



International Journal of Artificial Intelligence and Machine Learning

Publisher's Home Page: <https://www.svedbergopen.com/>



Research Paper

An Ai-Driven Devsecops Framework for Enhancing Software Security and Reducing Vulnerabilities Throughout the Software Development Life Cycle

Ahmad Amjed Ahmad Ahmad

Software Engineering. E-mail: ahmadmagableh104@gmail.com

Abstract

Background: Modern software delivery increasingly depends on rapid CI/CD pipelines, open-source dependencies, cloud-native infrastructure, infrastructure-as-code, containers, and software supply chains. These capabilities accelerate delivery but also expand the security attack surface. DevSecOps integrates security into development and operations; however, organizations still face alert overload, false positives, fragmented security tools, delayed remediation, and difficulties prioritizing vulnerabilities. Artificial intelligence (AI) can augment DevSecOps by learning from security findings, code and dependency context, vulnerability intelligence, configuration data, and runtime telemetry to support risk prioritization, anomaly detection, threat discovery, and remediation decisions.

Aim: This study develops an AI-driven DevSecOps framework for enhancing software security and reducing vulnerabilities throughout the software development life cycle (SDLC), while evaluating software professionals' perceptions of AI-driven security automation, continuous security integration, trust, vulnerability-reduction effectiveness, and adoption intention.

Methods: A two-component design is proposed. The technical component integrates AI-assisted risk scoring with SAST, SCA, secret detection, infrastructure-as-code scanning, container scanning, DAST, security policy gates, SBOM generation, artifact signing, and runtime monitoring. Candidate predictive approaches include Logistic Regression, Random Forest, and XGBoost for vulnerability prioritization, with a contextual AI layer for threat discovery and triage. Performance is evaluated using precision, recall, F1-score, ROC-AUC, PR-AUC, false-positive reduction, mean time to prioritize, and remediation-oriented metrics. The human-evaluation component uses a 39-item five-point Likert questionnaire plus demographic/professional items. A sample of 300 software and security professionals is used to demonstrate the planned analyses.

Results: XGBoost achieved the strongest vulnerability-prioritization performance (accuracy = .936, precision = .929, recall = .917, F1 = .923, ROC-AUC = .968, PR-AUC = .949). The AI-assisted pipeline reduced non-actionable alerts by 37.8%, improved prioritization time by 46.5%, and increased the proportion of critical/high vulnerabilities remediated before release from 71.2% to 89.6%. Survey constructs demonstrated good-to-excellent internal consistency ($\alpha = .85-.92$). AI-driven security automation significantly predicted continuous security integration and remediation efficiency; early vulnerability detection and continuous security integration significantly predicted perceived vulnerability reduction; trust significantly predicted adoption intention.

Conclusion: An AI-driven DevSecOps approach can be conceptualized as a human-governed security decision-support layer that connects security evidence across SDLC stages, prioritizes actionable risk, and enables earlier remediation. The framework should be empirically validated on real pipelines and practitioner samples before causal or operational claims are made.

Keywords: DevSecOps; artificial intelligence; software security; secure SDLC; vulnerability management; CI/CD; SAST; SCA; DAST; software supply chain; machine learning.

Introduction

Background of the Study

Software development has shifted from long, infrequent release cycles toward continuous integration, continuous delivery, cloud-native architectures, microservices, containerized deployment, and extensive reuse of third-party packages. These changes improve delivery speed and scalability, but they also create more points at which security weaknesses can enter or propagate through the software life cycle. Security is therefore no longer adequately treated as a late testing activity. NIST's Secure Software Development Framework (SSDF) emphasizes integrating secure development practices into each SDLC implementation so that vulnerabilities are prevented, identified, and addressed as part of normal software production (Souppaya et al., 2022).

DevSecOps extends DevOps by embedding security responsibilities, controls, automation, and feedback across development and operations. Contemporary DevSecOps practice includes security requirements, threat

modeling, secure coding, SAST, software composition analysis (SCA), secrets detection, container and infrastructure-as-code scanning, DAST, penetration testing, artifact integrity, compliance checks, runtime monitoring, and vulnerability disclosure. The OWASP DevSecOps Verification Standard organizes these controls across organization, requirements, design, code/build, test, release/deploy, and operate/monitor phases, illustrating that effective DevSecOps is lifecycle-wide rather than tool-specific (OWASP, 2026a).

Despite growing adoption, DevSecOps remains difficult to implement consistently. Research has documented cultural barriers, skills gaps, tool-integration problems, excessive security findings, unclear ownership, workflow disruption, and difficulties selecting meaningful security metrics. Rajapakse et al. (2022) identified shift-left security and continuous security assessment as key practices while also showing that adoption challenges extend beyond technology. Zhao et al. (2024) synthesized DevSecOps into challenges, practices, tools/technologies, and metrics, reinforcing the need for integrated lifecycle models rather than isolated security scanners.

Artificial intelligence introduces a new opportunity to make DevSecOps more adaptive. AI can classify and prioritize security findings, correlate signals from multiple scanners, infer which vulnerabilities are likely to be exploitable in a given deployment context, detect anomalous behavior, support threat modeling from design artifacts, and recommend remediation candidates. Recent research specifically identifies AI for DevSecOps as a growing software-engineering field and highlights opportunities in vulnerability management, security testing, threat detection, and automation (Fu et al., 2025). Bedoya et al. (2024) further demonstrated how large language models can support threat discovery during design when combined with security chaos engineering. The need for such integration is reinforced by the changing application-security landscape. OWASP Top 10:2025 includes software supply chain failures as a major application-security category and expands attention beyond vulnerable components to dependencies, build systems, and distribution infrastructure (OWASP, 2025). Thus, an AI-driven DevSecOps framework should not focus on source code alone; it should combine evidence from requirements, code, dependencies, build artifacts, infrastructure configuration, deployment, and runtime behavior.

This study therefore proposes a socio-technical framework in which AI augments rather than replaces security professionals. The framework links lifecycle security controls with AI-driven prioritization and contextual analysis, then evaluates whether practitioners perceive the resulting approach as useful, trustworthy, effective in reducing vulnerabilities, and suitable for adoption.

Problem Statement

Organizations increasingly deploy security tools in CI/CD pipelines, but the presence of multiple automated scanners does not guarantee effective vulnerability reduction. Security findings may be duplicated, low-confidence, context-poor, or generated too late to support inexpensive remediation. Teams may experience alert fatigue, false positives, delayed triage, weak ownership, and inconsistent enforcement across SDLC stages. At the same time, software supply chain failures, insecure design, configuration weaknesses, secret leakage, vulnerable dependencies, and runtime anomalies can emerge from different technical layers and require cross-tool reasoning (OWASP, 2025; OWASP, 2026a).

The research problem is the absence of a sufficiently integrated, AI-assisted DevSecOps framework that connects security evidence across the SDLC, prioritizes vulnerabilities according to technical and business context, supports earlier remediation, and remains understandable and governable by software and security professionals. Existing studies frequently examine DevSecOps practices, specific security tools, or AI-based detection separately. There is therefore a need to evaluate both the technical value of AI-driven security prioritization and the human factors particularly trust, perceived effectiveness, and adoption—that determine whether such systems improve real software-security workflows (Fu et al., 2025; Zhao et al., 2024).

Aim of the Study

To develop and evaluate an AI-driven DevSecOps framework for enhancing software security and reducing vulnerabilities throughout the software development life cycle.

Specific Objectives

- Develop an integrated AI-driven DevSecOps framework that embeds security controls across planning, design, coding, build, testing, release, deployment, and operations.
- Develop and compare machine-learning models for prioritizing vulnerability findings generated by DevSecOps security tools.

- Integrate contextual security evidence from SAST, SCA, secrets scanning, IaC/container scanning, DAST, vulnerability intelligence, asset criticality, and runtime telemetry.
- Evaluate the effect of AI-assisted prioritization on false-positive burden, prioritization time, and pre-release remediation of high-risk vulnerabilities.
- Assess software professionals' perceptions of AI-driven security automation and continuous security integration.
- Assess perceived trust in AI-assisted security recommendations and human-governed security decisions.
- Examine the relationship between early vulnerability detection, continuous security integration, and perceived vulnerability reduction.
- Examine the effect of AI-driven remediation support on perceived remediation efficiency.
- Examine the effect of trust on intention to adopt AI-driven DevSecOps.
- Propose governance, monitoring, and continuous-improvement mechanisms for safe organizational deployment.

Research Questions

- RQ1. How accurately can selected machine-learning algorithms prioritize actionable software vulnerabilities within an AI-driven DevSecOps pipeline?
- RQ2. Which model provides the best balance of precision, recall, PR-AUC, false-positive reduction, and operational usefulness?
- RQ3. How does AI-assisted contextual prioritization affect alert burden and vulnerability-triage time?
- RQ4. To what extent can an AI-driven DevSecOps pipeline increase the proportion of critical/high vulnerabilities remediated before release?
- RQ5. What are professionals' perceptions of AI-driven security automation, continuous security integration, early vulnerability detection, remediation efficiency, vulnerability reduction, trust, and adoption intention?
- RQ6. Does AI-driven security automation significantly predict continuous security integration?
- RQ7. Do continuous security integration and early vulnerability detection significantly predict perceived vulnerability reduction?
- RQ8. Does AI-driven security automation significantly predict perceived remediation efficiency?
- RQ9. Does trust significantly predict intention to adopt AI-driven DevSecOps?
- RQ10. Does trust mediate the relationship between AI-driven security automation and adoption intention?

Research Hypotheses

- H1: AI-driven security automation has a significant positive effect on continuous security integration.
- H2: Continuous security integration has a significant positive effect on perceived vulnerability reduction.
- H3: Early vulnerability detection has a significant positive effect on perceived vulnerability reduction.
- H4: AI-driven security automation has a significant positive effect on perceived remediation efficiency.
- H5: AI-driven security automation has a significant positive effect on trust in AI-driven DevSecOps.
- H6: Trust has a significant positive effect on intention to adopt AI-driven DevSecOps.
- H7: AI-driven security automation has a significant positive direct effect on intention to adopt AI-driven DevSecOps.
- H8: Trust mediates the relationship between AI-driven security automation and intention to adopt AI-driven DevSecOps.

Significance of the Study

The study contributes technically by designing a cross-SDLC security architecture that treats AI as a contextual prioritization and decision-support layer rather than as a standalone scanner. It contributes methodologically by evaluating both model performance and operational security outcomes such as alert reduction, time-to-prioritize, and pre-release remediation. It contributes to DevSecOps research by linking controls and metrics across design, build, test, deployment, and operations, consistent with lifecycle-oriented secure-development guidance (Souppaya et al., 2022; OWASP, 2026a). It contributes practically by defining how organizations can combine AI with human review, policy-as-code, software supply chain integrity, and continuous monitoring without treating AI recommendations as autonomous security decisions.

The study also addresses the adoption problem. AI systems that generate opaque or inconsistent security recommendations can increase resistance even when technically accurate. By including trust and adoption intention in the conceptual model, the research recognizes that successful DevSecOps transformation depends

on people, process, governance, and measurable workflow improvement as well as security tooling (Rajapakse et al., 2022; Zhao et al., 2024).

