International Journal of Science and Engineering Applications
Volume 11-Issue 12, 481 – 488, 2022, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1112.1069

Automated Incident Response in Hospital IT through AIOps-Driven DevOps Pipelines

Nagarjuna Nellutla
Independent Researcher
Eagan, MN, USA 55123

Abstract: Operational reliability in hospital information systems depends on the continuous availability of clinical software, network infrastructure, and bedside medical devices. Traditional incident response practices rely on manual escalation and reactive troubleshooting, which delay access to critical systems and disrupt patient workflows during infrastructure faults or abnormal device behavior. This paper proposes an AIOps-driven DevOps pipeline that automates incident detection and remediation across hospital IT environments by combining statistical anomaly analysis, event correlation, and rule-based remediation triggers. Classic machine learning models process device telemetry, network failures, and application health metrics to detect anomalies without requiring generative inference. Detected events are then executed as automated playbooks through DevOps pipelines to isolate faulty components, quarantine impacted nodes, initiate system restarts, or escalate alerts based on clinical priority. The proposed architecture treats operational reliability as a continuous deployment of corrective actions rather than a post-failure activity. By embedding anomaly detection into DevOps workflows, hospitals can reduce remediation time, improve service availability, and preserve clinical continuity without dependence on emerging generative models.

Keywords: AIOps, DevOps, Incident Response, Hospital IT, Healthcare Operations, Anomaly Detection, Clinical Reliability

1. INTRODUCTION

Modern hospitals depend on continuous access to clinical systems, biomedical equipment, and networked data services to support patient care. Electronic health record platforms, laboratory reporting systems, bedside telemetry devices, diagnostic imaging services, and clinical communication tools all operate within a shared digital ecosystem where failure in any component can immediately disrupt clinical workflows. Traditional incident response methods in these environments rely heavily on human intervention, where operations staff must interpret alerts, diagnose the root cause, and manually execute remediation steps. This reactive approach increases mean time to resolution and places clinical continuity at risk when failures occur during time-critical procedures.

The expansion of network-connected medical devices has amplified operational complexity. Biomedical sensors, infusion pumps, wearable monitors, and portable diagnostic machines generate continuous telemetry that must be processed to detect anomalies in performance or safety conditions. These devices produce irregular traffic patterns influenced by patient movement, clinical activities, and device configuration states, making it difficult to manage them with static rules alone.

Hospital network infrastructure also experiences fluctuating loads produced by imaging transfers, laboratory requests, telemedicine sessions, and electronic orders, all of which must remain reliably available.

AIOps introduces automated anomaly detection and response mechanisms to address this operational burden. Classic AIOps approaches employ statistical learning, clustering, and classification algorithms to identify abnormal patterns in system metrics without requiring generative inference or deep learning. Time-series models capture predictable variations in clinical workflows, allowing algorithms to anticipate workload trends in laboratory and EHR transaction volumes. Clustering techniques identify unusual communication behaviors in medical devices or network switches, while classification models distinguish between transient fluctuations and legitimate system faults that require remediation.

When integrated with DevOps practices, AIOps enables corrective actions to be executed as automated operational deployments rather than manual post-failure activities. Instead of waiting for technicians to evaluate an alert, anomaly detection models can trigger DevOps pipelines that execute predefined remediation playbooks. These include restarting degraded services, isolating network segments, reallocating compute resources, or initiating controlled device reboots. Each action becomes versioned, tested, and continuously improved like application code, ensuring that the same failure does not require repeated human intervention.

The benefits of automated incident response extend beyond reduced downtime. Hospitals operate under predictable yet critical workloads where delayed remediation can impact safety and treatment delivery. For instance, unresponsive EHR modules may delay medication reconciliation, network instability may disrupt telemetry feeds, and faulty laboratory services may stall result delivery. Automating corrective actions within the same ecosystem that deploys clinical software ensures that remediation follows standardized approval paths and does not impose additional operational risk.

Incident classification also improves when AIOps pipelines correlate multiple signals rather than reacting to single thresholds. Instead of triggering an alert based solely on CPU load or packet loss, multi-signal correlation uses concurrent

metrics such as device battery voltage, authentication failures, or elevated laboratory transaction delays to determine whether the event indicates a genuine fault. This reduces unnecessary escalations and prevents alert fatigue, enabling staff to focus on high-impact issues while automated pipelines address routine failures autonomously.

