Scholars Journal of Engineering and Technology
OPEN ACCESS
Abbreviated Key Title: Sch J Eng Tech
ISSN 2347-9523 (Print) | ISSN 2321-435X (Online)
Journal homepage: https://saspublishers.com

Secure Continuous Deployment Pipelines for Enterprise Cloud Applications

Mabu Hussain Shaik[1]*, Shujath Baig Mirza[2], Md Ariful Islam[3], Farhan Tariq[4]

[1]Master of Science in Information Technology Management, University- Campbellsville University, KY, United States
[2]Master of Science in Engineering Management, University- Saint Martin's University, Lacey, Washington
[3]MSC in Computer Science and Engineering, Jagannath University, Bangladesh
[4]Department of Software Engineering, University- University of Gujrat, Sialkot Campus, Pakistan

DOI: https://doi.org/10.36347/sjet.2026.v14i05.005 | **Received:** 03.03.2026 | **Accepted:** 20.04.2026 | **Published:** 13.05.2026

***Corresponding author:** Mabu Hussain Shaik
Master of Science in Information Technology Management, University- Campbellsville University, KY, United States

Abstract | **Original Research Article**

Secure continuous deployment has become an important requirement for enterprise cloud applications, where rapid software release must coexist with security control, service stability, and operational accountability. Conventional CI/CD pipelines often focus on build and test automation, yet they provide limited support for integrated release screening, staged production control, runtime-governed continuation, and traceable governance records. This paper proposes a secure continuous deployment framework for enterprise cloud environments that combines security inspection, policy-aware release admission, staged rollout, runtime observation, and rollback control within a single deployment process. The method evaluates release candidates through multi-stage validation of source code, dependencies, container artifacts, infrastructure definitions, and configuration state before controlled production exposure. Production progression occurs through canary and partial rollout phases, while runtime signals such as logs, metrics, traces, and security alerts affect continuation and recovery decisions. The results show that the proposed framework improves pre-release risk detection, reduces failed production releases, shortens rollback response time, increases runtime anomaly detection coverage, and improves audit trace completeness when compared with a baseline CI/CD pipeline. The study shows that secure deployment in enterprise cloud applications requires more than automated release execution. It requires coordinated release control in which security, runtime behavior, and governance evidence operate as connected parts of one deployment model.

Keywords: Secure Continuous Deployment, DevSecOps, Enterprise Cloud Applications, CI/CD Pipelines, Release Risk Management, Canary Deployment, Runtime Monitoring, Deployment Automation, Cloud Security, Software Delivery Systems.

I. INTRODUCTION

Enterprise software delivery has changed significantly with the growth of cloud computing, microservices, containers, and infrastructure automation. Release pipelines now influence not only development speed but also system reliability, security posture, operational control, and business continuity. In enterprise environments, deployments often affect customer-facing services, internal process platforms, identity systems, data operations, and regulated workflows. As a result, deployment can no longer be treated as a narrow technical handoff between development and operations. It has become a structured operational activity with direct consequences for service stability and organizational risk. Although continuous integration and continuous deployment have improved development efficiency, many pipelines still focus heavily on build completion and test success. That approach is often insufficient for enterprise cloud applications, where release risk may also come from vulnerable dependencies, insecure configurations, container image flaws, policy violations, or unstable runtime behavior. This paper addresses that gap through a secure continuous deployment framework that combines release screening, staged rollout, runtime control, and traceable governance within one deployment process.

A. Background and Motivation

Cloud-native enterprise applications operate in environments marked by rapid software change, distributed services, automated infrastructure, and continuous operational pressure. Development teams are expected to deliver updates quickly, while operations teams must maintain service continuity and policy compliance. This tension creates a strong need for

deployment pipelines that support both speed and disciplined release control. Traditional release processes depended on periodic deployment windows and substantial manual review. Modern enterprise systems no longer function well under that model because updates occur more frequently and involve many interacting components. A single release may include code revisions, dependency changes, infrastructure template updates, API modifications, and configuration adjustments. These changes do not carry the same risk, yet many pipelines still evaluate them through a limited set of standard checks. That limitation motivates this study. Organizations need deployment mechanisms that examine technical quality, security state, runtime readiness, and approval evidence in a coordinated manner. The motivation is therefore practical as well as technical: safe delivery matters because deployment failure can interrupt services, trigger compliance concerns, and increase recovery effort across enterprise cloud platforms.

B. Problem Statement

Despite broad adoption of automated CI/CD practices, many enterprise deployment pipelines still contain structural weaknesses that limit production safety. One major problem lies in release admission logic. In many cases, promotion decisions depend mainly on successful builds and test execution, even though unresolved dependency risk, container issues, insecure configuration states, or policy violations may still remain. A second problem concerns production exposure. Releases are often promoted too broadly and too quickly, with little opportunity to observe runtime behavior under limited traffic before wider rollout. When hidden issues emerge after deployment, faults can spread across services before operators intervene. A third problem involves the weak connection between runtime monitoring and deployment control. Logs, metrics, traces, and alerts may be collected, but they often support incident review only after release problems occur. They are not consistently used as direct signals for rollout continuation, pause, or rollback. Taken together, these weaknesses create a gap between automation and safe production entry. Enterprise organizations therefore face a recurring conflict between rapid delivery and controlled deployment.

C. Proposed Solution

This paper proposes a secure continuous deployment framework designed for enterprise cloud applications that require controlled release admission and stable production behavior. The framework treats deployment as a managed operational process rather than a direct path from build completion to production promotion. It integrates security inspection, release scoring, policy-aware approval, staged rollout, runtime observation, and rollback control within one pipeline. Each release candidate moves through multiple validation stages before broader production exposure. These stages examine source code, dependencies, container artifacts, infrastructure definitions, and configuration state. The framework then applies decision rules that consider both technical findings and release conditions. Production progression occurs gradually through canary deployment and partial rollout instead of direct full release. Runtime evidence collected during live exposure affects the continuation of that rollout. If service behavior degrades, the pipeline can pause the release or restore the last stable version. At the same time, the system records inspection output, artifact identity, deployment state, and approval evidence. This structure supports release control, operational visibility, and governance traceability within a single deployment model for enterprise use.

D. Contributions

This paper makes five main contributions to secure software delivery research for enterprise cloud systems. First, it presents a unified deployment framework that combines security inspection, release control, runtime observation, and governance tracking in one operational process. Second, it introduces a release admission model that depends on more than build completion and test success. The proposed method evaluates release readiness through technical checks, policy conditions, and runtime-related evidence. Third, the framework places staged deployment at the center of production control. Canary release and partial rollout are treated as core decision points rather than optional operational practices. Fourth, the method connects runtime telemetry directly to deployment progression and recovery actions. Monitoring therefore becomes an active part of release control rather than a passive post-release function. Fifth, the framework integrates audit evidence into the deployment lifecycle so that approvals, inspection records, release states, and rollback outcomes remain connected to each candidate. Together, these contributions address the need for a more disciplined and traceable release model suited to enterprise cloud applications with high operational and administrative demands.

E. Paper Organization

The remainder of this paper is structured to move from prior knowledge to system design, then to evaluation and interpretation. Section II reviews related work on secure software delivery, DevSecOps, deployment control, cloud operations, runtime observation, and enterprise governance. That section identifies the limits of existing studies and clarifies the gap addressed in this paper. Section III presents the methodology, including the deployment architecture, multi-stage validation process, release decision logic, staged rollout model, runtime control process, and evaluation framework. Section IV reports the results and discussion. It compares the baseline and proposed pipelines across release screening, deployment outcomes, runtime stability, recovery behavior, and governance visibility. The section also interprets those findings in terms of enterprise deployment practice and

the main contribution of the proposed framework. Section V concludes the paper with a summary of the major findings and a brief future work discussion. This sequence supports a clear progression from motivation and problem definition to technical design, observed outcomes, and final interpretation within the scope of secure continuous deployment.

