IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 28, Issue 5, Ser. I (Sep. – Oct. 2026), PP 47-61
www.iosrjournals.org

LLM-Based Root Cause Analysis and Automated Incident Response for Large-Scale Distributed Commerce Systems

Sohail Sayed; Nauman Sayed
California State University Fullerton

Abstract *— Production incidents in large-scale distributed commerce systems are expensive and inevitable: one hour of downtime is estimated to cost Amazon US$100 million on major shopping days (Ahmed et al., 2023), Amazon lost US$66,240 per minute of server downtime in early 2016, and the average US data center outage cost US$740,357 per event at roughly US$7,900 per minute (Nguyễn et al., 2018). Yet incident management remains dominated by manual toil: on-call engineers spend multiple rounds of back-and-forth communication to identify root causes and mitigation steps (Ahmed et al., 2023). This paper argues that large language models are the missing reasoning layer that converts incident management from a human-bottlenecked diagnostic process into an automated, governed loop: LLMs supply the causal reasoning, knowledge bases supply the evidence, and deterministic execution supplies the safety. We propose IRIS-SC, a closed-loop framework coupling (i) multi-source anomaly detection over metrics, logs, and traces, (ii) alert-type-matched, evidence-grounded candidate generation, (iii) LLM-based root cause analysis with knowledge grounding and provenance, (iv) automated incident response blending playbooks with learned recovery policies, (v) continual Ops-domain benchmarking, and (vi) guardrail governance with human-in-the-loop escalation. The framework consolidates the reported evidence envelope: an LLM-driven cloud-native observability system reduces mean time to resolution by 84.2% relative to rule-based methods, achieves 95% F1 in anomaly detection, and autonomously resolves 91% of common incidents (Wang et al., 2025); an LLM-plus-RAG self-healing architecture achieves up to 82% MTTR reduction with an 89.5% autonomous success rate (Sharma & Jain, 2026); RCACopilot automates root cause analysis on a year of Microsoft incidents with accuracy up to 0.766, its diagnostic component in production for over four years (Chen et al., 2024); a production AIOps observability framework improves MTTD by 68%, MTTR by 54%, and availability by 12% (Singh, 2025); industry syntheses report 40–60% MTTR reductions for LLM-powered incident assistants and a 30–70% reduction in incident response time across implementing organizations (Damarched, 2026); a graph-augmented LLM agentic workflow improves root-cause-identification accuracy by up to 42.22% over state-of-the-art methods (Tian et al., 2025); and OpsEval, a 7,334-question benchmark across 24 LLMs, demonstrates a 0.9185 human-agreement coefficient for Ops-domain evaluation (Liu et al., 2025). The paper argues that grounded diagnosis — not fluent generation — is what separates LLM assistants from LLM operators, and that automation rates near 90% are achievable only when every reasoning step is bound to evidence and every action passes governance.*

Keywords *— Root cause analysis, incident management, large language models, AIOps, microservices, cloud computing, anomaly detection, automated incident response, observability, self-healing systems.*

Date of Submission: 03-09-2026 Date of Acceptance: 13-09-2026

I. INTRODUCTION

Large-scale distributed commerce platforms — marketplaces, payment gateways, order orchestration, inventory and pricing engines — are microservice architectures whose failures propagate across service boundaries within seconds. The stakes are quantified in the incident literature. Cloud outages and performance degradations are expensive in service-level-agreement penalties and engineering effort, and one hour of downtime is estimated to cost Amazon US$100 million on major shopping days (Ahmed et al., 2023). Ponemon-style accounting puts the average unplanned data-center outage at US$740,357 per event — up 46% from US$505,502 in 2010 — at roughly US$7,900 per minute, with online businesses suffering the most severe revenue losses; Amazon alone lost US$66,240 per minute over approximately 15 minutes of server downtime in early 2016 (Nguyễn et al., 2018). Facebook lost US$99.75 million in revenue during its October 7, 2021 outage of more than six hours (Xie & Li, 2023). At the enterprise level, the average company experiences 14–18 hours of downtime annually at costs reaching $14,056 per minute (Damarched, 2026).

Traditional incident management fails at the two moments that matter most. First, diagnosis: identifying the root cause of a production incident typically requires significant time and communication before engineers converge (Ahmed et al., 2023), and manual root cause analysis and static troubleshooting guides struggle with the complexity of modern distributed systems (Damarched, 2026). Second, response: the industry remains dominated

by reactive remediation, with LLM-powered automation reporting 40–60% MTTR reductions where it has been deployed (Damarched, 2026). The result is a system in which detection is automated but action is not — the same prediction-without-action gap established for failure prediction in the portfolio's platform paper.