A combined architecture that integrates time-series predictions, clustering for telemetry outliers, and classification for fault discrimination yields a practical and explainable incident response mechanism suitable for hospitals. These algorithms can be tuned using operational history, do not require extensive training data, and produce interpretable outputs that align with clinical risk awareness. When paired with DevOps pipelines that manage corrective procedures, automated incident response evolves into a continuous operational workflow where learning, detection, and remediation operate in tandem.

By treating operational reliability as a continuous deployment of corrective measures, hospitals can reduce dependency on manual troubleshooting, shorten resolution windows, and improve the resilience of patient-supporting digital systems [1]. This approach elevates AIOps from a monitoring enhancement to an integral component of clinical infrastructure management, enabling predictable and explainable automation within regulated care environments.

2. AIOPS ARCHITECTURE FOR HOSPITAL IT INCIDENT AUTOMATION

Automated incident response in hospitals requires an architecture that captures telemetry from clinical systems, identifies abnormal conditions using explainable algorithms, and executes remediation tasks consistently. Hospital IT assets generate heterogeneous signals across three operational layers: enterprise software such as EHR and laboratory systems, network infrastructure that transports clinical transactions, and medical devices that stream patient data and equipment metrics. An AIOps-driven architecture must ingest and correlate telemetry from all three domains before deciding whether to initiate remediation or escalate to human operators.

The architecture begins with a multi-source observability layer that collects application logs, network metrics, and device telemetry. Time-series stream processors detect deviations from historical access patterns in health record services, while clustering algorithms isolate anomalous communication from bedside devices or gateways [2]. Classification algorithms then differentiate between transient variance and actionable faults [3]. This classification stage supports explainability, allowing operational staff to understand why a pipeline is triggered and what risk category it belongs to.

Hybrid remediation triggers prevent misclassification from causing unnecessary operational interventions. Machine learning models provide detection and confidence scoring, while rule-based evaluators confirm risk thresholds [4]. Remediation is only approved when anomaly scores exceed tuned thresholds and logical conditions are satisfied, such as prioritizing EHR failures over peripheral service abnormalities. The rule layer also directs which remediation playbook to execute, assigning automated actions and escalation priorities based on clinical sensitivity.

Once approved, DevOps pipelines execute remediation playbooks as structured release tasks. Actions may include restarting degraded services, isolating network segments, reallocating compute capacity, or rolling back unstable software versions. These pipelines are version-controlled and validated in staging environments, ensuring predictable impact before being deployed automatically in production. Fail-safe checks trigger rollback rules if remediation impacts system responsiveness or device connectivity during execution.

The service mesh and policy layer supports secure control operations between remediation pipelines and the hospital infrastructure they modify. Policies govern which systems can be restarted automatically, which actions require clinician visibility, and which functions must be quarantined rather than reset. This guardrail ensures that automated remediation respects regulated clinical environments by restricting actions that would affect live patient monitoring or interrupt medication workflows.

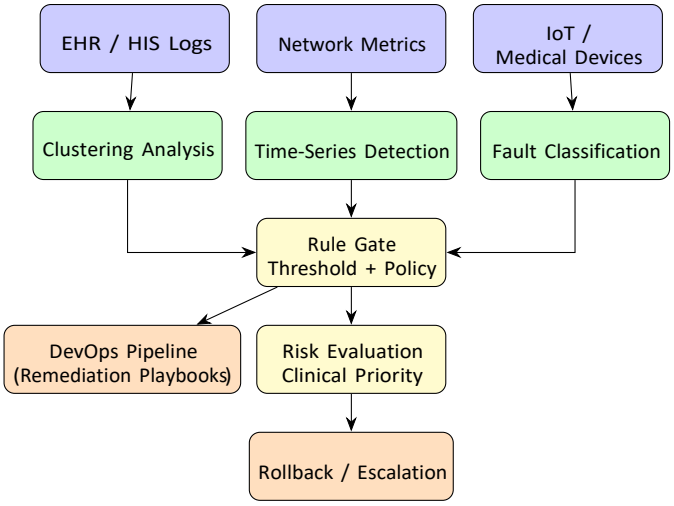

Fig. 1: AIOps-driven incident automation pipeline combining anomaly models, rule gating, and DevOps remediation playbooks for hospital IT.

