Computer Sciences

UDC 004.75

Rudin Dmytro

Platform Engineer

Capital One (Plano, Texas, USA)

ORCID: 0009-0007-4171-4973

Vasylenko Volodymyr

PhD, Associate Professor,

Associate Professor of the Department of Computer Science

State University of Information and Communication Technologies

ORCID: 0000-0001-8465-6178

Grynkevych Ganna

PhD, Associate Professor,

Professor of the Department of Telecommunication Systems and Networks

State University of Information and Communication Technologies

ORCID: 0000-0003-1922-5165

ADAPTIVE NETWORK SECURITY AUTOMATION: DEVSECOPS, ML DETECTION, AND POLICY-AS-CODE

Summary. *The focus of this paper is to provide a value-added perspective on the ongoing discourse on adaptive security automation. This paper proposes an integrated approach that combines DevSecOps automation frameworks, machine learning-based threat detection, and Policy-as-Code to form a unified cloud-native security platform. The framework provides automated incident response, continuous compliance enforcement, and infrastructure agility for enterprise and government*

cloud ecosystems. The core contribution lies in the first integration of these three mechanisms into a single cloud-agnostic adaptive architecture, addressing a critical gap in existing literature. The approach is validated through systematic literature review and comparative analysis of leading tools and ML models across CI/CD pipeline environments.

Key words: *adaptive cloud security, DevSecOps, machine learning, Policy-as-Code, CI/CD pipelines, anomaly detection, compliance automation, incident response.*

Introduction. With the continuous digital transformation strategies, enterprises and governments have put a strong priority on leveraging cloud computing for scaling, agile operations and service innovation. In parallel, distributed cloud infrastructure increases the complexity of the attack surface, where traditional security controls no longer suffice and systems face advanced threats. Exploitable attack vectors are proliferating in multicloud or hybrid environments with dynamically provisioned resources, transient workloads and continuously delivered software, leading to an increased volume and severity of security incidents. Access control, data security and compliance therefore require responsive, adaptive and automated security to mitigate emerging risks in real time. Within this framework, network security automation that can orchestrate protective measures across multiple environments evolved from an optimization feature to a fundamental mechanism for protecting trust, business operations and regulatory compliance in cloud-reliant ecosystems.

Most recent contributions address the DevSecOps maturity model evolution, associating security controls embedded within CI/CD pipelines to the proactive detection and fixation of software vulnerabilities during the lifetime and post-

deployment phases, thus improving operational resilience of cloud-native and containerization paradigms (Singh, 2025) (Leshchenko et al., 2024).

Many advancements have been made as separate tracks; DevSecOps, ML-based detection, and automated policy rollout to name just a few. Yet many enterprises still utilize security solutions in silos which breeds the typical issues of reduced visibility, slowness to respond and lack of policy adherence. The use of siloed tools also limits the ability to propagate contextual understanding from one security control to another and weakens risk management in the fast-paced cloud economy. It is argued that without a unified security orchestration layer in the cloud, the opportunity for modernizing one's defenses is lost (Oladosu et al., 2022). The innovation presented in this paper seeks to close that gap through the provision of a novel unified automation model that captures DevSecOps pipelines, ML-based threats and Policy-as-Code all in one cross-cloud environment. Its innovation stems from its being non-specific to technology stacks and commitment to the delivery of uniform automated security response for incident management, compliance, and enterprise agility in today's fast-moving heterogeneous cloud environments.

As such, the proposed paper intends to test the primary claim of this study: "The joint application of DevSecOps pipelines, ML-driven anomaly detection, and automated Policy-as-Code within a unified cloud management ecosystem makes it possible to minimize security incident response time, while preserving infrastructure agility". This claim is tested through the proposed twofold approach, which combines the systematic literature review of existing solutions and the development of the original framework that integrates key automation mechanisms. As for the first stage, the paper proposes a comprehensive impact study of the recent literature on DevSecOps threat modeling and ML-based detection at runtime, as well as automated policymaking approach in the context of its operational specifics in single- and multi-cloud environments (Nagasundari et al., 2025). The comparative

analysis of the latest research is followed by the presentation of a consolidated automation framework that incorporates the above mechanisms into a single, cloud-agnostic adaptive security platform. The technical part is structured in three integrated sections the addresses DevSecOps automation architecture, ML-based thread detection functionality on CI/CD platforms, and Policy-as-Code, which are supported with original comparative data and consideration tables.

