



Generative AI-Based Incident Management for Cloud Enterprise Systems Through Causal Inference and Automated Remediation

Dr. Femi Emmanuel Ayo

Department of Computer Science, Olabisi Onabanjo University, Ago-Iwoye, Nigeria

Publication History: Received: 02.08.2026; Revised: 24.09.2026; Accepted: 27.09.2026; Published: 08.10.2026.

ABSTRACT: Cloud enterprise systems increasingly depend on distributed applications, interconnected services, and dynamic infrastructure, making incident detection, diagnosis, and resolution complex operational challenges. Traditional incident management approaches frequently rely on predefined rules, manual investigation, and correlation-based analytics, which may struggle to identify underlying causes and recommend effective corrective actions. This research proposes a generative artificial intelligence-based incident management framework that integrates causal inference, intelligent diagnosis, and automated remediation for cloud enterprise environments. The framework collects infrastructure metrics, application logs, distributed traces, configuration changes, and service dependency information to construct a comprehensive representation of operational incidents. Causal inference techniques analyze relationships between system events and performance degradation, helping distinguish potential causes from coincidental correlations. Generative AI processes the resulting evidence to summarize incidents, explain diagnostic findings, recommend remediation procedures, and generate structured response plans. A policy-controlled execution layer validates proposed actions against operational constraints, security permissions, and service availability requirements before implementation. The methodology evaluates incident classification accuracy, root-cause identification quality, remediation success rate, response time, and operational overhead. Comparative experiments against conventional monitoring and rule-based incident management approaches are proposed to assess the framework's effectiveness. Human oversight, auditability, and rollback mechanisms are incorporated to minimize the risks associated with incorrect AI-generated recommendations. The proposed framework aims to improve incident response efficiency, strengthen service reliability, and support evidence-based operational decisions across cloud enterprise systems. Its practical benefits require validation through reproducible experiments and controlled incident scenarios.

KEYWORDS: Generative AI, Incident Management, Causal Inference, Automated Remediation, Cloud Enterprise Systems, Root Cause Analysis, AIOps

I. INTRODUCTION

Cloud enterprise systems have become fundamental to modern digital operations, supporting business applications, transactional processing, data integration, analytics, and distributed service delivery. Organizations increasingly deploy enterprise workloads across public clouds, private clouds, hybrid infrastructures, and containerized environments to achieve scalability, flexibility, and operational efficiency. However, the complexity of these architectures increases the difficulty of identifying, diagnosing, and resolving incidents. A single service interruption may involve multiple interacting factors, including resource exhaustion, network latency, configuration changes, database contention, authentication failures, and application defects. Conventional incident management systems commonly depend on predefined thresholds, monitoring alerts, correlation rules, and manual troubleshooting procedures. Although these approaches remain useful for detecting known failure patterns, they may generate excessive alerts, overlook relationships between dependent services, and require substantial engineering effort to identify underlying causes. Correlation-based monitoring can indicate that several events occurred together without establishing which event contributed to the failure. Consequently, incident responders may investigate symptoms instead of root causes, prolonging service disruptions and increasing operational costs. Artificial intelligence for IT operations, commonly known as AIOps, has introduced opportunities for automated event correlation, anomaly detection, predictive monitoring, and intelligent incident prioritization. Generative artificial intelligence extends these capabilities by interpreting technical evidence, synthesizing diagnostic information, and generating context-sensitive recommendations. Integrating generative AI with causal



inference and controlled automated remediation offers a potential approach to improving incident management in complex cloud enterprise environments.

Causal inference provides analytical methods for investigating relationships between system variables, distinguishing plausible causal mechanisms from associations, and evaluating the potential effects of interventions. When integrated with infrastructure telemetry, service dependency graphs, distributed traces, and configuration histories, causal analysis can help identify factors that contribute to operational failures. Generative AI can then transform these findings into readable incident summaries, diagnostic explanations, and structured remediation plans. For example, when an enterprise application experiences increased response times, a causal analysis component may examine database contention, connection-pool saturation, recent deployments, and upstream service delays to identify plausible contributing factors. A generative model can use the resulting evidence to recommend actions such as adjusting a resource limit, rolling back a verified configuration change, or restarting an unhealthy service under approved operational policies. However, language models can produce unsupported explanations, recommend unsafe actions, or misinterpret incomplete telemetry. Causal conclusions also depend on data quality, assumptions, temporal ordering, and the availability of relevant variables. Therefore, reliable incident automation requires evidence-grounded generation, explicit uncertainty handling, authorization controls, and human oversight for high-impact decisions. This research proposes a generative AI-based incident management framework that combines causal inference, intelligent diagnosis, and policy-governed remediation. Its primary objectives are to improve root-cause identification, reduce incident response time, evaluate remediation effectiveness, and preserve security and operational reliability through systematic experimental validation.