Fig. 1 illustrates the complete AIOps-driven pipeline for incident automation in hospital IT environments. It shows how multi-source telemetry feeds anomaly models, which are validated by rule gates, before DevOps pipelines perform controlled remediation. The architecture supports rapid correction without sacrificing interpretability or safety.

This hybrid architecture ensures that incident remediation is fast, controlled, and clinically aware. Automation reduces resolution time, while rule validation and scoped execution protect mission-critical systems such as bedside monitors and

medication ordering modules. As a result, AIOps functions not as a replacement for human expertise, but as an operational extension that reinforces clinical reliability within hospital infrastructure.

3. INCIDENT DETECTION MODELS FOR HOSPITAL IT AIOPS PIPELINES

Incident response automation requires anomaly models that are explainable, predictable, and consistent with operational constraints of regulated clinical environments. In hospital IT systems, anomaly detection cannot rely on opaque computations that cannot justify actions leading to system restarts, network isolation, or service suspensions. Instead, algorithms must support human interpretability while detecting abnormal conditions across three operational domains: enterprise clinical systems, network infrastructure, and medical device telemetry [5]. Classic anomaly models satisfy these requirements by providing structured, measurable deviations that can be correlated with incident categories.

Time-series models capture repeatable clinical workflows such as morning census documentation, laboratory batch processing times, radiology uploads, and sudden shifts triggered by emergency admissions. Forecasting approaches like ARIMA and Holt–Winters detect deviations in predictable patterns, allowing remediation triggers to differentiate between genuine faults and natural spikes in request volume. For example, an unexpected drop in EHR query rate despite high census volume may indicate system degradation rather than reduced usage. Time-series models therefore provide baseline operational behavior derived from historical workloads.

Clustering models complement time-series analysis by isolating anomalous device communication and irregular network activity. Medical IoT devices often generate irregular patterns because patient behavior, device calibration, or network connectivity can change dynamically. Clustering techniques group similar telemetry profiles and identify abnormal device states that do not fit any learned cluster. This helps detect faulty devices, suspicious network traffic, unstable gateway connections, or malformed data outputs, without requiring extensive classification training.

Classification models are used when hospital systems must differentiate between benign anomalies and actionable faults. Support vector machines, decision boundaries, and threshold hybrid classifiers determine whether the detected deviation should initiate automated remediation. For example, a transient network jitter in a departmental switch may not require remediation, whereas prolonged interruptions around bedside monitors must be escalated immediately. These classifiers use fault features such as packet loss rates, query latency, device voltage signals, or memory saturation to determine severity.

The combination of these models yields a multi-layered detection architecture that separates normal patterns, anomalous telemetry, and actionable faults. Table I compares the suitability of classic anomaly models along the dimensions of explainability, training requirements, data source applicability, and operational overhead in hospital IT.

As shown in Table I, no single model is sufficient to detect all operational anomalies in hospital IT. Each algorithm family targets a distinct behavior profile, enabling layered detection across clinical workflow patterns, device telemetry outliers,

Model Type	Training Needs	Suitable Domains	Explainability
(Time-Series (ARIMA, HW))	Low–Medium	EHR workflows, network load forecasting	High
(Clustering (K-Means, DBSCAN))	Medium	IoT telemetry, device health outliers, gateway activity	Medium
(SVM/ Threshold Hybrid) Medium		Fault classification, priority escalation, severity gating	High

TABLE I: Comparison of classic anomaly detection techniques for AIOps-driven incident response in hospital IT.

and fault severity boundaries. By combining these models, AIOps pipelines achieve a balance between interpretability and automation, ensuring that remediation decisions are supported by measurable evidence rather than heuristic triggers alone. This enables incident response actions to be trusted, validated, and integrated into regulated clinical DevOps environments.

4. HYBRID REMEDIATION PLAYBOOKS FOR AUTOMATED HOSPITAL INCIDENT RESPONSE

AIOps-based incident remediation must balance automation speed with patient safety requirements. Unlike consumer IT environments, where rapid self-healing is the only priority, hospital operations involve regulatory constraints, clinical risk considerations, and audit accountability. Automated actions must therefore differentiate between infrastructure components that directly affect patient care and those that operate indirectly. Remediation tasks must be executed as reproducible workflows but must also adhere to selective safeguards based on clinical impact, data sensitivity, and operational priority [6].