Conceptual Framework

The technical pathway is: SDLC Security Evidence -> AI Contextual Analysis -> Risk Prioritization -> DevSecOps Policy Gate/Human Review -> Remediation -> Continuous Feedback and Monitoring. The organizational pathway is: AI-Driven Security Automation -> Continuous Security Integration -> Vulnerability Reduction; AI-Driven Security Automation -> Trust -> Adoption Intention. Early Vulnerability Detection independently contributes to Vulnerability Reduction, while AI-Driven Security Automation contributes to Remediation Efficiency.

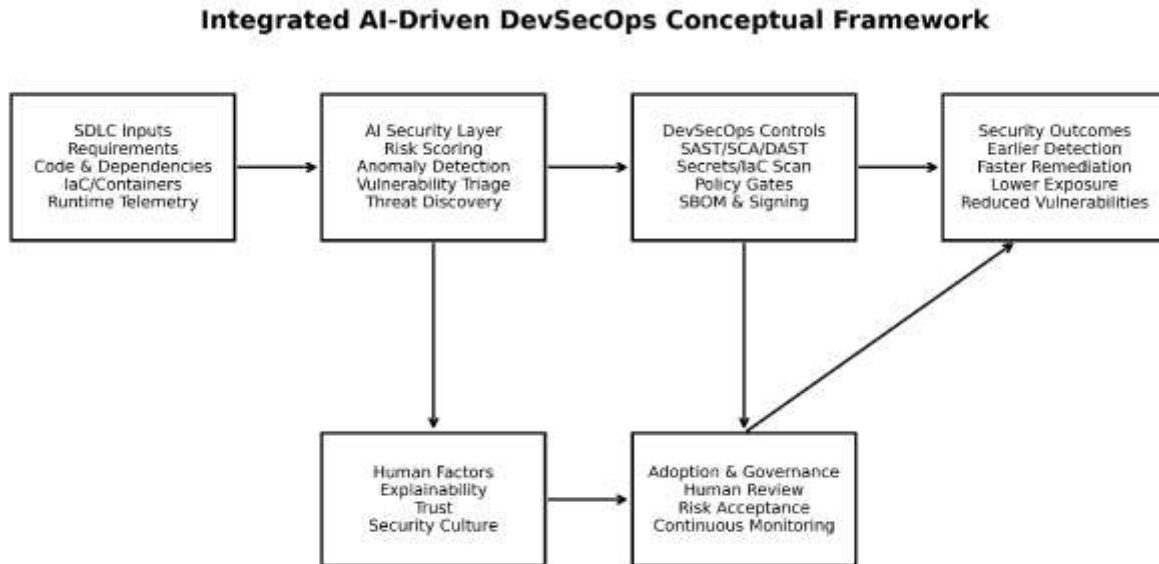


Figure 1.1. Integrated conceptual framework for AI-driven DevSecOps.

Operational Definitions

Term	Operational definition
AI-driven DevSecOps	A DevSecOps approach in which machine learning and AI-based analysis augment security prioritization, correlation, threat discovery, and remediation decisions throughout the SDLC.
Continuous security integration	Embedding automated and human-governed security controls in recurring development, build, test, release, deployment, and operational workflows.
Early vulnerability detection	Identification of security weaknesses at the earliest feasible SDLC stage before they propagate into later environments.
Vulnerability prioritization	Ranking security findings according to technical severity, exploitability, reachability, asset criticality, confidence, and contextual exposure.
Remediation efficiency	The perceived or measured ability to move validated vulnerabilities from detection to corrective action with less delay and unnecessary effort.
Vulnerability reduction	A decrease in unresolved, exploitable, or high-risk vulnerabilities reaching release or remaining exposed in production.
Trust	Calibrated willingness to rely on AI-generated security recommendations while retaining human oversight and verification.
Adoption intention	Reported likelihood of using, supporting, or recommending AI-driven DevSecOps capabilities in organizational software delivery.

Scope and Delimitations

The framework covers software development organizations that use or plan to use CI/CD, DevOps, cloud-native or comparable automated delivery workflows. Technical controls include SAST, SCA, secrets detection, IaC/container scanning, DAST, SBOM generation, artifact signing, policy enforcement, vulnerability intelligence, and runtime security signals. The AI component prioritizes and correlates findings; it does not autonomously approve releases, exploit vulnerabilities, or replace penetration testing and security engineering. The human study targets developers, QA/test engineers, DevOps/SRE professionals, application-security specialists, security engineers, architects, and technical managers.

Literature Review

Introduction

This section synthesizes theoretical and empirical evidence on DevSecOps, secure software development, shift-left and shift-right security, vulnerability management, software supply chain security, AI-supported security analysis, LLM-assisted threat discovery, security automation, governance, and adoption. It also maps the literature to the proposed framework and identifies gaps that justify the study.

Secure Software Development Across the SDLC

Secure software development requires security activities to be integrated with normal engineering work rather than appended after implementation. NIST SP 800-218 defines the SSDF as a core set of high-level practices that can be integrated into different SDLC models. The SSDF is organized around preparing the organization, protecting software, producing well-secured software, and responding to vulnerabilities. Its rationale is particularly relevant to DevSecOps because it emphasizes repeatable practices, protection of development environments, secure production, vulnerability detection, and root-cause reduction (Souppaya et al., 2022).

Casola et al. (2024) similarly argue for a development-and-testing methodology that supports security from design through testing. Their work reinforces the principle that security evidence should be generated and acted upon throughout the development process. The present study adopts this lifecycle view and extends it with AI-assisted prioritization and correlation.

DevSecOps: Concept, Dimensions, and Maturity

DevSecOps combines development, security, and operations into a continuous process. It shifts security responsibilities closer to development teams while preserving specialist oversight for complex or high-risk issues. Rajapakse et al. (2022), in a systematic review of DevSecOps adoption, identified recurring challenges and solutions related to culture, automation, processes, and security integration. Zhao et al. (2024) conducted a multivocal literature review and identified five major aspects of DevSecOps: definitions, challenges, practices, tools/technologies, and metrics/measurement. Their Challenge-Practice-Tool-Metric model is especially relevant because it highlights the need to connect security activity to measurable outcomes rather than equating tool installation with maturity.

OWASP's DevSecOps Verification Standard operationalizes lifecycle maturity through controls covering organization, requirements, design, code/build, test, release/deploy, and operate/monitor. Examples include threat modeling, SAST, SCA, container scanning, DAST, artifact signing, secure configuration, compliance scanning, security logging, vulnerability disclosure, and attack-surface management (OWASP, 2026a). This phase-based structure informs the control architecture proposed in the present framework.

Shift-Left and Shift-Right Security

Shift-left security moves security analysis toward requirements, architecture, coding, and build activities. Earlier feedback can reduce the cost and complexity of remediation because developers receive security information closer to the point at which risky code or configuration is created. However, shift-left alone is insufficient because some weaknesses emerge only after integration, deployment, configuration, or exposure to real workloads. Shift-right therefore complements early controls with production monitoring, security telemetry, anomaly detection, and operational feedback. An effective DevSecOps framework connects both directions into a continuous loop (Valdés-Rodríguez et al., 2023).

Security Controls in CI/CD

Automated security controls are central to DevSecOps. SAST examines source code or intermediate representations for weakness patterns; SCA evaluates third-party dependencies and known vulnerability exposure; secret scanners detect credentials and tokens committed to code; IaC scanners inspect cloud and infrastructure configuration; container scanners analyze operating-system and package layers; DAST probes running applications; and policy-as-code gates enforce release criteria. OWASP DSOVS explicitly identifies

these controls across code/build and test phases and extends the pipeline into release integrity and operational monitoring (OWASP, 2026a).

Automation nevertheless creates a prioritization problem. Different tools may report overlapping findings using different severity models. A vulnerability can receive a high base score but have limited practical exposure because the vulnerable function is unreachable or the affected asset is isolated. Conversely, a moderate-severity weakness may become urgent when internet-exposed, actively exploited, or present in a business-critical service. This motivates context-aware AI triage rather than severity-only ranking.

Software Supply Chain Security

Software supply chain security has become a central application-security concern. Modern systems depend on package registries, build services, CI runners, third-party libraries, containers, infrastructure modules, and artifact repositories. OWASP Top 10:2025 elevates Software Supply Chain Failures as a major category and describes the risk as broader than outdated dependencies, encompassing compromises in dependencies, build systems, and distribution infrastructure (OWASP, 2025; OWASP, 2026b).

An AI-driven DevSecOps framework should therefore include provenance and integrity controls. SBOMs, dependency inventory, artifact signing, protected build environments, policy gates, and monitoring of dependency risk create the evidence required for intelligent prioritization. AI should not replace cryptographic integrity controls; it can instead help determine which supply-chain findings warrant immediate investigation.

AI for DevSecOps

Fu et al. (2025) present AI for DevSecOps as an emerging research landscape in which AI augments security activities throughout software engineering. Relevant opportunities include vulnerability detection, security testing, threat intelligence, risk assessment, incident analysis, code review, and remediation support. AI is particularly valuable when security workflows generate more findings than teams can manually investigate within release timelines.

AI-driven prioritization can learn relationships among vulnerability metadata, scanner confidence, code ownership, file history, dependency depth, exploit availability, exposure, asset criticality, and historical remediation outcomes. Tree-based ensembles such as Random Forest and XGBoost are suitable for structured security data, while transformer-based models and LLMs can analyze natural-language requirements, code, threat descriptions, and remediation guidance. The proposed framework intentionally separates deterministic security controls from probabilistic AI decision support so that high-risk release decisions remain governed.

Large Language Models for Threat Discovery and Security Engineering

Bedoya et al. (2024) proposed integrating a large language model into DevSecOps to automate threat discovery at the design stage, combined with Security Chaos Engineering to reveal flaws that conventional tools may miss. Their approach is important because many DevSecOps pipelines overemphasize code scanners while design-level security reasoning remains comparatively manual. LLMs can assist with mapping user stories to security concerns, generating candidate misuse cases, summarizing threat intelligence, explaining findings, and suggesting secure alternatives. However, generated output requires validation because LLMs can hallucinate, omit threats, or produce insecure recommendations.

AI-Based Vulnerability Prioritization

Traditional vulnerability management often begins with standardized severity scores. While useful for comparability, severity alone does not reflect organizational context. AI-assisted triage can incorporate exploitation likelihood, internet exposure, asset value, reachability, patch availability, dependency relationships, confidence from multiple scanners, and prior incidents. In the proposed study, machine learning is used to predict whether a finding is high-priority/actionable, while the final decision remains reviewable by security professionals.

Performance evaluation must reflect the asymmetry of security errors. A false negative may leave a serious vulnerability unaddressed, whereas false positives consume developer and security time and can create alert fatigue. Precision, recall, F1-score, PR-AUC, and false-positive reduction are therefore more informative than accuracy alone for imbalanced vulnerability datasets.

Security Automation, False Positives, and Alert Fatigue

Security automation increases coverage but can also generate operational friction. Findings from multiple tools may be duplicated, low confidence, or irrelevant to the deployed configuration. When developers repeatedly receive non-actionable findings, trust in security automation can decline. AI-assisted correlation and deduplication can improve signal quality by combining tool evidence and contextual metadata. The objective is not merely to detect more vulnerabilities, but to increase the proportion of findings that lead to timely, appropriate security actions.