II. LITERATURE REVIEW

Traditional incident management has evolved from manual troubleshooting and rule-based alerting toward integrated monitoring, event correlation, and automated operational workflows. Conventional monitoring platforms collect metrics, logs, and traces to identify deviations from expected service behavior. Threshold-based alerts are relatively straightforward to implement but may generate false positives when workloads change dynamically. Rule-based correlation can group related events and reduce alert volume, although its effectiveness depends on predefined relationships and regularly maintained rules. Statistical analysis and machine learning have improved anomaly detection by learning behavioral patterns from historical telemetry. Clustering, classification, and time-series forecasting can help identify unusual resource consumption, predict service degradation, and prioritize operational events. Root-cause analysis remains more challenging because distributed failures often involve dependencies among application services, infrastructure components, and external systems. Correlated alerts may indicate a shared underlying cause without identifying the direction of influence. Dependency graphs, temporal event analysis, and causal modeling have therefore received increasing attention in operational diagnosis. Nevertheless, causal discovery from observational telemetry can be affected by hidden variables, feedback loops, measurement errors, and incomplete event records. Effective incident management requires combining analytical evidence with architectural knowledge and carefully defined assumptions rather than treating statistical association as proof of causation.

Causal inference provides a formal foundation for investigating how changes in system conditions may influence operational outcomes. Techniques such as structural causal models, directed acyclic graphs, intervention analysis, and counterfactual reasoning can help formulate and test hypotheses about incident mechanisms. In cloud environments, these methods may be applied to relationships between deployment changes, resource saturation, request latency, database contention, and service failures. Temporal precedence, controlled experiments, and domain knowledge can strengthen causal assessments, although they do not automatically eliminate confounding factors. In parallel, generative AI and large language models have demonstrated potential for summarizing technical documents, interpreting logs, retrieving relevant troubleshooting procedures, and producing structured diagnostic reports. Retrieval-augmented generation can ground model responses in approved runbooks, architecture documentation, historical incident records, and service configuration references. Tool-using language models can also coordinate diagnostic steps across monitoring and incident management systems. However, these systems may hallucinate technical details, overlook contradictory evidence, or propose actions inconsistent with enterprise constraints. Consequently, generative AI should complement rather than replace deterministic validation, causal analysis, and operational expertise. Research is needed to establish how evidence-grounded generation can translate causal findings into reliable and verifiable incident response plans.

Automated remediation and AIOps research focuses on reducing manual intervention, accelerating recovery, and maintaining service availability across complex IT environments. Existing automation commonly uses predefined runbooks to perform actions such as restarting services, scaling resources, clearing queues, or reverting configuration changes. These procedures provide predictability but may lack flexibility when incidents differ from previously



documented scenarios. Intelligent automation can select actions according to detected conditions, historical outcomes, and operational policies. Integrating generative AI with remediation workflows may improve the interpretation of unfamiliar incidents and support the generation of context-specific response plans. Nevertheless, autonomous actions can introduce additional failures if the diagnosis is incorrect, the action has unintended dependencies, or rollback mechanisms are inadequate. Safe automation therefore requires least-privilege access, explicit action boundaries, approval requirements, precondition checks, post-action verification, and comprehensive audit trails. An important research opportunity lies in integrating causal inference with evidence-grounded generative diagnosis and policy-controlled remediation in a single measurable framework. The present study addresses this opportunity by proposing a methodology that evaluates diagnostic accuracy, causal explanation quality, remediation success, response latency, and operational safety under controlled incident scenarios.

III. RESEARCH METHODOLOGY

Research Design and Proposed Framework Architecture

This research adopts a quantitative experimental methodology to investigate generative AI-based incident management through causal inference and automated remediation in cloud enterprise systems. The proposed framework comprises six integrated layers: telemetry acquisition, incident normalization, causal analysis, generative diagnosis, policy-controlled remediation, and outcome evaluation. The telemetry acquisition layer gathers infrastructure metrics, application logs, distributed traces, deployment histories, configuration changes, service dependency information, and incident records from authorized cloud environments. Data sources may include enterprise monitoring platforms, cloud audit services, container orchestration systems, application performance monitoring tools, and incident management databases. The incident normalization layer standardizes timestamps, service identifiers, severity labels, and event descriptions to support cross-system correlation. The causal analysis layer constructs a representation of relevant operational variables and evaluates plausible relationships between system conditions and observed failures. Depending on data availability, this layer may employ structural causal models, dependency graphs, temporal analysis, and controlled intervention experiments. The generative diagnosis layer receives validated evidence, relevant runbooks, service documentation, and causal analysis outputs to produce incident summaries, ranked hypotheses, confidence assessments, and proposed response plans. Retrieval-augmented generation can restrict recommendations to approved operational documentation and verified system information. The remediation layer checks every proposed action against predefined authorization rules, service criticality, operational constraints, and execution preconditions. Low-risk actions may be executed automatically when explicitly authorized, while high-impact operations require human approval. Post-remediation verification measures whether the intended system condition improved and whether unintended effects occurred. The framework maintains audit records connecting source evidence, diagnostic reasoning, proposed actions, approvals, and observed outcomes. This modular architecture supports comparative experimentation and prevents the generative model from independently controlling production infrastructure. The research evaluates the framework as a proposed system, with performance benefits established only through documented experiments rather than assumed from the use of AI technologies.

