Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

AI-Driven Incident Management for Distributed Cloud Systems: Detection, Mitigation, and Root Cause Automation

Harpreet Paramjeet Singh

Microsoft Corp., USA

ARTICLE INFO

Received: 19 Jan 2026

Revised: 23 Jan 2026

Accepted: 02 Feb 2026

ABSTRACT

Artificial intelligence for IT operations signifies a paradigmatic shift in managing hyperscale cloud distributions because manual incident management strategies fail to scale. This rises from the growing complexity of service dependencies, increasingly high numbers of alerts, and failure propagation patterns. Multiple learning anomaly detection strategies use self-adjusting thresholds and combined signal analysis to isolate occurrences of operation beyond the norm on an operational timeline. Meanwhile, intelligent alert consolidation strategies use intelligence to group alerts based on commonalities. Autopsy-style diagnosis uses causal analysis along with large language models to synthesize incident information from a data mashup of various telemetry data. Meanwhile, predictive repair uses time-series forecasting along with reinforcement learning to predict repair strategies in the form of proactive repair before the materialization of operational impacts. Similarly, incident lookup uses the same approach to quickly restore operations by accessing organizational memory. Of course, the convergence of the approaches makes possible a fully closed-loop automated system, where detection and repair occur independently for appropriately modeled operational failure modes. A satisfactory evaluation framework focuses upon the detection accuracy of the system, operational efficiency gains, and reduction of cognitive loads to show significant improvement in mean time to detection and mean time to repair while maintaining human oversight thresholds for high-risk operational scenarios.

Keywords: AIOps, Anomaly Detection, Automated Root Cause Analysis, Predictive Mitigation, Incident Management

1. Introduction: Challenges in Hyperscale Incident Management

However, it has been observed that current cloud systems running in a distributed fashion handle a huge volume of services, and it has been found that each cloud deployment contains a huge number of microservices, making it challenging to manage them centrally due to their massive size spread across geographically scattered data centers. Moreover, dependency graphs found in cloud production systems are observed to be quite deep, leading to complex propagation paths for failures, where degradation in one cloud component results in rapid propagation in dependent cloud services due to their interconnected nature, and degradation in one cloud software layer causes cascading failures in multiple software layers [1].

Alert volumes are a significant challenge in modern cloud systems. The classical method of issuing alerts based on thresholds generates high levels of noise in the alerts. This means that the operators receive high numbers of false alerts to those that require actual attention. Alert fatigue has a

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

straightforward correlation with the mean time to detect in these systems. This is because significant alerts are masked by the noise patterns. Studies attest that decision-making performances are hampered by the high number of alerts in human operators. Environments in production frequently generate high numbers of alerts that are beyond the capabilities of human operators to handle [2].

Cascading failures in distributed systems have highly complex characteristics that cannot be analyzed linearly. For database clusters that see latency degradation, secondary services have shown timeout characteristics that experience complete availability loss instead of a corresponding performance degradation. Due to the complexities associated with these interdependencies, it is highly complex to identify the root causes, requiring the diagnosis of correlation patterns associated with multiple different streams simultaneously, such as infrastructure data, application logs, network flow data, and user experience signals. Manual diagnosis processes involve high complexity that consumes considerable time for critical events, during which the services continue to experience degradation for user populations. This reactive approach associated with present diagnosis processes creates limitations for achieving service-level objectives at hyperscale [3].

Being able to combine AIOps techniques with the strengths of large language models has the potential for revolutionizing automated incident response. Machine learning algorithms can analyze high-dimensional telemetry data online at speeds sufficient to isolate deviations that would be manually time-consuming. Predictive analysis also allows proactive fixing by taking remedial actions before the end user is affected by pre-failure indicators. The strengths of large language models include natural language support for incident data synthesis. The integration of these techniques has made possible the automation of end-to-end incident response using detection, diagnosis, and remediation without human intervention for known types of failures, leaving human intelligence for ambiguous instances [4].