Hybrid remediation playbooks enable this behavior by categorizing actions according to their clinical dependencies. Device or network operations that do not interact with clinical records or monitoring can be executed automatically without

blocking audit approval queues. Examples include resetting a non-clinical network controller, reallocating compute resources for laboratory queue processing, or isolating faulty IoT gateways. These actions reduce response time and prevent minor failures from cascading into wider service degradation. Since they operate outside the clinical record ecosystem, they can safely resolve incidents while only requiring audit logging after execution.

In contrast, actions that could impact clinical documentation, medication orders, telemetry streams, or patient metadata must be audited before they are performed. These include restarting EHR modules, clearing clinical record caches, modifying hospital identity services, or reconfiguring telemetry routing. Since errors in these systems can disrupt medication reconciliation, diagnostic workflows, or patient monitoring, hybrid playbooks introduce a pre-execution logging step that documents the automated response for compliance and traceability. This approach protects clinical continuity while preserving automation advantages.

The remediation engine uses the same DevOps pipeline infrastructure that deploys software updates. Playbooks are version-controlled, tested in staging environments, and linked to rollback actions that restore services if remediation impacts performance. Each playbook defines detection inputs, validation logic, action sequences, success criteria, and rollback conditions. Automated rollback allows pipelines to revert modifications if system responsiveness degrades or clinical services become unstable during execution. This ensures that remediation, like deployments, remains reversible and controlled.

Remediation tasks are divided into three categories: *restart, isolate, and recover*. Restart actions reset malfunctioning components such as application services, device applications, or overloaded network functions. Isolation actions remove impacted nodes from the service mesh, preventing disruption to other components. Recovery tasks restore system state through rollback, replenishing compute resources, or refreshing expired credentials. Each task executes with bounded autonomy, constrained by clinical awareness and audit policies implemented as rule gates.

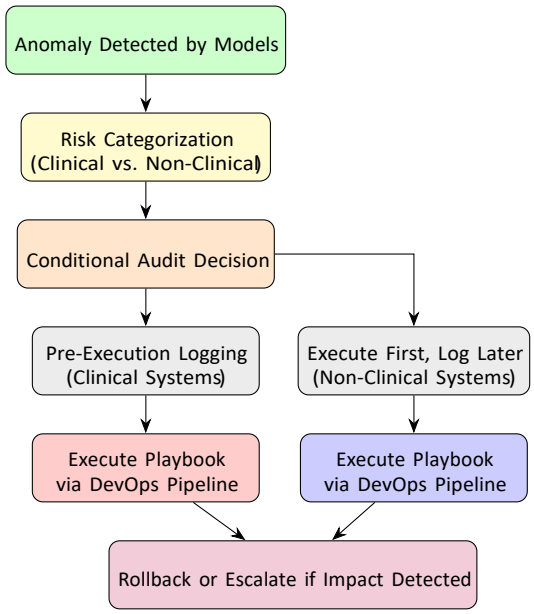

Fig. 2: Hybrid remediation flow: conditional logging and DevOps-based execution for AIOps-driven hospital incident response.

Fig. 2 illustrates the hybrid flow of automated incident remediation. It shows how anomaly models initiate remediation requests, how rule gates enforce conditional logging, and how DevOps pipelines execute actions according to their clinical risk category. The flowchart emphasizes that automation does not replace human oversight; instead, it formalizes and accelerates response procedures while preserving clinical accountability.

This hybrid playbook strategy prevents over-automation while accelerating response to operational faults. By embedding conditional logging and bounded autonomy into DevOps pipelines, hospitals preserve clinical accountability without compromising remediation speed. Automated incident response therefore becomes reproducible, auditable, and clinically aware, enabling dependable infrastructure behavior in environments where service disruptions directly affect patient care [7].

5. OBSERVABILITY AND SIGNAL CORRELATION FOR HOSPITAL AIOPS

Effective AIOps in hospital environments depends on rich, high-quality observability rather than isolated alerts. Logs, metrics, and traces generated by clinical applications, network infrastructure, and medical devices must be collected in a consistent format and time-aligned to support correlation [8]. Without this alignment, anomaly models and remediation logic operate on partial information, potentially misclassifying transient noise as faults or overlooking slow-developing failures [9]. Observability therefore serves as the foundational layer that translates raw operational data into actionable signals.