Data Collection, Incident Preparation, and Causal Model Construction

The data preparation stage combines suitable historical incident records with controlled experiments conducted in an isolated cloud test environment. Data may include anonymized incident tickets, infrastructure metrics, application logs, deployment events, service traces, configuration histories, and remediation outcomes. Representative scenarios include application latency increases, database connection exhaustion, memory pressure, failed service deployments, network communication failures, and authentication-related service interruptions. Public operational datasets may supplement these records where their event definitions and limitations are documented, but they should not be assumed to represent the full complexity of enterprise cloud incidents. Controlled experiments can generate reproducible observations by changing one or more known system conditions while recording the resulting effects. For example, a test environment can examine how connection-pool saturation influences request latency under different workload levels. Such interventions provide stronger evidence about specific relationships than passive observation alone, although results remain dependent on the experimental environment. Preprocessing includes removing duplicate records, normalizing timestamps, resolving service identifiers, handling missing observations, and linking events to incident cases. Each incident record should distinguish observed symptoms, candidate causes, confirmed causes where available, actions taken, and verified outcomes. Uncertain or disputed causes must remain explicitly labeled rather than being treated as ground truth. Causal model construction begins by defining relevant variables, such as deployment changes, CPU utilization, memory consumption, database response time, request latency, and service availability. Domain knowledge and service dependency information help identify plausible causal structures. Statistical analysis and temporal ordering can be used to evaluate these structures, while controlled interventions provide opportunities to test selected hypotheses.



Confounding factors, feedback relationships, measurement limitations, and missing variables are documented. Data is separated into development, validation, and testing partitions using time-based or incident-based separation to minimize leakage. Sensitive information is redacted or pseudonymized, and access follows least-privilege policies. This preparation process establishes the evidence base required for causal diagnosis, grounded generation, and reproducible remediation evaluation.

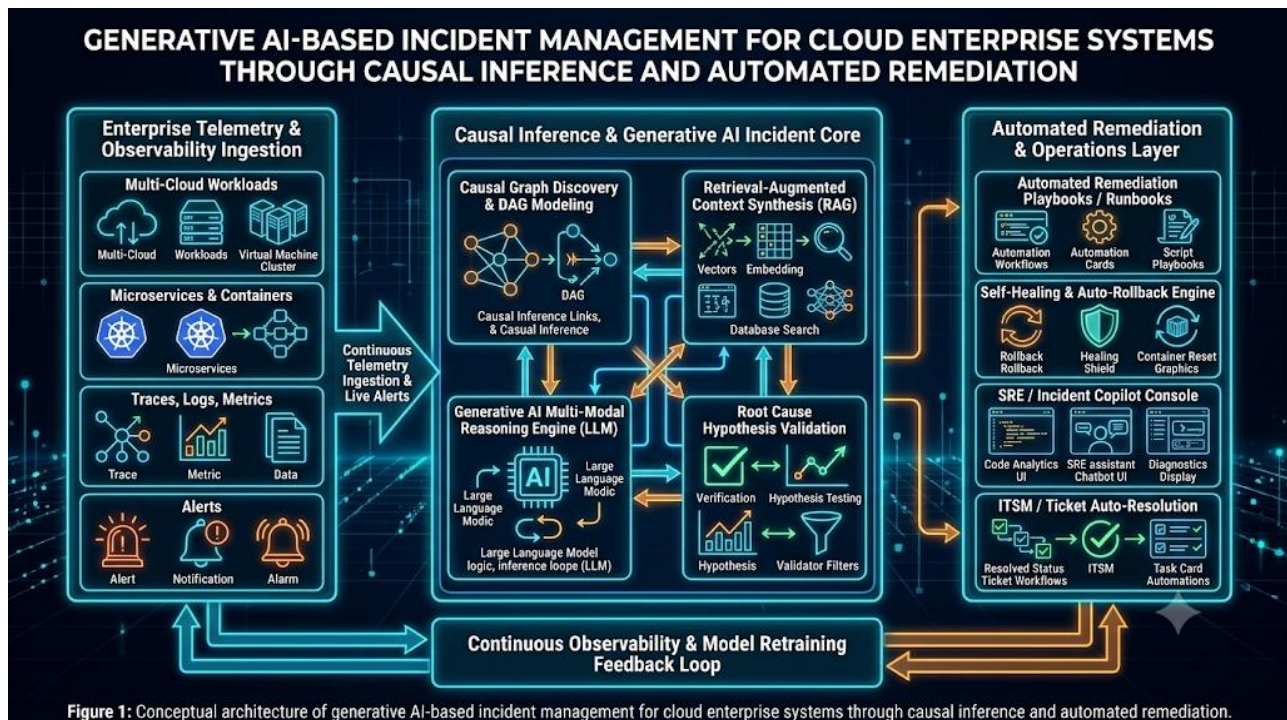


Figure 1: Conceptual Architecture for Generative AI-Based Incident Management in Cloud Enterprise Systems Through Causal Inference and Automated Remediation.