F. Research Objective

The objective of this paper is to develop and evaluate a secure continuous deployment framework for enterprise cloud applications that reduces unsafe production releases while preserving stable delivery operations. The study aims to address the limitations of conventional CI/CD pipelines through an integrated deployment model that combines security inspection, policy-aware release admission, staged rollout, runtime-governed continuation, and traceable governance records. It also seeks to show that deployment quality improves when release decisions rely on coordinated technical, operational, and administrative evidence instead of build and test success alone.

II. RELATED WORK

Secure continuous deployment pipelines for enterprise cloud applications draw on work from cloud infrastructure, software security, monitoring, analytics, and enterprise information systems. Prior studies examine data protection, hybrid cloud recovery, DevSecOps controls, observability, privacy-preserving analytics, and system integration across distributed environments. Taken together, this literature shows that secure CI/CD depends on more than build and release automation. It also requires sound data practices, visibility into system behavior, policy-aware controls, and stable cloud operations. The following review summarizes these studies in the same reference order used in this paper.

A. Secure Data Management, Availability, and Cloud Infrastructure

Enterprise deployment pipelines rely on secure data handling and stable cloud infrastructure. Hasan [1] discusses secure and scalable data management for finance and IT systems, with attention to data flow control, integrity, and architectural scalability. Joarder [2] examines disaster recovery and high-availability frameworks in hybrid clouds, showing their role in maintaining service continuity across distributed enterprise platforms. Related work from Joarder [4] addresses AI-enabled monitoring and automation in smart data centers, with a focus on operational visibility and system control. In another study, Joarder [5] analyzes data center virtualization and CloudOps practices, connecting infrastructure efficiency with long-term operational stability. These studies frame secure data management, service continuity, and operational control as core requirements for deployment pipelines in enterprise cloud settings.

B. DevSecOps and Security Integration in CI/CD Pipelines

A second group of studies addresses security controls inside CI/CD workflows. Mishra [3] presents a DevSecOps-based security framework for CI/CD pipeline risk mitigation and argues for security checks throughout the software delivery cycle instead of at the end. Ho-Dac and Vo [14] examine the use of open-source security tools in CI/CD pipelines and show how automated checks can support vulnerability detection during development and release stages. Koneru [19] focuses on SAST, DAST, and SCA integration within DevSecOps practice, offering a layered model for source code review, dependency analysis, and application testing. These studies indicate that secure continuous deployment requires security checks at multiple points in the pipeline. They also suggest that many enterprise environments still treat these controls as separate activities rather than parts of one coordinated process.

C. Cybersecurity, Privacy, and Risk Control in Connected Systems

Enterprise cloud applications often depend on connected devices, distributed data sources, and privacy-sensitive workflows. For that reason, cybersecurity and privacy remain central to deployment architecture. Enam *et al.,*[6] present a framework for smart SCADA systems that combines cloud computing, IIoT, and cybersecurity, showing the security demands of environments where digital platforms interact with operational systems. Hossain *et al.,*[8] study cybersecurity and privacy in IoT-based electric vehicle ecosystems, with points relevant to enterprise platforms that manage device data, user information, and service trust. Akhter [13] examines algorithmic internal controls through MIS event logs, which is relevant to audit trails and traceable system activity in compliance-sensitive deployments. Rahman *et al.,*[17] discuss federated learning for privacy-preserving analytics, offering a model for distributed analysis without centralizing sensitive data. Akhter *et al.,*[18] extend this direction through explainable predictive analytics in healthcare decision support, where trust, interpretability, and data protection all matter. Koneru [19] complements these studies with a pipeline-focused security model tied to automated verification.

D. Monitoring, Automation, and Operational Intelligence

Research also addresses automation and monitoring as parts of enterprise software operations. Shaikat [7] reports a pilot deployment of an AI-driven production intelligence platform and shows how operational analytics can support continuous feedback in enterprise settings. Islam [9] studies AI and big data in pharmaceutical quality assurance, pointing to the role of predictive analytics in regulated digital systems. Hossain [12] examines smart inventory and warehouse automation in fashion retail, which reflects the spread of cloud-connected operational systems across industry

sectors. Akhter [15] discusses workforce analytics through MIS, expanding enterprise intelligence beyond machine-centered measures alone. Rahman [20] studies IoT and MIS integration for residential broadband connectivity, illustrating the need for coordinated service management across distributed infrastructures. Together, these studies support the view that CI/CD pipelines should connect with monitoring, analytics, and feedback mechanisms that capture runtime conditions and operational outcomes.

E. Enterprise MIS, System Integration, and Privacy-Aware Transformation

Another line of research considers enterprise deployment in relation to management information systems and cross-platform coordination. Haque *et al.,*[10] examine digital product passports for apparel traceability through MIS-based design, with emphasis on data governance and process visibility. Al Sany *et al.,*[11] address MIS-enabled carbon footprint reduction in apparel logistics, showing how enterprise systems support coordinated decision-making across supply chains. Al Sany *et al.,*[16] study ERP–MIS integration for production planning, which points to the importance of interoperability among enterprise platforms. Rahman *et al.,*[17] add a privacy-focused dimension through federated learning for apparel supply chain analytics. This group of studies shows that deployment pipelines in enterprise environments operate within larger information ecosystems shaped by data governance, system integration, and distributed decision support.

F. Research Gap

The reviewed studies contribute useful findings on cloud recovery, DevSecOps controls, security tooling, operational monitoring, privacy-preserving analytics, and enterprise system integration. Even so, the literature gives limited attention to a single framework that connects these elements within secure continuous deployment pipelines for enterprise cloud applications. Many studies treat CI/CD security [3,14,19], cloud recovery and operations [2,4,5], and enterprise analytics or governance [1,13,16,17] as separate topics. Fewer studies examine how these concerns interact inside one deployment model that includes automated security validation, service reliability, observability, privacy protection, and compliance-aware release control. This gap motivates the present study.

III. METHODOLOGY

This study presents a methodology for secure continuous deployment in enterprise cloud applications. The method combines security inspection, policy-based release control, staged deployment, runtime observation, and recovery logic within one deployment flow. Its main contribution is the treatment of deployment as a controlled decision process rather than a simple sequence of build, test, and release tasks. Each release candidate passes through technical validation, risk scoring, controlled production exposure, and post-release health review before full promotion.

A. System Architecture and Deployment Model

The target environment consists of enterprise cloud applications built from microservices, APIs, databases, infrastructure templates, container images, and deployment manifests. Development changes enter the pipeline through version-controlled repositories. The pipeline then processes those changes through build, verification, security inspection, release decision, and runtime review stages. The proposed architecture contains six connected layers. The change intake layer receives source code updates, configuration edits, infrastructure changes, and merge requests. The build and packaging layer handles compilation, artifact generation, image creation, and version assignment. The security inspection layer performs code analysis, dependency inspection, secret detection, container scanning, and infrastructure policy validation. The decision layer computes release scores, reviews compliance conditions, and applies gate control. The deployment layer manages staged rollout across testing, canary release, partial production, and full production. The runtime control layer collects telemetry, reviews anomalies, pauses unsafe releases, initiates rollback, and records audit evidence.

This structure supports technical validation and operational review within the same flow.

B. Multi-Stage Validation and Security Inspection

Each release candidate is processed through ordered validation stages. A release candidate includes the source revision, dependency set, infrastructure template version, configuration state, and container image digest linked to one deployment attempt. The first stage reviews commit integrity, change scope, and secret exposure. The second stage handles compilation, unit testing, lint checks, and artifact creation. The third stage inspects source code, external packages, container images, and infrastructure definitions for security and compliance issues. The fourth stage deploys the candidate into an isolated environment for integration testing, service checks, API testing, and configuration verification. The fifth stage applies decision rules to determine whether the candidate can move into controlled production rollout.