2. Intelligent Detection & Alert Consolidation

Anomaly detection models used in distributed systems utilize unsupervised learning algorithms that learn to thrive in varying baselines without needing large training datasets. Machine learning models used for anomaly detection in cloud systems make use of statistical models and neural network designs that focus on detecting anomalies within the operational patterns of systems. Such models take multidimensional streams of information, such as system metrics, performance, and resource usage patterns, to derive baselines to learn to thrive in varying workload characteristics. However, the core difficulty in this anomaly detection system is to learn to separate normal workload differences that might otherwise be identified as anomalies when a system degrades [3].

Dynamic threshold adaptation mechanisms have shown a major improvement over static threshold setting mechanisms, which often produce a high number of random false positives. Adaptivity techniques adjust the threshold values dynamically using the historic data windows to account for changes in capacity and usage patterns. Time-series analysis techniques decompose the signals to identify components such as the trend, seasonal variation, and residuals of the signal, which indicate anomalies. Time-series analysis algorithms have shown the ability to adapt to the characteristics of normal behavior, allowing the anomaly detection algorithms to detect the significance of the anomalies while incorporating the natural variations of the normal behavior [5].

Multi-dimensional architectures process signals from several types of telemetry data, including telemetry related to infrastructure, application performance, distributed tracing, and synthetic monitoring feedback. Each type of signal has different statistical patterns that need different processing. Metric streams come at a regular time interval and have high cardinality because each service can produce many unique metric time series. The log stream has a far higher volume, with active services log events that produce a voluminous number of log entries that can produce

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

actionable signals using streaming aggregation and sampling techniques to avoid overwhelming storage and processing systems while providing a complete view of system conditions along various operating dimensions [6].

Alert consolidation using intelligent grouping algorithms tackles the problem of alert proliferation by understanding the relationships among related anomalous signals. In cases where there are multiple anomalous signals from different components within temporal correlation windows, grouping algorithms form incident clusters, thereby not generating alerts for individual signals. Temporal and spatial relationships among anomalous events modeled using graph algorithms for cluster formation take into consideration the topology of the services and their past occurrence frequencies. Grouping algorithms greatly help in the reduction of alerts given to human analysts to process by taking individual alerts and generating groups, thereby allowing quick diagnosis due to the immediate understanding of the impact of the events by the human operator, rather than the operator taking time to investigate the different affected components. Cross-service dependency monitoring preserves the real-time topology of the services' communication dependencies, thereby allowing topology-based detection systems to generate alerts with their contexts by propagating the dependency relationships among the services to identify the affected services that may encounter impact due to the anomalous events in other services [4].

Detection Approach	Threshold Type	False Positive Reduction	Alert Consolidation	Processing Capability
Traditional Monitoring	Static thresholds	Baseline	Individual alerts	Limited manual processing
Adaptive ML Detection	Dynamic baselines	Substantial reduction	Minimal grouping	High-cardinality streams
Intelligent Grouping	Context-aware	Major reduction	Cluster-based	Multi-dimensional signals
Topology-Aware Systems	Service mesh integrated	Optimized filtering	Dependency-based	Real-time correlation

Table 1: Anomaly Detection and Alert Management Performance [3-5]

3. Automated Diagnosis & Contextual Synthesis

Correlation engines that deal with multi-signal pattern recognition are statistical and machine learning-based approaches that look for correlations between abnormal behavior observations from varied telemetry signals. Such approaches focus on time-based observations that can segregate symptom observations from root cause observations in complex cascaded observations. Causal inference is the process of analyzing the flow of information from time series observations and building models that statistically determine the cause and effect of varied signals. The direction of causation of anomaly between varied services is analyzed on the basis of cross-correlations and the flow of information between potentially related signals [5].

Large language model integration makes it possible to automatically generate the context of incidents based on a variety of sources of disparate data into a narrative form of natural language. Modern developments in the domain of natural language processing and deep learning made it possible for the system to process structured inputs such as anomaly data, the latest log records with error messages, distributed trace examples illustrating the pattern of request traces, and the history of past incidents with descriptions of similar events. The language models are capable of preparing the diagnosis

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

summary with the symptoms of the issue, possibilities of the root cause correlated with the results of the correlation analysis, and steps for further investigation that can be used for mitigation. These models use generation with the help of knowledge bases of past incidents to refer to the generated content based on the verified knowledge of the system, instead of referring to speculative information [6].