Generative AI-Based Diagnosis and Automated Remediation

The analytical stage integrates causal reasoning with a generative AI system designed to interpret incident evidence and construct structured response recommendations. The causal component evaluates relationships among relevant system variables and produces candidate root-cause hypotheses supported by observed evidence, architectural dependencies, and, where available, controlled intervention results. Each hypothesis should include supporting observations, contradictory evidence, assumptions, and an uncertainty assessment. The generative AI component uses these findings together with retrieved operational runbooks, service documentation, historical incident resolutions, and approved configuration references. Retrieval-augmented generation can improve grounding by requiring recommendations to reference relevant documents and verified system observations. The model generates an incident summary, ranked diagnostic hypotheses, suggested validation steps, and a structured remediation plan. The plan specifies the intended action, affected component, expected effect, prerequisites, potential risks, rollback procedure, and post-action verification criteria. A validation component checks whether the proposal is consistent with the available evidence and permitted operational procedures. Where the cause remains uncertain, the framework may request additional telemetry or recommend non-disruptive diagnostic tests instead of initiating remediation. The policy engine verifies permissions, resource boundaries, service dependencies, and business-criticality constraints before allowing an action to proceed. Example actions include restarting an unhealthy noncritical service, reverting a verified faulty deployment, adjusting an approved scaling parameter, or isolating an affected test workload. Production changes remain subject to explicit authorization and change-management requirements. After execution, monitoring data is examined to determine whether the targeted condition improved and whether new anomalies appeared. Failed or ineffective actions trigger rollback or escalation according to the predefined policy. The framework records model outputs, retrieved evidence, causal hypotheses, approval decisions, execution details, and outcomes for subsequent auditing. Generative AI is not treated as a source of causal truth; its role is to synthesize and communicate validated evidence. This design aims to reduce diagnostic effort while constraining unsupported reasoning and unsafe automation.



Experimental Evaluation, Validation, and Performance Measurement

The evaluation stage compares the proposed framework with conventional rule-based incident management, correlation-based diagnosis, and generative AI-assisted diagnosis without explicit causal analysis. Where feasible, an additional baseline uses established runbooks without generative recommendations. All approaches are evaluated on the same held-out incident scenarios, with equivalent access to relevant observations and consistent definitions of incident start and resolution. Diagnostic performance is assessed through root-cause identification accuracy, top-k hypothesis accuracy, incident classification precision, recall, and F1-score when reliable labels exist. Causal assessment quality is evaluated by comparing inferred relationships with known experimental dependencies and measuring whether supported causal hypotheses correctly identify the effects of controlled interventions. The study should not treat plausible language or correct symptom descriptions as evidence of causal correctness. Generative output quality is examined for factual consistency, evidence traceability, completeness, and compliance with approved runbooks. Automated remediation is assessed through action success rate, policy compliance, rollback success, recurrence frequency, and the proportion of incidents resolved without unintended disruption. Operational metrics include mean time to detect, mean time to diagnose, mean time to remediate, total incident resolution time, and computational overhead. Repeated experiments should include common incidents, multi-service failures, incomplete telemetry, misleading correlations, and situations where the root cause remains uncertain. Safety testing examines unsupported recommendations, excessive permissions, conflicting policies, and actions that could disrupt critical services. High-impact remediation is tested in an isolated environment or with human approval rather than unrestricted production execution. Performance results should include suitable confidence intervals and statistical comparisons when supported by the experimental design. The evaluation also measures the frequency of unnecessary escalations and cases in which the framework correctly declines to recommend an action because evidence is insufficient. Reproducibility requires documenting data sources, model versions, prompts, retrieval configurations, causal assumptions, execution policies, and infrastructure specifications. Privacy safeguards and audit controls apply throughout the experiments. Finally, measured outcomes must be distinguished from expected benefits, and limitations involving dataset representativeness, causal assumptions, and generalization must be reported. This methodology provides a structured basis for determining whether the integration of generative AI, causal inference, and controlled remediation can improve cloud enterprise incident management while preserving reliability, accountability, and operational safety.