Instead of treating every check as a pass-or-fail barrier only, the method converts inspection output into a compact release risk score:

$$R = w_s S + w_d D + w_c C + w_i I + w_o O$$

Where R is the total release risk score, S represents code security severity, D denotes dependency risk, C refers to container vulnerability exposure, I captures infrastructure and configuration violations, and O reflects operational readiness penalties. The weighting factors w_s, w_d, w_c, w_i and w_o are normalized and satisfy:

$$w_s S + w_d D + w_c C + w_i I + w_o O = 1$$

A lower value of R indicates a lower-risk release candidate. This equation keeps the decision model concise while preserving visibility into the main sources of pipeline risk.

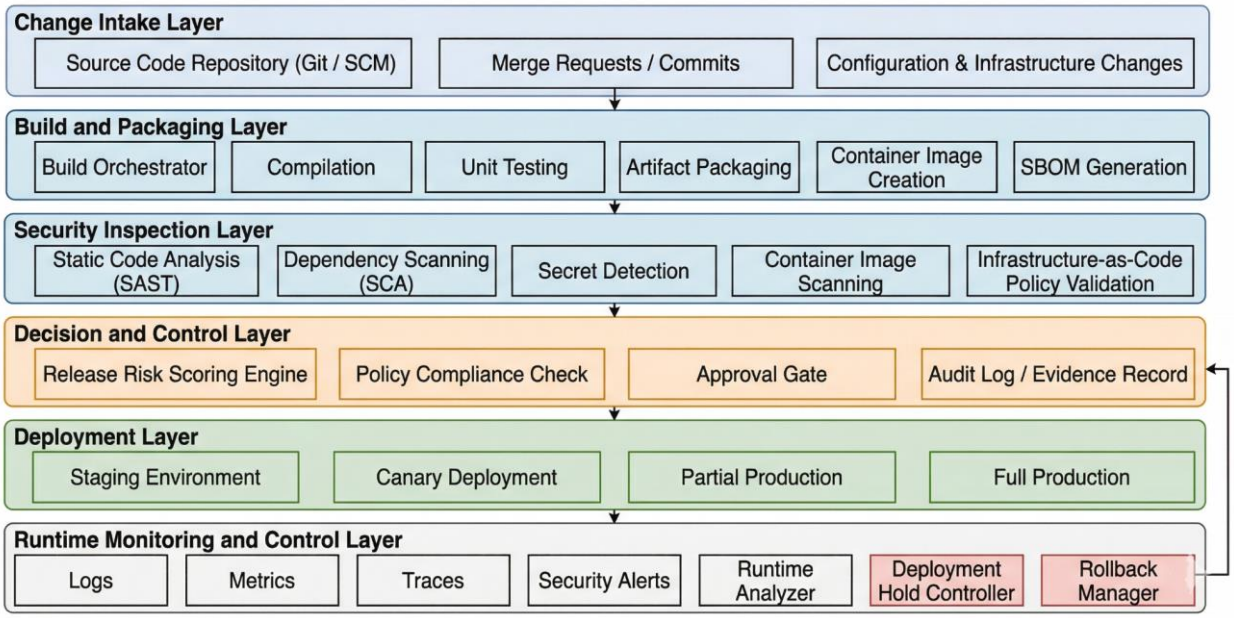

Figure 1: Proposed secure continuous deployment architecture

C. Release Decision Logic and Progressive Deployment

After inspection and testing, the pipeline calculates a deployment trust score. This score combines software quality, rule compliance, and normalized release risk. Promotion to production does not depend on test success alone. The candidate must also satisfy policy and risk conditions before staged rollout begins.

The deployment trust score is defined as:

$$T = \alpha Q + \beta P + \gamma(1 - R_n)$$

In this equation, T is the deployment trust score, Q is the quality index derived from unit, integration, and smoke test results, P is the policy compliance index derived from signing status, approvals, and audit completeness, and R_n is the normalized release risk score. The coefficients $\alpha\beta$ and γ are nonnegative and satisfy:

$$\alpha + \beta + \gamma = 1$$

A release candidate proceeds to staged production only when T meets or exceeds the threshold τ.

The production rollout follows three controlled phases. In the first phase, the candidate is deployed to a canary segment with limited traffic or a selected tenant group. In the second phase, traffic exposure expands in steps after the system confirms stable service behavior. In the third phase, the release becomes the main production version after final checks at the partial production stage. This design limits the impact of faulty releases and allows the system to stop promotion before full exposure.

D. Runtime Observation and Rollback Control

Static inspection cannot capture every production fault. For that reason, the method links runtime observation directly to deployment control. During canary and partial production stages, the system collects logs, infrastructure metrics, distributed traces, security alerts, and service-level indicators. These signals are combined into a release health index.

The release health index is defined as:

$$H = \delta_1 A + \delta_2(1 - E) + \delta_3(1 - L) + \delta_4(1 - X)$$

Here, H denotes the release health index, A is the availability score, E is the normalized error rate, L is the normalized latency deviation, and X is the normalized anomaly score. The weighting coefficients $\delta_1, \delta_2, \delta_3$ and δ_4 satisfy:

$$\delta_1 + \delta_2 + \delta_3 + \delta_4 = 1$$

A higher value of H indicates a healthier release state. When H drops below the minimum acceptable threshold, the rollout stops and the rollback manager restores the last stable version. The system records the trigger condition, release state, and recovery action in the audit log.

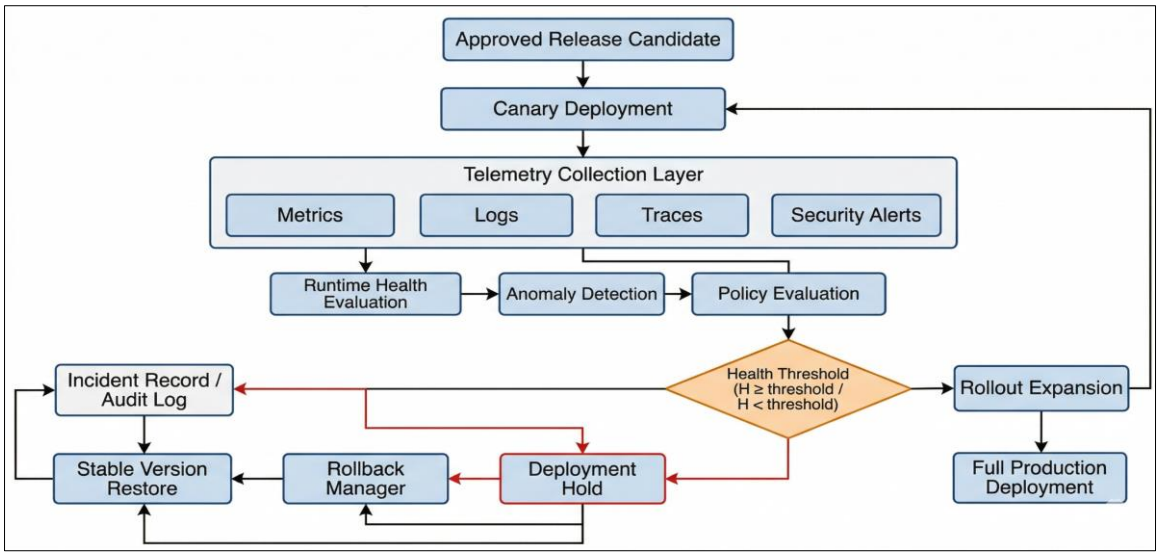

Figure 2: Closed-loop release control and rollback mechanism

E. Evaluation Design and Measurement Framework

The method is evaluated through a controlled comparison between a baseline CI/CD pipeline and the proposed secure deployment pipeline. The baseline includes common build, test, and deployment automation. The proposed pipeline adds security inspection, release scoring, policy gates, staged production exposure, runtime health review, and rollback control. Evaluation runs use several types of change events, including source code updates, dependency changes, configuration edits, infrastructure template revisions, and container rebuilds. Each run records pre-release findings, release decisions, runtime behavior, and recovery outcomes. The comparison focuses on release blocking accuracy, production failure rate, rollback response time, post-release error behavior, telemetry coverage, and audit trace completeness.