Hospital information systems produce verbose application logs, transaction timings, and user activity metrics. These streams capture how clinicians interact with EHR modules, order entry systems, and departmental applications. Network devices contribute telemetry about throughput, packet loss, interface errors, and routing events that affect connectivity between hospital systems and external services [10]. Medical devices generate continuous telemetry such as signal quality, connectivity status, battery levels, and internal error codes [11]. Each of these domains offers a partial view of infrastructure health only by correlating them can AIOps pipelines reliably infer incident scope and impact [12].

International Journal of Science and Engineering Applications
Volume 11-Issue 12, 481 – 488, 2022, ISSN:- 2319 - 7560
DOI: 10.7753/IJSEA1112.1069

Signal correlation engines aggregate these heterogeneous sources into unified event timelines. Rather than evaluating an increase in EHR latency or device disconnects individually, correlation logic identifies shared patterns—for example, simultaneous increases in lab result delays, elevated packet loss on a core switch, and frequent device reconnect attempts on a ward. Taken together, these signals indicate a localized network issue rather than independent application problem [13]s. This multi-signal approach reduces false positives and helps models distinguish between isolated component stress and systemic infrastructure failures.

Time synchronization is essential for meaningful correlation. Hospital AIOps systems must normalize timestamps across applications, switches, gateways, and devices, accounting for clock drift and differing logging intervals [14]. Once aligned, correlation engines can form "incident windows," periods during which related anomalies emerge across disparate systems. Within these windows, models compute relationships such as co-occurrence frequency, temporal ordering, and lag between precursor signals and visible failures. These relationships then inform fault classification and determine which remediation playbooks are appropriate.

The correlation process extends beyond simple co-occurrence metrics. Feature engineering incorporates signal types such as latency, error counts, throughput drops, disconnect events, and resource utilization into composite indicators. For instance, sustained latency increases accompanied by unchanged throughput may indicate application-level issues, while joint increases in latency and packet loss suggest network congestion or hardware degradation. By encoding such patterns, the correlation engine provides interpretable evidence that incident responders and automated pipelines can trust.

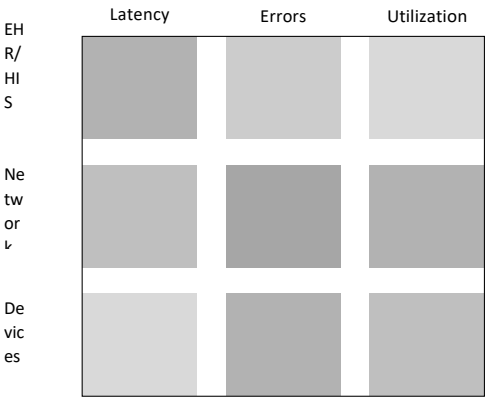

Fig. 3: Example correlation matrix indicating relative diagnostic value of signal types across EHR, network, and medical device domains. Darker cells represent stronger relevance for incident detection.

Fig. 3 illustrates a compact correlation matrix that represents how different hospital signal domains interact. Rows correspond to data sources, and columns represent key signal types used in anomaly analysis. Darker cells indicate stronger correlation or diagnostic value for incident classification, guiding engineers on where to invest observability and modeling effort.

By structuring observability around correlated signals instead of isolated metrics, hospital AIOps platforms can detect incidents earlier and classify them more accurately. Correlated patterns guide which anomaly models should be applied, which systems warrant priority investigation, and which remediation actions are appropriate. This correlation-centric approach transforms monitoring from a collection of independent dashboards into an integrated decision engine that supports rapid, automated incident response while preserving clinical safety.

6. EVALUATION DIMENSIONS AND EXAMPLE SCENARIOS

Assessing the effectiveness of automated incident response in hospital IT demands evaluation metrics that reflect both operational resilience and clinical safety. Unlike commercial AIOps deployments, which often prioritize uptime and cost efficiency, hospital environments require assurances that remediation actions do not compromise patient care. Therefore, evaluation must encompass performance, reliability of anomaly classification, alignment with clinical priorities, and the safety of automated decision-making. These dimensions collectively determine whether an AIOps pipeline strengthens hospital operations or introduces hidden risks.