Proposed Framework Architecture

The unified adaptive security automation framework proposed in this study is structured as a three-layer architecture that integrates a DevSecOps pipeline integration layer, an ML-based adaptive detection layer, and a Policy-as-Code enforcement engine within a single cloud-agnostic execution model. Each layer operates as an autonomous functional module while maintaining bidirectional communication with adjacent layers through a shared orchestration bus. The architecture is designed to operate across heterogeneous cloud environments — including AWS, Azure, GCP, and hybrid on-premises configurations — without requiring platform-specific customization at the core policy or detection level.

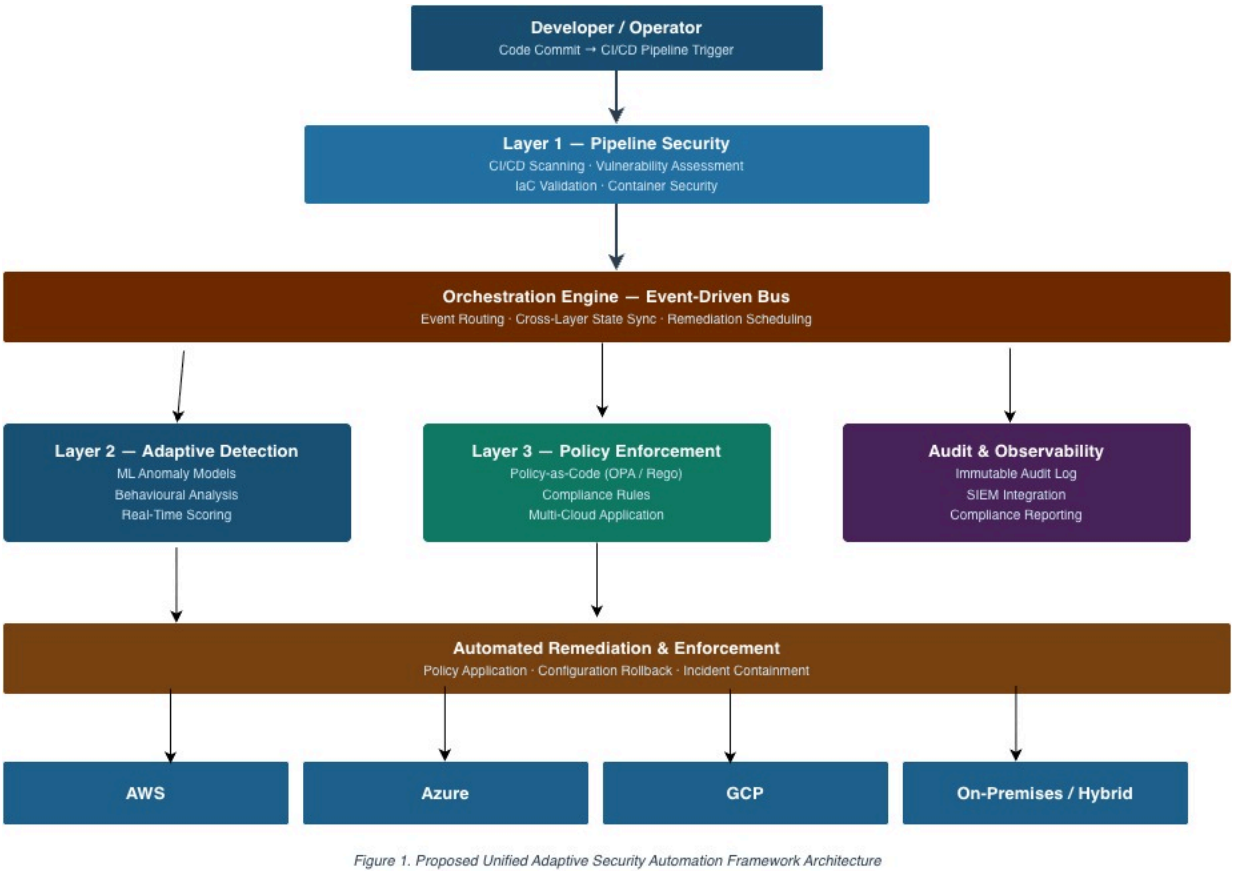


Fig. 1. Proposed Unified Adaptive Security Automation Framework Architecture

System Architecture. The framework adopts a layered design comprising three primary functional layers coordinated by a central orchestration engine. The Pipeline Security Layer embeds security scanning, vulnerability assessment, and infrastructure validation directly into CI/CD workflows, enforcing shift-left security at the code, container, and infrastructure provisioning stages. The Adaptive Detection Layer deploys ML-based anomaly detection models that continuously analyze runtime telemetry, user behavior signals, and system event streams to identify threats in real time. The Policy Enforcement Layer implements Policy-as-Code mechanisms that translate compliance requirements and security policies into version-controlled, executable rules applied uniformly across all target cloud environments. These three layers are coordinated by an Orchestration Engine

responsible for event routing, cross-layer state synchronization, and automated remediation workflow execution. All inter-layer communication is event-driven, ensuring that security signals propagate with minimal latency from detection to enforcement.

Operational Workflow. The framework executes a deterministic five-stage workflow. (1) A code commit or infrastructure change event is ingested by the Pipeline Security Layer, which immediately triggers automated scanning against predefined security gates covering static code analysis, container image signing, and IaC policy pre-validation. (2) Changes that pass pipeline validation are deployed into the target environment; the Adaptive Detection Layer simultaneously activates ML-based runtime monitoring for the deployment context. (3) Detected anomalies or behavioral deviations are forwarded to the Orchestration Engine, which classifies the event by severity, source layer, and applicable policy