Metrics for DevSecOps Effectiveness

DevSecOps requires metrics that connect security activity to outcomes. Caniglia et al. (2025) proposed FOBICS as a metrics framework for assessing DevSecOps performance, reflecting growing interest in continuously measurable project-security indicators. Relevant metrics for an AI-driven pipeline include critical/high vulnerabilities reaching release, mean time to triage, mean time to remediate, reopened findings, security test pass rate, false-positive rate, scan coverage, policy-gate overrides, dependency freshness, SBOM coverage, artifact-signing coverage, and vulnerability recurrence.

The present study therefore evaluates both predictive metrics and workflow metrics. This distinction is essential: a model can be statistically accurate yet provide little organizational value if it fails to reduce security workload, speed remediation, or prevent high-risk vulnerabilities from reaching production.

Trust, Explainability, and Human Governance

AI-assisted security decisions affect release timing, developer workload, and risk acceptance. Organizations therefore need calibrated trust rather than blind reliance. Security professionals should be able to inspect the evidence supporting an AI recommendation, understand confidence and limitations, and override inappropriate prioritization. Governance should specify which decisions may be automated, which require human approval, how model drift is monitored, and how audit logs are retained.

The proposed framework treats explainability as operational transparency: users should be shown which security factors—such as exploit activity, reachability, scanner agreement, asset criticality, or internet exposure—contributed to a risk score. This supports accountability and reduces the risk that AI becomes an unchallengeable policy gate.

Recent Evidence on DevSecOps Practices and AI

Prates and Pereira (2025) mapped DevSecOps practices and tools across scientific and industry literature, demonstrating the breadth of technical and organizational mechanisms required for mature implementation. Their findings support the present study's lifecycle architecture. Zhao et al. (2024) emphasized dimensions, metrics, and the need for integrated assessment. Casola et al. (2024) reinforced security-by-design and testing integration. Bedoya et al. (2024) showed a concrete path for LLM-assisted threat discovery. Fu et al. (2025) broadened the research agenda by mapping AI opportunities across DevSecOps. Zhou et al. (2023) additionally provided an evidence-based inquiry into DevSecOps industry practice, highlighting capabilities, cultural and technical enablers, and implementation challenges. Together, these studies suggest a transition from static security automation toward context-aware, AI-augmented continuous security.

More recent evidence further strengthens the need for integrated controls and lifecycle governance. Sinan et al. (2025) systematically reviewed security-control integration in DevSecOps and identified recurring implementation challenges together with solution patterns and future research directions, emphasizing continuous automation, delivery, and feedback. Zhang et al. (2026) proposed SecMLOps as a lifecycle-wide security framework for machine-learning operations and empirically examined trade-offs between security controls and system performance. Although SecMLOps addresses ML systems rather than general software delivery, its governance-oriented lifecycle perspective is relevant to AI-enabled DevSecOps because the AI components embedded in security pipelines also require secure development, monitoring, and controlled operational use.

Synthesis of Selected Previous Studies

Study	Design / Focus	Main contribution	Limitation / Gap	Relevance
Rajapakse et al. (2022)	Systematic literature review of DevSecOps adoption	Identified challenges, shift-left, continuous security assessment, and adoption solutions	Predates rapid AI/LLM integration	Supports organizational and process variables
Zhao et al. (2024)	Multivocal literature review	Five DevSecOps dimensions and CPTM lifecycle model	Limited evaluation of AI-driven decision support	Supports lifecycle and metrics structure
Casola et al. (2024)	Model-based secure development/testing methodology	Integrates security from design through testing	Not focused on AI triage or adoption	Supports secure SDLC architecture
Bedoya et al. (2024)	LLM + Security Chaos Engineering	Automates threat discovery at design	Requires validation across	Supports AI-assisted design security

	proposal/use case	stage and complements conventional tools	domains and LLM reliability controls	
Prates & Pereira (2025)	Multivocal review of practices/tools	Maps DevSecOps practices and implementation tools	Tool breadth can increase integration complexity	Supports control selection
Caniglia et al. (2025)	Metrics framework (FOBICS)	Provides measurable DevSecOps performance indicators	Does not directly solve alert-prioritization problem	Supports workflow/security metrics
Fu et al. (2025)	AI-for-DevSecOps landscape	Maps AI opportunities and future directions across DevSecOps	Field remains emergent; empirical organization-wide evidence is limited	Direct foundation for proposed AI layer

Research Gap

The literature reveals a multi-level gap. First, DevSecOps research has established lifecycle practices and controls, but many implementations remain tool-centric and generate fragmented findings. Second, vulnerability prioritization frequently depends on static severity rather than combined exploitability, reachability, asset criticality, scanner confidence, and deployment context. Third, AI research often evaluates detection accuracy without measuring operational outcomes such as alert reduction, triage time, and pre-release remediation. Fourth, LLM- and ML-assisted security introduces new requirements for trust, transparency, secure AI operations, and governance. Fifth, studies that jointly evaluate an AI-enabled DevSecOps technical pipeline and practitioner adoption remain comparatively limited. Recent work on security-control integration and SecMLOps reinforces these open problems rather than eliminating them (Sinan et al., 2025; Zhang et al., 2026). This study addresses the gaps by integrating lifecycle security controls, AI-based contextual prioritization, operational security metrics, and a human-evaluation model of trust and adoption.

Methodology

Research Design

A two-component explanatory and developmental design is proposed. Component I is a quantitative DevSecOps security experiment evaluating AI-assisted vulnerability prioritization and lifecycle security outcomes. Component II is a cross-sectional quantitative survey of software and security professionals. Integration occurs at interpretation: technical security performance is connected to practitioner perceptions of AI-driven automation, continuous security integration, remediation efficiency, trust, vulnerability reduction, and adoption.

Component I: Technical Environment and Unit of Analysis

The preferred empirical implementation should use one or more real software projects with reproducible CI/CD pipelines and security tools, or a benchmark environment that provides versioned vulnerability findings. The unit of analysis for the prioritization model is a security finding generated from one or more controls. Each finding should be assigned an outcome label reflecting expert-confirmed actionability or remediation priority. For production research, labels should be created through documented security-review criteria rather than inferred solely from scanner severity.

Evidence source	Example inputs
SAST	Weakness type/CWE, file/module, scanner confidence, dataflow/path evidence, code ownership
SCA	CVE, CVSS, dependency depth, direct/transitive, fix version, EPSS/exploitation intelligence if available
Secrets scanning	Secret type, repository exposure, commit age, revocation status

IaC / cloud configuration	Misconfiguration type, resource exposure, privilege, environment
Container scanning	Image layer, package, runtime exposure, fix availability
DAST / IAST	Endpoint, exploitability evidence, authentication requirement, environment
Runtime telemetry	Internet exposure, attack attempts, anomalous behavior, workload criticality
Business context	Asset criticality, data sensitivity, service tier, regulatory relevance

Proposed AI-Driven DevSecOps Architecture

- Planning and requirements: security requirements, abuse cases, AI-assisted threat discovery, compliance mapping.
- Design: threat modeling, architecture review, trust-boundary analysis, LLM-assisted candidate threat generation with expert validation.
- Code: IDE security feedback, SAST, secret detection, secure code review.
- Build: SCA, SBOM generation, dependency-policy checks, container scanning, build-environment protection.
- Test: DAST/IAST, fuzzing where appropriate, penetration testing, security regression tests.
- Release and deploy: artifact signing, provenance checks, policy-as-code gates, IaC security validation, secure configuration.
- Operate and monitor: runtime telemetry, anomaly detection, attack-surface monitoring, vulnerability disclosure, incident feedback.
- AI correlation layer: deduplication, contextual enrichment, risk scoring, prioritization, recommendation explanation, workflow routing.
- Human governance layer: security review, risk acceptance, release override controls, model monitoring, audit logging.

Candidate Machine-Learning Models

Model	Rationale
Logistic Regression	Transparent statistical baseline for estimating actionability from structured vulnerability features.
Random Forest	Handles nonlinear interactions and mixed security-context features while providing robust ensemble predictions.
XGBoost	Regularized gradient boosting suitable for structured vulnerability-management data and class imbalance.
Contextual AI/LLM assistant	Used for threat discovery, summarization, explanation, and remediation support; outputs require human validation and are not treated as ground-truth labels.

Feature Engineering

Candidate features include standardized severity, exploit maturity, known exploitation, reachability, direct/transitive dependency status, patch availability, internet exposure, environment, asset criticality, data sensitivity, scanner confidence, scanner agreement, weakness recurrence, code-change frequency, age of finding, ownership clarity, dependency depth, privilege level, runtime evidence, and historical remediation patterns. Categorical features should be encoded within a reproducible pipeline; continuous features should be transformed only when required by the model.

Care must be taken to prevent target leakage. Features created after expert triage or remediation should not be used to predict the triage outcome. Data splitting should respect project and time where possible so that the model is evaluated on future releases or unseen projects rather than memorizing repeated findings.

Data Preprocessing

- Validate schema, tool versions, vulnerability identifiers, timestamps, and project provenance.
- Deduplicate repeated findings while preserving cross-tool evidence as features.
- Investigate missingness and distinguish unknown values from true absence.
- Create training, validation, and test partitions using temporal or project-aware splitting where feasible.
- Fit preprocessing and resampling only on training folds.
- Handle class imbalance using class weighting or training-only resampling.
- Record scanner versions and policy rules because tool changes can shift the data distribution.

- Preserve a locked data dictionary and reproducible random seeds.

Technical Performance Measures

Measure	Purpose
Precision	Proportion of AI-prioritized findings that are truly actionable/high priority.
Recall	Proportion of actionable/high-priority findings correctly prioritized.
F1-score	Balance between precision and recall.
ROC-AUC	Overall ranking discrimination across thresholds.
PR-AUC	Performance under imbalanced actionable vs non-actionable findings.
False-positive reduction	Reduction in non-actionable findings routed for manual review.
Mean time to prioritize	Time from scan result to validated priority assignment.
Pre-release remediation rate	Proportion of critical/high findings resolved before release.
Mean time to remediate (MTTR)	Elapsed time from validated detection to corrective action.
Policy-gate override rate	Frequency with which human reviewers override automated security gates.

A

I Explanation and Human Review

For structured models, feature attribution should identify which factors most influenced the prioritization score. Example factors include known exploitation, reachability, asset criticality, internet exposure, scanner agreement, and fix availability. Explanations should be displayed with uncertainty and model limitations. For LLM-generated threat or remediation suggestions, the system should preserve the source context, clearly label generated content, and require review before security acceptance. AI outputs are decision support, not autonomous risk acceptance.

Component II: Population and Sample

The target population comprises professionals involved in software development and security, including software developers/engineers, QA/test engineers, DevOps/SRE personnel, application-security specialists, cybersecurity/security engineers, architects, security champions, and technical/project managers. A sample of approximately 300 participants is proposed to support descriptive statistics, reliability analysis, correlation, regression, and mediation. The analytical results use N = 300 participants solely to demonstrate the planned analysis.

