically review the application of DevSecOps practices to real-time IoT security monitoring, finding that embedding automated security testing and machine learning into CI/CD pipelines facilitates early attack detection and continuous vulnerability assessment across varied IoT landscapes. Their review demonstrates how DevSecOps can enhance collaboration between development and security teams, streamline device onboarding, and enable faster response to threats; nonetheless, the literature reveals ongoing challenges including a shortage of security experts, limited resources, scalability issues for heterogeneous device ecosystems, high false positive rates, and insufficient standardisation of evaluation metrics. These unresolved issues emphasise the importance of further research to improve scalability, establish standards, and develop organisational strategies that support the effective deployment of DevSecOps for adaptive, real-time IoT security.

Sinan et al. [35] conduct a systematic literature review on integrating security controls within DevSecOps, systematically identifying numerous challenges including toolchain complexity, inconsistent security culture, limited automation capabilities, and a lack of standardised practices for embedding security throughout the software development lifecycle. Their review catalogues 19 unique challenges and 18 emerging solutions, such as automation toolkits, security-aware CI/CD processes, and targeted training programmes, yet notes that organisations still struggle with high false positive rates, inefficient manual interventions, scalability limits across heterogeneous environments, and gaps in adapting controls to evolving regulatory requirements. Critically, the authors emphasise that empirical validation is sparse, and most evaluated approaches are context-specific, underscoring the need for further rigorous studies, generalizable frameworks, and cross-industry benchmarks to enhance the practical adoption and long-term effectiveness of security controls in DevSecOps.

Pranav et al. [36] and Zhou et al. [38] both examine the evolution and industry adoption of DevSecOps, showing that automation significantly boosts cybersecurity by enabling rapid vulnerability detection and more efficient compliance processes, yet the literature identifies persistent limitations including tool incompatibility, challenges with legacy systems, fragmented security policies, and resistance to change within organisations. Additionally, Zhou et al.'s industry-focused analysis reveals further issues such as vague definitions, skill gaps, inconsistent security practices, and a lack of consensus on DevSecOps standards, highlighting the continued fragmentation of approaches. These findings indicate that, despite progress, substantive gaps in tooling, culture, scalability, and standardisation remain, and more robust empirical research and unified frameworks are needed for DevSecOps to reach its full potential in enterprise settings.

While the reviewed literature collectively advances understanding of DevSecOps and its integration into modern software engineering, each study presents limitations related to scope, methodology, or context. Addressing these gaps will be critical for developing more comprehensive and universally applicable DevSecOps strategies.

3.2. Shift-Left Security

DevSecOps aims to embed security at every stage of the Software Development Life Cycle (SDLC), emphasising the "Shift Left" approach to detect risks early and continuous vigilance, or "Securing Right", to ensure ongoing protection. The integration of automated tools such as Static Application Security Testing (SAST), Dynamic Application Security

Testing (DAST), vulnerability scanning, and threat modelling within CI/CD pipelines accelerates development, reduces costs, supports compliance, and strengthens security posture. Early risk mitigation minimises rework and potential breaches, while collaboration among development, security, and operations teams ensures the delivery of high-quality, secure software at scale by Manchana [48]. The author provides a broad overview of DevSecOps benefits and practices but lacks empirical data or case studies to substantiate the claimed improvements in efficiency, cost savings, and security outcomes. The discussion remains largely conceptual, limiting its practical applicability for organisations seeking concrete implementation guidance. Figure 5 shows how security integrates at every stage in the DevSecOps pipeline.

Kahan [49] explores the protection of cloud-native technologies through DevSecOps, highlighting security integration across all SDLC stages. The paper emphasises shifting left for early detection and securing right for continuous runtime protection, with automation and continuous monitoring as core practices. The study also addresses cultural barriers, skill gaps, and tool integration challenges, offering insights into methods and best practices for building scalable, secure cloud-native systems. While Kahan [49] offers valuable guidance on DevSecOps implementation in cloud-native environments, the analysis is primarily descriptive and lacks detailed empirical evaluation or comparative studies. The recommendations may not fully address the unique needs of organisations with legacy systems or hybrid infrastructures.

Sandu [50] discusses the transformative impact of DevSecOps in embedding security into every phase of the DevOps lifecycle, supported by real-world case studies. The paper covers automation, resilience testing, cultural change, and regulatory compliance, providing practical strategies and examples for adopting DevSecOps principles to counter evolving cyber threats. Although Sandu [50] presents practical insights and case studies, the scope is limited to select organisations and scenarios, which may not be representative of all industry contexts. Additionally, the paper does not systematically evaluate the long-term effectiveness of the recommended strategies.

Saurabh & Kumar [51] focus on Static Application Security Testing (SAST) and its integration into the development process using tools like SonarQube. The study highlights that traditional SAST is often performed too late in the DevSecOps pipeline, resulting in build delays and increased project costs. The authors advocate for earlier static analysis using platforms such as GitHub and Azure DevOps to enhance quality, efficiency, and reliability. The findings of Saurabh & Kumar [51] are primarily based on tool-centric analysis and do not comprehensively address the organisational or process-related challenges of shifting SAST earlier in the pipeline. The study's focus on specific tools may limit its generalisability to other environments or toolchains.

Achuthan & Alimohideen [14] examine the automotive industry's transition toward Software-Defined Vehicles (SDVs) and centralised E/E architectures, which increase system complexity and vulnerability exposure. The paper argues for proactive, "shift-left" security strategies and the integration of cybersecurity tools into CI/CD pipelines, noting the absence of automated security audits in current practices. A methodology is proposed to help OEMs and suppliers enhance product security. This work is industry-specific, focusing on automotive OEMs and suppliers, which may limit the applicability of its recommendations to other sectors. Furthermore, the proposed methodology lacks empirical validation through large-scale implementation or longitudinal studies.

Lee & Liu [52] stress the necessity of embedding security within DevOps lifecycles, particularly in fast-paced CI/CD environments. The paper introduces the CodeHawk platform, which automates security testing in DevSecOps pipelines, allowing developers to focus on coding while maintaining security at reasonable operational costs. The evaluation

of the CodeHawk platform is limited in scope, with insufficient detail on its performance across diverse organisational contexts or against a wide range of security threats. The study does not provide a comparative analysis with other automated security solutions.

Lombardi & Fanton [39] propose CyberDevOps, a more extreme shift-left architecture designed to fully integrate cybersecurity into every stage of the development pipeline, addressing gaps such as insufficient automation, lack of standardised security metrics, integration complexity, and limited empirical validation in practice. However, systematic reviews underscore ongoing challenges including operational friction when adopting automation tools, skills gaps, trade-offs between security and system performance, and difficulties in privacy compliance within multi-cloud environments, indicating that current frameworks, including CyberDevOps, require broader empirical studies and standardisation before widespread industry adoption can be achieved.

The reviewed literature collectively underscores the importance of integrating security into every phase of the SDLC through DevSecOps, with automation and early risk detection as key enablers. However, each study exhibits limitations, including a lack of empirical validation, narrow focus on specific tools or industries, and limited generalisability. Addressing these gaps through broader, data-driven research and cross-industry studies will be crucial for advancing effective and universally applicable DevSecOps practices.

3.3. Frameworks

Zhao et al. [41] examine DevSecOps as a security-centric evolution of DevOps, aiming to address the security deficiencies found in fast-paced DevOps environments. Through a multi-vocal literature review of 147 sources published between 2012 and 2021, the study identifies five key dimensions: definitions, challenges, practices, tools/technologies, and metrics/measurement. The authors introduce the CPTM model, a lifecycle-based framework designed to guide the understanding and application of DevSecOps, and emphasise the importance of collaboration among development, operations, and security teams. While the CPTM model provides a comprehensive roadmap for advancing DevSecOps research and practice, the study's reliance on literature up to 2021 may limit its relevance to the most recent technological advancements and emerging practices. Additionally, the model's applicability across diverse organisational and industry contexts remains insufficiently validated.