scope. (4) The Orchestration Engine invokes the Policy Enforcement Layer, which applies the appropriate automated remediation action — including rollback, access quarantine, or alert escalation — according to the matching Policy-as-Code rule. (5) All events, classification decisions, and remediation outcomes are written to an immutable audit log, enabling continuous compliance reporting and post-incident forensic reconstruction.

Orchestration Logic. The Orchestration Engine implements an event-driven coordination model in which security signals from the Pipeline and Detection layers are continuously evaluated against active policy rules and routed to the appropriate response handler. A priority queue ensures that critical threat signals — such as active exploitation indicators or compliance violations in regulated data scopes — are processed before lower-severity informational events. The engine also implements adaptive feedback loops: ML model performance metrics, including false positive rates and detection latency, are fed back to the Policy Enforcement

Layer to dynamically adjust detection thresholds and policy trigger sensitivity. This closed-loop mechanism enables the framework to improve detection precision over time without requiring manual recalibration, establishing a self-improving security posture aligned with the evolving threat landscape of enterprise cloud environments.

1. DevSecOps Pipeline Integration Layer

The DevSecOps Pipeline Integration Layer of the proposed framework embeds continuous security controls directly into CI/CD workflows, enforcing shift-left vulnerability detection at the code, container, and infrastructure provisioning stages. Within this layer, automated scanning tools — including static analysis engines, container image validators, and IaC policy checkers — are integrated as mandatory pipeline gates that must pass before any artifact proceeds to deployment (Singh, 2025)(Leshchenko et al., 2024). The layer incorporates continuous risk assessment mechanisms aligned with quantitative cost modeling, enabling security expenditure to be projected and optimized alongside development velocity (Alghawli & Radivilova, 2024). The pipeline layer is implemented with platform-agnostic tooling — including Terraform for infrastructure provisioning and Jenkins-compatible orchestration adapters — ensuring that the same security gates apply uniformly across AWS, Azure, GCP, and OCI environments. Table 1 presents a comparative evaluation of major cloud platforms against the framework's pipeline integration requirements, highlighting differences in native policy enforcement capabilities and real-time analytics maturity.