Large language models change the diagnosis equation. LLMs are increasingly explored as general-purpose assistants for infrastructure operations — automating querying data, analyzing logs, and suggesting fixes — and the ambitious version of the problem is fully automating RCA, where the LLM must collect information, reason about it, and interact with the environment to detect, localize, and resolve issues (Cornacchia et al., 2025). Production evidence is decisive on magnitude: a domain-adapted LLM over OpenTelemetry telemetry reduces MTTR by 84.2% versus rule-based methods, achieves 95% F1 in anomaly detection, and automates 91% of common incidents without human intervention (Wang et al., 2025); RCACopilot has run its diagnostic-information component in Microsoft production for over four years while predicting root-cause categories with accuracy up to 0.766 (Chen et al., 2024). The conversation has shifted from whether LLMs belong in incident management to how to make them safe there.

This paper is the tenth in a research program on resilient, AI-driven commerce. Paper 6 established failure prediction and self-healing as the platform's closed-loop control problem; Paper 7 built retrieval-grounded real-time intelligence; Paper 8 formalized multi-agent autonomy with consensus; Paper 9 made risk intelligence explainable via knowledge graphs. This paper completes the platform thread: it takes the prediction layer of Paper 6, the retrieval grounding of Paper 7, the agent orchestration of Paper 8, and the evidence discipline of Paper 9, and formalizes the two capabilities that convert a self-healing framework into a self-healing operation — root cause analysis and automated incident response, with evaluation as a first-class engineering artifact.

The contributions of this paper are:

- A formalization of incident management for distributed commerce systems as a grounded diagnosis-and-response loop, with explicit evidence-boundness, automation, and safety constraints, and a stochastic downtime-cost model (Sections III–IV).
- IRIS-SC, a framework coupling multi-source anomaly detection, alert-type-matched evidence generation, LLM-based RCA with provenance, automated response blending playbooks and learned policies, continual Ops benchmarking, and governance.
- A consolidated analysis of reported performance — downtime economics, MTTD/MTTR reductions, automation rates, diagnosis-accuracy gains, and benchmark scale — with worked calculations and clearly flagged illustrative extensions (Sections V–VI).
- A research agenda on grounded diagnosis versus hallucinated reasoning, Ops-domain evaluation standards, and governed automation (Sections VII–VIII).

II. RELATED WORK

A. The economics and structure of incidents

Incident cost is the load-bearing fact of this research area. Production incidents adversely impact customers and are expensive in terms of SLA penalties and engineering effort (Ahmed et al., 2023). The empirical base is now quantified across three levels: hyperscale commerce (US$100 million per hour on major shopping days at Amazon (Ahmed et al., 2023); US$66,240 per minute in 2016 (Nguyễn et al., 2018)); data centers (average US$740,357 per unplanned outage at ~US$7,900/minute, up 46% from US$505,502 in 2010 (Nguyễn et al., 2018); ~$8,000 per minute with an average incident length of 95 minutes (Xie & Li, 2023)); and the enterprise average of 14–18 hours of annual downtime at costs reaching $14,056 per minute (Damarched, 2026). The formal structure of these costs matters for design: modeling aggregate annual downtime cost S as a compound Poisson process yields mean $E[S] = \lambda \cdot c$ and variance $Var[S] = \lambda \cdot c^2$, so variance-weighted, shorter outages should be preferred when outage cost is proportional to duration (Wang & Franke, 2020), and empirical recovery-time data from more than 1,800 incidents (over 11,000 hours of recorded downtime in a large Nordic bank) fit a log-normal distribution, motivating MTTR modeling as a log-normal random variable rather than a point estimate (Franke et al., 2014). In commerce specifically, transactional bottlenecks, network downtimes, and server anomalies directly degrade consumer experience, making proactive maintenance a strategic capability (Mayo et al., 2023). These economics drive every design decision in this paper: the value of automation is measured in minutes, and the value of correctness in customer trust.