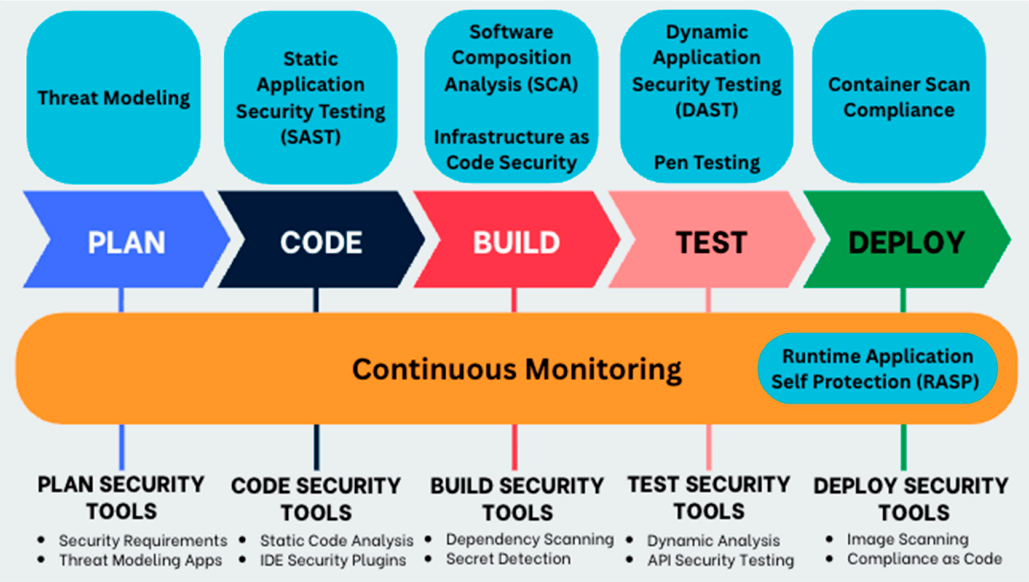


Figure 5. Security Integration Stages in DevSecOps Pipeline [40,42,43].

Abiona et al. [53] explore the transformative nature of DevSecOps, tracing its evolution, core principles, and the need to address vulnerabilities arising from outdated security practices. The paper highlights the integration of automated security testing, multi-stakeholder engagement, and a cultural shift towards prioritising security. DevSecOps is presented as a methodology that enhances cyber resilience, regulatory compliance, and customer trust while balancing speed and innovation. However, the study primarily offers a conceptual discussion with limited empirical evidence or case studies, which may restrict its practical applicability and the generalisability of its recommendations.

Nikolov & Aleksieva-Petrova [54] propose a framework for integrating threat modelling into Jenkins CI/CD pipelines, enabling early identification of vulnerabilities by incorporating threat data into automated security scans. The study identifies three main challenges in embedding security within DevOps pipelines. Despite its practical focus, the framework is validated only in the context of Jenkins, which may limit its transferability to other CI/CD platforms or development environments. Furthermore, the study does not provide a comprehensive evaluation of the framework's effectiveness in large-scale or heterogeneous settings.

Islam & Chadee [55] investigate the role of adaptive governance in building resilience among global value chain (GVC) suppliers, particularly in response to external shocks such as the COVID-19 pandemic. Using a case study of Bangladeshi apparel suppliers, the research demonstrates the importance of resilience and governance for sustaining GVC performance. While the findings offer valuable insights into organisational resilience, the study is industry-specific and does not directly address DevSecOps or application security, limiting its relevance to the core focus of this review.

Rizvi et al. [56] address the security challenges posed by the proliferation of IoT devices in enterprise networks, noting that traditional IT auditing standards are inadequate for assessing IoT infrastructure. The authors propose a modular IoT auditing framework to evaluate various security aspects, including firmware, hardware, data privacy, and communication. This framework aims to equip IT auditors with tools for compliance and breach prevention in IoT-enabled environments. However, the framework's effectiveness is not empirically validated across a wide range of IoT implementations, and its direct connection to DevSecOps practices is not explicitly established. See Table 3, which summarises industry-specific information security frameworks to address the threat with a DevSecOps focus.

Table 3. Industry-specific information security frameworks.

Industry	Information Security Framework	Primary Threats	Top DevSecOps Focus
Financial Services	PCI-DSS SOX Compliance APRA CPS 234 (Australia) Gramm-Leach-Bliley Act (USA)	Fraud Data Theft Ransomware	Transaction Security API Protection Audit Trials
Healthcare	HIPAA/HITECH FDA Regulations (USA) GDPR (EU)/CCPA (USA) FedRAMP (USA)	Patient Data Breach Medical Device Hacking Insider Threats	PHI Protection Access Controls Device Security
Government	ASD Essential Eight (Australia) FISMA (USA)/NIST State/Local Regulations	Nation-State Attacks Data Exfiltration Supply Chain Attacks	Classification Controls Advanced Monitoring Supply Chain Validation

Table 3. *Cont.*

Industry	Information Security Framework	Primary Threats	Top DevSecOps Focus
Retail/E-Commerce	PCI-DSS GDPR (EU)/CCPA (USA) ADA Compliance (USA)	Card Skimming Customer Data Theft DDoS Attacks	Payment Security Website Protection Fraud Prevention
SaaS Providers	SOC 2 ISO 27001 Multi-Tenant Controls	Data Isolation Failure API Vulnerabilities Privilege Escalation	Multi-Tenancy Security API Security Access Management

Overall, while these studies collectively advance the understanding of DevSecOps and related security challenges, each exhibits limitations in terms of scope, empirical validation, or direct relevance to DevSecOps. Addressing these gaps will be essential for developing more robust, adaptable, and universally applicable security practices in modern software development.

3.4. Infrastructure as Code (IaC)

Alonso et al. [30] examine the challenges of establishing trusted Infrastructure-as-Code (IaC) within a DevSecOps framework. The study highlights how IaC automates deployment, configuration, and management tasks, promoting repeatability, security by design, and auditability. However, the inherent complexity of IaC can introduce errors that compromise reliability. To address this, the authors propose a holistic framework that integrates DevSecOps principles, emphasising security, integrity, and self-healing throughout the IaC lifecycle. This approach combines automation, safe practices, and error prevention to ensure consistency across heterogeneous environments, ultimately enhancing robustness, reliability, and trust in dynamic IT ecosystems. The framework is primarily conceptual and lacks empirical validation through real-world case studies or quantitative assessments, which limits its proven applicability in diverse operational contexts.

Zeini et al. [57] investigate security threats within the IaC lifecycle, focusing on the importance of early collaboration among development, security, and operations teams for effective risk analysis and mitigation. The study advocates for a tailored risk management framework that integrates IaC practices into the DevSecOps culture, aiming to improve security and safe implementation across the SDLC. While the paper provides a strong conceptual foundation, it does not present empirical results or practical evaluations of the proposed framework, making it difficult to assess its effectiveness in varied organisational settings.

Ramaj et al. [58] present a conceptual framework for managing risk in critical infrastructures within a DevSecOps context. The framework is structured around action, state, and contrivance, and is designed to enhance risk understanding, incentivise best practices, and facilitate human–machine collaboration by integrating security into software pipelines. This integration aims to reduce business interruptions caused by security risks. The framework remains theoretical, with limited empirical evidence or case studies to demonstrate its practical impact or scalability in real-world critical infrastructure environments.

Nisha & Khandebharad [31] trace the evolution of software development methodologies from the waterfall model to DevOps and DevSecOps, highlighting the growing need to integrate security into agile and DevOps practices. The authors introduce a migration framework for transitioning from DevOps to DevSecOps, identify key evaluation attributes, and discuss migration challenges, strategies, and tools. The paper also examines the shift from Software as a Product (SaaP) to Software as a Service (SaaS), emphasising the transformation of delivery and deployment models in security-focused environments. This study

is largely descriptive and lacks in-depth empirical analysis or longitudinal studies on the success and challenges of migration in diverse industry settings.

Rajapakse et al. [28] provide a systematic review of 54 peer-reviewed studies to identify challenges and solutions related to DevSecOps adoption. The review categorises findings into four themes: People, Practices, Tools, and Infrastructure, identifying 21 challenges and 31 solutions. Tool-related challenges, particularly around automation, are most frequently reported, while people-related factors are highlighted as a significant area for further research. Recommendations include the adoption of shift-left security and continuous security assessment, as well as the need for user-friendly tools and the automation of manual security practices. The review is limited by the scope of the literature analysed, which may not capture the latest developments or all industry perspectives. Additionally, the study calls for more research on people-related factors, indicating a current gap in understanding the human and cultural dimensions of DevSecOps adoption.

Collectively, these studies advance the understanding of DevSecOps frameworks, risk management, and migration strategies, but each is constrained by limitations such as conceptual focus, lack of empirical validation, or limited generalisability. Addressing these gaps through practical case studies and broader industry engagement is essential for the continued evolution and effective implementation of DevSecOps practices.