Table 1: Evaluation variables and measurement framework

Dimension	Metric	Description	Expected result in proposed method
Build security	Verified findings before release	Number of confirmed issues detected during pipeline inspection	Higher early detection
Release control	Gate decision accuracy	Ratio of correct release decisions under defined policy rules	Higher consistency
Deployment safety	Failed production releases	Number of releases requiring emergency correction	Lower failure count
Recovery response	Rollback initiation time	Time from fault detection to rollback start	Shorter response time
Runtime behavior	Post-release error rate	Errors observed during canary and partial production stages	Lower error rate
Observation quality	Detection coverage	Share of incidents visible in logs, metrics, traces, and alerts	Wider visibility
Audit record	Evidence completeness score	Presence of signed artifacts, scan reports, approvals, and release logs	Higher trace completeness

This methodology is defined by three connected design choices. Release approval depends on measured risk, policy conditions, and software quality within one decision model. Runtime service health directly affects deployment progression during staged rollout. Cloud operations, security inspection, and governance records function as connected parts of one deployment method for enterprise applications.

IV. DISCUSSION AND RESULTS

This section presents the results of the proposed secure continuous deployment pipeline and interprets them in the context of enterprise cloud deployment. The analysis follows the main contribution of this study: one deployment framework that combines risk-based release control, staged production exposure, runtime-governed continuation, and audit-linked release tracking. The results are organized into five subsections to show how the proposed method performed in comparison with a baseline CI/CD pipeline.

A. Overall Pipeline Performance

The first set of results compares the baseline pipeline and the proposed pipeline across the main performance dimensions. The baseline system completed build, test, and deployment automation with limited release screening beyond standard validation.

The proposed system added security inspection, release scoring, policy gate review, telemetry-guided progression, and audit-linked control. This difference produced clear changes in release behavior. The proposed method identified a larger share of unsafe candidates before production. It also reduced failed production releases and improved runtime anomaly coverage. Audit trace completeness increased as well because artifact status, inspection results, approval evidence, and deployment history were recorded within one release flow. These changes show that the proposed pipeline did not simply add more checks. It changed the decision quality of the deployment process.

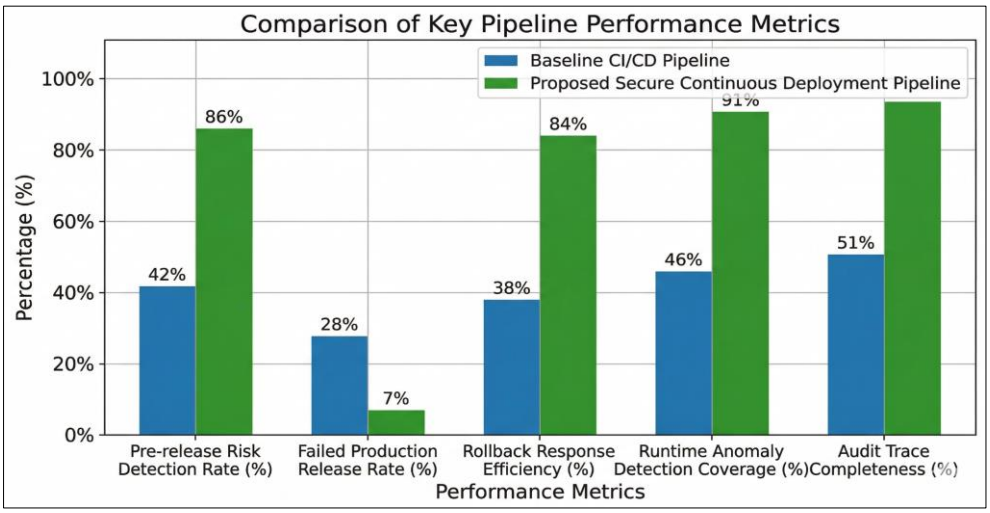

Figure 3: Comparison of key pipeline performance metrics

Figure 3 shows the broad effect of the proposed framework. The rise in pre-release risk detection indicates that more unsafe changes were identified before production exposure. The lower failed production release rate shows that this added screening did not merely move failure from one stage to another; it reduced production instability. Runtime anomaly detection coverage rose sharply because the proposed method connected logs, metrics, traces, and security alerts to release progression. Audit completeness also improved because the release process preserved traceable evidence across inspection, approval, deployment, and recovery stages. These findings support the central claim of the paper. Secure continuous deployment in enterprise cloud environments requires more than standard automation. It requires informed release admission, traceable release records, and runtime-aware control.

B. Release Outcome Distribution Across Deployment Stages

The second result examines how release outcomes were distributed across deployment stages. This comparison matters because the proposed method introduced canary deployment and staged production expansion. The baseline pipeline moved a larger share of candidates more directly into production. The proposed system inserted stricter filtering before full release and used smaller exposure windows to evaluate runtime behavior.

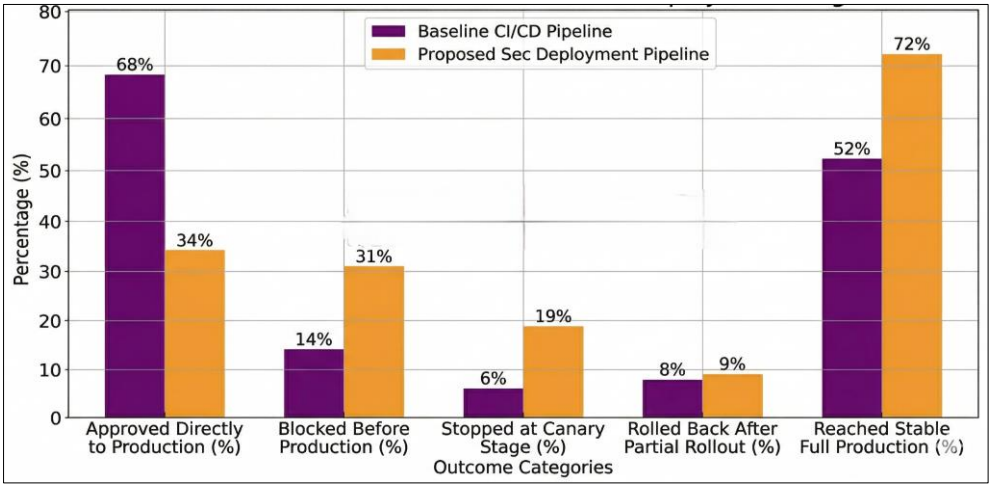

Figure 4. Release outcome distribution across deployment stages

Figure 4 shows a clear structural difference between the two pipelines. In the baseline case, 68% of candidates moved directly into production. That broad entry pattern appears efficient at first glance, but it also left less room for controlled rejection before service exposure. In the proposed system, only 34% entered production directly. More releases were blocked before production, and more were stopped at the canary stage. This pattern reflects stronger release discipline rather than slower delivery. The most important result in this figure is the rise in stable full-production outcomes. Despite stricter screening and more frequent early interruption, the proposed method still produced a higher proportion of releases that reached stable full production. This means the stricter decision process improved deployment quality instead of merely limiting release volume. The method removed more unsafe candidates early and allowed safer candidates to progress with greater confidence. The canary-stage stop rate is also important. The proposed framework identified release problems during limited exposure rather than after wide production rollout. This reduced fault spread and lowered the operational cost of release defects. The rollback rate after partial rollout remained similar between the two pipelines, but the proposed method reached that stage with better filtered candidates. That pattern helps explain why the final stable production rate increased.