B. From log anomaly detection to root cause localization

The detection substrate is mature. Deep learning transformed log analysis: DeepLog models logs as natural-language sequences with an LSTM, learns normal patterns, supports online updates, and constructs workflows for root-cause analysis (Du et al., 2017); Transformer-based Loader captures global dependencies with top-p candidate selection (Xiao et al., 2023); and deep-learning log detectors demonstrate superior performance over conventional ML (He et al., 2023; Kavlak & Sezen, 2026; Landauer et al., 2023; Lv & Min, 2023; Shi et al., 2024). LLMs now extend this line: LogReasoner adds coarse-to-fine reasoning enhancement that mirrors expert

cognition, achieving state-of-the-art performance on four distinct log analysis tasks with open-source models including Qwen-2.5 and Llama-3 (Ma et al., 2026); ScalaLog builds failure diagnosis on RAG with LLM-based summarization, sample augmentation, and chain-of-thought prompting, improving accuracy without training or log parsing (Zhang et al., 2025); and LoFI extracts fault-indicating descriptions and parameters directly from logs, achieving an absolute F1 improvement of 25.8–37.9 over the best baseline and deploying successfully at CloudA (Huang et al., 2024). The production economics of LLM log analysis are now documented: a tool in production since March 2024, scaled across 70 software products and more than 2,000 tickets, saved over 300 man-hours and approximately $15,444 per month in manpower costs (Gupta et al., 2026).

The harder problem is localization. Root cause localization in microservices is arduous because frequent updates produce large-scale data and complex dependencies that defeat predefined rules and manual experience (Fu et al., 2025). Two design lessons dominate. First, multi-source data matters — Eadro integrates traces, logs, and KPIs with multi-task learning, outperforming state-of-the-art approaches on two benchmark microservices (Lee et al., 2023), and tracing-based NLP frameworks generalize across applications without prior knowledge (Kohyarnejadfard et al., 2022). Second, method choice must be evidence-driven: a controlled evaluation of 24 RCA methods (20 metric-based, two trace-based, two multi-source) on a best-practice LLM inference deployment under failure injection finds that multi-source approaches achieve the highest accuracy, metric-based methods show fault-type-dependent performance, and trace-based methods largely fail — evidence that existing RCA tools do not generalize automatically and that multi-source design is the robust default (Scheinert et al., 2026). Scale must also be production-grade: LogRCA identifies a minimal set of log lines describing a root cause on a production dataset of 44.3 million log lines with 80 expert-labeled failures, consistently outperforming deep-learning and statistical baselines (Wittkopp et al., 2024). The failure-prediction layers feeding these systems are equally mature — ensemble voting at 97.97% accuracy on the Google 2019 Cluster Sample (Padhy et al., 2026), multilayer voting at 99.83% detection and 99.97% typing (Elkaradwy et al., 2025), and generic cross-platform models near-perfect on Grid5000 (Jagannathan et al., 2025). For this paper's purposes, detection is assumed available; the open problem is what happens after the alert.

C. LLMs for incident diagnosis and root cause analysis

The Microsoft Research line established the agenda: applying GPT-3-class models to find root causes and recommend mitigation steps on real cloud incidents, with a rigorous large-scale evaluation over 44,340 incidents from 1,759 Microsoft services and human validation by the original incident owners (Ahmed et al., 2023). RCACopilot industrialized the pattern: it matches incoming incidents to incident handlers based on alert types, aggregates critical runtime diagnostic information, predicts the root-cause category, and provides an explanatory narrative, achieving RCA accuracy up to 0.766 on a year of real Microsoft incidents, with its diagnostic-information collection component in production for over four years (Chen et al., 2024). The broader state of the art has since accelerated. An LLM-driven observability framework integrates OpenTelemetry-based telemetry collection with a domain-adapted LLM performing multimodal analysis over metrics, logs, and traces; fine-tuned on operational data with chain-of-thought reasoning, it generates explainable root-cause hypotheses and actionable remediation plans, reducing MTTR by 84.2% versus rule-based methods, achieving 95% F1 in anomaly detection, and automating 91% of common incidents without human intervention (Wang et al., 2025). Graph-augmented agentic workflows push diagnosis quality further: GALA combines statistical causal inference with LLM-driven iterative reasoning, improving accuracy over state-of-the-art methods by up to 42.22% on an open benchmark while producing more causally sound and actionable diagnostic outputs (Tian et al., 2025). Model Context Protocol pipelines institutionalize engineer knowledge — schemas, queries, architecture, operational patterns — in knowledge bases, enabling automated multi-step diagnosis that mimics engineer behavior and converting expert tasks into tasks executable by less-experienced engineers (Gujarathi, 2026). LLM-plus-RAG autonomous healing (the ARCH framework) achieves up to 82% MTTR reduction and an 89.5% autonomous success rate (Sharma & Jain, 2026). The honest evaluation also exists systematic evaluation of multiple LLM agent architectures across incident scenarios exposes common failure modes — hallucinated reasoning paths and inefficient use of context — revealing both the promise and the limitations of current approaches (Cornacchia et al., 2025). Unified anomaly-detection-plus-RCA frameworks using ensembles of statistical and deep models contextualized through service dependency graphs with causal inference further reduce RCA latency and mean time to recovery in real deployments (Meyer et al., 2025). At the survey level, LLM4AIOps — 183 articles published between January 2020 and December 2024 — maps failure-data sources, task evolution, LLM-based methods, and evaluation methodologies across AIOps (Zhang et al., 2025).