Mean Time to Resolution (MTTR) remains a central evaluation metric, but its interpretation differs in clinical environments. Traditional MTTR improvement reflects reduced downtime and improved service delivery. In hospitals, shorter MTTR directly influences care continuity. Faster remediation of EHR slowdowns prevents delays in order entry, specimen processing, and medication verification. Reduced network recovery time stabilizes telemetry systems and remote diagnostic imaging, lowering the likelihood of delayed clinical decisions. Evaluation must therefore measure MTTR improvements per system type, prioritizing platforms whose failure directly affects patient-critical workflows.

False-positive rate reduction is another key dimension in evaluating AIOps systems. Excessive alerting leads to alert fatigue and wasted engineering effort; however, in hospitals, false positives can generate unnecessary remediation actions that disrupt care. Automated restarts or isolation events triggered by false-positive anomalies may disconnect bedside monitors or delay medication documentation. Evaluating false positives must therefore measure not only the accuracy of anomaly detection but also the clinical consequences of unnecessary remediation execution. AIOps pipelines must be tuned to favor human escalation rather than automation when uncertainty risks service disruption.

Safety compliance must be measured alongside automation performance. Evaluating safety involves verifying that remediation pipelines correctly enforce logging rules, policy constraints, and rollback conditions. Metrics should quantify the frequency with which unsafe remediation attempts are blocked, the time required to perform a rollback after adverse

impact, and whether logs fully capture required clinical events. Successful evaluation ensures that automation remains reversible, traceable, and aligned with regulatory expectations for digital clinical systems.

Example evaluation scenarios provide practical insight into model reliability. In one scenario, a spike in laboratory order throughput causes EHR latency to increase. An AIOps-enabled pipeline detects anomalous traffic, classifies it as benign surge driven latency, and suppresses remediation to avoid unnecessary service resets. The pipeline performance is evaluated by confirming that no restart occurred, latency recovered as surge subsided, and no care delays were observed. This scenario validates false-positive suppression and decision correctness under high load, demonstrating that non-action can be the safest remediation outcome.

Another scenario involves erratic telemetry behavior from a bedside monitor due to unstable Wi-Fi handoffs. Clustering models detect an outlier communication pattern, classify it as device-specific rather than systemic, and trigger an automated gateway reset while logging the recovery afterward. Evaluation measures MTTR improvement since gateway remediation occurred immediately without awaiting audit approval. This scenario illustrates conditional logging effectiveness and validates the autonomy of non-clinical remediation playbooks.

A more complex scenario concerns an EHR module failure in a medication ordering service. Hybrid remediation logic evaluates severity, logs the event before execution, and rolls back changes when system responsiveness drops further during recovery. Evaluation metrics include rollback execution time, impact on order latency, and completeness of audit entries [15]. This scenario tests both decision safety and operational reversibility, reinforcing that effective evaluation must confirm whether automation preserves clinical trust as well as system stability.

These scenarios illustrate that automated incident response cannot be judged solely by technical measures such as uptime or throughput. Evaluation must reflect clinical outcomes, safety guarantees, and the reliability of remediation decisions. By treating resilience as a multidimensional performance objective, hospitals can reliably deploy AIOps solutions that accelerate recovery while protecting clinical workflows from unintentional disruption.

7. GOVERNANCE, DEPLOYMENT STRATEGY, AND COMPARATIVE IMPACT OF AIOPS IN HOSPITAL IT

Introducing automated incident response into hospital IT requires more than model accuracy or pipeline reliability; it demands a governance framework that controls where automation is allowed, how it evolves, and who is accountable when actions affect clinical workflows. Governance defines boundaries around automation, establishes approval processes, and clarifies which operations teams, biomedical engineers, and clinical stakeholders must be involved at each maturity stage. Without formal governance, even technically correct automation can erode trust if remediation steps impact patient facing systems without clear justification or traceability.

A practical governance model assigns responsibility across three roles: operational engineering, clinical safety oversight, and compliance auditing. Operational engineers design AIOps models and remediation playbooks, clinical representatives validate that proposed actions do not interfere with care delivery, and compliance teams ensure that logging, access, and rollback behavior align with regulatory requirements. Automated pipelines must respect these role boundaries by enforcing approval gates for high-risk systems and exposing configuration changes through auditable change logs. Governance thus acts as a control plane over the automation layer rather than an afterthought.