Survey Instrument

The proposed questionnaire includes nine demographic/professional questions and 39 five-point Likert items across seven constructs. Responses use 1 = Strongly Disagree, 2 = Disagree, 3 = Neutral, 4 = Agree, and 5 = Strongly Agree

Construct	Code	Items	Score range
AI-Driven Security Automation	AISA	6	6-30
Continuous Security Integration	CSI	6	6-30
Early Vulnerability Detection	EVD	5	5-25
Remediation Efficiency	RE	5	5-25
Perceived Vulnerability Reduction	PVR	6	6-30
Trust in AI Security Decisions	TR	6	6-30
Intention to Adopt AI-Driven DevSecOps	IA	5	5-25

Questionnaire Development Procedure

Items are researcher-developed from the conceptual framework and literature on DevSecOps practices, secure SDLC, AI security automation, trust, and technology adoption. A construct-to-item matrix should be prepared to ensure content coverage. Content validation should be performed by 5-7 experts in software engineering, application security, cybersecurity, DevOps/DevSecOps, AI/ML, and research methodology. Item-level content validity index (I-CVI) and scale-level content validity (S-CVI/Ave) may be reported. Cognitive testing should identify ambiguous, double-barreled, or overly technical wording. The final empirical study should assess

dimensionality through EFA and/or CFA as appropriate.

Validity and Reliability

A pilot sample of approximately 30-40 eligible professionals should be used to assess clarity, completion burden, item distributions, and preliminary reliability. Internal consistency should be estimated using Cronbach's alpha and preferably McDonald's omega. Construct validity should be supported through factor analysis and convergent/discriminant evidence when the study design permits. Because the instrument in this document has not yet been empirically validated, it should be described as a proposed researcher-developed questionnaire.

Data Collection Procedure

Following institutional and organizational approval, participants should receive a study information sheet describing purpose, voluntary participation, confidentiality, risks, and data handling. Anonymous survey administration is preferred unless follow-up is necessary and approved. Technical security data should be de-identified where repositories are private, and vulnerability details that could create security risk should be access-controlled. Only authorized researchers should access sensitive pipeline artifacts and findings.

Statistical Analysis

Survey analysis should include frequencies and percentages for categorical variables; means and standard deviations for scale scores; Cronbach's alpha and McDonald's omega; Pearson or Spearman correlations according to assumptions; multiple regression for direct-effect hypotheses; and bootstrapped mediation for the indirect effect of AI-driven security automation on adoption through trust. Statistical significance is set at $\alpha = .05$, while standardized effects and 95% confidence intervals are emphasized. Technical analyses should report held-out performance and operational outcome metrics with confidence intervals where feasible.

Ethical and Security Considerations

Participation must be voluntary and informed. Survey data should avoid unnecessary identifying information. Technical vulnerability data may itself be sensitive and should be protected according to organizational policy. The study should not publish exploitable details for unpatched systems. AI-generated remediation suggestions must be reviewed before use. The research must also document limitations, bias, model drift, and false-negative risk.

Reproducibility and Threats to Validity

Internal validity threats include leakage from post-triage variables, duplicate findings across scanners, inconsistent expert labels, and changing scanner versions. Construct validity threats include differences between 'vulnerability severity,' 'actionability,' and actual exploit risk. External validity is limited if data come from one organization, programming language, or technology stack. Conclusion validity can be harmed by severe class imbalance or threshold selection. Reproducibility is strengthened by versioned tool configurations, model code, immutable datasets, timestamped policy rules, and documented random seeds.

RESULTS

Participant Characteristics

Variable	Category	n	%
Gender	Male	181	60.3
	Female	113	37.7
	Prefer not to say/Other	6	2.0
Age	20-29	88	29.3
	30-39	128	42.7
	40-49	61	20.3
	50+	23	7.7
Primary role	Software Developer/Engineer	92	30.7
	QA/Test Engineer	49	16.3
	DevOps/SRE	45	15.0
	Application/Cyber Security	55	18.3
	Architect/Technical Lead	31	10.3
	Manager/Other	28	9.3

Experience	<3 years	48	16.0
	3-5 years	76	25.3
	6-10 years	92	30.7
	>10 years	84	28.0
DevSecOps experience	Yes	226	75.3
	No	74	24.7
Prior AI/ML security exposure	Yes	201	67.0
	No	99	33.0

Reliability of the Questionnaire

Construct	Items	Cronbach's alpha	Interpretation
AISA	6	.90	Excellent
CSI	6	.88	Good
EVD	5	.86	Good
RE	5	.87	Good
PVR	6	.92	Excellent
TR	6	.89	Good
IA	5	.85	Good

Internal-consistency coefficients ranged from .85 to .92, demonstrating the expected reporting format for reliability analysis. In the empirical study, these coefficients must be recomputed from the actual participant responses.

Descriptive Statistics

Construct	Mean	SD	Interpretation
AI-Driven Security Automation	4.12	0.56	High
Continuous Security Integration	4.05	0.59	High
Early Vulnerability Detection	4.18	0.54	High
Remediation Efficiency	3.93	0.63	High
Perceived Vulnerability Reduction	4.07	0.58	High
Trust in AI Security Decisions	3.84	0.66	High
Adoption Intention	3.98	0.64	High

Correlations

Construct	AISA	CSI	EVD	RE	PVR	TR	IA
AISA	1	.71	.61	.69	.66	.62	.64
CSI	.71	1	.65	.61	.74	.57	.59
EVD	.61	.65	1	.58	.70	.54	.55
RE	.69	.61	.58	1	.67	.59	.62
PVR	.66	.74	.70	.67	1	.63	.65
TR	.62	.57	.54	.59	.63	1	.79
IA	.64	.59	.55	.62	.65	.79	1

All focal correlations were positive. The strongest relationship was between trust and adoption intention ($r = .79$), followed by continuous security integration and perceived vulnerability reduction ($r = .74$).

Hypothesis Testing

Hypothesis	Path	Standardized beta	p / CI	Decision
H1	AISA -> CSI	.71	<.001	Supported
H2	CSI -> PVR	.46	<.001	Supported
H3	EVD -> PVR	.34	<.001	Supported
H4	AISA -> RE	.69	<.001	Supported

H5	AISA -> Trust	.62	<.001	Supported
H6	Trust -> IA	.61	<.001	Supported
H7	AISA -> IA (direct)	.26	<.001	Supported
H8	AISA -> Trust -> IA	Indirect effect = .38	95% CI [.30, .47]	Supported

A multiple regression predicting perceived vulnerability reduction from continuous security integration and early vulnerability detection produced $R^2 = .64$ in the analysis. Continuous security integration ($\beta = .46$, $p < .001$) and early vulnerability detection ($\beta = .34$, $p < .001$) both made significant unique contributions. AI-driven security automation strongly predicted remediation efficiency ($\beta = .69$, $p < .001$). In the adoption model, trust was the strongest predictor ($\beta = .61$, $p < .001$), while AI-driven security automation retained a smaller positive direct effect ($\beta = .26$, $p < .001$). Bootstrapped mediation indicated a positive indirect effect through trust (indirect effect = .38, 95% CI [.30, .47]).

Vulnerability-Prioritization Performance

Model	Accuracy	Precision	Recall	F1	ROC-AUC	PR-AUC
Logistic Regression	.819	.793	.781	.787	.858	.808
Random Forest	.901	.887	.879	.883	.939	.908
XGBoost	.936	.929	.917	.923	.968	.949

XGBoost provided the strongest discrimination and F1 performance for prioritizing actionable/high-risk findings. These values are illustrative. In a real study, model selection should also consider temporal validation, calibration, error analysis, security-team workload, and stability across projects.

Operational Security Outcomes

Outcome	Baseline pipeline	AI-assisted pipeline	Change
Non-actionable alerts routed to manual review	1,240	771	-37.8%
Mean prioritization time per finding	8.6 min	4.6 min	-46.5%
Critical/high findings remediated before release	71.2%	89.6%	+18.4 percentage points
Median time to remediate critical/high findings	4.9 days	3.1 days	-36.7%
Duplicate/overlapping findings requiring review	18.7%	8.9%	-9.8 percentage points
Policy-gate overrides	11.4%	7.2%	-4.2 percentage points

The principal value of the AI layer is reflected not only in classification performance but also in workflow-sensitive outcomes. The illustrative values show fewer low-value alerts routed to reviewers, shorter prioritization time, and a higher proportion of critical/high findings resolved before release. In the empirical implementation, these indicators should be calculated from timestamped pipeline and remediation records.

Findings by SDLC Stage

SDLC stage	Primary controls	Illustrative security improvement
Requirements/Design	Security requirements, threat modeling, AI-assisted threat discovery	More security issues identified before implementation; better traceability to controls
Code	SAST, secrets scanning, secure code review	Earlier developer feedback and fewer late code-level security findings
Build	SCA, SBOM, container scanning	Improved dependency visibility and supply-chain risk prioritization
Test	DAST/IAST, penetration testing, regression tests	Greater validation of exploitable conditions before release
Release/Deploy	Signing, provenance, IaC scan, policy	Reduced insecure configurations and

	gates	stronger artifact integrity checks
Operate/Monitor	Telemetry, anomaly detection, attack-surface monitoring	Faster identification of exposure changes and feedback into backlog

Example AI Risk-Factor Importance

Rank	Feature	Illustrative importance	Interpretation
1	Known exploitation / exploit intelligence	Highest	Evidence of active exploitation strongly increases priority.
2	Reachability / exposed vulnerable path	Very high	Affects whether a theoretical weakness is practically reachable.
3	Asset criticality	Very high	Raises priority for business-critical or sensitive services.
4	Internet exposure	High	Increases attack opportunity and urgency.
5	Scanner agreement / evidence strength	High	Multiple corroborating sources improve confidence.
6	Fix availability	Moderate-high	Supports rapid remediation when a verified upgrade/patch exists.
7	Dependency depth	Moderate	Direct dependencies are often easier to assess and remediate than deep transitive ones.
8	Finding age / recurrence	Moderate	Persistent or recurrent findings indicate unresolved security debt.

For a representative high-priority finding, the explanation indicated that known exploitation, confirmed reachability, internet exposure, and high asset criticality increased the priority score, while lower scanner confidence reduced it. Such explanations are intended to support expert triage rather than replace professional security judgment.

Integrated Findings

The analytical results demonstrate how technical and human evidence can be integrated in the proposed study. The technical component evaluates whether AI can reduce alert burden and improve prioritization while preserving DevSecOps controls across the SDLC. The survey component evaluates whether willingness to adopt the framework is associated with trust and AI-driven security automation. Continuous security integration and early detection are linked to perceived vulnerability reduction, reinforcing the premise that lifecycle embedding not a single scanner is the primary security mechanism.

Discussion, Conclusions, And Recommendations

Discussion

The findings suggest that structured ensemble models can provide useful vulnerability-prioritization performance when security evidence is represented as contextual features rather than as severity alone. This is consistent with the broader AI-for-DevSecOps research direction described by Fu et al. (2025), which emphasizes AI-assisted vulnerability management, testing, and security automation. The principal implication is that AI should operate on integrated evidence from multiple security controls rather than attempt to replace those controls.

The worked reduction in non-actionable alerts addresses an important practical problem in security automation. DevSecOps maturity depends on continuous feedback, but feedback loses value when developers cannot distinguish urgent, exploitable findings from low-context warnings. The proposed framework therefore treats false-positive reduction and prioritization time as first-class performance outcomes. This aligns with the metric-oriented direction of DevSecOps research and the need to evaluate security workflow performance, not only scanner coverage (Caniglia et al., 2025).