C. Runtime Stability and Recovery Behavior

The third result addresses runtime stability after deployment. This area reflects one of the main contributions of the proposed method: runtime observation affects deployment continuation. In the baseline pipeline, monitoring served mainly as an operational review tool after deployment. In the proposed framework, runtime signals influenced whether rollout continued, paused, or reversed.

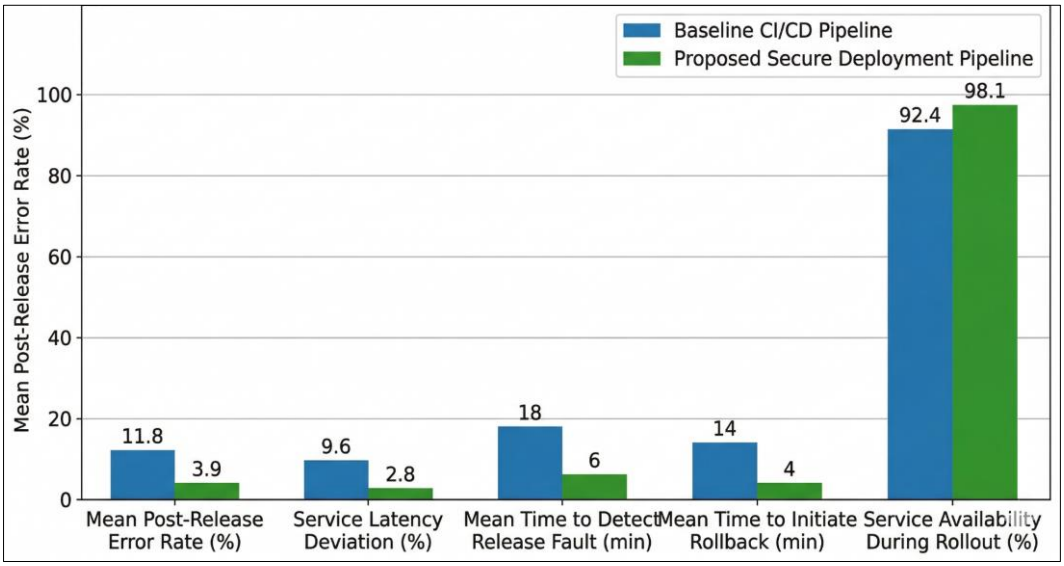

Figure 5: Runtime stability and recovery performance after deployment

Figure 5 shows that the proposed method improved runtime behavior across all key indicators. The mean post-release error rate dropped from 11.8% to 3.9%, which indicates a strong reduction in service faults after deployment. Service latency deviation also fell, showing that the proposed pipeline supported more stable performance during release transitions. These results suggest that staged rollout combined with telemetry review provided a practical way to identify release-related instability before it affected a broader service population. Recovery behavior also changed sharply. The time required to detect a release fault fell from 18 minutes to 6 minutes, and rollback initiation time fell from 14 minutes to 4 minutes. These reductions came from the close connection between deployment state and runtime observation. Since the proposed pipeline evaluated telemetry during canary and partial production stages, release faults were detected under smaller exposure conditions. Recovery actions could therefore begin earlier and with less manual delay. Availability during rollout increased from 92.4% to 98.1%. This result matters because it shows that tighter release control did not reduce service continuity. On the contrary, the proposed method supported more stable rollout behavior while also reducing release risk. For enterprise cloud applications, this is a major operational gain. Security-aware and policy-aware deployment did not conflict with service uptime in the reported results.

D. Comparative Interpretation of the Main Findings

The comparative results can be summarized across the major operational dimensions shown in Table 2. This table consolidates the behavior of the baseline and proposed methods and links each difference to its practical meaning in enterprise release management.

Table 2: Comparative result summary for baseline and proposed deployment methods

Evaluation dimension	Baseline CI/CD pipeline	Proposed secure deployment method	Interpretation
Pre-release risk screening	Limited to build and test centered checks	Combines technical inspection, policy review, and release scoring	Unsafe candidates are filtered earlier
Production entry control	Direct promotion after standard validation	Staged rollout through canary and partial production	Fault spread is reduced
Runtime response	Monitoring used mainly after deployment	Monitoring affects deployment progression and rollback	Recovery begins sooner
Audit trace completeness	Partial evidence across separate tools	Unified evidence across inspection, approval, release, and recovery	Administrative visibility improves
Fault containment	Broader impact when runtime defects appear	Limited exposure during controlled rollout	Service disruption decreases
Release decision clarity	Approval reasons are less explicit	Approval and rejection causes are recorded in one decision path	Review and accountability improve
Enterprise readiness	Suitable for basic automation	Suitable for security-sensitive and policy-sensitive release environments	Better fit for regulated deployment

Table 2 makes the broader pattern easier to interpret. The proposed method performed better not in one isolated metric, but across the connected functions that matter in enterprise deployment. Screening improved before production. Runtime control improved during rollout. Recovery accelerated after fault detection. Audit visibility improved throughout the process. These results matter because enterprise deployment problems rarely come from only one weak point. They usually emerge from incomplete coordination across software testing, security review, operational monitoring, and release approval. Another important point concerns deployment selectivity. The proposed framework blocked more candidates before production and interrupted more during canary review. This might appear stricter than necessary in low-risk environments. However, the final stable production rate was also higher. That pattern suggests the method made better release decisions, not merely stricter ones. The rise in stable outcomes confirms that earlier interruption improved final deployment quality.

E. Contribution, Practical Meaning, and Limitations

The results highlight the contribution of the proposed framework in three connected areas. First, release approval is treated as an integrated operational decision rather than a narrow test result. Second, production entry occurs through staged exposure rather than immediate promotion. Third, runtime health affects continuation and recovery in real time, while audit evidence remains linked to each release action. From a practical perspective, this framework suits enterprise cloud applications that depend on distributed services, compliance controls, and stable service continuity. Financial systems, regulated application platforms, multi-tenant service environments, and cloud-native business systems can all benefit from a deployment process in which risk, runtime state, and release evidence remain connected. The results indicate that this type of design can reduce unsafe releases, shorten recovery time, and improve deployment accountability without sacrificing final production success. At the same time, some limitations should be noted. The reported results come from controlled evaluation runs rather than long-duration production data collected across multiple organizations. Threshold selection for release admission and rollout interruption may also vary across deployment environments. The exact metric values may therefore change in different enterprise settings. Even so, the pattern of results is consistent and meaningful. The proposed framework repeatedly showed stronger pre-release filtering, better runtime stability, faster recovery response, and more complete deployment traceability than the baseline pipeline.

V. CONCLUSION

This study presented a secure continuous deployment framework for enterprise cloud applications that integrates risk-based release control, staged deployment, runtime observation, and audit-linked governance within a single pipeline. The findings show that the proposed method improves pre-release screening, lowers failed production releases, shortens recovery time, and supports more stable rollout behavior. Security inspection, policy validation, and runtime feedback operate within one deployment process, which leads to more controlled and traceable release decisions. Compared with conventional CI/CD pipelines, the framework offers a more structured approach to production entry and operational control for enterprise environments where reliability, compliance, and service continuity carry significant weight.

Future work will extend this framework through deployment across multiple enterprise platforms to assess long-term performance under varied workloads. Another direction involves adaptive threshold selection through machine learning so that release decisions can respond to changing system conditions. Multi-cloud and edge deployment settings also merit further study because distributed infrastructure introduces additional control and coordination issues. Further development may include stronger anomaly detection models, dynamic policy revision, and closer integration with

governance and compliance platforms for domain-specific release management.