Deployment strategy must reflect this governance structure. A balanced rollout approach begins with an observation-only phase, where AIOps models analyze signals and generate recommendations without executing any remediation. This allows teams to evaluate detection quality, correlation logic, and projected playbook behavior without affecting live systems. In the next phase, automation is enabled for low-risk, non-clinical infrastructure such as file transfer services, cache tiers, or departmental support applications. Only after these pipelines demonstrate stable performance does automation extend to hybrid mode for clinical-adjacent systems, where some actions are fully automated and others require human approval.

For core clinical systems like EHR components, order entry engines, and bedside telemetry aggregation, automation remains gated until both model performance and operational guardrails are proven. Even when automated remediation is enabled, it should start with narrow, low-impact playbooks such as cache clears, service restarts behind redundant endpoints, or controlled failover to known-stable nodes. Governance policies can then progressively authorize broader actions based on historical safety evidence, while retaining manual override capabilities and clearly documented escalation procedures.

Continuous monitoring and feedback loops are integral to both governance and deployment strategy [16]. Evaluation metrics such as MTTR improvement, false-positive remediation attempts, rollback frequency, and clinical incident reports feed into governance reviews. If remediation actions are rolled back too frequently or correlate with clinical complaints, governance bodies can restrict automation scope, tighten thresholds, or revert specific playbooks to manual approval. Likewise, consistently successful automated responses build the evidence needed to expand automation safely into additional domains.

The impact of AIOps relative to traditional manual operations can be summarized along several dimensions, including response time, consistency, auditability, and risk management. Table II provides a structured contrast that highlights both the

strengths and trade-offs of automated incident response in hospital environments.

This comparison shows that AIOps delivers substantial gains in response speed, consistency, and auditability but introduces new governance responsibilities around safety and threshold calibration. A mature deployment therefore combines automation with strong policy controls, staged rollout, and continuous review. When governance and deployment strategies are aligned, hospitals can harness AIOps to improve operational reliability while preserving the safeguards and accountability required in clinical environments.

8. CONCLUSION

Automated incident response represents a transformative evolution for hospital IT operations, bridging the gap between scalable digital infrastructure and the safety requirements of clinical environments. As healthcare systems continue to expand their reliance on interconnected software, network services, and bedside medical devices, operational failures increasingly threaten not only information accessibility but also the continuity of patient care. Traditional incident handling, dependent on human availability and reactive troubleshooting, cannot reliably meet the demands of hospitals in which delayed system recovery may compromise diagnostic accuracy, medication workflows, or real-time telemetry

TABLE II: Comparison of manual incident handling and AIOps-driven remediation across key operational dimensions in hospital IT.

Dimension	Manual Operations	AIOps-Driven Pipelines
MTTR	Dependent on staff availability and expertise; variable resolution times	Consistent, predictable response once playbooks are validated and triggered automatically
Consistency	Procedures may differ between shifts or teams; risk of ad-hoc fixes	Playbooks enforce standardized remediation sequences across all incidents of a given class
Auditability	Logging quality depends on human documentation practices	Automated actions, parameters, and timing recorded consistently through pipeline logs
Safety Control	High human oversight but prone to delays during critical incidents	Guardrails enforced by policy gates; requires careful threshold tuning to avoid over-automation
Scalability	Difficult to scale during surges or widespread incidents	Naturally scales with event volume, as pipelines can process many incidents in parallel

Oversight. AIOps-driven remediation, when implemented with appropriate safeguards, offers a more predictable and resilient foundation for supporting clinical systems.

The use of classic anomaly detection techniques, rather than opaque generative models, enables explainability and interpretability that are essential in regulated care settings. Time-series forecasting, clustering for telemetry deviations, and fault-classification algorithms collectively deliver a structured and transparent mechanism for identifying abnormal behavior across medical devices, networks, and EHR applications. When combined with hybrid, rule-gated remediation playbooks, these algorithms ensure that automated actions are neither unrestricted nor overly conservative. Instead, response mechanisms become context-aware, prioritizing clinical safety without sacrificing incident resolution speed.

Beyond algorithm design, the success of AIOps in healthcare depends on operational governance. Hospitals must embrace a deployment strategy in which automation expands gradually, guided by measurable reliability improvements and evidence of safety. Selective autonomy, conditional logging, and reversible remediation workflows enable continuous correction of system faults without undermining clinical trust. DevOps pipelines serve as the enforcement mechanism for these safeguards, providing version control, standardized execution, audit traceability, and immediate rollback capacity when clinical impact becomes detectable.