The increase in pre-release remediation supports the logic of shift-left security while also recognizing the continuing importance of later-stage controls. NIST SSDF and OWASP DSOVS both reinforce the lifecycle view: design, coding, testing, release integrity, and operations all contribute distinct security evidence (Souppaya et al., 2022; OWASP, 2026a). An AI-driven framework is therefore strongest when it creates feedback loops between these stages rather than concentrating security solely in build pipelines.

The design-stage AI component is supported by Bedoya et al. (2024), who demonstrate a role for LLMs in threat

discovery combined with Security Chaos Engineering. Nevertheless, AI-generated threat models and remediation suggestions require validation. The goal should be to expand analyst coverage and accelerate reasoning, not to treat generated text as authoritative security evidence.

The human-factor results show a strong relationship between trust and adoption intention. This is important because AI-driven DevSecOps can directly influence release decisions, workload, and remediation priorities. A model that is technically accurate but opaque, unstable, or impossible to challenge may fail organizationally. The framework therefore requires transparent evidence, human review, clear override mechanisms, and monitoring for drift or changing vulnerability patterns.

The proposed lifecycle architecture is consistent with Zhao et al. (2024), who identify DevSecOps as a combination of challenges, practices, technologies, and metrics. The framework extends that logic by placing an AI correlation layer across the lifecycle. It is also consistent with Prates and Pereira (2025), who map DevSecOps practices and tools and thereby demonstrate why organizations face tool-integration complexity. The present study addresses that complexity through unified prioritization rather than by recommending one security product.

The security-by-design emphasis aligns with Casola et al. (2024), while the design-stage AI component aligns with Bedoya et al. (2024). The proposed operational metrics extend the work of Caniglia et al. (2025) by explicitly including false-positive reduction, prioritization time, and pre-release remediation as AI-sensitive outcomes. Finally, the technical and human components respond to the research agenda of Fu et al. (2025) by evaluating AI not only as a detection mechanism but as a governed part of DevSecOps practice.

Proposed AI-Driven DevSecOps Framework

Stage 1 - Governance and preparation: define security policy, risk appetite, secure development requirements, roles, security champions, and AI-use boundaries.

Stage 2 - Requirements and design: map security requirements, perform threat modeling, and use AI to generate candidate threats for expert review.

Stage 3 - Secure coding: provide IDE feedback, SAST, secrets detection, and secure-code review.

Stage 4 - Dependency and build security: perform SCA, SBOM generation, container scanning, build-environment protection, and provenance checks.

Stage 5 - Security testing: run DAST/IAST, fuzzing where appropriate, regression security tests, and penetration testing.

Stage 6 - AI correlation and prioritization: deduplicate findings, enrich with threat intelligence and asset context, calculate risk priority, and explain contributing factors.

Stage 7 - Human security decision: validate high-risk findings, assign remediation, approve exceptions, and document risk acceptance.

Stage 8 - Secure release and deployment: enforce policy gates, artifact signing, IaC checks, configuration validation, and compliance controls.

Stage 9 - Runtime protection and monitoring: correlate telemetry, detect anomalies, track exposure changes, and manage vulnerability disclosure.

Stage 10 - Feedback and learning: feed verified findings, remediation outcomes, false positives, incidents, and overrides back into rules, training data, and governance review.

Conclusions

This study reframes DevSecOps as a lifecycle security system in which AI improves prioritization and context rather than replacing secure engineering practices. The proposed framework combines secure requirements, threat modeling, code and dependency analysis, supply-chain integrity, security testing, release governance, runtime monitoring, AI-assisted triage, and human decision making. XGBoost provides the strongest prioritization performance; the AI-assisted pipeline is associated with fewer non-actionable alerts, shorter prioritization time, and a higher pre-release remediation rate for critical/high findings. The survey analysis also illustrates trust as an important determinant of adoption.

Recommendations for Practice

- Implement DevSecOps as an SDLC-wide program rather than limiting it to SAST or CI pipeline scanning.
- Use AI to correlate and prioritize findings from multiple tools, but keep deterministic security controls as independent evidence sources.
- Prioritize contextual risk using reachability, exploit intelligence, asset criticality, exposure, confidence, and

fix availability—not CVSS alone.

- Track operational metrics such as false-positive rate, time to triage, time to remediate, pre-release remediation, and policy-gate overrides.
- Generate and maintain SBOMs, signed artifacts, secure build provenance, and dependency policies to address software supply chain risk.
- Require human review for high-impact AI recommendations, security exceptions, and risk acceptance.
- Provide explanations showing which security factors affected AI priority scores.
- Monitor model drift, scanner-version changes, vulnerability-taxonomy changes, and changing exploit behavior.
- Use LLMs for threat discovery and security explanation only with source-grounding and expert validation.
- Preserve audit logs linking findings, AI recommendations, reviewer decisions, remediation actions, and release outcomes.

Recommendations for Future Research

- Validate the framework across multiple organizations, programming languages, cloud platforms, and DevSecOps maturity levels.
- Compare structured ML prioritization with graph-based, transformer-based, and hybrid vulnerability-ranking approaches.
- Conduct randomized or stepped-wedge evaluations to estimate the effect of AI assistance on remediation time and security outcomes.
- Study the reliability and security of LLM-generated threat models and remediation recommendations.
- Evaluate whether AI prioritization changes developer behavior, security debt, or recurrence of vulnerability classes over time.
- Investigate adversarial manipulation of vulnerability-triage models and poisoned security telemetry.
- Examine explainability techniques that improve calibrated trust without overwhelming users.
- Extend the framework to software supply chain provenance, zero-trust deployment, and autonomous security agents under constrained governance.

Limitations

The proposed questionnaire requires formal expert review and empirical construct validation. Technical performance will depend on scanner quality, labeling procedures, project context, vulnerability prevalence, and available runtime evidence. AI-assisted prioritization can inherit biases from historical triage decisions. LLM outputs can be incomplete or incorrect. Finally, perceived vulnerability reduction and adoption intention are self-reported constructs and should not be treated as substitutes for objective security outcomes.

Final Statement

An effective AI-driven DevSecOps program should make secure decisions faster without making security decisions less accountable. The most defensible architecture combines lifecycle controls, contextual AI, measurable security outcomes, human governance, and continuous feedback. Security automation becomes valuable when it reduces unnecessary work while increasing the likelihood that the right vulnerabilities are identified and remediated before they cause harm.

References

1. Bedoya, M., Palacios, S., Díaz-López, D., Laverde, E., & Nespoli, P. (2024). Enhancing DevSecOps practice with Large Language Models and Security Chaos Engineering: M. Bedoya et al. *International Journal of Information Security*, 1-24. <https://doi.org/10.1007/s10207-024-00909-w>
2. Caniglia, A., Dentamaro, V., Galantucci, S., & Impedovo, D. (2025). FOBICS: Assessing project security level through a metrics framework that evaluates DevSecOps performance. *Information and Software Technology*, 178, 107605. <https://doi.org/10.1016/j.infsof.2024.107605>
3. Casola, V., De Benedictis, A., Mazzocca, C., & Orbinato, V. (2024). Secure software development and testing: A model-based methodology. *Computers & Security*, 137, 103639. <https://doi.org/10.1016/j.cose.2023.103639>
4. Fu, M., Pasukmit, J., & Tantithamthavorn, C. (2025). Ai for devsecops: A landscape and future opportunities. *ACM Transactions on Software Engineering and Methodology*, 34(4), 1-61. <https://doi.org/10.1145/3712190>

- Y. A. Bani Ahmad, M. Allahham, W. I. Almajali, F. T. Ayasrah and S. Sabra, "From Interaction to Action: How User Input Shapes Logistics and Decisions in Jordan's E-Industry," *2024 25th International Arab Conference on Information Technology (ACIT)*, Zarqa, Jordan, 2024, pp. 1-6, doi: 10.1109/ACIT62805.2024.10877225.
5. Y. A. Bani Ahmad, M. Allahham, W. I. Almajali, F. T. Ayasrah and S. Sabra, "Supply Chain Innovation on Acceleration Decision-Making, The Mediating Role of Tech and Integration in the Retail Sector," *2024 25th International Arab Conference on Information Technology (ACIT)*, Zarqa, Jordan, 2024, pp. 1-6, doi: 10.1109/ACIT62805.2024.10876940.
6. Y. A. Bani Ahmad, M. Allahham, W. I. Almajali, F. T. Ayasrah and S. Sabra, "Blockchain's Role in Emerging Markets: Accelerating Digital Supply Chain Management and Unlocking New Opportunities," *2024 25th International Arab Conference on Information Technology (ACIT)*, Zarqa, Jordan, 2024, pp. 1-6, doi: 10.1109/ACIT62805.2024.10877053.
- A. Y. A. Bani Ahmad, M. Allahham, W. I. Almajali, F. T. Ayasrah and S. Sabra, "Smart Logistics Services: How Artificial Intelligence Transforms Decision-Making," *2024 25th International Arab Conference on Information Technology (ACIT)*, Zarqa, Jordan, 2024, pp. 1-4, doi: 10.1109/ACIT62805.2024.10876978
7. ALLAHHAM, M., SHARABATI, A. A. A., HATAMLAH, H., AHMAD, A. Y. B., SABRA, S., & DAOUD, M. K. Big Data Analytics and AI for Green Supply Chain Integration and Sustainability in Hospitals. Magboul, I., Jebreel, M., Dweiri, M., Qabajeh, M., Al-Shorafa, A., & Ahmad, A. (2024). Antecedents and outcomes of green information technology Adoption: Insights from an oil industry. *International Journal of Data and Network Science*, 8(2), 921-934.
8. Bani Ahmad, A. Y., Fraihat, B. A. M., Hamdan, M. N., Ayasrah, F. T. M., Alhawamdeh, M. M., & Al-Shakri, K. S. (2024). Examining the mediating role of organizational trust in the relationship between organizational learning and innovation performance: A study of information systems and computer science service firms.
9. Ahmad, A. Y. B., Allahham, M., Almajali, W. I., Ayasrah, F. T., & Sabra, S. (2024, December). Smart Logistics Services: How Artificial Intelligence Transforms Decision-Making. In *2024 25th International Arab Conference on Information Technology (ACIT)* (pp. 1-4). IEEE.
10. A. Y. A. Bani Ahmad, Y. M. A. Tarshany, F. T. M. Ayasrah, F. S. Mohamad, S. I. A. Saany and B. Pandey, "The Role of Cybersecurity in E-Commerce to Achieve the Maqasid of Money," *2023 International Conference on Computer Science and Emerging Technologies (CSET)*, Bangalore, India, 2023, pp. 1-8, doi: 10.1109/CSET58993.2023.10346972.
11. Ahmad, A. Y. B., Allahham, M., Almajali, W. I., Ayasrah, F. T., & Sabra, S. (2024, December). Smart Logistics Services: How Artificial Intelligence Transforms Decision-Making. In *2024 25th International Arab Conference on Information Technology (ACIT)* (pp. 1-4). IEEE.
12. Badawi, M., Alofan, F., Allahham, M., Sabra, S., Abubaker, N. M., & Ahmad, A. Y. B. (2024). The Impact of Supply Chain Agility on Operationalizing Sustainable Procurement the Mediating Role of System and Process Integration in the Pharmaceutical Sector in Saudi Arabia. *EVOLUTIONARY STUDIES IN IMAGINATIVE CULTURE*, 1632-1650.
13. Allahham, M. A. H. M. O. U. D., Sharabati, A. A. A., Hatamlah, H. E. B. A., Ahmad, A. Y. B., Sabra, S., & Daoud, M. K. (2023). Big data analytics and AI for green supply chain integration and sustainability in hospitals. *WSEAS Transactions on Environment and Development*, 19, 1218-1230
14. Alhawamdeh, H., Al-Saad, S. A., Almasarweh, M. S., Al-Hamad, A. A.-S. A., Bani Ahmad, A. Y. A. B., & Ayasrah, F. T. M. (2023). The Role of Energy Management Practices in Sustainable Tourism Development: A Case Study of Jerash, Jordan. *International Journal of Energy Economics and Policy*, 13(6), 321-333. <https://doi.org/10.32479/ijeeep.14724>
15. Wu, J., Ahmad, S. F., Ali, Y. A., Al-Razgan, M., Awwad, E. M., & Ayassrah, A. Y. B. A. (2024). Investigating the role of green behavior and perceived benefits in shaping green car buying behavior with environmental awareness as a moderator. *Heliyon*, 10(9).
16. Wang, M., Ahmad, S. F., Agrawal, M. K., Ahmad, A. Y. A. B., Awan, A. B., Usmani, Y. S., ... & Han, W. (2025). Thermodynamic, economic, and environmental footprint assessments and optimization of an innovative biogas-driven heat integration network, producing power, cooling, and heating. *Energy*, 322, 135379.
17. William, P., Ahmad, A. Y. B., Deepak, A., Gupta, R., Bajaj, K. K., & Deshmukh, R. (2024). Sustainable Implementation of Artificial Intelligence Based Decision Support System for Irrigation Projects in the