These multi-modal data ingestion systems address the structured and unstructured data—such as text—of the system's telemetry and logs. The structured data involves time-series normalization and the extraction of features based on the calculation of percentiles and rate of change values. The unstructured logs are processed using natural language processing techniques like tokenization and named-entity recognition to identify service and resource names. These named-entity recognitions help to form the embeddings. These embeddings help to perform a similarity search and allow a system to search and identify previously similar patterns in the logs that have a known cause. A large amount of raw system telemetry in a production system calls for a distributed system to process the significant volume of events [9].

Automated diagnostic processes utilize the knowledge base of past incident experiences that are stored in a structured format consisting of past instances of specific symptoms, diagnosed root causes, mitigations applied, and outcome measures. Given the occurrence of novel instances of anomalies, similarity search algorithms identify similar past instances by computing distances between similarity vector features for affected components, metric deviation profiles, and time features. The past instances supply diagnostic blueprints for the automation process, indicating the queries that need to be executed against the monitoring infrastructure or focused components for investigation. Case-based reasoning techniques offer high levels of diagnostic accuracy for familiar types of failures that have past instances, meaning that the automatic systems are capable of pinpointing the root cause for routine instances without any human involvement. Real-time impact assessment combines user experience signals with infrastructure monitoring, enabling the computation of incident severity in business terms, correlating past infrastructural anomaly instances with corresponding user journey signals for affected workflows, and estimating the population affected by the workflows [10].

Diagnostic Component	Data Input Type	Processing Method	Knowledge Source	Output Format
Correlation Engines	Time-series metrics	Causal inference	Statistical models	Causation graphs
LLM Synthesizers	Multi-modal telemetry	NLP generation	Historical incidents	Natural language summaries
Log Analyzers	Unstructured text	Tokenization & NER	Pattern embeddings	Similarity matches
Case-Based Reasoning	Incident vectors	Distance computation	Knowledge base	Diagnostic blueprints

Table 2: Diagnostic Automation Components and Capabilities [5, 6, 9, 10]

4. Predictive Mitigation & Autonomous Response

Fault prediction models in fault-forecast approaches utilize time series models to forecast future potential failures based on their likely onset ahead of service impact experienced by end users. These models make predictions based on early warning signs such as systematic trending towards resource depletion, nearing capacity limits, and early phases in elevated levels of errors forecasted ahead of

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

total failures. Predictive maintenance in a cloud setup uses actual experiences in system operations to develop models to predict signs associated with performance degradation based on systematic observations. Neural networks, such as recurrent networks, are used to learn how to make predictions based on behavior observations in system performance over observed history, predicting future states based on present and past states [7].

The capacity plan integration process makes it possible to perform proactive resource allocation according to patterns in expected demands and growth rates that are observed to ensure the efficiency of plans in the long term. The forecasting method considers historical patterns with seasonality in time to provide data on resource allocation in the long term to prevent demands that surpass the initially planned capacity, initiating resource scale operations in resource launches, database capacity allocation, and network bandwidth reservations to counter capacity resource violations rather than treating decreasing performance in resource services visible in user metrics [8]. Proactive capacity plans in resource management show a significant decrease in resource-related incidents compared to reactive resource capacity plans in treating resource performance observed in user metrics [8].

Reinforcement learning methods for choosing mitigation strategies have modeled incident response as sequential decision problems, with decisions leading to consequences or outcomes that come with certain costs. The reinforcement learning model recognizes the present system state, which encompasses telemetry data, service health, and past experiences, and chooses decisions using available remediation strategies, which might involve service retries, resource scaling, activation of backup systems, or traffic directioning. The learning agent gets rewards depending on the success rate and resource utilization. The learning can be achieved by incident replay learning, learning past events with defined decisions and their consequences in historical incidents, as well as experimental learning in test environments [8].