As hospitals adopt increasingly complex digital ecosystems, resilience becomes a function of automation maturity rather than human capacity alone. The framework presented in this work demonstrates that AIOps is not merely a monitoring enhancement but a disciplined operational model that unites observability, anomaly detection, remediation orchestration, and safety governance. When applied with structured controls, AIOps-driven incident response turns infrastructure reliability into a continuous service rather than a best-effort procedure, allowing hospital IT to scale with clinical demand while maintaining the precision and accountability expected in healthcare environments.

9. REFERENCES

[1] L. Chen, “Continuous delivery: Overcoming adoption challenges.” *J. Syst. Softw.*, vol. 128, pp. 72–86, 2017.

[2] V. J. M. Watzlaf, L. Zhou, D. R. DeAlmeida, and L. M. Hartman, “A systematic review of telehealth privacy and security,” *International Journal of Telerehabilitation*, vol. 9, no. 2, pp. 39–58, 2017. [Online]. Available: https://doi.org/10.5195/ijt.2017.6231

[3] C.-C. Chang, S.-R. Yang, E.-H. Yeh, P. Lin, and J.-Y. Jeng, “A kubernetes-based monitoring platform for dynamic cloud resource provisioning,” in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, 2017, pp. 1–6.

[4] A. A. Khaleq and I. Ra, "Intelligent autoscaling of microservices in the cloud for real-time applications," *IEEE Access*, vol. 9, pp. 35464– 35476, 2021.

[5] B. Kaplan, "Ethical, legal, and social issues in telehealth and telemedicine," *International Journal of Medical Informatics*, vol. 143, p. 104239, 2020. [Online]. Available: https://doi.org/10.1016/j.ijmedinf. 2020.104239

[6] N. Cruz Coulson, S. Sotiriadis, and N. Bessis, "Adaptive microservice scaling for elastic applications," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 4195–4202, 2020.

[7] A. Rahman and L. Williams, "Source code properties of defective infrastructure as code scripts," *Information and Software Technology*, vol. 112, pp. 148–163, 2019. [Online]. Available: https://doi.org/10. 1016/j.infsof.2019.04.013

[8] M. M. R. Hasan, E. L. Asaf, B. Paik, and J.-F. Letarte, "Testing practices for infrastructure as code," *ACM Software Engineering Notes*, vol. 45, no. 4, pp. 1–7, 2020. [Online]. Available: https: //doi.org/10.1145/3416504.3424334

[9] R. Mahindru, H. Kumar, and S. Bansal, "Log anomaly to resolution: Ai based proactive incident remediation," in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2021, pp. 1353–1357.

[10] I. Keshta and A. Odeh, "Security and privacy of electronic health records: Concerns and challenges," *Egyptian Informatics Journal*, vol. 22, no. 2, pp. 177–183, 2021. [Online]. Available: https://doi.org/10.1016/j.eij.2020.07.003

[11] A. Rahman, C. Parnin, and L. Williams, "The seven sins: Security smells in infrastructure as code scripts," in *2019 IEEE/ACM 41st International Conference on Software Engineering (ICSE)*, 2019, pp. 164–175.

[12] T. M. Hale and J. C. Kvedar, "Privacy and security concerns in telehealth," *AMA Journal of Ethics*, vol. 16, no. 12, pp. 981–985, 2014. [Online]. Available: https://doi.org/10.1001/virtualmentor.2014. 16.12.jdsc1-1412

[13] T. Kanwal, A. Anjum, and A. Khan, "Privacy preservation in e-health cloud: Taxonomy, privacy requirements, feasibility analysis, and opportunities," *Cluster Computing*, 2020. [Online]. Available: https://doi.org/10.1007/s10586-020-03106-1"

[14] Y. Dang, Q. Lin, and P. Huang, "Aiops: Real-world challenges and research innovations," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion)*, 2019, pp. 4–5.

[15] A. M. Y. Kuo, "Opportunities and challenges of cloud computing to improve health care services," *Journal of Medical Internet Research*, vol. 13, no. 3, p. e67, 2011. [Online]. Available: https://doi.org/10.2196/jmir.1867

[16] M. Shahin, M. A. Babar, and L. Zhu, "Continuous integration, delivery and deployment: A systematic review," *IEEE Access*, vol. 5, pp. 3909– 3943, 2017. [Online]. Available: https://doi.org/10.1109/ACCESS.2017. 2685629