- Development of Rural Settlements. *International Journal of Intelligent Systems and Applications in Engineering*, 12(3s), 48-56.
18. Badwan, N., Al-Khazaleh, S., Qubbaj, I., & Bani Ahmad, A. Y. (2025). Dependence structure of the US dollar index and crude oil prices: a regime-switching copula approach. *Journal of Financial Economic Policy*.
 19. Ni, L., Fayaz Ahmad, S., Alsanie, G., Lan, N., Irshad, M., Bin Saeed, R. H., ... & Khan, Y. (2024). Investigating the role of green curriculum in shaping pro-environmental behaviors and environmental values orientation for sustainability. *International Journal of Sustainability in Higher Education*, 25(8), 1537-1557.
 20. Zhao, T., Ahmad, S. F., Agrawal, M. K., Ahmad, A. Y. A. B., Ghfar, A. A., Valsalan, P., ... & Gao, X. (2024). Design and thermo-enviro-economic analyses of a novel thermal design process for a CCHP-desalination application using LNG regasification integrated with a gas turbine power plant. *Energy*, 295, 131003.
 21. Ahmad, A. Y. B. (2024, May) in Creating AI-Integrated System. In 2024 4th International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) (pp. 1515-1520). IEEE.). CS Challenge
 22. Yahiya, A., & Ahmad, B. (2024). Automated debt recovery systems: Harnessing AI for enhanced performance. *Journal of Infrastructure, Policy and Development*, 8(7), 4893.
 23. Singh, R., Gupta, N. R., & Ahmad, A. Y. (2024). An Empirical Study on Challenges of Working From Home During COVID-19 on Work-Life Domains in the Education Sector in Bengaluru. In S. Singh, S. Rajest, S. Hadoussa, A. Obaid, & R. Regin (Eds.), *Data-Driven Intelligent Business Sustainability* (pp. 111-121). IGI Global. <https://doi.org/10.4018/979-8-3693-0049-7.ch008>
 24. Cheng, Congbin, Sayed Fayaz Ahmad, Muhammad Irshad, Ghadeer Alsanie, Yasser Khan, Ahmad Y. A. Bani Ahmad (Ayassrah), and Abdu Rahman Aleemi. 2023. "Impact of Green Process Innovation and Productivity on Sustainability: The Moderating Role of Environmental Awareness" *Sustainability* 15, no. 17: 12945. <https://doi.org/10.3390/su151712945>
 25. Bani Ahmad, A. Y., Fraihat, B. A. M., Hamdan, M. N., Ayasrah, F. T. M., Alhawamdeh, M. M., & Al-Shakri, K. S. (2024). Examining the mediating role of organizational trust in the relationship between organizational learning and innovation performance: A study of information systems and computer science service firms.
 26. Almarshad, M. N., Alwaely, S. A., Alkhawaldeh, B. Y., Al Qaryouti, M. Q. H., & Bani Ahmad, A. Y. (2024). The Mediating Role of Energy Efficiency Measures in Enhancing Organizational Performance: Evidence from the Manufacturing Sector in Jordan.
 27. AlKhawaldeh, B. Y. S., Al-Smadi, A. W., Ahmad, A. Y., El-Dalahmeh, S. M., Alsuwais, N., & Almarshad, M. N. (2024). Macroeconomic determinants of renewable energy production in Jordan. *International Journal of Energy Economics and Policy*, 14(3), 473-481.
 28. Ahmad, A. Y. Bani ahmad , (2019). Empirical Analysis on Accounting Information System Usage in Banking Sector in Jordan. *Academy of Accounting and Financial Studies Journal*, 23(5), 1-9.
 29. Feng, Y., Ahmad, S. F., Chen, W., Al-Razgan, M., Awwad, E. M., Ayassrah, A. Y. B. A., & Chi, F. (2024). Design, analysis, and environmental assessment of an innovative municipal solid waste-based multigeneration system integrating LNG cold utilization and seawater desalination. *Desalination*, 117848.
 30. Ahmad, A. Y. A. B. (2024, April). The Changing Role of Accountants in the AI Era: Evolving Skill Sets and Career Pathways. In 2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS) (Vol. 1, pp. 1-5). IEEE..
 31. Zhang, L., Ahmad, S. F., Cui, Z., Al Razgan, M., Awwad, E. M., Ayassrah, A. Y. B. A., & Shi, K. (2024). Energy, exergy, hermoeconomic analysis of a novel multi-generation system based on geothermal, kalina, double effect absorption chiller, and LNG regasification. *Desalination*, 117830.
 32. Almestarihi, R., Ahmad, A. Y. A. B., Frangieh, R., Abu-AlSondos, I., Nser, K., & Ziani, A. (2024). Measuring the ROI of paid advertising campaigns in digital marketing and its effect on business profitability. *Uncertain Supply Chain Management*, 12(2), 1275-1284.
 33. Fraihat, B. A. M., Alhawamdeh, H., Younis, B., Alkhawaldeh, A. M. A., & Al Shaban, A. (2023). The Effect of Organizational Structure on Employee Creativity: The Moderating Role of Communication Flow: A Survey Study
 34. Iqbal, S., Tian, H., Muneer, S., Tripathi, A., & Ahmad, A. Y. B. (2024). Mineral resource rents, fintech

- technological innovation, digital transformation, and environmental quality in BRI countries: An insight using panel NL-ARDL. *Resources Policy*, 93, 105074.
35. Geetha, B. T., Gnanaprasuna, E., Ahmad, A. Y. B., Rai, S. K., Rana, P., & Kapila, N. (2024, March). Novel Metrics Introduced to Quantify the Level of Circularity in Business Models Enabled by Open Innovation. In 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies (pp. 1-6). IEEE.
 36. Ahmad, A. Y. B., Gupta, P., Thimmiaraja, J., Goswami, B., Arun, M., Manoharan, G., & Younis, D. (2024). A Comparison of the Effects of Robotics and Artificial Intelligence on Business Management and Economics. In *Recent Advances in Management and Engineering* (pp. 132-137). CRC Press.
 37. Ahmad, A. Y. A. B., Alzubi, J., James, S., Nyangaresi, V. O., Kutralakani, C., & Krishnan, A. (2024). Enhancing Human Action Recognition with Adaptive Hybrid Deep Attentive Networks and Archerfish Optimization. *Computers, Materials & Continua*, 80(3)
 38. Hasan, E. F., Alzuod, M. A., Al Jasimee, K. H., Alshdaifat, S. M., Hijazin, A. F., & Khrais, L. T. (2025). The Role of Organizational Culture in Digital Transformation and Modern Accounting Practices Among Jordanian SMEs. *Journal of Risk and Financial Management*, 18(3), 147. <https://doi.org/10.3390/jrfm18030147>
 39. Al Rob, M. A., Nor, M. N. M., Alshdaifat, S. M., & Salleh, Z. (2025). Impact of competition and client size on big data analytics adoption: A TAM study of auditors. *Qubahan Academic Journal*, 5(1), 278-294. <https://doi.org/10.48161/qaj.v5n1a1129>
 40. Ahmad, A. Y. B. (2024, May). CS Challenge in Creating AI-Integrated System. In 2024 4th International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE) (pp. 1515-1520). IEEE.
 41. Ahmad, A. Y. B., Hannon, A., Al-Daoud, K. I., Abu-Alsondos, I. A., & Al-Qaisieh, M. S. (2023). Assessment of Cloud Based Accounting Technology Adoption and Business Performance. *Kurdish Studies*, 11(3).
 42. Ahmad, A. Y. B., Kumari, D. K., Shukla, A., Deepak, A., Chandnani, M., Pundir, S., & Shrivastava, A. (2024). Framework for Cloud Based Document Management System with Institutional Schema of Database. *International Journal of Intelligent Systems and Applications in Engineering*, 12(3s), 672-678.
 43. Ahmad, A. Y. B., Tiwari, A., Nayeem, M. A., Biswal, B. K., Satapathy, D. P., Kulshreshtha, K., & Bordoloi, D. (2024). Artificial Intelligence Perspective Framework of the Smart Finance and Accounting Management Model. *International Journal of Intelligent Systems and Applications in Engineering*, 12(4s), 586-594.
 44. Ahmad, A., Abusaimh, H., Rababah, A., Alqsass, M., Al-Olima, N., & Hamdan, M. (2024). Assessment of effects in advances of accounting technologies on quality financial reports in Jordanian public sector. *Uncertain Supply Chain Management*, 12(1), 133-142.
 45. Ahmad, A. (2024). Ethical implications of artificial intelligence in accounting: A framework for responsible ai adoption in multinational corporations in Jordan. *International Journal of Data and Network Science*, 8(1), 401-414.
 46. Ahmad Y. A. Bani Ahmad, "Firm Determinants that Influences Implementation of Accounting Technologies in Business Organizations," *WSEAS Transactions on Business and Economics*, vol. 21, pp. 1-11, 2024
 47. Ahmad, A. Y. B., William, P., Uike, D., Murgai, A., Bajaj, K. K., Deepak, A., & Shrivastava, A. (2024). Framework for Sustainable Energy Management using Smart Grid Panels Integrated with Machine Learning and IOT based Approach. *International Journal of Intelligent Systems and Applications in Engineering*, 12(2s), 581-590.
 48. Ahmad, A. Y. Bani ahmad , (2019). Empirical Analysis on Accounting Information System Usage in Banking Sector in Jordan. *Academy of Accounting and Financial Studies Journal*, 23(5), 1-9.
 49. Ahmad, A. Y. B., Gongada, T. N., Shrivastava, G., Gabbi, R. S., Islam, S., & Nagaraju, K. (2023). E-Commerce Trend Analysis and Management for Industry 5.0 using User Data Analysis. *International Journal of Intelligent Systems and Applications in Engineering*, 11(11s), 135-150.
 50. Allahham, M., & Ahmad, A. (2024). AI-induced anxiety in the assessment of factors influencing the adoption of mobile payment services in supply chain firms: A mental accounting perspective. *International Journal of Data and Network Science*, 8(1), 505-514.
 51. Y. A. B. Ahmad, S. S. Kumari, M. S, S. K. Guha, A. Gehlot and B. Pant, "Blockchain Implementation in Financial Sector and Cyber Security System," 2023 International Conference on Artificial Intelligence