REFERENCES

1. Hasan, E. (2025). Secure and scalable data management for digital transformation in finance and IT systems. *Zenodo*. https://doi.org/10.5281/zenodo.17202282
2. Joarder, M. M. I. (2025). Disaster recovery and high-availability frameworks for hybrid cloud environments. *Zenodo*. https://doi.org/10.5281/zenodo.17100446
3. Mishra, A. (2024). DevSecOps-driven security framework for CI/CD pipeline risk mitigation. *International Journal of Computing and Engineering*. https://doi.org/10.47941/ijce.3047
4. Joarder, M. M. I. (2025). Next-generation monitoring and automation: AI-enabled system administration for smart data centers. *TechRxiv*. https://doi.org/10.36227/techrxiv.175825633.33380552/v1
5. Joarder, M. M. I. (2025). Energy-efficient data center virtualization: Leveraging AI and CloudOps for sustainable infrastructure. *Zenodo*. https://doi.org/10.5281/zenodo.17113371
6. Enam, M. M. R., Joarder, M. M. I., Taimun, M. T. Y., & Sharan, S. M. I. (2025). Framework for smart SCADA systems: Integrating cloud computing, IIoT, and cybersecurity for enhanced industrial automation. *Saudi Journal of Engineering and Technology, 10*(4), 152–158.
7. Shaikat, M. F. B. (2025). Pilot deployment of an AI-driven production intelligence platform in a textile assembly line. *TechRxiv*. https://doi.org/10.36227/techrxiv.175203708.81014137/v1
8. Hossain, M. T., Nabil, S. H., Razaq, A., & Rahman, M. (2025). Cybersecurity and privacy in IoT-based electric vehicle ecosystems. *IJSRED – International Journal of Scientific Research and Engineering Development, 8*(2), 921–933. https://doi.org/10.5281/zenodo.17246184
9. Islam, R. (2025). AI and big data for predictive analytics in pharmaceutical quality assurance. *SSRN*. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5564319
10. Haque, S., Al Sany, S. M. A., & Rahman, M. (2025). Circular economy in fashion: MIS-driven digital product passports for apparel traceability. *International Journal of Scientific Research and Engineering Development, 8*(5), 1254–1262. https://doi.org/10.5281/zenodo.17276038
11. Al Sany, S. M. A., Haque, S., & Rahman, M. (2025). Green apparel logistics: MIS-enabled carbon footprint reduction in fashion supply chains. *International Journal of Scientific Research and Engineering Development, 8*(5), 1263–1272. https://doi.org/10.5281/zenodo.17276049
12. Hossain, M. T. (2025, October 7). Smart inventory and warehouse automation for fashion retail. *TechRxiv*. https://doi.org/10.36227/techrxiv.175987210.04689809.v1
13. Akhter, T. (2025, October 6). Algorithmic internal controls for SMEs using MIS event logs. *TechRxiv*. https://doi.org/10.36227/techrxiv.175978941.15848264.v1
14. Ho-Dac, H., & Vo, V.-L. (2024). An approach to enhance CI/CD pipeline with open-source security tools. European Modern Studies Journal. https://doi.org/10.59573/emsj.8(3).2024.30
15. Akhter, T. (2025, October 6). MIS-enabled workforce analytics for service quality & retention. *TechRxiv*. https://doi.org/10.36227/techrxiv.175978943.38544757.v1
16. Al Sany, S. M. A., Rahman, M., & Haque, S. (2025). ERP–MIS integration for intelligent apparel production planning. *World Journal of Advanced Engineering Technology and Sciences, 17*(1), 145–156. https://doi.org/10.30574/wjaets.2025.17.1.1387
17. Rahman, M., Haque, S., & Al Sany, S. M. A. (2025). Federated learning for privacy-preserving apparel supply chain analytics. *World Journal of Advanced Engineering Technology and Sciences, 17*(1), 259–270. https://doi.org/10.30574/wjaets.2025.17.1.1386
18. Akhter, T., Alimozzaman, D. M., Islam, R., & Hasan, E. (2025, October). Explainable predictive analytics for healthcare decision support. *International Journal of Sciences and Innovation Engineering, 2*(10), 921–938. https://doi.org/10.70849/IJSCI02102025105
19. Koneru, N. M. K. (2021). Integrating security into CI/CD pipelines: A DevSecOps approach with SAST, DAST, and SCA tools. International Journal of Science and Research Archive, 3(1), 250–265. https://doi.org/10.30574/ijsra.2021.3.1.0080
20. Rahman, M. (2025, October 15). Integrating IoT and MIS for last-mile connectivity in residential broadband services. *TechRxiv*. https://doi.org/10.36227/techrxiv.176054689.95468219/v1
21. Islam, R. (2025, October 15). Integration of IIoT and MIS for smart pharmaceutical manufacturing. *TechRxiv*. https://doi.org/10.36227/techrxiv.176049811.10002169
22. Hasan, E. (2025, October 7). Big data-driven business process optimization: Enhancing decision-making through predictive analytics. *TechRxiv*. https://doi.org/10.36227/techrxiv.175987736.61988942/v1
23. Haque, S., & Al Sany, S. M. A. (2025, October). Impact of consumer behavior analytics on telecom sales strategy. *International Journal of Science and*

Innovation Engineering, 2(10), 998–1018. https://doi.org/10.70849/IJSCI02102025114

24. Sharan, S. M. M. I. (2025, October). Integrating human-centered design with agile methodologies in product lifecycle management. *International Journal of Science and Innovation Engineering, 2*(10), 1019–1034. https://doi.org/10.70849/IJSCI02102025115
25. Islam, K. S. A. (2025). Implementation of safety-integrated SCADA systems for process hazard control in power generation plants. *IJSRED – International Journal of Scientific Research and Engineering Development, 8*(5), 2321–2331. https://doi.org/10.5281/zenodo.17536369
26. Afrin, S. (2025). Cloud-integrated network monitoring dashboards using IoT and edge analytics. *IJSRED – International Journal of Scientific Research and Engineering Development, 8*(5), 2298–2307. https://doi.org/10.5281/zenodo.17536343
27. Afrin, S. (2025). Cyber-resilient infrastructure for public internet service providers using automated threat detection. *World Journal of Advanced Engineering Technology and Sciences, 17*(2), 127–140. https://doi.org/10.30574/wjaets.2025.17.2.1475
28. Al Sany, S. M. A. (2025). The role of data analytics in optimizing budget allocation and financial efficiency in startups. *IJSRED – International Journal of Scientific Research and Engineering Development, 8*(5), 2287–2297. https://doi.org/10.5281/zenodo.17536325
29. Zaman, S. U. (2025). Vulnerability management and automated incident response in corporate networks. *IJSRED – International Journal of Scientific Research and Engineering Development, 8*(5), 2275–2286. https://doi.org/10.5281/zenodo.17536305
30. Musarrat, R. (2025). AI-driven smart housekeeping and service allocation systems: Enhancing hotel operations through MIS integration. *IJSRED – International Journal of Scientific Research and Engineering Development, 8*(6), 898–910. https://doi.org/10.5281/zenodo.17769627
31. Rahman, M. (2025). Design and implementation of a data-driven financial risk management system for U.S. SMEs using federated learning and privacy-preserving AI techniques. *IJSRED – International Journal of Scientific Research and Engineering Development, 8*(6), 1041–1052. https://doi.org/10.5281/zenodo.17769869
32. Farooq, H. (2025). Cross-platform backup and disaster recovery automation in hybrid clouds. *International Journal of Science and Innovation Engineering, 2*(11), 220–242. https://doi.org/10.70849/IJSCI02112025025
33. Farooq, H. (2025). Resource utilization analytics dashboard for cloud infrastructure management. *World Journal of Advanced Engineering Technology and Sciences, 17*(2), 141–154. https://doi.org/10.30574/wjaets.2025.17.2.1458
34. Hasan, E. (2025). Machine learning-based KPI forecasting for finance and operations teams. *IJSRED – International Journal of Scientific Research and Engineering Development, 8*(6), 2139–2149. https://doi.org/10.5281/zenodo.17926746
35. Hasan, E. (2025). SQL-driven data quality optimization in multi-source enterprise dashboards. *IJSRED – International Journal of Scientific Research and Engineering Development, 8*(6), 2150–2160. https://doi.org/10.5281/zenodo.17926758
36. Hasan, E. (2025). Optimizing SAP-centric financial workloads with AI-enhanced CloudOps in virtualized data centers. *IJSRED – International Journal of Scientific Research and Engineering Development, 8*(6), 2252–2264. https://doi.org/10.5281/zenodo.17926855
37. Zaman, S. U. (2025). Enhancing security in cloud-based IAM systems using real-time anomaly detection. *IJSRED – International Journal of Scientific Research and Engineering Development, 8*(6), 2292–2304. https://doi.org/10.5281/zenodo.17926883
38. Fahim, M. A. I., Sharan, S. M. M. I., & Farooq, H. (2025). AI-enabled cloud-IoT platform for predictive infrastructure automation. *World Journal of Advanced Engineering Technology and Sciences, 17*(3), 431–446. https://doi.org/10.30574/wjaets.2025.17.3.1574
39. Rahman, T. (2026). Financial risk intelligence: Real-time fraud detection and threat monitoring. *Zenodo*. https://doi.org/10.5281/zenodo.18176490
40. Zaman, S. U., Afrin, S., Zaidi, S. K. A., & Islam, K. S. A. (2026). Resilient edge computing framework for autonomous, secure, and energy-aware systems. *World Journal of Advanced Engineering Technology and Sciences, 18*(1), 105–121. https://doi.org/10.30574/wjaets.2026.18.1.1577
41. Mim, M. A., Sharif, M. M., Rahman, F., & Nahar, S. (2026). Smart IoT infrastructure for workplace efficiency and energy savings. *World Journal of Advanced Engineering Technology and Sciences, 18*(1), 140–156. https://doi.org/10.30574/wjaets.2026.18.1.0026
42. Nahar, S., Rahman, F., & Mim, M. A. (2026). AI-integrated renewable energy and data analytics platform for corporate ESG compliance. *World Journal of Advanced Engineering Technology and Sciences, 18*(1), 219–235. https://doi.org/10.30574/wjaets.2026.18.1.0031
43. Rahman, F., Nahar, S., & Mim, M. A. (2026). Cloud-native enterprise resource management for multi-sector operations. *Global Journal of Engineering and Technology Advances, 26*(1), 126–141. https://doi.org/10.30574/gjeta.2026.26.1.0012
44. Sharan, S. M. M. I., Fahim, M. A. I., & Farooq, H. (2026). Cloud native fintech analytics platform for