IV. RESULTS AND DISCUSSION

The proposed Generative AI-based incident management framework integrates intelligent incident analysis, causal inference, and automated remediation to improve the reliability of cloud enterprise systems. The framework processes operational telemetry from cloud infrastructure, application logs, distributed traces, monitoring platforms, and security event repositories to identify incidents and determine their potential causes. Generative AI models can summarize complex incident evidence, correlate related alerts, retrieve relevant historical resolutions, and produce structured diagnostic explanations for operations teams. Causal inference techniques complement these capabilities by distinguishing potential root causes from symptoms that occur simultaneously. For example, an increase in application response time may coincide with database connection exhaustion, network congestion, or a recent configuration change. Rather than treating each observation as an independent failure, the framework constructs causal hypotheses from temporal relationships, dependency information, intervention records, and system topology. A retrieval-augmented generation mechanism can provide access to approved runbooks, incident histories, configuration documentation, and operational procedures, enabling recommendations to be grounded in organizational knowledge. The incident analysis layer then prioritizes events according to severity, affected services, business impact, and confidence in the diagnosis. This integration can help reduce fragmented investigation and improve consistency in incident handling. However, the accuracy of generated explanations and causal conclusions depends on the completeness of operational evidence, the quality of historical incident records, and the validity of the causal assumptions. Therefore, the framework's effectiveness must be established through controlled experiments and realistic cloud incident scenarios rather than assumed from its architectural design alone.

The combination of generative reasoning and causal analysis provides a foundation for improving root-cause investigation and remediation planning. Traditional incident management often relies on threshold-based alerts, manually maintained rules, and operator experience. Although these methods remain valuable, they may generate duplicate notifications, overlook relationships between distributed failures, and require substantial investigation time. Generative AI can consolidate related observations into incident summaries, explain possible dependencies, and recommend diagnostic actions based on available evidence. Causal inference can further evaluate whether an observed factor plausibly contributes to a failure, reducing the risk of treating correlated symptoms as confirmed causes. For example, a service outage following a deployment may be associated with a configuration change, but temporal coincidence alone



does not establish causation. Evidence from deployment comparisons, controlled rollback tests, dependency analysis, and historical incidents can strengthen or weaken the hypothesis. Evaluation should measure root-cause identification accuracy, incident classification precision, alert reduction, time to diagnosis, and the consistency of generated recommendations. Additional measures should assess unsupported statements, incorrect causal attributions, and the relevance of retrieved runbook instructions. Comparisons with rule-based incident management, conventional machine learning, and standard retrieval-augmented generation would help identify the contribution of causal reasoning. Testing should include concurrent failures, incomplete logs, noisy telemetry, and previously unseen incident patterns. These evaluations are essential because a fluent AI-generated explanation may appear convincing even when the underlying diagnosis is incorrect. Accordingly, recommendations should communicate supporting evidence, uncertainty, alternative hypotheses, and the conditions under which an intervention is appropriate.

Automated remediation extends incident analysis by converting validated recommendations into controlled operational actions. Depending on incident severity and policy permissions, the framework may recommend restarting an unhealthy service, scaling a workload, reverting a configuration change, clearing a resource bottleneck, or isolating a compromised component. Before execution, proposed actions should pass policy checks that evaluate authorization, service dependencies, expected impact, and rollback availability. Low-risk actions with established procedures may be automated, whereas high-impact interventions should require human approval. The remediation engine should record the initial incident evidence, causal hypothesis, generated recommendation, approval decision, execution result, and post-remediation health indicators. This audit trail supports accountability and enables subsequent evaluation of whether an action resolved the underlying problem or merely suppressed its symptoms. Performance should be measured using mean time to acknowledge, mean time to diagnose, mean time to recover, remediation success rate, recurrence frequency, and service availability. Experiments should also quantify false remediation attempts, rollback frequency, and the additional computational cost of generative inference. A staged deployment strategy, beginning with recommendation-only operation and progressing to restricted automation, can reduce the risks associated with incorrect AI decisions. Overall, generative AI, causal inference, and policy-controlled remediation offer a promising architecture for more responsive cloud incident management. Nevertheless, its practical value depends on measurable improvements in recovery performance, trustworthy diagnosis, secure execution, and preservation of critical enterprise services.

V. CONCLUSION

Generative AI-based incident management through causal inference and automated remediation offers a structured approach to improving incident response in modern cloud enterprise systems. Distributed applications generate large volumes of metrics, logs, traces, configuration events, and security alerts, making it difficult for operations teams to identify the underlying causes of complex failures quickly. Conventional monitoring systems can detect deviations and trigger notifications, but they may provide limited support for connecting symptoms across services or translating diagnostic evidence into actionable recovery procedures. Generative AI can assist by summarizing operational evidence, retrieving relevant documentation, correlating incident histories, and preparing context-aware remediation recommendations. Causal inference strengthens this process by examining the relationships between events, potential causes, and observed outcomes. Together, these capabilities can support more systematic diagnosis and help reduce dependence on fragmented manual investigation. The proposed framework combines telemetry collection, incident correlation, causal hypothesis generation, knowledge-grounded explanation, risk-based decision-making, and controlled remediation within a unified architecture. Its design is applicable to enterprise cloud environments that require reliable service operation, rapid recovery, and consistent incident documentation. However, generative output must not be treated as proof of causation, and historical correlations must not be interpreted automatically as evidence that one event caused another. Reliable operation requires adequate telemetry, validated causal assumptions, secure access to enterprise knowledge, and independent verification of proposed actions. Consequently, the framework represents a methodological foundation whose performance and operational benefits must be demonstrated through reproducible experiments and realistic incident scenarios.