The AUTO REMED stands for "remediation action execution," which follows safe and proven intervention procedures that adjust either configurations or resource allocations of pre-authorized types of change. Typical examples of AUTO REMEDs include service restarts that help resolve issues related to state corruption in a transient state, traffic rerouting that helps redirect loads away from problematically functioning instances, resource scaling that adds resources to provide greater capacity about increased demand, config rollbacks that reverse changes recently deployed that are causing regression issues, and cache invalidations that help remove stale data resulting in sending out incorrect responses. Each procedure is risk-assessed; high-risk procedures require human approval, but low-risk procedures are automatically executed. Resiliency in operation improved through predictive intervention shows that significant reductions in noticeable effects are achieved when intervention takes place ahead of noticeable symptom manifestation. This proactive operation prevents cascading failures that are potentially generated by amplification of early degradations in dependent services that are amplified through other services on which they depend. Scaling prevents potential effects on users by ensuring that resources are provided ahead of actual demand [7].

Mitigation Approach	Prediction Method	Intervention Timing	Action Type	Failure Prevention
Fault Forecasting	Time-series LSTM	Pre-symptom warning	Predictive alerts	Resource exhaustion
Capacity Planning	Seasonal forecasting	Hours ahead	Proactive scaling	Capacity violations
RL-Based Selection	Sequential decisions	State-responsive	Optimal strategies	Multi-failure types
Automated Remediation	Risk-assessed execution	Immediate response	Config adjustments	Cascading failures

Table 3: Predictive Mitigation Strategy Framework [7, 8]

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

5. Generative Root Cause Analysis

Causal correlation root cause localization uses causative inference methodologies to separate symptoms from actual root causes within complex failure chains. For multiple services experiencing simultaneous anomalies, the determination of causative direction is beyond the realm of simple correlation measurement. Advanced statistical techniques measure directed information flow between time series signals, picking those signals that possess predictive value for others. Such methodologies examine time-differenced relationships for potentially relevant signals, building directed graphs whose links denote likely causative paths with corresponding confidence values. Actual root causes are pinpointed within nodes having multiple downstream effects but few, if any, upstream causes, thus condensing the search scope from multiple anomalous signals to merely several likely root causes [9].

The large-scale architectures in report generation, which analyze root cause, provide a natural language processing generation of diagnostic information from multiple sources, which form a structured format of incident description related to failure mechanisms in natural language form. The process involves correlation of analytical results that point towards possible occurrences of root cause, log information relevant to errors, summarized deviations in metric performances related to magnitude and duration, topology information related to dependencies, and previous relevant incidents in knowledge repositories. The result is in a structured form that involves summary reports related to incidents, which include a description of what failed and when, symptoms in detail related to incidents, root cause analyses, and contributors that pertain to conditions or new changes related to failures in increased likelihood [10].

Incident recall systems use semantic searches based on embeddings to find similar incidents from the organizational knowledge base. The historical incidents are modeled as vector embeddings through transformer models trained on incident attributes such as affected services, symptom expressions, root causes, and actions. When there are new incidents, their attributes are modeled as embeddings, and the most similar incidents are then searched. The similar incidents are important in incident diagnosis, as the new incidents that match historical ones and are of higher similarity measures imply that the root causes and mitigations of the historical ones are of high probability in the current incident context. Systems in production environments show that similar incident identification can improve diagnosis time when there are repeat patterns of failures [11].

Groundedness validation techniques will ensure that the diagnostic text generated will never become factually inaccurate and will be supported by actual telemetry evidence, as opposed to generating speculative text. Systematic validation of generated text helps to attribute claims to actual evidence, such as metric values, log records, or historical events retrieved based on the evidence provided. Claims that are unsupported by evidence will be labeled as such, and generation parameters will be modified to become even more cautious in generating text, giving higher priority to statements supported by strong evidence as opposed to generating explanations supported by speculative information. Attribution systems will make use of inference models, which ensure that logical connections between evidence statements and assertions are valid. Feedback loops will allow systems to continuously learn, which will increase diagnostic system accuracy as experts continue to validate, correct, and enhance automatically generated diagnostic text after an incident has been resolved. Engineers will submit feedback in the form of correctness feedback for automatically identified root causes, additional contributing issues identified through manual investigation, and effectiveness feedback for automatically suggested mitigations after an incident has been resolved, as feedback to enhance system performance, as mentioned in [12].