3.5. AI and ML

Pakalapati, Konidena, et al. [33] explore the integration of AI and ML within DevSecOps to enhance security, efficiency, and innovation in software development. The study details approaches such as automated threat detection, predictive analytics for vulnerabilities, and intelligent automation within CI/CD pipelines. Case studies are used to illustrate the practical challenges of data privacy, algorithm transparency, and ethical considerations. The authors emphasise the transformative potential of AI/ML for fostering innovation, resilience, and agility in DevSecOps initiatives. The paper's reliance on selected case studies may limit the generalisability of its findings, and it provides limited empirical evaluation of long-term impacts across diverse organisational contexts.

Camacho [59] discusses the revolutionary impact of integrating AI and ML with DevSecOps, highlighting strategies like automated threat detection, predictive analytics, and intelligent automation in CI/CD processes. The study also addresses challenges related to data privacy, algorithm transparency, and ethics, using case studies to demonstrate how AI/ML streamlines the pipeline, reduces risk, and supports continuous improvement. The research primarily focuses on conceptual strategies and illustrative case studies, lacking comprehensive quantitative analysis or cross-industry validation to substantiate its claims.

Jose & Poulose [60] examine the application of AI techniques, specifically machine learning and natural language processing, in securing cloud environments through DevSecOps. The study highlights AI's role in enhancing threat detection, risk assessment, and automated incident response, while also addressing scalability and ethical challenges. Case studies demonstrate AI's capacity to create adaptive, secure, and efficient cloud infrastructures. The findings are based on a limited set of case studies, and the study does not provide a detailed comparative analysis of AI techniques or their effectiveness relative to traditional security approaches.

Pakalapati, Venkatasubbu, et al. [61] focus on the transformative integration of AI/ML into DevSecOps, emphasising automated threat detection, predictive analytics, and intelligent automation. The paper uses case studies to illustrate improvements in security, accelerated software delivery, and the continuous enhancement of development practices. The scope is restricted to specific case studies, with limited discussion of scalability, poten-

tial drawbacks, or challenges in broader implementation across different organisational or regulatory environments.

Al-Garadi et al. [32] provide an overview of machine learning (ML) and deep learning (DL) applications for improving IoT security. The paper reviews the rapid expansion of IoT, associated security vulnerabilities, and the application of ML/DL techniques for threat mitigation. Strengths, weaknesses, and future research directions are discussed in the context of IoT security. The review is primarily focused on IoT environments and does not directly address the integration of ML/DL into DevSecOps pipelines, which may limit its relevance for general DevSecOps practices.

Petrović [62] investigates the use of machine learning-based runtime DevSecOps, particularly the role of ChatGPT-4 in enhancing application security. The study assesses ChatGPT's potential for automating security practices, detecting vulnerabilities, and proactively mitigating threats. Findings indicate improved real-time security integration, reduced manual intervention, and increased operational resilience in DevSecOps pipelines. The evaluation is largely exploratory and does not provide extensive empirical validation or benchmarking against other AI-driven security tools.

Petrović [63] further introduces a Python API for integrating ChatGPT into DevSecOps workflows, focusing on the static analysis of Infrastructure as Code (IaC) scripts. The tool automates vulnerability checks for Ansible and Terraform scripts, aggregating and post-processing results to enhance usability for engineers and system administrators. The tool's effectiveness is demonstrated in a limited context, with insufficient assessment of its scalability, adaptability to other IaC tools, or performance in large-scale enterprise environments.

Rangaraju et al. [37] examine the integration of AI-driven strategies into DevSecOps for cloud security, emphasising how automation in vulnerability detection, threat intelligence, and compliance can improve risk management, developer collaboration, and real-time response within CI/CD pipelines. Despite these advantages, systematic literature reviews reveal key limitations such as high false positive rates, interoperability issues with cloud-native environments, increased computational demands, and a lack of standardised benchmarks and empirical validation for AI-based practices, underscoring the necessity for more scalable, privacy-conscious frameworks and extensive cross-sector evaluation to achieve broader adoption and effectiveness in robust cloud security.

Bedoya et al. [40] investigate the integration of large language models (LLMs) and security chaos engineering in DevSecOps, finding that LLMs can automate early threat identification and chaos engineering can reveal vulnerabilities not detected by standard tools, thus enhancing threat detection and system resilience in fast-paced software development environments. Despite these advancements, systematic literature indicates several limitations, including excessive reliance on automation, LLMs' limited ability to account for project-specific nuances, the risk of errors and false positives from generative tools, and difficulties in conducting scalable chaos engineering in realistic production contexts. These findings underscore the necessity for additional research into integrating human oversight, improving chaos engineering strategies, and empirically validating LLM-based security in varied real-world pipelines before widespread adoption is feasible.

Collectively, these studies highlight the significant promise of AI and ML in advancing DevSecOps practices, particularly in areas such as automated threat detection, predictive analytics, and intelligent workflow automation. However, limitations such as reliance on case studies, lack of broad empirical validation, and restricted focus on specific environments or tools underscore the need for further research to establish generalisable best practices and evaluate long-term impacts across diverse organisational settings.

Table 4 summarises research on DevSecOps across key areas such as lifecycle models, shift-left security, frameworks, Infrastructure as Code (IaC), AI/ML integration, and associated challenges. The table highlights how DevSecOps enables the seamless integration of security throughout the software development lifecycle, thereby enhancing system resilience and optimising outcomes through automation. Notably, it also identifies persistent challenges, including cultural resistance, increased complexity, ethical considerations, and the ongoing tension between development speed and robust security.

Table 4. Literature on DevSecOps Evolution, Shift-Left Security, Frameworks, Infrastructure as Code (IaC), AI/ML, and Challenges.

Category	Study	Focus Area	Key Contributions	Applications	Challenges Addressed
DevSecOps Evolution	[41]	DevSecOps Lifecycle Model	CPTM model for understanding and applying DevSecOps across lifecycle stages	Research and implementation of DevSecOps in organisations	Collaboration gaps, security integration challenges across SDLC
	[53]	Evolution and Principles of DevSecOps	Automation, multi-stakeholder engagement, security integration in DevOps	DevSecOps in modern software development	Cultural transformation, balancing speed with security priorities
Shift-Left Security	[48]	Shift-Left Security	Early risk identification, automation of SAST, DAST, vulnerability scanning in CI/CD	Accelerating software delivery, improved security posture	Delayed traditional security measures, lack of early intervention in development processes
	[49]	Shift-Left Security in Cloud	Infusing security into cloud-native technologies through DevSecOps practices	Secure cloud-native software development	Integration of security measures into CI/CD pipelines, overcoming resistance to cultural changes
	[50]	Shift-Left Security in DevSecOps	Building secure agile software development through early inclusion of security	Real-world case studies of DevSecOps adoption	Organisational resistance, regulatory compliance, resources for security automation
Frameworks	[54]	Threat Modelling in DevSecOps	Integrating threat modelling in Jenkins pipelines, automated security scans	Continuous security in CI/CD pipelines	Threat data management, early identification of vulnerabilities
	[55]	Adaptive Governance Framework	Resilience and governance in GVCs through DevSecOps, managing external shocks	Governance frameworks for resilience	Resource reconfiguration, disruption orientation, sustainability of supply chain performance
	[56]	IoT Security Framework	Modular IoT auditing framework for evaluating security features of IoT devices	IoT device security in enterprise networks	Traditional IT auditing standards, data privacy, and communication security in IoT