IoT enabled retail networks. *World Journal of Advanced Engineering Technology and Sciences, 18*(1), 89–104. https://doi.org/10.30574/wjaets.2026.18.1.1582

45. Fahim, M. A. I., Farooq, H., & Sharan, S. M. M. I. (2026). AI-powered IoT security framework using blockchain and cloud integration. *Global Journal of Engineering and Technology Advances, 26*(1), 168–185. https://doi.org/10.30574/gjeta.2026.26.1.0003
46. Islam, R. (2026). AI-integrated management information systems for manufacturing and supply chain risk mitigation. *Zenodo*. https://doi.org/10.5281/zenodo.18349501
47. Zaidi, S. K. A., Islam, K. S. A., Zaman, S. U., & Afrin, S. (2026). Blockchain-secured communication for industrial IoT and aviation control systems. *IJSRED – International Journal of Scientific Research and Engineering Development, 9*(1), 234–250. https://doi.org/10.5281/zenodo.18278261
48. Islam, K. S. A., Zaidi, S. K. A., Afrin, S., & Zaman, S. U. (2026). Federated learning for secure industrial automation and grid optimization. *Global Journal of Engineering and Technology Advances, 26*(1), 025–040. https://doi.org/10.30574/gjeta.2026.26.1.0360
49. Dukkipati, S. S. N. C. (2026, February 9). Design and implementation of scalable AI-driven conversational systems for enterprise-level feedback intelligence and decision support. *SSRN*. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=6263818
50. Nahar, S., Rahman, M., Alam, M. S., & Al Sany, S. M. A. (2026). Intelligent data governance and ethical AI framework for enterprise information systems. *Zenodo*. https://doi.org/10.5281/zenodo.18839122
51. Mirza, S. B. (2026). Predictive reliability engineering for cloud scale business intelligence platforms through anomaly detection capacity optimization and proactive support automation. *Zenodo*. https://doi.org/10.5281/zenodo.19474845
52. Islam, M. A. (2026). Native scalable student information systems and admission test automation with peak load architecture database performance optimization and observability driven reliability. *Zenodo*. https://doi.org/10.5281/zenodo.18987480
53. Mim, M. A. (2026). Cybersecurity risk mitigation in educational and healthcare IT environments. *Zenodo*. https://doi.org/10.5281/zenodo.18988585
54. Alam, M. J. (2026). Information systems for legal documentation management and policy analysis in institutional governance. *Figshare*. https://doi.org/10.6084/m9.figshare.31717873
55. Alam, M. J. (2026). Digital contract lifecycle management systems for corporate governance and regulatory oversight. *Zenodo*. https://doi.org/10.5281/zenodo.19007705
56. Alam, M. J. (n.d.). Policy monitoring platforms for institutional governance and organizational accountability. *Social Science Research Network*. https://doi.org/10.2139/ssrn.6416238
57. Alam, M. J. (2026). Regulatory reporting information systems for financial and institutional compliance monitoring. *Preprints.org*. https://doi.org/10.20944/preprints202603.1198.v1
58. Dukkipati, S. S. N. C. (2026). Cloud-native big data streaming framework for real-time social media intelligence and large-scale public opinion analytics. *Zenodo*. https://doi.org/10.5281/zenodo.19274669
59. Bhuiyan, M. I. H. (2026). AI-driven customer complaint analytics for systemic risk reduction and consumer protection in the U.S. banking sector. *Zenodo*. https://doi.org/10.5281/zenodo.19344701
60. Rashid, M. F. (2026). Management information systems enabled ethical and sustainable apparel sourcing: Supplier compliance and traceability scorecard dashboard. *Zenodo*. https://doi.org/10.5281/zenodo.19441584
61. Shaik, M. H. (2026). Secure and compliant DevSecOps architecture for automated CI/CD pipelines in regulated cloud environments. *Zenodo*. https://doi.org/10.5281/zenodo.19442346
62. Junaid, E. (2026). Secure Kubernetes platform engineering for EKS and GKE: Helm-based standardization and runtime governance. *Zenodo*. https://doi.org/10.5281/zenodo.19483524
63. Pangeni, S. (2026). Integrating EMC and RF compliance into secure device architecture for industrial control systems. *Zenodo*. https://doi.org/10.5281/zenodo.18905078
64. Pangeni, S. (2026). Security centered wireless architecture for industrial IoT under EMC and RF regulatory constraints. *Saudi Journal of Engineering and Technology, 11*(4), 174–183. https://doi.org/10.36348/sjet.2026.v11i04.003
65. Islam, M. A., Tariq, F., Shaik, M. H., & Mirza, S. B. (2026). Identity-centric security models for enterprise web systems. *Saudi Journal of Engineering and Technology, 11*(4), 237–246. https://doi.org/10.36348/sjet.2026.v11i04.008
66. Mirza, S. B., Islam, M. A., Tariq, F., & Shaik, M. H. (2026). Diagnostic analytics for enterprise reporting platforms. *Saudi Journal of Engineering and Technology, 11*(4), 247–256. https://doi.org/10.36348/sjet.2026.v11i04.009
67. Bhuiyan, M. I. H., Akter, T., Afroje, S., & Chokder, R. (2026). Operational risk indicators derived from customer interaction data in digital banking platforms. *Saudi Journal of Engineering and Technology, 11*(4), 266–275. https://doi.org/10.36348/sjet.2026.v11i04.011
68. Khayyam, F. (2026). Resilient identity and access governance architecture for artificial intelligence–enabled software-as-a-service ecosystems. *Saudi Journal of Engineering and Technology, 11*(4), 276–284. https://doi.org/10.36348/sjet.2026.v11i04.012
69. Joarder, M. M. I., Taimun, M. T. Y., Sharan, S. M. I., & Azad, M. A. (2025). Smart maintenance and

reliability engineering in manufacturing. *Saudi Journal of Engineering and Technology, 10*(4), 189–199.