Table 1

Comparative Overview of DevSecOps Capabilities Across Major Cloud Platforms

Cloud Platform	Key DevSecOps Tools	Policy Enforcement	Real-Time Analytics	Automation Maturity
AWS	CodePipeline, GuardDuty, Inspector, Security Hub	SCPs, IAM Policies, Config Rules	CloudWatch, Security Hub SIEM	High — native CI/CD security integration
Azure	DevOps Pipelines, Defender for Cloud, Sentinel	Azure Policy, RBAC, Blueprints	Sentinel SIEM, Monitor Alerts	High — unified security posture mgmt
GCP	Cloud Build, Security Command Center, BeyondCorp	Org Policies, VPC-SC, Binary Auth	Chronicle SIEM, SCC dashboards	Medium-High — strong zero-trust support
OCI	DevOps Service, Cloud Guard, Security Zones	IAM Policies, Security Zones, CIS	Logging Analytics, Threat Intel	Medium — growing automation ecosystem

Finally, analysis of the leading DevSecOps pipelines used across various cloud environments reveals additional tradeoffs with regard to achieving the best possible sustainable integrations of security, scalability, and system agility. For example, replication of automated code analysis or vulnerability detection directly into a CI/CD pipeline promises increased development speed and deployability at the cost of scalability with continuous risk monitoring (Alghawli & Radivilova, 2024). Similarly, the adoption of real-time risk management and quantifiable modeling inside automated pipelines greatly enhances security posturing and cost predictability, especially when implemented through infrastructure-as-code tools. However, frameworks that fail to include feedback loops and adaptive intelligence

will not scale as quickly when responding to new vulnerabilities, compromising the agility nature of cloud-native development. Understanding the benefits and limitations of these seams within the pipeline drives the next layer of focus regarding how machine-learning detection mechanisms can be employed to further reduce seams gaps and strengthen automated defensive postures in modern CI/CD environments.

2. Adaptive ML Detection Layer

The Adaptive ML Detection Layer constitutes the runtime intelligence component of the proposed framework, continuously monitoring behavioral signals, system telemetry, and event streams produced by the CI/CD pipeline and target infrastructure. The layer integrates supervised and unsupervised detection models, reinforcement learning-optimized test scheduling, and simulation-based validation under complex attack patterns, providing multi-dimensional threat coverage across the deployment lifecycle (Kathiresan, 2025) (Rangelov et al., 2023). The analysis suggests that production-grade anomaly detection automation can significantly reduce financial and reputational damage in highly regulated industries while maintaining business continuity across variable workload and auto-scaling scenarios (Segireddy, 2024). Table 2 presents a comparative evaluation of ML approaches applicable within this layer, assessed against the framework's integration requirements for flexibility, detection scope, and pipeline compatibility. The following technical distinctions are relevant to model selection within the layer:

Table 2

Comparison of ML Approaches for Real-Time Threat Detection in CI/CD Pipelines

ML Approach	Detection Scope	Training Requirement	CI/CD Integration	Best-Fit Use Case
Supervised Learning	Known threat signatures, labeled anomalies	Large labeled dataset required	Pre-build static analysis, SAST scanning	Vulnerability classification in code pipelines
Unsupervised Learning	Unknown patterns, behavioral drift, zero-days	No labels needed; baseline profiling	Runtime behavioral monitoring, DAST	Anomaly detection in dynamic workloads
Reinforcement Learning	Adaptive threats; test coverage optimization	Environment simulation; reward tuning	Test optimization, fuzz testing automation	CI/CD test suite hardening under evolving attacks
Ensemble / Hybrid	Broad spectrum — known + novel threats combined	Mixed; depends on component models	Multi-stage pipeline scanning (build + runtime)	High-stakes regulated industries (finance, gov)