Table 4. *Cont.*

Category	Study	Focus Area	Key Contributions	Applications	Challenges Addressed
Infrastructure as Code (IaC)	[30]	Trusted IaC in DevSecOps	Holistic framework integrating DevSecOps with IaC principles for security and auditability	Consistent IaC practices across environments	IaC complexity, security by design, maintaining trust in evolving IT ecosystems
	[57]	Risk Management in IaC	Tailored risk management framework for IaC lifecycle	IaC in dynamic, risk-aware environments	Early-stage risk management, integration of IaC with DevSecOps culture
	[58]	Risk Management in Critical Infrastructures	Conceptual framework for managing risks in critical infrastructure under DevSecOps	Risk mitigation in critical infrastructures	High-quality information sharing, collaboration between human and machine, risk incentives
AI and ML	[33]	AI/ML in DevSecOps	Automated threat detection, predictive analytics for vulnerabilities, intelligent automation	Enhancing DevSecOps pipelines with AI/ML	Data privacy concerns, algorithm transparency, ethical AI challenges
	[59]	AI/ML in DevSecOps	Strategies for integrating AI/ML, predictive analytics, and continuous improvement	AI-optimised DevSecOps practices	Ethical concerns, scalability issues, balancing AI integration in DevSecOps workflows
	[60]	AI for Cloud Security	Enhancing threat detection, risk assessment, and incident response with AI/ML	Secure cloud environments	Scalability of AI, ethics in cloud-based security, ensuring real-time effectiveness
	[63]	ChatGPT for IaC Security	Python API for automating IaC security checks in DevSecOps	IaC scripts (Ansible, Terraform)	Vulnerability detection in IaC scripts, reducing manual intervention in security processes
Challenges	[31]	DevSecOps Migration Challenges	Framework for migrating from DevOps to DevSecOps, including integration strategies	Transitioning software delivery models	Integration of security in operations, collaboration among teams, policy updating
	[28]	DevSecOps Adoption Challenges	Identified 21 challenges and 31 solutions related to people, practices, and tools	Adoption of DevSecOps across organisations	Automation gaps, developer tool usability, balancing speed of delivery with robust security

3.6. Summary of Limitations

The reviewed literature highlights that the evolution of DevSecOps reveals several methodological and contextual limitations that constrain the broader applicability and practical impact of existing research. A key concern is the heavy reliance on expert opinion for identifying and prioritising implementation challenges. While valuable, this approach introduces subjectivity and potential bias, as the findings often reflect the perspectives of a

limited set of practitioners or organisations. Consequently, the insights may overrepresent specific organisational experiences, thereby reducing their generalisability across industries and diverse operational contexts. Similarly, some frameworks, though innovative in their integration of security and business metrics, are often validated only through a narrow set of case studies. Such validations fail to capture the wide variety of complexities present across different sectors, team structures, and project environments, further limiting the universality of proposed solutions.

The emphasis on automation, particularly in research centred on cloud-native environments, creates additional challenges to broad applicability. Many studies are conducted within contexts where organisations have already adopted advanced tooling and infrastructure, leading to insights tailored predominantly to mature DevSecOps ecosystems. Recommendations derived from environments reliant on modern tools and real-time risk assessment frameworks may not translate effectively to enterprises operating in hybrid or on-premises settings. This over-specialisation risks marginalising sectors that are still in earlier stages of DevSecOps or cloud adoption, while leaving legacy or mixed technological environments underexplored.

A recurring limitation across the literature is the absence of rigorous empirical validation for proposed frameworks, toolchains, and methodologies. Although numerous conceptual models and pipeline-level approaches demonstrate theoretical promise, their practical effectiveness remains largely untested in real-world applications. The lack of large-scale, cross-industry, or longitudinal case studies restricts the ability to assess scalability, sustainability, and adaptability in diverse organisational and cultural contexts. Studies that take a domain-specific or tool-centric perspective also tend to generate insights that are difficult to generalise, as their conclusions remain tied to particular technologies, sectors, or environmental settings.

The diversity in proposed technical solutions further contributes to uneven coverage of the DevSecOps landscape. Many frameworks focus disproportionately on either automation and measurable technical outcomes or broad conceptual guidance, while overlooking critical dimensions such as team culture, collaboration, and continuous organisational learning. Aspects like feedback mechanisms from security incidents, iterative improvement processes, and adaptation to evolving regulatory requirements are often insufficiently addressed. This creates a gap in understanding the interplay between cultural, human, and technological factors that influence successful adoption.

The body of work exploring emerging technologies, including artificial intelligence and machine learning, highlights their potential for enhancing security automation and threat detection. However, existing research in this area is largely conceptual, based on narrow case studies or highly specific applications. Demonstrations of AI- and ML-enabled solutions are typically constrained to small-scale contexts, offering limited evidence of scalability, cross-industry applicability, or feasibility at enterprise level. Moreover, ethical, operational, and integration challenges associated with deploying these technologies within DevSecOps pipelines are rarely examined in detail.

Overall, the current literature is characterised by over-dependence on conceptual work, case-specific studies, and technology-centric explorations, with insufficient empirical validation across varied contexts. This results in limitations in generalisability, scalability, and adaptability, underscoring the urgent need for comprehensive, multi-contextual, and empirical approaches to studying DevSecOps. Such advancements are essential to ensure that proposed practices and frameworks can deliver practical, sustainable, and universally relevant outcomes in the face of diverse sectoral requirements and an evolving threat landscape.

Figure 6 shows the heat map analysis reveals DevSecOps research as a rapidly evolving field characterised by distinct temporal phases and categorical specialisations. The visualisation demonstrates clear progression from foundational challenge identification in 2021–2022 toward advanced implementation strategies and emerging technology integration in 2023–2024. The uneven distribution across categories highlights both research strengths in areas like AI/ML integration and potential gaps in sustained Infrastructure as Code investigation. These patterns provide valuable guidance for researchers and practitioners seeking to identify emerging opportunities and underexplored domains within the DevSecOps landscape, ultimately contributing to more informed research prioritisation and resource allocation decisions. (Source: Heat Map generated using claude.ai)

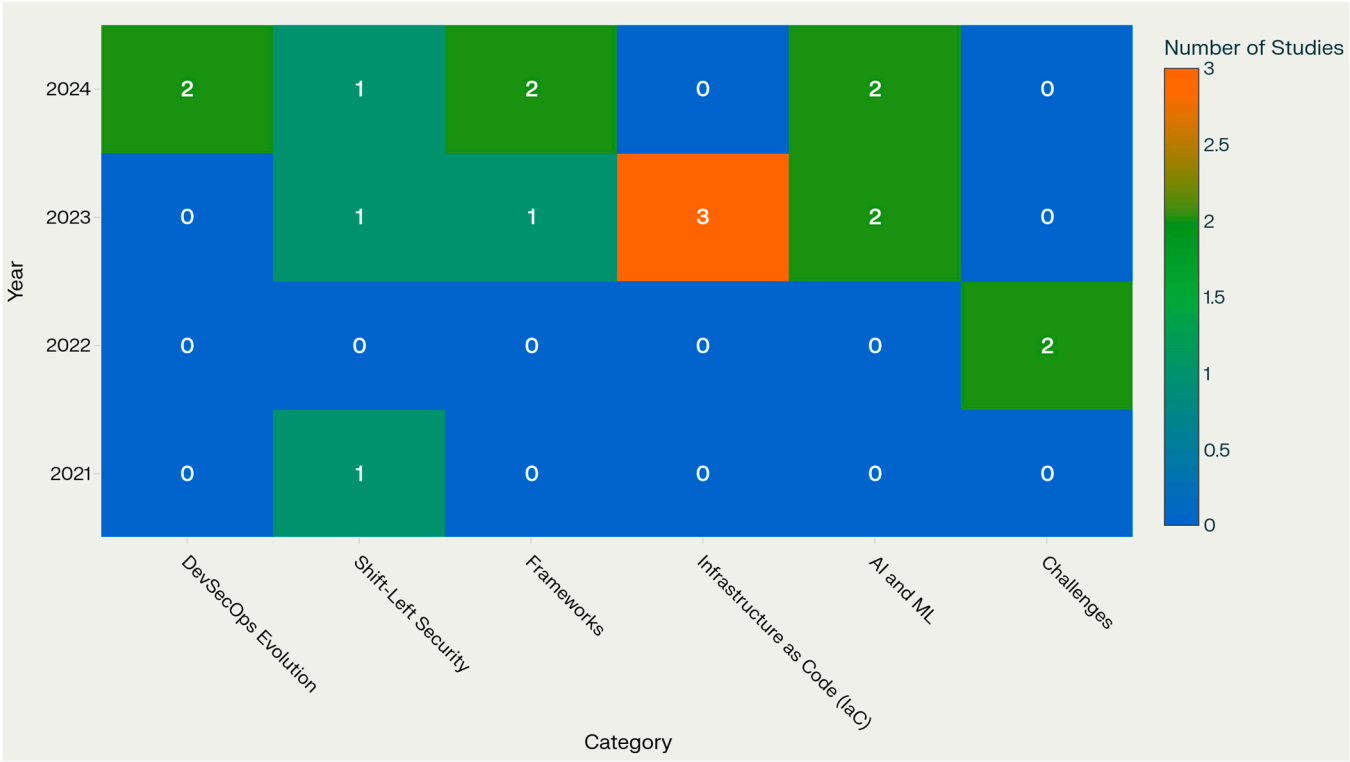


Figure 6. Year-wise heat map analysis of DevSecOps.