Journal of Information Systems Engineering and Management
2026, 11(1s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

RCA Component	Analysis Technique	Information Source	Validation Method	Accuracy Improvement
Causation Localization	Directed graphs	Multi-signal correlation	Confidence scoring	Root cause isolation
Narrative Generation	LLM synthesis	Structured incident data	Groundedness checks	Contextual accuracy
Similar Incident Retrieval	Embedding search	Historical knowledge base	Similarity scoring	Pattern recognition
Continuous Learning	Feedback integration	Post-incident reviews	Expert validation	Progressive refinement

Table 4: Root Cause Analysis Generation Pipeline [9-12]

6. Evaluation Framework and Future Directions

IS-GG Detection performance metrics reflect the quality of anomaly detection solutions by precision, measuring the ratio of alerts raised per true incident, recall, measuring the ratio of identified instances of true incidents, and the rate of false positives, measuring the ratio of cases of normal operation mistakenly identified as anomalies. Production-level anomaly detection solutions aimed at optimizing service availability requirements handle the trade-off between being responsive to real anomalies and being minimally affected by alert fatigue from excessive notification of normal operating conditions. Critical incident recall rates guarantee the identification of real high-impact events, but lower-impact anomalies may have irregular levels of successful detection. A limiting factor in detecting anomalies is keeping acceptable levels of false positives per service being watched [11].

Operational efficiency metrics can be used to measure business value for AIOps solution deployments based on mean time to detection, mean time to resolution, and volume of alerts presented to human analysts. Manual baseline detection capabilities have a large detection latency for critical events, as human analysts take notice of anomalies in monitoring consoles or user complaints for anomalous behavior. Automated detection capabilities have low detection latencies as monitoring is done instantaneously with sub-minute detection response. Resolution latencies for critical events have seen a large reduction as manual analyses and resolutions take a prolonged period for events in complex distributed systems. Automated analyses and resolution tools have small resolution latencies for known events for which automated resolutions can be done. Alert volumes have become a critical aspect for efficient operations as smart alert correlation and groupings have decreased alert volumes faced by human analysts, lessening them from manual baseline alert volumes to manageable groupings of alerts for events [12].

By cognitive load reduction measurement, the amount of mental effort involved in managing incidents is measured based on the number of data sources for which manual research is needed in the diagnosis stage, the number of decisions that demand human judgment, and the number of incidents that can be handled simultaneously by the same operator. By the introduction of automation, the number of heterogeneous sources for the monitor to consult is decreased as it enables the use of common operating interfaces, the number of decisions for which the system has to consult human judgment is decreased as it includes decisions supported by recommendations generated with the help of artificial intelligence, and the number of incidents that can be simultaneously handled by the system with the help of human operators when it handles routine tasks is increased.

The cognitive load of human operators is measured based on the results of on-call engineer surveys on cognitive load for assessing the decrease in mental effort, temporal demand, and frustration due to the automation of detection, diagnosis, and routine repair of incidents without human intervention for

critical judgment-demanding incidents involving human experts. Agentic autonomous remediation with multi-step reasoning is the next generation past one-click automated fixes, where systems utilize reasoning engines to develop multi-step solve-the-problem strategies, which may include seeking supplemental diagnostic data, hypothesizing based on possible root causes, testing hypotheses using controlled experiments or config changes, and adjusting strategy based on outcomes if the first steps are unsuccessful. Cross-cloud multi-dependency root cause analysis recognizes a reality where cloud-based applications now exist on multiple clouds and on-premises systems, causing dependencies to "stretch across organizational and technical divides," where looking ahead, "new frameworks will use federated protocols for sharing anonymized dependency data and alert information, allowing cross-organizational causation analysis while keeping proprietary information demarcations intact," where safety guardrails and human-in-the-loop oversight "prevent AI systems from taking actions with risky profiles w/o human oversight and approvals," [11].