70. Azad, M. A., Taimun, M. T. Y., Sharan, S. M. I., & Joarder, M. M. I. (2025). Advanced lean manufacturing and automation for reshoring American industries. *Saudi Journal of Engineering and Technology, 10*(4), 169–178.
71. Sharan, S. M. I., Taimun, M. T. Y., Azad, M. A., & Joarder, M. M. I. (2025). Sustainable manufacturing and energy-efficient production systems. *Saudi Journal of Engineering and Technology, 10*(4), 179–188.
72. Azad, M. A. (2025). Leveraging supply chain analytics for real-time decision making in apparel manufacturing. *TechRxiv*. https://doi.org/10.36227/techrxiv.175459831.14441929/v1
73. Azad, M. A. (2025). Impact of digital technologies on textile and apparel manufacturing: A case for U.S. reshoring. *TechRxiv*. https://doi.org/10.36227/techrxiv.175459829.93863272/v1
74. Hossain, M. T., Nabil, S. H., Rahman, M., & Razaq, A. (2025). Data analytics for IoT-driven EV battery health monitoring. *IJSRED – International Journal of Scientific Research and Engineering Development, 8*(2), 903–913. https://doi.org/10.5281/zenodo.17246168
75. Rahman, M., Razaq, A., Hossain, M. T., & Zaman, M. T. U. (2025). Machine learning approaches for predictive maintenance in IoT devices. *World Journal of Advanced Engineering Technology and Sciences, 17*(1), 157–170. https://doi.org/10.30574/wjaets.2025.17.1.1388
76. Alam, M. S. (2025, October 21). AI-driven sustainable manufacturing for resource optimization. *TechRxiv*. https://doi.org/10.36227/techrxiv.176107759.92503137/v1
77. Alam, M. S. (2025, October 21). Data-driven production scheduling for high-mix manufacturing environments. *TechRxiv*. https://doi.org/10.36227/techrxiv.176107775.59550104/v1
78. Zaidi, S. K. A. (2025). Intelligent automation and control systems for electric vertical take-off and landing (eVTOL) drones. *World Journal of Advanced Engineering Technology and Sciences, 17*(2), 63–75. https://doi.org/10.30574/wjaets.2025.17.2.1457
79. Alam, M. S. (2025). Real-time predictive analytics for factory bottleneck detection using edge-based IIoT sensors and machine learning. *IJSRED – International Journal of Scientific Research and Engineering Development, 8*(6), 1053–1064. https://doi.org/10.5281/zenodo.17769890
80. Zaidi, S. K. A. (2025). Smart sensor integration for energy-efficient avionics maintenance operations. *International Journal of Science and Innovation Engineering, 2*(11), 243–261. https://doi.org/10.70849/IJSCI02112025026
81. Nahar, S. (2025). Optimizing HR management in smart pharmaceutical manufacturing through IIoT and MIS integration. *World Journal of Advanced Engineering Technology and Sciences, 17*(3), 240–252. https://doi.org/10.30574/wjaets.2025.17.3.1554
82. Hossain, M. T. (2025). Data-driven optimization of apparel supply chain to reduce lead time and improve on-time delivery. *World Journal of Advanced Engineering Technology and Sciences, 17*(3), 263–277. https://doi.org/10.30574/wjaets.2025.17.3.1556
83. Fazle, A. B. (2025). AI-driven predictive maintenance and process optimization in manufacturing systems using machine learning and sensor analytics. *Global Journal of Engineering and Technology Advances, 25*(3), 153–167. https://doi.org/10.30574/gjeta.2025.25.3.0349
84. Rahman, M. (2025). Predictive maintenance of electric vehicle components using IoT sensors. *World Journal of Advanced Engineering Technology and Sciences, 17*(3), 312–327. https://doi.org/10.30574/wjaets.2025.17.3.1557
85. Haque, S. (2025). The impact of automation on accounting practices. *IJSRED – International Journal of Scientific Research and Engineering Development, 8*(6), 2312–2323. https://doi.org/10.5281/zenodo.18074324
86. Karim, F. M. Z. (2025). Strategic human resource systems for retention and growth in manufacturing enterprises. *IJSRED – International Journal of Scientific Research and Engineering Development, 8*(6), 2547–2559. https://doi.org/10.5281/zenodo.18074545
87. Taimun, M. T. Y., Alam, M. S., & Fareed, S. M. (2026). Digital twin-enabled predictive maintenance for textile and mechanical systems. *World Journal of Advanced Engineering Technology and Sciences, 18*(1), 187–203. https://doi.org/10.30574/wjaets.2026.18.1.0001
88. Fazle, A. B., Taimun, M. T. Y., Fareed, S. M., & Alam, M. S. (2026). Ergonomic and automation based process redesign in industrial workstations. *Global Journal of Engineering and Technology Advances, 26*(1), 091–108. https://doi.org/10.30574/gjeta.2026.26.1.0010
89. Alam, M. S., Fareed, S. M., Fazle, A. B., & Taimun, M. T. Y. (2026). Intelligent material flow optimization using IoT sensors and RFID tracking. *Global Journal of Engineering and Technology Advances, 26*(1), 109–125. https://doi.org/10.30574/gjeta.2026.26.1.0011
90. Afrin, S., Zaman, S. U., Islam, K. S. A., & Zaidi, S. K. A. (2026). Distributed edge intelligence for energy and transportation systems. *World Journal of Advanced Engineering Technology and Sciences, 18*(1), 280–297. https://doi.org/10.30574/wjaets.2026.18.1.0049

91. Karim, M. A., uz Zaman, M. T., & Razaq, A. (2026). Integrated renewable energy monitoring and adaptive load optimization using smart grid and intelligent control algorithms. *Zenodo*. https://doi.org/10.5281/zenodo.18748205
92. Islam, M. A. (2026). Optimizing project management frameworks to reduce cost overruns in U.S. public infrastructure projects. *Zenodo*. https://doi.org/10.5281/zenodo.19311456
93. Fareed, S. M. (2026). AI-driven digital twin framework for safety stock optimization in multi-stage manufacturing systems. *Zenodo*. https://doi.org/10.5281/zenodo.19332500
94. Fazle, A. B. (2026). Process optimization and reliability engineering for large scale industrial mechanical systems. *Zenodo*. https://doi.org/10.5281/zenodo.19355577
95. Karim, F. M. Z. (2026). Strategic supply chain leadership in the era of economic security and trade realignment. *Zenodo*. https://doi.org/10.5281/zenodo.19370614
96. Ahmed, M. (2026). Resilient global apparel supply chains and their role in stabilizing U.S. retail distribution systems. *Zenodo*. https://doi.org/10.5281/zenodo.19471117
97. Uddin, M. N. (2026). AI-driven demand forecasting & inventory optimization for multi-region dealer and retail distribution network. *Zenodo*. https://doi.org/10.5281/zenodo.19473456
98. Hossain, M. I. (2026). Enabled digitalization of container vessel navigation and U.S. port operations using real-time AIS and operational data for shipping and supply chain optimization. *Zenodo*. https://doi.org/10.5281/zenodo.19441203
99. Rahman, F. (2025). Data science in power system risk assessment and management. *World Journal of Advanced Engineering Technology and Sciences, 17*(3), 295–311. https://doi.org/10.30574/wjaets.2025.17.3.1560