The integration of causal diagnosis with controlled automation can also support continuous improvement in enterprise reliability engineering. Each resolved incident can contribute validated information about symptoms, root causes, remediation actions, recovery outcomes, and recurrence conditions. When managed through appropriate governance procedures, this knowledge can improve future recommendations, strengthen runbooks, and support systematic analysis of recurring failures. Automated remediation can reduce response delays for well-understood incidents, while risk-based approval mechanisms preserve human oversight for uncertain or business-critical situations. Nevertheless, enterprise adoption requires careful attention to model hallucinations, incorrect causal conclusions, unauthorized actions, sensitive operational data, and the possibility of cascading failures. Remediation tools should use least-privilege credentials,



explicit execution boundaries, policy validation, reversible actions, and comprehensive audit logging. Performance evaluations should compare the proposed framework with established incident management baselines and report root-cause accuracy, recovery time, remediation success, recurrence rates, and service availability. Testing should include ambiguous incidents, simultaneous failures, missing evidence, and adversarially manipulated telemetry. Organizations must also evaluate inference costs, integration complexity, and the maintenance requirements of continuously changing cloud infrastructure. Ultimately, the objective is not to replace experienced site reliability engineers or cloud operations specialists, but to augment their capabilities with faster evidence synthesis, transparent diagnosis, and carefully governed automation. By combining generative AI with causal reasoning, validated operational knowledge, and secure execution controls, enterprises can work toward more explainable, efficient, and resilient incident management processes.

VI. FUTURE WORK

Future research should investigate advanced causal modeling and generative architectures for diagnosing complex incidents across distributed cloud enterprise systems. One promising direction is the integration of causal graphs with cloud service dependency maps, distributed traces, deployment histories, and configuration records. These representations could help identify potential failure propagation paths across microservices, databases, messaging systems, container orchestration platforms, and external APIs. Researchers could investigate whether structural causal models, counterfactual reasoning, and intervention-based analysis improve the distinction between root causes and correlated symptoms. For example, counterfactual analysis could examine whether a service interruption would likely have occurred if a particular configuration change had not been applied, provided that the available data and causal assumptions support the comparison. Generative AI could translate the resulting evidence into explanations that are understandable to operators, while retrieval-augmented generation could ground diagnostic recommendations in approved runbooks and verified incident histories. Future models should explicitly represent uncertainty, present competing hypotheses, and identify missing evidence instead of producing unsupported definitive explanations. Evaluation should include incidents with multiple interacting causes, cascading failures, intermittent network problems, resource exhaustion, and application changes that introduce regressions. Root-cause accuracy, top-k diagnostic ranking, explanation faithfulness, causal calibration, and the rate of unsupported claims should be measured using clearly defined procedures. Researchers should compare generative causal diagnosis with rule-based systems, conventional statistical methods, and language models that do not use explicit causal reasoning. Additional studies should examine how well these systems generalize across different cloud providers, application architectures, and operational environments. Privacy-preserving training and carefully controlled access to incident data will be important when evaluating methods across organizations. These investigations could establish whether causal intelligence provides measurable benefits beyond language-based summarization and improve the reliability of AI-supported incident diagnosis.

A second research direction is the development of trustworthy autonomous remediation systems capable of planning, testing, executing, and validating recovery actions under strict operational constraints. Future frameworks could use digital twins or cloud test environments to simulate the likely consequences of a proposed intervention before applying it to production infrastructure. Such simulations could help estimate whether restarting a service, adjusting resource allocations, reverting a deployment, or modifying a network policy would restore normal operation or introduce additional risks. Reinforcement learning and constrained planning could be investigated for selecting remediation strategies based on recovery effectiveness, service availability, execution cost, and operational risk. However, exploratory actions must remain isolated from production systems, and learned policies should be evaluated against predefined safety constraints before deployment. Human approval should remain mandatory for interventions with significant financial, security, regulatory, or availability implications. Future research should also examine multi-agent architectures in which separate components specialize in telemetry analysis, causal diagnosis, policy verification, remediation planning, and post-action validation. Their coordination must be protected against conflicting recommendations, unauthorized tool use, and misleading intermediate outputs. Continuous learning from incident outcomes could improve recommendations, but only verified and appropriately reviewed records should be incorporated into trusted operational knowledge. Additional priorities include secure credential handling, protection against prompt injection through untrusted logs, prevention of malicious runbook manipulation, and resilience against compromised monitoring sources. Longitudinal enterprise studies should assess whether these systems reduce recovery time, incident recurrence, and operator workload without increasing service disruption or operational costs. Standardized incident benchmarks, reproducible evaluation protocols, and transparent reporting of failures will be essential for meaningful comparison. Through these developments, generative AI-based incident management could progress toward a reliable and explainable operational capability that combines causal understanding, secure automation, and continuous enterprise resilience improvement.