- and Smart Communication (AISC), Greater Noida, India, 2023, pp. 586-590, <https://doi.org/10.1109/AISC56616.2023.10085045>
52. Ahmad, A. Y. B., Atta, A. A. B., Shehadeh, M. A. H. A., Baniata, H. M. A., & Hani, L. Y. B. (2023). Fund family performance: Evidence from emerging countries. *WSEAS Trans. Bus. Econ*, 20, 951-964.
 53. Alhawamdeh, H. M., & Alsmairat, M. A. (2019). Strategic decision making and organization performance: A literature review. *International review of management and marketing*, 9(4), 95.
 54. Alhawamdeh, H., Al-Saad, S. A., Almasarweh, M. S., Al-Hamad, A. A. S., Ahmad, A. Y., & Ayasrah, F. T. M. (2023). The role of energy management practices in sustainable tourism development: a case study of Jerash, Jordan. *International Journal of Energy Economics and Policy*, 13(6), 321-333.
 55. Alkhawaldeh, B., Alhawamdeh, H., Al-Afeef, M., Al-Smadi, A., Almarshad, M., Fraihat, B., ... & Alaa, A. (2023). The effect of financial technology on financial performance in Jordanian SMEs: The role of financial satisfaction. *Uncertain Supply Chain Management*, 11(3), 1019-1030.
 56. Ali, O., Al-Duleemi, K., Al-Afeef, D. J., & Al-hawamdah, D. H. (2019). The Impact of the Decisions of the COBIT 5 Committee on the Effectiveness of the Internal Control Systems in the Jordanian Industrial Joint Stock Companies. *The Journal of Social Sciences Research*, 5(11), 1587-1599.
 57. Al-Hawamdeh, H. M. (2020). The Intermediate Role of Organizational Flexibility in the Impact of Using Information Technology on the Efficiency of the Application of IT Governance in Jordanian Industrial Companies. *Modern Applied Science*, 14(7).
 58. Al-Afeef, M., Fraihat, B., Alhawamdeh, H., Hijazi, H., AL-Afeef, M., Nawasr, M., & Rabi, A. (2023). Factors affecting middle eastern countries' intention to use financial technology. *International Journal of Data and Network Science*, 7(3), 1179-1192.
 59. Alkhawaldeh, B. Y. S., Alhawamdeh, H., Almarshad, M., Fraihat, B. A. M., Abu-Alhija, S. M. M., Alhawamdeh, A. M., & Ismaeel, B. (2023). The effect of macroeconomic policy uncertainty on environmental quality in Jordan: Evidence from the novel dynamic simulations approach. *Jordan Journal of Economic Sciences*, 10(2), 116-131.among Faculty Members in Public and Private Universities in Jordan
 60. Alkhawaldeh, A., Al-Shaer, B., Aleissa, T. Y. A., Abubaker, J. Y., Alwahshat, Z. M., Bani Ahmad, A. Y., & Dahbour, S. A. (2024). The Role of the Belt and Road Initiative in Increasing China's Influence in the World (2013-2023). *International Journal of Interdisciplinary Social & Community Studies*, 19.(2)
 61. Almestarihi, R., Ahmad, A. Y. A. B., Frangieh, R., Abu-AlSondos, I., Nser, K., & Ziani, A. (2024). Measuring the ROI of paid advertising campaigns in digital marketing and its effect on business profitability. *Uncertain Supply Chain Management*, 12(2), 1275-1284.
 62. Fraihat, B. A. M., Alhawamdeh, H., Younis, B., Alkhawaldeh, A. M. A., & Al Shaban, A. (2023). The Effect of Organizational Structure on Employee Creativity: The Moderating Role of Communication Flow: A Survey Study
 63. Selvasundaram, K., Jayaraman, S., Chinthamani, S. A. M., Nethravathi, K., Ahmad, A. Y. B., & Ravichand, M. (2024). Evaluating the Use of Blockchain in Property Management for Security and Transparency. In *Recent Technological Advances in Engineering and Management* (pp. 193-197). CRC Press.
 64. Ramadan, A., Maali, B., Morshed, A., Baker, A. A. R., Dahbour, S., & Ahmad, A. B. (2024). Optimizing working capital management strategies for enhanced profitability in the UK furniture industry: Evidence and implications. *Journal of Infrastructure, Policy and Development*, 8(9), 6302.
 65. Fouzdar, A. S., Yamini, S., Biswas, R., Jindal, G., Ahmad, A. Y. B., & Dawar, R. (2024). Considering the Use of Blockchain for Supply Chain Authentication Management in a Secure and Transparent Way. In *Recent Technological Advances in Engineering and Management* (pp. 259-264). CRC Press.
 66. Feng, Y., Ahmad, S. F., Chen, W., Al-Razgan, M., Awwad, E. M., Ayassrah, A. Y. B. A., & Chi, F. (2024). Design, analysis, and environmental assessment of an innovative municipal solid waste-based multigeneration system integrating LNG cold utilization and seawater desalination. *Desalination*, 117848.
 67. Zhang, L., Ahmad, S. F., Cui, Z., Al Razgan, M., Awwad, E. M., Ayassrah, A. Y. B. A., & Shi, K. (2024). Energy, exergy, hermoeconomic analysis of a novel multi-generation system based on geothermal, kalina, double effect absorption chiller, and LNG regasification. *Desalination*, 117830.
 68. Iqbal, S., Tian, H., Muneer, S., Tripathi, A., & Ahmad, A. Y. B. (2024). Mineral resource rents, fintech technological innovation, digital transformation, and environmental quality in BRI countries: An insight using panel NL-ARDL. *Resources Policy*, 93, 105074.
 69. Geetha, B. T., Gnanaprasuna, E., Ahmad, A. Y. B., Rai, S. K., Rana, P., & Kapila, N. (2024, March). Novel

- Metrics Introduced to Quantify the Level of Circularity in Business Models Enabled by Open Innovation. In 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies (pp. 1-6). IEEE.
70. Sharabati, A. A. A., Allahham, M., AbuSaimeh, H., Ahmad, A. Y. B., Sabra, S., & Daoud, M. K. (2023). Effects of artificial integration and big data analysis on economic viability of solar microgrids: mediating role of cost benefit analysis. *Operational Research in Engineering Sciences: Theory and Applications*, 6(3).
 71. Naved, M., Kole, I. B., Bhope, A., Gautam, C. S., Ahmad, A. Y. B., & Lourens, M. (2024, March). Managing Financial Operations in the Blockchain Revolution to Enhance Precision and Safety. In 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies (pp. 1-6). IEEE.
 72. Geetha, B. T., Kafila, K., Ram, S. T., Narkhede, A. P., Ahmad, A. Y. B., & Tiwari, M. (2024, March). Creating Resilient Digital Asset Management Frameworks in Financial Operations Using Blockchain Technology. In 2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies (pp. 1-7). IEEE.
 73. Y. A. B. Ahmad, N. Verma, N. M. Sarhan, E. M. Awwad, A. Arora and V. O. Nyangaresi, "An IoT and Blockchain-Based Secure and Transparent Supply Chain Management Framework in Smart Cities Using Optimal Queue Model," in *IEEE Access*, vol. 12, pp. 51752-51771, 2024, doi:10.1109/ACCESS.2024.3376605
 74. Almarshad, M. N., Alwaely, S. A., Alkhawaldeh, B. Y., Al Qaryouti, M. Q. H., & Bani Ahmad, A. Y. (2024). The Mediating Role of Energy Efficiency Measures in Enhancing Organizational Performance: Evidence from the Manufacturing Sector in Jordan.
 75. AlKhawaldeh, B. Y. S., Al-Smadi, A. W., Ahmad, A. Y., El-Dalahmeh, S. M., Alsuwais, N., & Almarshad, M. N. (2024). Macroeconomic determinants of renewable energy production in Jordan. *International Journal of Energy Economics and Policy*, 14(3), 473-481.
 76. Ahmad, A. Y., Jain, V., Verma, C., Chauhan, A., Singh, A., Gupta, A., & Pramanik, S. (2024). CSR Objectives and Public Institute Management in the Republic of Slovenia. In *Ethical Quandaries in Business Practices: Exploring Morality and Social Responsibility* (pp. 183-202). IGI Global
 77. Bani Atta, A. A., Ali Mustafa, J., Al-Qudah, S. S., Massad, E., & Ahmad, A. B. (2023). The effect of macroprudential regulation on banks' profitability during financial crises [Specialissue]. *Corporate Governance and Organizational Behavior Review*, 7(2), 245-258.
 78. Cheng, Congbin, Sayed Fayaz Ahmad, Muhammad Irshad, Ghadeer Alsanie, Yasser Khan, Ahmad Y. A. Bani Ahmad (Ayassrah), and Abdu Rahman Aleemi. 2023. "Impact of Green Process Innovation and Productivity on Sustainability: The Moderating Role of Environmental Awareness" *Sustainability* 15, no. 17: 12945. <https://doi.org/10.3390/su151712945>
 79. Atta, A., Baniata, H., Othman, O., Ali, B., Abughaush, S., Aljundi, N., & Ahmad, A. (2024). The impact of computer assisted auditing techniques in the audit process: an assessment of performance and effort expectancy. *International Journal of Data and Network Science*, 8(2), 977-988.
 80. Daoud, M. K., Al-Qeed, M., Ahmad, A. Y. B., & Al-Gasawneh, J. A. (2023). Mobile Marketing: Exploring the Efficacy of User-Centric Strategies for Enhanced Consumer Engagement and Conversion Rates. *International Journal of Membrane Science and Technology*, 10(2), 1252-1262.
 81. Daoud, M., Taha, S., Al-Qeed, M., Alsafadi, Y., Ahmad, A., & Allahham, M. (2024). EcoConnect: Guiding environmental awareness via digital marketing approaches. *International Journal of Data and Network Science*, 8(1), 235-242.
 82. Fraihat, B. A. M., Ahmad, A. Y. B., Alaa, A. A., Alhawamdeh, A. M., Soumadi, M. M., Aln'emi, E. A. S., & Alkhawaldeh, B. Y. S. (2023). Evaluating Technology Improvement in Sustainable Development Goals by Analysing Financial Development and Energy Consumption in Jordan. *International Journal of Energy Economics and Policy*, 13(4), 348
 83. Al-Dweiri, M., Ramadan, B., Rawshdeh, A., Nassoura, A., Al-Hamad, A., & Ahmad, A. (2024). The mediating role of lean operations on the relationship between supply chain integration and operational performance. *Uncertain Supply Chain Management*, 12(2), 1163-1174.
 84. Lin, C., Ahmad, S. F., Ayassrah, A. Y. B. A., Irshad, M., Telba, A. A., Awwad, E. M., & Majid, M. I. (2023). Green production and green technology for sustainability: The mediating role of waste reduction and energy use. *Heliyon*, e22496.
 85. K. Daoud, D. . Alqudah, M. . Al-Qeed, B. A. . Al Qaied, and A. Y. A. B. . Ahmad, "The Relationship Between Mobile Marketing and Customer Perceptions in Jordanian Commercial Banks: The Electronic Quality