Figure 7 shows the heat map visualises the distribution of DevSecOps research studies across six key research categories and focus areas, providing a comprehensive overview of current research trends. The visualisation encompasses six research categories arranged as rows: DevSecOps Evolution (2 studies), Shift-Left Security (3 studies), Frameworks (3 studies), Infrastructure as Code (3 studies), AI and ML (4 studies), and Implementation Challenges (2 studies), which are analysed across six focus areas represented as columns: Security Integration, Automation & Tools, Risk Management, AI/ML Integration, Frameworks, and Implementation. The heat map employs a colour-coded intensity scale ranging from grey (0 studies) through blue (1 study), orange (2 studies), and red (3 studies) to purple (4+ studies), effectively highlighting research concentration patterns. The analysis reveals that AI/ML Integration commands the highest research attention with four studies, while Risk Management demonstrates particularly strong focus within Infrastructure as Code and AI/ML applications. Security Integration emerges as a consistently addressed theme across most research categories, and Implementation Challenges are primarily concentrated on integration and adoption aspects, indicating these as critical areas of ongoing scholarly investigation in the DevSecOps domain. (Source: Heat Map generated using claude.ai)



Figure 7. Heat map analysis of DevSecOps based on categories.

Figure 8 shows the VOS Viewer network visualisation presents a comprehensive mapping of DevSecOps research relationships, illustrating the interconnected nature of key concepts, methodologies, and research contributions within the field, though it reveals several critical gaps and areas requiring deeper analytical scrutiny. The network displays several distinct clusters of research themes, with "Security Integration" serving as a central hub connecting various specialised areas including Infrastructure as Code, Risk Management, and AI/ML applications; however, this centralisation may indicate an over-reliance on integration as a solution without sufficient focus on foundational security principles or emerging threat landscapes. The visualisation reveals strong connections between foundational DevSecOps concepts such as SAST/DAST security testing, Shift-Left Security practices, and CI/CD pipeline automation, while also highlighting emerging research areas like predictive analytics, threat modelling, and ChatGPT applications in security frameworks, yet the apparent disconnect between traditional security practices and AI-driven approaches suggests a potential fragmentation in research methodology and practical implementation strategies. Notable research contributions are represented by author nodes (including Zhao et al. [41], Manchana [48], Kahan [49], and others) that are strategically positioned to show their relationships to specific research themes and methodologies, but the temporal distribution and citation patterns suggest possible research silos where recent AI/ML developments may not be adequately building upon established security frameworks. The network demonstrates the evolution of DevSecOps research from traditional security practices toward more sophisticated approaches incorpo-

rating machine learning, cloud-native technologies, and automated threat detection, with particular emphasis on collaboration frameworks, cultural transformation challenges, and the integration of AI/ML technologies into security workflows; however, the visualisation exposes a concerning gap between technological advancement and human factors research, potentially overlooking critical implementation barriers and organisational resistance patterns. The spatial arrangement of nodes and connecting edges effectively illustrates how contemporary DevSecOps research bridges multiple domains, from infrastructure management and risk assessment to advanced analytics and framework development, reflecting the interdisciplinary nature of modern cybersecurity practices, yet the apparent clustering around specific methodologies may indicate echo chambers within the research community that could limit cross-pollination of ideas and comprehensive validation of proposed solutions across diverse organisational contexts and threat environments.

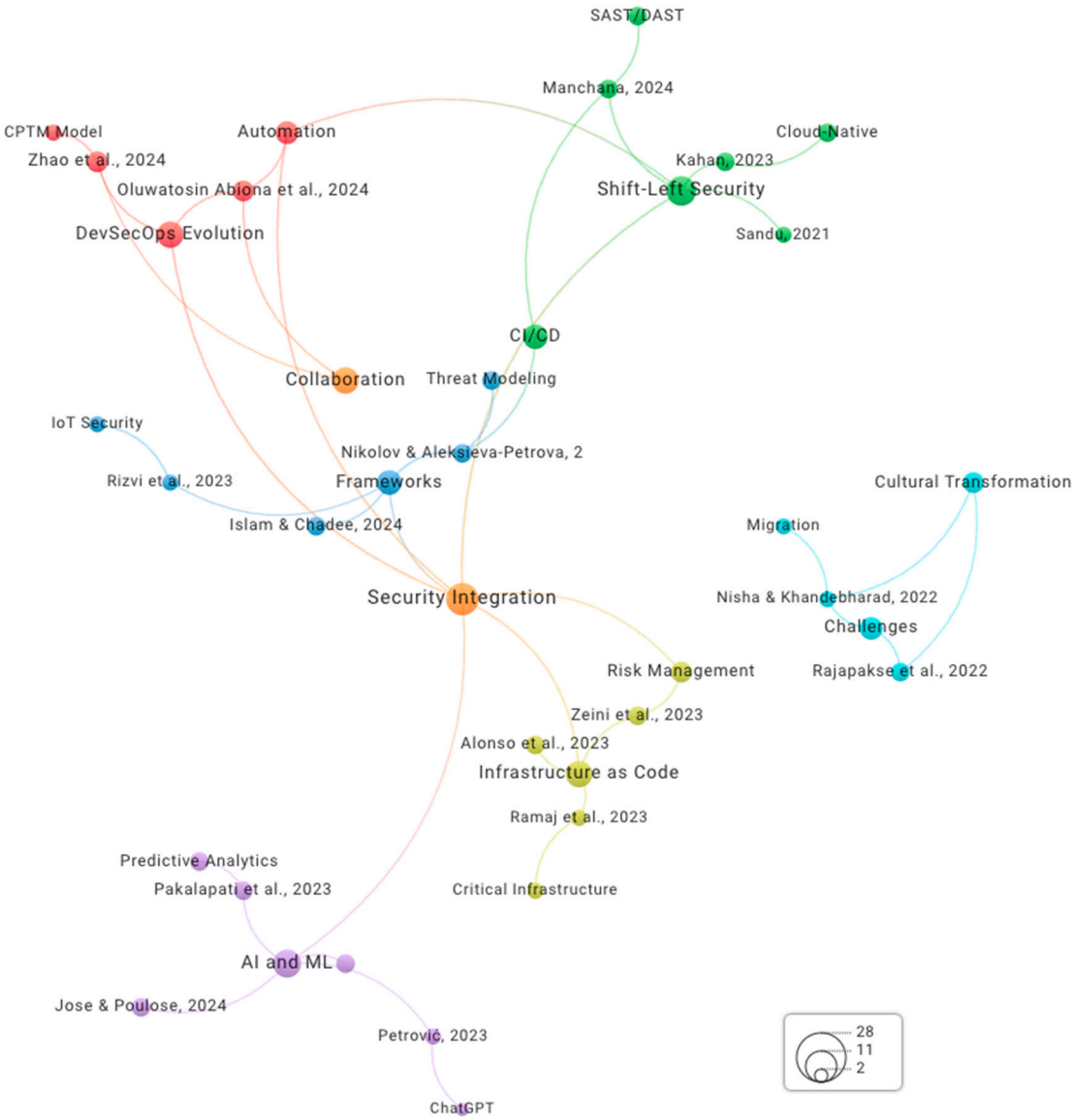


Figure 8. Visualisation using VOS Viewer [28,30,31,41,48,49,53–58,60,61,63].

4. Research Gaps

Despite significant advancements in the understanding and implementation of DevSecOps, several critical gaps persist within both research and practice. The seamless integration of security tools into CI/CD pipelines remains under-explored, especially in

complex, real-world environments. Although AI and ML technologies are widely recognised as transformative for automating threat detection and enhancing security processes, there is a notable lack of comprehensive frameworks to guide their effective deployment within DevSecOps ecosystems.

Furthermore, limited research addresses the organisational and cultural barriers to DevSecOps adoption, such as resistance to change, skill shortages, and the need for enhanced collaboration among development, security, and operations teams. The scalability of DevSecOps practices in emerging domains like IoT and cloud-native technologies also lacks robust, empirically validated models. Additionally, there is an absence of standardised metrics and evaluation methodologies for assessing the effectiveness and industry-wide impact of DevSecOps implementations.