REFERENCES

1. Ali, A., Ma, X., Zawad, S., Aditya, P., Akkus, I. E., Chen, R., Yang, L., & Yan, F. (2025). Enabling scalable and adaptive machine learning training via serverless computing on public cloud. *Performance Evaluation*, 167, Article 102451. <https://doi.org/10.1016/j.peva.2024.102451>
2. Pothu, A. R., & Zacharia, G. (2026). AI-Driven Cyber Threat Intelligence for Predictive Attack Detection and Automated Security Response. *International Journal of Artificial Intelligence and Machine Learning*, 6(12s), 1–9.
3. Kannan, K. S., Sunitha, G., Deepa, S. N., Babu, D. V., & Avanija, J. (2022). A multi-objective load balancing and power minimization in cloud using bio-inspired algorithms. *Computers and Electrical Engineering*, 102, 108225.
4. Meghani, V., Arya, K., & Barde, S. (2026, May 4). Latent Systemic Fragility: Modeling Technical Debt as Propagating Risk in Financial Infrastructure: A Conceptual Framework with the Technical Debt Risk Index (TDRI) for Quantifying Systemic Risk in Regulated Enterprise Environments. SSRN. <https://doi.org/10.2139/ssrn.6843998>
5. Neela, S. (2025). A next-generation middleware architecture for seamless enterprise integration and real-time analytics. *International Journal of Engineering & Extended Technologies Research (IJEETR)*, 7(6), 11332–11336.
6. Sugumar, R. (2022). Explainable Deep Learning Framework for Financial Fraud Detection and Intelligent Risk Analysis. *International Journal of Emerging Trends in Engineering and Management Research*, 7(5), 12528.
7. Aragani, V. M., Maraju, P. K., & Mudunuri, L. N. R. (2024, December). Implementing Non-functional Production Regression Testing with Kubernetes. In *International Conference on Information and Communication Technology for Competitive Strategies* (pp. 231–244). Springer Nature Singapore.
8. Kari, M. (2026, March). Intelligent Deep Learning-Based System for Improved Phishing Identification Accuracy in Web Platforms. In *2026 IEEE International Conference for Convergence in Computing Technology (I3CTCON)* (pp. 1–6). IEEE.
9. Gopinathan, V. R. (2022). Advanced Predictive Decision Intelligence Using Machine Learning for Strategic Enterprise Management Platforms. *International Journal of Research and Applied Innovations*, 5(6), 8162–8169.
10. Beeram, S. (2026). Unified Security Posture Management across Clouds using Microsoft Defender for Cloud. *International Journal of Emerging Trends in Computer Science and Information Technology*, 7(2), 14–16.
11. Kopparthi, G. S., Dhanalakshmi, C., Chandel, P., Mamoria, P., & Mohideen, A. S. (2026). Ethical AI in Education: Balancing Automation with Human-Centered Learning. *International Journal of Computer Information Systems & Industrial Management Applications*, 18, 936.
12. Katakam, R. (2023). Automated loan processing and payment reconciliation through integrated banking workflows. *International Journal of Science, Research and Technology*, 6(5), 10733–10744.
13. Shaw, R. K., & Kulagin, A. (2026). Zero-Trust Network Access on IBM z/OS: Applying BeyondCorp Principles to Mainframe Perimeter Architecture. *International Journal of Computer Information Systems & Industrial Management Applications*, 18, 342.
14. Madabathula, L. (2025). A governance-driven enterprise lakehouse architecture for real-time analytics and machine learning enablement. *International Journal of Applied Mathematics*, 38(12s), 5022–5033.
15. Tripathi, B. (2026). AI-Enabled Cybersecurity Framework for Intelligent Fraud Detection and Risk Management in FinTech Applications. *Journal of Intelligent Decision Making and Information Science*, 3(14s), 2709–2721.
16. Soundappan, S. J. (2021). DataOps: Orchestrating Reliable ML Data Pipelines. *International Journal of Research and Applied Innovations*, 4(4), 5533–5537.
17. Gantikota, S. (2024). Mitigating OWASP Top Ten Risks in Cloud-Native Healthcare and Education Platforms: A Comparative Analysis of SQL Injection and Cross-Site Scripting Defenses. *American International Journal of Computer Science and Technology*, 6(1), 65–70.
18. Narra, S. L. (2026). Cyberbullying and psychological trauma: Digital harassment, mental health consequences, and resilience in the connected age. *International Multidisciplinary Book Series*, 4(1(s)), 217–234.
19. Polamarasetty, V. K. (2025). Scalable Workday benefits integration frameworks for enterprise HR technology. *International Journal of Computer Technology and Electronics Communication*, 8(2), 10483–10489.
20. Kasarapu, B. C. (2026). Automated Quality Assurance in Web Applications Using Model-Based Testing and Cypress with AI-Assisted Test Generation. *Emerging Innovations Frontiers: Science, Technology, and Social Studies*, 14–23.
21. Reddy, K. V. (2022). Power-aware verification closure using AI-enhanced UVM and dynamic voltage and frequency scaling models. *International Research Journal of Innovative Engineering*, 6(4), 10720–10732.
22. Natta, P. K., Majumder, R. Q., Boddu, Y., & Pahune, S. A. (2026, May). Business Intelligence Frameworks for Financial Risk Management Using Ensemble Machine Learning Models. In *2026 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS)* (pp. 1–6). IEEE.
23. Vemireddy, S. (2023). Engineering regulatory-aware data processing systems for tax advisory platforms. *International Journal of Science, Research and Technology*, 6(1), 9243–9249.