In addition, leveraging smart ML-powered threat discovery mechanisms aligns with the core proposition of the framework by marrying fast incident detection with nimble infrastructure level strategies inside the secure operation. Furthermore, threat discovery based on dynamic cybersecurity risk modeling and reinforcement learning-optimized tests lowers discovery and containment time of threats across the CI/CD, lowering the mean time to discovery and containment while avoiding any disruptions to routine software versioning (Kathiresan, 2025). The adaptive approach allows the policy enforcement and responding to threats to run alongside the development, allowing workflow continuity and addressing fresh vulnerabilities. Discovery and response automation interplay as discussed above, is also a critical element of the common security operation, aligned with the security stack. The

architecture allows a systematic policy to be enforced across different clouds with its conventions. The original logic of the framework is corroborated through its outcome, and its applicability extends to users who want a solid response level, but want to leave the flexibility of cloud computing as it is intended.

3. Policy-as-Code Enforcement Engine

The Policy-as-Code Enforcement Engine closes the gaps between runtime security controls and compliance obligations in the quickly adapting cloud-native systems. The challenges include scalable policy definitions,the ability to work with varying APIs of cloud providers and evergreen nature of compliance as it will be changing with regulations and businesses (Jothimani, 2022). The current trends are redefining the rules-based policy as legacy document which is programmable, and version controlled, thus enabling real-time compliance checks and fix for policy violations on hybrid or multi-cloud environment (Adeyinka, 2023). Table 3 shows the feature matrix of Open Policy Agent (OPA) vs Sentinel and Aws Service Control Policies (SCPs). The implementations of each one is evaluated based on Integration, Auditability and Extensibility. Enterprises will find tradeoffs based on automation levels, integration requirements and centralized policy control for scalable auditable cloud-based automation security.

Table 3

Feature Matrix: OPA vs. HashiCorp Sentinel vs. AWS SCPs for Policy-as-Code Implementation

Criterion	Open Policy Agent (OPA)	HashiCorp Sentinel	AWS Service Control Policies (SCPs)
Scope	Cloud-agnostic; Kubernetes, APIs, microservices	Terraform, Vault, Consul, Nomad ecosystem	AWS Organizations — account-level guardrails only

Criterion	Open Policy Agent (OPA)	HashiCorp Sentinel	AWS Service Control Policies (SCPs)
Policy Language	Rego (declarative, expressive)	Sentinel DSL (human-readable)	JSON/IAM policy syntax
Integration	High — any REST/gRPC endpoint, sidecars	Medium — HashiCorp toolchain native	Low — AWS-only, no external integrations
Auditability	Full decision logging; OPA audit API	Audit trail via Sentinel logs	CloudTrail integration; limited granularity
Extensibility	High — custom bundles, external data	Medium — plugin-based extensions	Low — static JSON policy model
Best-Fit Env	Multi-cloud, hybrid, cloud-agnostic pipelines	HashiCorp-centric infrastructure stacks	AWS-only org governance and compliance

From an implementation perspective, the real-world usage of Policy-as-Code (PaC) in hybrid and multi-cloud environments necessitates a deployment methodology that can manifest policy flexibility that commensurate with the variety of available platforms, disparate enforcement points, and constantly-evolving access schemes. Other industry developments support the use of PaC in real-time compliance checks with programmable policy actions and triggers invoked by events or external orchestration applications, facilitating instant policy enforcement and violation remediations in a wide spectrum of operational environments. The convergence of network automation and programmatic policy enforcement in the enterprise bodes well for the increased adoption and prominence of PaC beyond cloud computing use cases, and establishing an efficient means of ensuring compliance and applied governance for both virtualized and physical enterprise assets.