This review underscores the necessity for a systematic approach that addresses these foundational gaps. It advocates for research and practical strategies focused on overcoming integration challenges, leveraging AI/ML applications, fostering organisational and cultural adoption, and ensuring scalability. By highlighting these areas, the study aims to advance the maturity and effectiveness of DevSecOps across diverse technological and organisational landscapes.

5. Future Work

Given the complex, evolving, and highly regulated environments characteristic of industries such as banking and financial services, government, healthcare, and critical infrastructure, future research must prioritise a comprehensive and methodologically rigorous approach to advancing DevSecOps practices. Researchers should first conduct an exhaustive systematic literature review spanning multiple regulated sectors to establish a baseline of current DevSecOps adoption, implementation maturity, and the specific constraints imposed by regulatory requirements. This review should map prevailing trends, recurrent challenges, research gaps, and novel frameworks or models adapted to these industries, allowing for a nuanced understanding of how regulations shape security and operational processes. The current review applied strict inclusion criteria to ensure rigour—considering only peer-reviewed studies with clear relevance and methodological transparency. Future research could extend this study by incorporating a structured quality assessment framework (e.g., CASP or AMSTAR) to evaluate methodological robustness in greater depth. Such an approach would strengthen the validity of findings, provide richer insights into the reliability of existing evidence, and guide more informed adoption of DevSecOps practices across diverse contexts.

Following this foundation, empirical research should employ robust quantitative methods, including the deployment of industry-wide, well-structured surveys targeting IT professionals, cybersecurity experts, compliance officers, and strategic leadership within various regulated domains. These surveys should gather comprehensive, sector-specific data about real-world challenges, barriers to adoption, benefits realised, and context-specific factors influencing DevSecOps integration. Analysing this data using advanced statistical techniques will help uncover correlations, dependencies, and causal relationships between technologies, organisational cultures, implementation frameworks, and compliance processes unique to regulated settings.

Through such analyses, researchers can systematically identify critical enablers and inhibitors of DevSecOps, such as the impact of stringent legal mandates (e.g., GDPR, HIPAA, or government security directives), institutional risk management strategies, integration of security with agile development, and organisational willingness to innovate within tight regulatory confines. The output of this data-driven inquiry should inform the design of evidence-based, domain-specific DevSecOps frameworks that balance the dual imperatives

of robust security and operational efficiency, tailored to the compliance, privacy, and risk profiles of each industry. Ultimately, sustained, iterative investigation anchored in both comprehensive literature mapping and rigorous quantitative study will be vital for defining best practices, enhancing resilience, and fostering a culture of innovation and adaptive security throughout the most regulation-intensive sectors.

6. Conclusions

The evolution of DevSecOps marks a transformative shift in software development, embedding security into every phase of the software development lifecycle. This approach responds to the increasing complexity of modern software systems by prioritising early and continuous incorporation of security measures. DevSecOps is strategically significant for overcoming the limitations of traditional security practices, enabling organisations to achieve faster delivery cycles without compromising on security. Central to this paradigm is the "shift-left" principle, which emphasises the early identification and mitigation of security risks during development. By integrating automated tools such as SAST and DAST into CI/CD pipelines, shift-left practices enhance both development efficiency and security. However, persistent challenges remain, including skill gaps, cultural resistance, and the complexity of integrating diverse security tools. Successfully scaling shift-left security requires organisations to foster collaboration among development, security, and operations teams.

This review also highlights the importance of comprehensive frameworks, such as the CPTM model and advanced threat modelling methodologies, for streamlining DevSecOps practices. These frameworks provide structured, lifecycle-based approaches to security integration and automated vulnerability management. Despite their promise, gaps in standardisation and limited global adoption persist, underscoring the need for more universally applicable solutions to support organisations in scaling and optimising their security processes.

IaC further enhances DevSecOps by automating the deployment, configuration, and management of secure environments. IaC ensures repeatability and reliability across heterogeneous systems, reducing manual errors and improving operational efficiency. Nevertheless, it introduces new risks related to complexity, early-stage collaboration, and error prevention. Integrating DevSecOps principles into IaC practices is essential for establishing security and trust in dynamic IT ecosystems.

The integration of AI and ML into DevSecOps is emerging as a game-changer, enabling automated threat detection, predictive analytics, and intelligent CI/CD automation. These technologies address scalability and efficiency challenges, as demonstrated by case studies showing increased operational resilience and reduced manual intervention. However, ethical concerns, algorithmic transparency, and data privacy remain significant hurdles that must be addressed for widespread adoption.

Despite these advancements, DevSecOps continues to face challenges such as organisational resistance, skill shortages, and the ongoing tension between rapid delivery and robust security. Addressing these issues requires a cultural shift toward shared security responsibility and targeted up-skilling initiatives. The development of user-friendly, automated tools can further bridge the gap between speed and security.

In conclusion, DevSecOps represents a critical paradigm shift for building scalable, secure, and agile software systems in an increasingly complex technological landscape. While substantial progress has been made, addressing the identified challenges through enhanced collaboration, advanced technologies, and standardised frameworks remains essential for broader adoption and maximising the effectiveness of DevSecOps in mitigating emerging cyber threats.

Author Contributions: Conceptualisation: K.I.M. and B.S.; Methodology: K.I.M. and B.S.; Validation: K.I.M. and B.S.; Formal Analysis: K.I.M. and B.S.; Resource: K.I.M.; Writing—Original Draft Preparation: K.I.M.; Writing—Review and Editing: K.I.M., B.S. and J.E.-D.; Supervision: B.S. and J.E.-D.; Project Administration: J.E.-D.; Funding: B.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SDLC	Software Development Life Cycle
IaC	Infrastructure as Code
AI	Artificial Intelligence
ML	Machine Learning
CPTM	Collaboration, Practices, Tools, and Metrics
CI/CD	Continuous Integration/Continuous Deployment
CI	Continuous Integration
CD	Continuous Deployment
DevOps	Development Operations
DevSecOps	Development Security Operations
SLR	Systematic Literature Review
LLMs	Large Language Models
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
ISM	Interpretive Structural Modelling
TOPSIS	Technique for Order Preference by Similarity to Ideal Solution
FAIR	Factor Analysis of Information Risk
FOBICS	Framework of Business Index Concerning Security
ADOC	Automated DevSecOps using Open-source software over Cloud
MLR	Multi-vocal Literature Review
AIOps	Artificial Intelligence for IT Operations
MLOps	Machine Learning Operations
XOps	Cross functional Operations
GSD	Global Software Development
COVID-19	Corona Virus Disease 2019
SAST	Static Application Security Testing
DAST	Dynamic Application Security Testing
SCA	Software Composition Analysis
RASP	Runtime Application Self Protection
SDVs	Software-Defined Vehicles
GVC	Global Value Chain
SaaP	Software as a Product
SaaS	Software as a Service
DL	Deep Learning
DDoS	Distributed Denial of Service
PCI-DSS	Payment Card Industry Data Security Standard
SOX Compliance	Sarbanes-Oxley Act
APRA CPS 234	Australian Prudential Regulation Authority's Prudential Standard CPS 234 Information Security
GLBA	Gramm-Leach-Bliley Act
HIPAA/HITECH	Health Insurance Portability and Accountability Act/Health Information Technology for Economic and Clinical Health
FDA Regulations	Food and Drug Administration Regulations

GDPR	General Data Protection Regulation
CCPA	California Consumer Privacy Act
FedRAMP	Federal Risk and Authorization Management Program
FISMA	Federal Information Security Management Act
ASD	Australian Signals Directorate
ADA Compliance	Americans with Disabilities Act
SOC 2	Service Organization Control 2