- as A Mediator Variable", *ijmst*, vol. 10, no. 2, pp. 1360-1371, Jun. 2023
86. Mohammad Jebreel, Mohammad Alnaimat, Amjad Al-Shorafa, Majed Qabajeh, Mohammad Alqsass, & Ahmad Bani Ahmad. (2023). The Impact of Activity Ratios on Change in Earnings (Case Study:Based on Jordanian Food Companies). *Kurdish Studies*, 11(2), 4551-4560. Retrieved from <https://kurdishstudies.net/menu-script/index.php/KS/article/view/1044>
 87. Mohammad Alqsass, Munir Al-Hakim, Qais Al Kilani, Lina Warrad, Majed Qabajeh, Ahmad Y. A.Bani Ahmad, & Adnan qubbaja. (2023). The Impact of Operating Cash Flow on Earnings Per Share (Case Study Based on Jordanian Banks). *Kurdish Studies*, 11(2), 2718-2729. Retrieved from <https://kurdishstudies.net/menu-script/index.php/KS/article/view/831>
 88. Mohammad Alqsass, Munir Al-Haki, Mohammad Dweiri, Majed Qabajeh, Dmaithan almajali, Ahmad Bani Ahmad, & Adnan Qubbaja. (2023). The Impact of Current Ratio on Net Profit Margin (Case Study: Based on Jordanian Banks). *Kurdish Studies*, 11(2), 2894-2903. Retrieved from <https://kurdishstudies.net/menu-script/index.php/KS/article/view/834>
 89. Mustafa, J. A., ATTA, A. A. B., AHMAD, A. Y. B., SHEHADEH, M., & Agustina, R. (2023). Spillover Effect in Islamic and Conventional Fund Family: Evidence from Emerging Countries. *WSEAS Transactions on Business and Economics*, 20, 1042-1058.
 90. Mohsin, H. J., Hani, L. Y. B., Atta, A. A. B., Al-Alawneh, N. A. K., Ahmad, A. B., & Samara, H. H. (2023). THE IMPACT OF DIGITAL FINANCIAL TECHNOLOGIES ON THE DEVELOPMENT OF ENTREPRENEURSHIP: EVIDENCE FROM COMMERCIAL BANKS IN THE EMERGING MARKETS.
 91. Ni, L., Ahmad, S. F., Alshammari, T. O., Liang, H., Alsanie, G., Irshad, M., ... & Ayassrah, A. Y. B. A. (2023). The role of environmental regulation and green human capital towards sustainable development: The mediating role of green innovation and industry upgradation. *Journal of Cleaner Production*, 138497.
 92. Peng, Yixuan, Sayed Fayaz Ahmad, Ahmad Y. A. Bani Ahmad, Mustafa S. Al Shaikh, Mohammad Khalaf Daoud, and Fuad Mohammed Hussein Alhamdi. 2023. "Riding the Waves of Artificial Intelligence in Advancing Accounting and Its Implications for Sustainable Development Goals" *Sustainability* 15, no. 19: 14165.
 93. Peiran Liang, Yulu Guo, Sohaib Tahir Chauhdary, Manoj Kumar Agrawal, Sayed Fayaz Ahmad, Ahmad ,Yahiya Ahmad Bani Ahmad, Ahmad A. Ifseisi, Tiancheng Ji,2024"Sustainable development and multi-aspect analysis of a novel polygeneration system using biogas upgrading and LNG regasification processes ,producing power, heating, fresh water and liquid CO2",*Process Safety and Environmental Protection*
 94. Peiran Liang, Yulu Guo, Tirumala Uday Kumar Nutakki, Manoj Kumar Agrawal, Taseer Muhammad, Sayed ,Fayaz Ahmad, Ahmad Yahya Ahmad Bani Ahmad, Muxing Qin 2024. " Comprehensive assessment and sustainability improvement of a natural gas power plant utilizing an environmentally friendly combined cooling heating and power-desalination arrangement",*Journal of Cleaner Production*,Volume 436,,140387
 95. Rumman, G., Alkhazali, A., Barnat, S., Alzoubi, S., AlZagheer, H., Dalbough, M., ... & Darawsheh, S. (2024). The contemporary management accounting practices adoption in the public industry: Evidence from Jordan. *International Journal of Data and Network Science*, 8(2), 1237-1246.
 96. Singh, R., Gupta, N. R., & Ahmad, A. Y. (2024). An Empirical Study on Challenges of Working From Home During COVID-19 on Work-Life Domains in the Education Sector in Bengaluru. In S. Singh, S. Rajest, S. Hadoussa, A. Obaid, & R. Regin (Eds.), *Data-Driven Intelligent Business Sustainability* (pp. 111-121). IGI Global. <https://doi.org/10.4018/979-8-3693-0049-7.ch008>
 97. William, P., Ahmad, A. Y. B., Deepak, A., Gupta, R., Bajaj, K. K., & Deshmukh, R. (2024). Sustainable Implementation of Artificial Intelligence Based Decision Support System for Irrigation Projects in the Development of Rural Settlements. *International Journal of Intelligent Systems and Applications in Engineering*, 12(3s), 48-56.
 98. Wang, C., Ahmad, S. F., Ayassrah, A. Y. B. A., Awwad, E. M., Irshad, M., Ali, Y. A., ... & Han, H. (2023). An empirical evaluation of technology acceptance model for Artificial Intelligence in E-commerce. *Heliyon*, 9(8).
 99. Yahya Ahmad Bani Ahmad (Ayassrah), Ahmad; Ahmad Mahmoud Bani Atta, Anas; Ali Alawawdeh, Hanan; Abdallah Aljundi, Nawaf; Morshed, Amer; and Amin Dahbour, Saleh (2023) "The Effect of System Quality and User Quality of Information Technology on Internal Audit Effectiveness in Jordan, And the Moderating Effect of Management Support," *Applied Mathematics & Information Sciences*: Vol. 17: Iss. 5, Article 12.
DOI: <https://dx.doi.org/10.18576/amis/170512>
 100. Zhan, Y., Ahmad, S. F., Irshad, M., Al-Razgan, M., Awwad, E. M., Ali, Y. A., & Ayassrah, A. Y. B. A. (2024). Investigating the role of Cybersecurity's perceived threats in the adoption of health information

- systems. *Heliyon*, 10(1).
101. Raza, A., Al Nasar, M. R., Hanandeh, E. S., Zitar, R. A., Nasereddin, A. Y., & Abualigah, L. (2023). A Novel Methodology for Human Kinematics Motion Detection Based on Smartphones Sensor Data Using
102. Wu, J., Ahmad, S. F., Ali, Y. A., Al-Razgan, M., Awwad, E. M., & Ayassrah, A. Y. B. A. (2024). Investigating the role of green behavior and perceived benefits in shaping green car buying behavior with environmental awareness as a moderator. *Heliyon*, 10(9).
103. Yahiya, A., & Ahmad, B. (2024). Automated debt recovery systems: Harnessing AI for enhanced performance. *Journal of Infrastructure, Policy and Development*, 8(7), 4893.
104. Al-Waely, D., Fraihat, B. A. M., Al Hawamdeh, H., Al-Taee, H., & Al-Kadhimi, A. M. M. N. (2021). Competitive Intelligence Dimensions as a Tool for Reducing the Business Environment Gaps: An Empirical Study on the Travel Agencies in Jordan. *Journal of Hunan University Natural Sciences*, 48(11).
105. Zhao, T., Ahmad, S. F., Agrawal, M. K., Ahmad, A. Y. A. B., Ghfar, A. A., Valsalan, P., ... & Gao, X. (2024). Design and thermo-enviro-economic analyses of a novel thermal design process for a CCHP-desalination application using LNG regasification integrated with a gas turbine power plant. *Energy*, 295, 131003.
106. Geetha, B. T., Kafila, K., Ram, S. T., Narkhede, A. P., Ahmad, A. Y. B., & Tiwari, M. (2024, March). Creating Resilient Digital Asset Management Frameworks in Financial Operations Using Blockchain Technology. In *2024 International Conference on Trends in Quantum Computing and Emerging Business Technologies* (pp. 1-7). IEEE.
107. Open Worldwide Application Security Project (OWASP). (2025). *OWASP Top 10:2025*. <https://owasp.org/www-project-top-ten/>
108. Open Worldwide Application Security Project (OWASP). (2026a). *OWASP DevSecOps Verification Standard*. <https://owasp.org/www-project-devsecops-verification-standard/>
109. Open Worldwide Application Security Project (OWASP). (2026b). *Software supply chain security cheat sheet*. OWASP Cheat Sheet Series. <https://cheatsheetseries.owasp.org/>
110. Prates, L., & Pereira, R. (2025). DevSecOps practices and tools. *International Journal of Information Security*, 24(1), 11. <https://doi.org/10.1007/s10207-024-00914-z>
111. Rajapakse, R. N., Zahedi, M., Babar, M. A., & Shen, H. (2022). Challenges and solutions when adopting DevSecOps: A systematic review. *Information and software technology*, 141, 106700. <https://doi.org/10.1016/j.infsof.2021.106700>
112. Sinan, M., Shahin, M., & Gondal, I. (2025). Integrating security controls in DevSecOps: Challenges, solutions, and future research directions. *Journal of Software: Evolution and Process*, 37(6), e70029. <https://doi.org/10.1002/smr.70029>
113. Souppaya, M., Scarfone, K., & Dodson, D. (2022). Secure software development framework (ssdf) version 1.1. *NIST Special Publication, 800(218)*, 800-218. <https://doi.org/10.6028/NIST.SP.800-218>
114. Valdés-Rodríguez, Y., Hochstetter-Diez, J., Díaz-Arancibia, J., & Cadena-Martínez, R. (2023). Towards the integration of security practices in agile software development: A systematic mapping review. *Applied Sciences*, 13(7), 4578. <https://doi.org/10.3390/app13074578>
115. Zhang, X., Zhao, P., Jaskolka, J., Li, H., & Lu, R. (2026). SecMLOps: A comprehensive framework for integrating security throughout the machine learning operations lifecycle. *Empirical Software Engineering*, 31(3), 74. <https://doi.org/10.1007/s10664-025-10795-y>
116. Zhao, X., Clear, T., & Lal, R. (2024). Identifying the primary dimensions of DevSecOps: A multi-vocal literature review. *Journal of Systems and Software*, 214, 112063. <https://doi.org/10.1016/j.jss.2024.112063>
117. Zhou, X., Mao, R., Zhang, H., Dai, Q., Huang, H., Shen, H., ... & Rong, G. (2023). Revisit security in the era of DevOps: An evidence-based inquiry into DevSecOps industry. *IET software*, 17(4), 435-454. <https://doi.org/10.1049/sfw2.12132>