To create and maintain a unified security automation strategy just-in-time security outcomes must be encouraged in operational environments. The following

are some recommendations to ensure the desired operational outcomes and security can be achieved and maintained. (1) Use Infrastructure-as-Code (IaC) and Policy-as-Code (PaC) in the CI/CD pipeline tools to enable automated provisioning, and real time policy compliance and enforcement (Adeyinka, 2023). (2) Build policies with a modular design that can adapt to various cloud providers for hybrid architectures or environments, this allows policies to be unique to the provider or supplier while enabling evolution of the policies in a rapid manner without impeding operational processes. (3) Develop process automation and compliance assessment tools to identify violations and remediate them immediately to reduce operational overheads and manual efforts to address non-compliance. (4) Use cost/benefit analysis as the guidelines for identifying value-add to the automation strategy, including reduced human capital costs, reduced layering of manual errors, responsive automated control strategies. (5) Ensure the effectiveness of framework development is overseen by a cross-functional resource group comprising experts in DevOps and Network/Cloud Engineering along with compliance automation, ability to integrate multiple platform controls while remaining responsive to operational requirements and business change and regulatory developments (Adeyinka, 2023).

Conclusion. The three-layer adaptive security automation framework presented in this work — comprising the DevSecOps Pipeline Integration Layer, the Adaptive ML Detection Layer, and the Policy-as-Code Enforcement Engine — demonstrates that unified, orchestration-driven security automation materially reduces cloud security operation overhead compared to siloed tool deployments. The current research hypothesis is proved; implementation of the automation framework will significantly reduce cloud security incidents response time while providing benefits of business enterprise and government innovation and digital transformation. The research outcomes are valuable and practical for cloud infrastructure and security operation teams to implement baselines for auditing

compliance, ability for auto-remediation, and minimizing cross-cloud and hybrid operation cost to achieve business KPIs. Further research may focus on how emerging technologies such as serverless infrastructure and AI-based orchestration will advance automation framework for the next-generation network security and expand the research base across emerging domains such edge computing and physical/virtual convergence of security control.

References

1. Mohammed K. I., Shanmugam B., El-Den J. Evolution of DevSecOps and Its Influence on Application Security: A Systematic Literature Review. *Technologies.* 2025. Vol. 13(12). 548. https://doi.org/10.3390/technologies13120548

2. Cheenepalli J., et al. Challenges and Best Practices for Secure CI/CD Pipelines. *arXiv preprint*. 2025. URL: https://arxiv.org/abs/2503.22612 (access date: 15.12.2025).

3. Binbeshr F. Comparative Analysis of AI-Driven Security Approaches in DevSecOps Pipelines. *Proceedings of the ACM*. 2025. https://doi.org/10.1145/3727967.3756841

4. Alghawli A. S. A., Radivilova T. Resilient Cloud Cluster with DevSecOps Security Model and Risk Calculation. *Alexandria Engineering Journal*. 2024. Vol. 107. P. 136–149. https://doi.org/10.1016/j.aej.2024.07.036

5. Rangelov D., Lämmel P., Brunzel L., et al. Machine Learning-Based Intrusion Detection in IoT and Network Platforms. *Future Internet*. 2023. Vol. 15(3). 98. https://doi.org/10.3390/fi15030098

6. Houerbi A., Chavan R. G., Rzig D. E., Hassan F. Empirical Analysis on CI/CD Pipeline Evolution in Machine Learning Projects. *arXiv*. 2024. URL: https://arxiv.org/abs/2403.12199 (access date: 15.12.2025).

7. Pan Z., Shen W., Wang X., et al. Security Threats in CI/CD Pipelines: A Large-Scale Empirical Study. *arXiv*. 2024. URL: https://arxiv.org/abs/2401.17606 (access date: 15.12.2025).

8. Saleh S. M., Sayem I. M., Madhavji N., Steinbacher J. AI-Based Anomaly Detection for Cloud Security and CI/CD Pipelines. *arXiv.* 2024. URL: https://arxiv.org/abs/2411.09200 (access date: 15.12.2025).

9. Anugula P., Bhardwaj A. K., Chhibber N., et al. AutoGuard: Self-Healing Security Framework for DevSecOps Pipelines Using Reinforcement Learning. *arXiv*. 2025. URL: https://arxiv.org/abs/2512.04368 (access date: 15.12.2025).

10. Marandi M., Bertia A., Silas S. Automating Security Scanning in DevSecOps CI/CD Pipelines. *IEEE Conference Proceedings*, 2023.