References

1. BBC News. HSBC Online Banking Is 'Attacked'. Available online: https://www.bbc.com/news/business-35438159 (accessed on 12 July 2025).
2. JISS. Cybercrime or Political Warfare? Available online: https://jiss.org.il/en/davidi-cybercrime-or-political-warfare/ (accessed on 12 July 2025).
3. Dhaka Tribune. The Great Bangladesh Cyber Heist Shows Truth is Stranger Than Fiction. Available online: https://www.dhakatribune.com/opinion/op-ed/122939/the-great-bangladesh-cyber-heist-shows-truth-is (accessed on 12 July 2025).
4. The Economic Times. 3.2 million Debit Cards Compromised; SBI, HDFC Bank, ICICI, YES Bank and Axis Worst Hit. Available online: https://economictimes.indiatimes.com/industry/banking/finance/banking/3-2-million-debit-cards-compromised-sbi-hdfc-bank-icici-yes-bank-and-axis-worst-hit/articleshow/54945561.cms (accessed on 12 July 2025).
5. CNN. A Hacker Gained Access to 100 Million Capital One Credit Card Applications and Accounts. Available online: https://www.cnn.com/2019/07/29/business/capital-one-data-breach (accessed on 29 July 2019).
6. Security Affairs. Chilean Bank BancoEstado Hit By REVil Ransomware. Available online: https://securityaffairs.com/108014/cyber-crime/bancoestado-ransomware.html (accessed on 6 September 2020).
7. Bleeping Computer. Interbank Confirms Data Breach Following Failed Extortion, Data Leak. Available online: https://www.bleepingcomputer.com/news/security/interbank-confirms-data-breach-following-failed-extortion-data-leak/ (accessed on 30 October 2024).
8. IBM. What is DevSecOps? Available online: https://www.ibm.com/think/topics/devsecops (accessed on 5 October 2021).
9. IBM. What is DevSecOps and Why Is It So Important? Available online: https://developer.ibm.com/articles/devsecops-what-and-why/ (accessed on 10 March 2022).
10. Göttel, C.; Kabir-Querrec, M.; Kozhaya, D.; Sivanthi, T.; Vuković, O. Qualitative Analysis for Validating IEC 62443-4-2 Requirements in DevSecOps. In Proceedings of the IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, Porto, Portugal, 9–12 September 2023. [CrossRef]
11. Afifah, A.S.; Kabetta, H.; Setia Buana, I.K.; Setiawan, H. Code Obfuscation in CI/CD Pipelines for Enhanced DevOps Security. In Proceedings of the 2024 International Conference on Artificial Intelligence, Blockchain, Cloud Computing, and Data Analytics, ICoABCD, Bali, Indonesia, 20–21 August 2024; pp. 137–142. [CrossRef]
12. Aktas, O.; Can, A.B. Making JavaScript Render Decisions to Optimize Security-Oriented Crawler Process. *IEEE Access* **2024**, *12*, 161688–161696. [CrossRef]
13. Gartner. DevSecOps: How to Seamlessly Integrate Security Into DevOps. Available online: https://www.gartner.com/en/documents/3463417 (accessed on 12 July 2025).
14. Achuthan, B.; Alimohideen, M.A. Shifting Gears: Integrating Security Audits into Automotive DevSecOps. In Proceedings of the 2024 International Conference on Vehicular Technology and Transportation Systems (ICVTTS), Bangalore, India, 27–28 September 2024; pp. 1–6. [CrossRef]
15. Lazarus, J.I.; Truett, L.; Fischer, B.; Kershner, C. DevSecOps Process Assessment Collaboration Tool: A Novel Method to Inject R&M Into Agile Development. In Proceedings of the Annual Reliability and Maintainability Symposium, Albuquerque, NM, USA, 22–25 January 2024. [CrossRef]
16. Yu, W.; Qian, J.; Xu, R.; Jin, C.; Fang, H.; Shi, X. Improving Substation Network Security with DevSecOps and AIOps. In Proceedings of the 2024 IEEE 10th Conference on Big Data Security on Cloud, BigDataSecurity, New York, NY, USA, 10–12 May 2024; pp. 113–118. [CrossRef]
17. Cankar, M.; Petrovic, N.; Pita Costa, J.; Cernivec, A.; Antic, J.; Martincic, T.; Stepec, D. Security in DevSecOps: Applying Tools and Machine Learning to Verification and Monitoring Steps. In Proceedings of the ICPE 2023—Companion of the 2023 ACM/SPEC International Conference on Performance Engineering, Coimbra, Portugal, 15–19 April 2023; pp. 201–205. [CrossRef]
18. Chen, T.; Suo, H. Design and Practice of Security Architecture via DevSecOps Technology. In *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, Beijing, China, 21–23 October 2022*; ICSESS; IEEE Computer Society: Washington, DC, USA, 2022; pp. 310–313. [CrossRef]

19. David, P.; Kushwaha, M.K.; Suseela, G. DevSecOps in Finance: Strengthening the Security Model of Applications. In Proceedings of the IEEE International Conference on Data Engineering and Communication Systems, ICDECS, Bengaluru, India, 22–23 March 2024. [CrossRef]
20. Ibrahim, A.; Yousef, A.H.; Medhat, W. DevSecOps: A Security Model for Infrastructure as Code over the Cloud. In *Proceedings of the MIUCC 2022—2nd International Mobile, Intelligent, and Ubiquitous Computing Conference, Cairo, Egypt, 8–9 May 2022*; Bahaa-Eldin, A., AbdelRaouf, A., Shorim, N., Refaat, S., Elbohy, S.E., Eds.; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2022; pp. 284–288. [CrossRef]
21. Leite, L.; Rocha, C.; Kon, F.; Milojicic, D.; Meirelles, P. A Survey of DevOps Concepts and Challenges. *ACM Comput. Surv.* **2019**, *52*, 127. [CrossRef]
22. Niu, X.; Yang, L.; Liu, K.; Liu, Z. Research on the Transformation Path of DevOps in the Digital Era. In Proceedings of the International Conference on Advanced Communication Technology, ICACT, Pyeong Chang, Republic of Korea, 4–7 February 2024; pp. 248–251. [CrossRef]
23. Verderame, L.; Caviglione, L.; Carbone, R.; Merlo, A. SecCo: Automated Services to Secure Containers in the DevOps Paradigm. In Proceedings of the 2023 International Conference on Research in Adaptive and Convergent Systems, Gdansk, Poland, 6–10 August 2023. [CrossRef]
24. Naidoo, R.; Möller, N. Building Software Applications Securely with DevSecOps: A Socio-Technical Perspective. In *Proceedings of the European Conference on Information Warfare and Security, ECCWS, Chester, UK, 16–17 June 2022*; Eze, T., Khan, N., Onwubiko, C., Onwubiko, C., Eds.; Curran Associates Inc.: New York, NY, USA, 2022; pp. 198–205. [CrossRef]
25. Ramaj, X.; Sánchez-Gordón, M.; Gkioulos, V.; Chockalingam, S.; Colomo-Palacios, R. Holding on to Compliance While Adopting DevSecOps: An SLR. *Electronics* **2022**, *11*, 3707. [CrossRef]
26. Aljohani, M.A.; Alqahtani, S.S. A Unified Framework for Automating Software Security Analysis in DevSecOps. In Proceedings of the International Conference on Smart Computing and Application, ICSCA, Hail, Saudi Arabia, 5–6 February 2023. [CrossRef]
27. Yadav, B.; Choudhary, G.; Shandilya, S.K.; Dragoni, N. AI Empowered DevSecOps Security for Next Generation Development. In *Communications in Computer and Information Science*; Succi, G., Kruglov, A., Ciancarini, P., Eds.; Springer Science and Business Media: Berlin/Heidelberg, Germany, 2021; Volume 1523 CCIS, pp. 32–46. [CrossRef]
28. Rajapakse, R.N.; Zahedi, M.; Babar, M.A.; Shen, H. Challenges and solutions when adopting DevSecOps: A systematic review. *Inf. Softw. Technol.* **2022**, *141*, 106700. [CrossRef]
29. Akbar, M.A.; Smolander, K.; Mahmood, S.; Alsanad, A. Toward successful DevSecOps in software development organizations: A decision-making framework. *Inf. Softw. Technol.* **2022**, *147*, 106894. [CrossRef]
30. Alonso, J.; Piliszek, R.; Cankar, M. Embracing IaC Through the DevSecOps Philosophy: Concepts, Challenges, and a Reference Framework. *IEEE Softw.* **2023**, *40*, 56–62. [CrossRef]
31. Nisha, T.N.; Khandebharad, A. Migration from DevOps to DevSecOps: A complete migration framework, challenges, and evaluation. *Int. J. Cloud Appl. Comput.* **2022**, *12*, 1–15. [CrossRef]
32. Al-Garadi, M.A.; Mohamed, A.; Al-Ali, A.K.; Du, X.; Ali, I.; Guizani, M. A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 1646–1685. [CrossRef]
33. Pakalapati, N.; Konidena, B.K.; Mohamed, I.A. Unlocking the Power of AI/ML in DevSecOps: Strategies and Best Practices. *J. Knowl. Learn. Sci. Technol.* **2023**, *2*, 176–188, ISSN: 2959-6386. [CrossRef]
34. Bahaa, A.; Abdelaziz, A.; Sayed, A.; Elfangary, L.; Fahmy, H. Monitoring real time security attacks for IoT systems using DevSecOps: A systematic literature review. *Information* **2021**, *12*, 154. [CrossRef]
35. Sinan, M.; Shahin, M.; Gondal, I. Integrating security controls in DevSecOps: Challenges, solutions, and future research directions. *J. Softw. Evol. Process* **2025**, *37*, e70029. [CrossRef]
36. Pranav, M.; Madhesh, I.; Lenin, J.; Sasikumar, R. *Advances in Devsecops and the Future of Cybersecurity Using Automation*; EAI Endorsed Transactions: Gent, Belgium, 2025. [CrossRef]
37. Rangaraju, S.; Ness, S.; Dharmalingam, R. Incorporating AI-driven strategies in DevSecOps for robust cloud security. *Int. J. Innov. Sci. Res. Technol.* **2023**, *8*, 7. [CrossRef]
38. Zhou, X.; Mao, R.; Zhang, H.; Dai, Q.; Huang, H.; Shen, H.; Li, J.; Rong, G. Revisit security in the era of DevOps: An evidence-based inquiry into DevSecOps industry. *IET Softw.* **2023**, *17*, 435–454. [CrossRef]
39. Lombardi, F.; Fanton, A. From DevOps to DevSecOps is not enough. CyberDevOps: An extreme shifting-left architecture to bring cybersecurity within software security lifecycle pipeline. *Softw. Qual. J.* **2023**, *31*, 619–654. [CrossRef]
40. Bedoya, M.; Palacios, S.; Díaz-López, D.; Vargas-Rosales, C.; Perez-Diaz, J.A. Enhancing DevSecOps practice with Large Language Models and Security Chaos Engineering. *Int. J. Inf. Secur.* **2024**, *23*, 3765–3788. [CrossRef]
41. Zhao, X.; Clear, T.; Lal, R. Identifying the primary dimensions of DevSecOps: A multi-vocal literature review. *J. Syst. Softw.* **2024**, *214*, 112063. [CrossRef]