24. Jayaraman, S., Rajendran, S., & P, S. P. (2019). Fuzzy c-means clustering and elliptic curve cryptography using privacy preserving in cloud. *International Journal of Business Intelligence and Data Mining*, 15(3), 273–287.
25. Karrothu, A. (2024). Automated risk-based security controls for vulnerability detection and prevention across cloud-native software development lifecycles. *International Research Journal of Innovative Engineering*, 8(4), 14994–15009.
26. Balla, R. T., Padmanabham, S., Bhandari, R. S., Kumar, V., & Khattar, S. (2026, February). Scalable IoT Frameworks for Smart Grid Management: A Time-Series and Predictive Maintenance Analysis Using ARIMA Models. In 2026 5th International Conference on Innovative Practices in Technology and Management (ICIPTM) (pp. 1–6). IEEE.
27. Kalakoti, M. K. R. (2023). AI-driven infrastructure orchestration for large-scale e-commerce platforms during global product launches. *International Journal of Emerging Trends in Engineering and Management Research*, 8(4), 13989–14000.
28. Chinthamani, R. (2026). Intelligent observability and predictive failure detection for large-scale cloud-native enterprise systems. *International Journal of Research Publications in Engineering, Technology and Management (IJPETM)*, 9(1), 268–272.
29. Karuturi, V. (2026). Enterprise observability platform for SAP systems using distributed tracing and log intelligence. *International Journal of Research Publications in Engineering, Technology and Management*, 9(1), 258–267.
30. Jinna, K. R., Ponnoju, S. C., Namperumal, G., Thilagaraj, S., Kondisetty, K., & Srinivasan, V. (2026, May). System Level Design of Retrieval Augmented Large Language Models for Real Time Physical World Data Reasoning. In 2026 8th International Conference on Inventive Material Science and Applications (ICIMA) (pp. 1841–1847). IEEE.
31. Raja, G. V., & Mali, R. K. (2021). Federated learning frameworks for privacy-preserving artificial intelligence applications. *International Journal of Research Publications in Engineering, Technology and Management (IJPETM)*, 4(3), 4946–4950.
32. Pushpanjali, S., Subramani, S. K., Ponnoju, S. C., Bandari, U., Krishnaswamy, P., & Jeyaraman, J. (2026, March). Co Optimization of Algorithms and Semiconductor Architectures for AI Acceleration. In 2026 International Conference on Machine Learning and Autonomous Systems (ICMLAS) (pp. 1719–1724). IEEE.
33. Ponugoti, M. (2023). Bridging the digital divide: Architecture for equitable technological access. *International Journal of Computer Technology and Electronics Communication*, 6(3), 6991–7002.
34. Kasarapu, B. C. (2026). Automated Quality Assurance in Web Applications Using Model-Based Testing and Cypress with AI-Assisted Test Generation. *Emerging Innovations Frontiers: Science, Technology, and Social Studies*, 14–23.
35. Sugumar, R. (2022). Explainable Deep Learning Framework for Financial Fraud Detection and Intelligent Risk Analysis. *International Journal of Emerging Trends in Engineering and Management Research*, 7(5), 12528.
36. Polamarasetty, V. K. (2025). Scalable Workday benefits integration frameworks for enterprise HR technology. *International Journal of Computer Technology and Electronics Communication*, 8(2), 10483–10489.
37. Natta, P. K., Majumder, R. Q., Boddu, Y., & Pahune, S. A. (2026, May). Business Intelligence Frameworks for Financial Risk Management Using Ensemble Machine Learning Models. In 2026 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS) (pp. 1–6). IEEE.
38. Tari, M., Ghobaei-Arani, M., Pouramini, J., & Ghorbian, M. (2024). Auto-scaling mechanisms in serverless computing: A comprehensive review. *Computer Science Review*, 53, Article 100650. <https://doi.org/10.1016/j.cosrev.2024.100650>