Conclusion

The inclusion of artificial intelligence capabilities within incident response systems has brought about a paradigm shift in the best practices of operational reliability for hyperscale distributed cloud systems. Machine learning algorithms for anomaly detection with adaptive thresholding and selective intelligent alert consolidation provide relief from the problem of alert fatigue by efficiently filtering noise while being highly sensitive to real service degradation. Automated diagnostic pipelines incorporating causal inference techniques, synthesis, and large language model synthesis enable quick root cause isolation by efficiently evaluating multi-modal time series, amalgamated with accumulated knowledge graphs of past incidents. Strategic mitigation through time series forecasting, futures forecasting, and reinforcement learning methods for strategy selection enables prescriptive action before failures start affecting users, another drastic shift from reactive operational methodologies. Correlation-driven root cause localization, embedding-driven retrieval of similar past incidents, and validation for groundedness against the real world ensure factually accurate, actionable, and reliable diagnostic storytelling. Future developments towards autonomous remediation with multi-step reasoning, cloud-wise dependencies via federated cloud telemetry, with appropriate choice of safety mechanisms, are expected to promote reliability, utilization, and continued human involvement for complex judgment-intensive scenarios.

References

[1] Fotios Voutsas, John Violos, and Aris Leivadeas, "Mitigating Alert Fatigue in Cloud Monitoring Systems: A Machine Learning Perspective," Computer Networks, Volume 250, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S138912862400375X

[2] Yingnong Dang et al., "AIOps: Real-World Challenges and Research Innovations," IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8802836

[3] Camilo Quiroz-Vázquez, "Anomaly detection in machine learning: Finding outliers for optimization of business functions," IBM Think Topics. [Online]. Available: https://www.ibm.com/think/topics/machine-learning-for-anomaly-detection

[4] Bronagh Lanigan et al., "Alert correlation for intelligent threat detection and response," Intelligent Systems with Applications, Volume 28, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2667305325001322

[5] Nina Marwede et al., "Automatic Failure Diagnosis Support in Distributed Large-Scale Software Systems Based on Timing Behavior Anomaly Correlation," 13th European Conference on Software

Maintenance and Reengineering, 2009. [Online]. Available: https://ieeexplore.ieee.org/document/4812738

[6] Shilin He et al., "A Survey on Automated Log Analysis for Reliability Engineering," arXiv:2009.07237, 2021. [Online]. Available:https://arxiv.org/abs/2009.07237

[7] Dinh Dai Vu et al., "Deep Learning-based fault prediction in cloud system," International Conference on Information and Communication Technology Convergence (ICTC), 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9620802

[8] RamaKrishnaReddy Muthyam, "Self-Healing Systems: Reinforcement Learning For Cloud Resilience," International Journal of Environmental Sciences, 2025. [Online]. Available: https://theaspd.com/index.php/ijes/article/view/11368/8141

[9] Jacopo Soldani and Antonio Brogi, "Anomaly Detection and Failure Root Cause Analysis in (Micro)Service-Based Cloud Applications: A Survey," arXiv:2105.12378, 2021. [Online]. Available: https://arxiv.org/abs/2105.12378

[10] Yitao Yang et al., "AidAI: Automated Incident Diagnosis for AI Workloads in the Cloud," arXiv:2506.01481v1, 2025. [Online]. Available: https://arxiv.org/pdf/2506.01481

[11] Amol D. Vibhute et al., "Network anomaly detection and performance evaluation of Convolutional Neural Networks on UNSW-NB15 dataset," Procedia Computer Science, Volume 235, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050924008871

[12] Hyunjong Shin and Vittaldas V. Prabhu, "Evaluating Impact of AI on Cognitive Load of Technicians During Diagnosis Tasks in Maintenance," Springer, 2018. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-99707-0_4