42. Page, M.J.; McKenzie, J.E.; Bossuyt, P.M.; Boutron, I.; Hoffmann, T.C.; Mulrow, C.D.; Shamseer, L.; Tetzlaff, J.M.; Akl, E.A.; Brennan, S.E.; et al. The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ* **2021**, *372*, 71. [CrossRef]
43. Alghawli, A.S.A.; Radivilova, T. Resilient cloud cluster with DevSecOps security model, automates a data analysis, vulnerability search and risk calculation. *Alex. Eng. J.* **2024**, *107*, 136–149. [CrossRef]
44. Caniglia, A.; Dentamaro, V.; Galantucci, S.; Impedovo, D. FOBICS: Assessing project security level through a metrics framework that evaluates DevSecOps performance. *Inf. Softw. Technol.* **2025**, *178*, 107605. [CrossRef]
45. Kumar, R.; Goyal, R. Modeling continuous security: A conceptual model for automated DevSecOps using open-source software over cloud (ADOC). *Comput. Secur.* **2020**, *97*, 101967. [CrossRef]
46. Zhang, X.; Zhao, P.; Jaskolka, J. Navigating the DevOps landscape. *J. Syst. Softw.* **2025**, *223*, 112331. [CrossRef]
47. Grande, R.; Vizcaíno, A.; García, F.O. Is it worth adopting DevOps practices in Global Software Engineering? Possible challenges and benefits. *Comput. Stand. Interfaces* **2024**, *87*, 103767. [CrossRef]
48. Manchana, R. DevSecOps in Cloud Native CyberSecurity: Shifting Left for Early Security, Securing Right with Continuous Protection. *Int. J. Sci. Res.* **2024**, *13*, 1374–1382. [CrossRef]
49. Kahan, N. DevSecOps in Action: Shifting Left and Securing Right for Next-Gen Cloud-Native Security. ResearchGate. 2023. Available online: https://www.researchgate.net/publication/386076256 (accessed on 10 June 2025).
50. Sandu, A.K. DevSecOps: Integrating Security into the DevOps Lifecycle for Enhanced Resilience. *Technol. Manag. Rev.* **2021**, *6*, 1–19. Available online: https://upright.pub/index.php/tmr/article/view/131 (accessed on 10 June 2025).
51. Saurabh, S.K.; Kumar, D. Model to reduce DevOps pipeline execution time using SAST. *Int. J. Syst. Assur. Eng. Manag.* **2024**, *15*, 1999–2009. [CrossRef]
52. Lee, W.-T.; Liu, Z.-W. Microservices-based DevSecOps Platform using Pipeline and Open Source Software. *J. Inf. Sci. Eng.* **2023**, *39*, 1117–1128. [CrossRef]
53. Abiona, O.O.; Oladapo, O.J.; Modupe, O.T.; Oyeniran, O.C.; Adewusi, A.O.; Komolafe, A.M. The emergence and importance of DevSecOps: Integrating and reviewing security practices within the DevOps pipeline. *World J. Adv. Eng. Technol. Sci.* **2024**, *11*, 127–133. [CrossRef]
54. Nikolov, L.; Aleksieva-Petrova, A. Framework for Integrating Threat Modeling into a DevOps Pipeline for Enhanced Software Development. In *Proceedings of the 2024 32nd International Conference on Software, Telecommunications and Computer Networks, SoftCOM, Split, Croatia, 26–28 September 2024*; Begusic, D., Radic, J., Saric, M., Eds.; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2024. [CrossRef]
55. Islam, M.T.; Chadee, D. Adaptive governance and resilience of global value chains: A framework for sustaining the performance of developing-country suppliers during exogenous shocks. *Int. Bus. Rev.* **2024**, *33*, 102248. [CrossRef]
56. Rizvi, S.; Zwerling, T.; Thompson, B.; Faiola, S.; Campbell, S.; Fisanick, S.; Hutnick, C. A modular framework for auditing IoT devices and networks. *Comput. Secur.* **2023**, *132*, 103327. [CrossRef]
57. Zeini, A.; Lennon, R.G.; Lennon, P. Securing Infrastructure as Code (IaC) through DevSecOps:A Comprehensive Risk Management Framework. In Proceedings of the 2023 Cyber Research Conference (Cyber-RCI), Letterkenny, Ireland, 24 November 2023; pp. 1–11. [CrossRef]
58. Ramaj, X.; Colomo-Palacios, R.; Sánchez-Gordón, M.; Gkioulos, V. Towards a DevSecOps-Enabled Framework for Risk Management of Critical Infrastructures. In *Communications in Computer and Information Science*; Yilmaz, M., Clarke, P., Riel, A., Messnarz, R., Eds.; Springer Science and Business Media: Berlin/Heidelberg, Germany, 2023; Volume 1890 CCIS, pp. 47–58. [CrossRef]
59. Camacho, N.G. Unlocking the Potential of AI/ML in DevSecOps: Effective Strategies and Optimal Practices. *J. Artif. Intell. Gen. Sci.* **2024**, *2*, 79–89, ISSN:3006-4023. [CrossRef]
60. Jose, A.; Poulose, J. Harnessing the Power of AI: Transforming DevSecOps for Enhanced Cloud Security. *Int. J. Comput. Inf. Eng.* **2024**, *18*, 335–341.
61. Pakalapati, N.; Venkatasubbu, S.; Sistla, S.M.K. The Convergence of AI/ML and DevSecOps: Revolutionizing Software Development. *J. Knowl. Learn. Sci. Technol.* **2023**, *2*, 189–212, ISSN: 2959-6386. [CrossRef]
62. Petrović, N. Machine Learning-Based Run-Time DevSecOps: ChatGPT Against Traditional Approach. In Proceedings of the 10th International Conference on Electrical, Electronic and Computing Engineering, IcETRAN, East Sarajevo, Bosnia and Herzegovina, 5–8 June 2023. [CrossRef]
63. Petrovic, N. Chat GPT-Based Design-Time DevSecOps. In *Proceedings of the 2023 58th International Scientific Conference on Information, Communication and Energy Systems and Technologies, ICEST, Nis, Serbia, 29 June–1 July 2023*; Doncov, N.S., Stankovic, Z.Z., Stosic, B.P., Eds.; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2023; pp. 143–146. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.