D. Automated incident response and self-healing

Response automation converts diagnosis into action. AIOps frameworks for hyperscale clouds combine self-adjusting-threshold anomaly detection, intelligent alert consolidation, autopsy-style diagnosis with causal analysis and LLMs over telemetry mashups, predictive repair pairing time-series forecasting with reinforcement

learning, and incident lookup through organizational memory — a fully closed-loop system where detection and repair proceed with human oversight thresholds for high-risk scenarios (Singh, 2026). Incident-response automation frameworks document reduced downtime in IT service operations (Babatope et al., 2023). ML-based self-healing systems integrate log/metric anomaly detection, root-cause analysis, and proactive recovery with reinforcement-learning policy decisions in a continuous learning loop (Konda, 2021); FinTech deployments reduce downtime and resolution time across latency spikes, memory leaks, and node crashes (Singh, 2024); Kubernetes-native healing executes pod restarts, cordoning, rescheduling, rollback, and policy-driven scaling (Nunavath, 2025; USA & Sannareddy, 2025). The quantitative anchors are strong: an integrated predictive-plus-RL-plus-playbook system reduced MTTR by 85% (from 90 to 13.5 minutes) across 220 fault scenarios with uptime above 98%, reliability above 95%, and resource overhead under 10%, outperforming rule-based-only (60% reduction) and RL-only (50%) variants (Laheri, 2025); deep-reinforcement-learning cluster orchestration (AutoPilot-K8s) reduces average latency by up to 42% and improves SLA adherence by over 60% (Kripa, 2025); and a production-oriented AIOps observability framework reports MTTD improved by 68%, MTTR by 54%, and overall availability by 12% (Singh, 2025). In commerce-specific deployments, a zero-touch-reliability case study on a large-scale e-commerce platform shows self-healing technology reducing incident resolution times by more than 80% and slashing downtime (Allam, 2024), while AI-augmented CI/CD pipelines at a major retail corporation use LSTM networks and autoencoders to detect subtle degradations and knowledge-graph with causal-inference models to dramatically reduce incident resolution times (Baswareddy, 2025). LLM-powered self-healing interfaces for application support integrate telemetry aggregation, intelligent RCA, and automated runbook execution to minimize downtime while maintaining transparency (Singh, 2025). Chaos engineering provides the validation substrate: AI-driven fault injection improves fault-detection accuracy by 28% and cuts recovery time by 35% versus static methods (Gunawat et al., 2025), generative-AI chaos frameworks cut experiment setup time by 80%, increase scenario diversity by 71%, and cut fault-diagnosis time by 67% (Pandey et al., 2025), and LLM-powered chaos automates the full experiment cycle at low cost (Basiri et al., 2019; Kikuta et al., 2025).

E. Commerce-specific evidence

The commerce context sharpens both requirements and constraints. A case study of a prominent retail organization documents the failure profile this paper targets: transaction completion rates above industry averages, platform instability during peak traffic periods, incident resolution times significantly exceeding e-commerce industry benchmarks, and a reactive posture in which most problems were identified only after customer impact — with customer-support satisfaction below industry standards as the downstream symptom (Sachdeva, 2024). A production-oriented AIOps program for high-velocity e-commerce ("Protecting Revenue at Scale") advances observability into pre-detecting anomalies early enough to prevent customer impact, combining multi-layer health monitoring, two-stage thresholding, time-aware seasonal baselines, cross-stack delta localization, event-aware reasoning, and confidence-weighted recommendations — with human approvals deliberately retained for high blast-radius actions (Jayakumar & Chan, 2026). The design implication is direct: commerce incident systems must operate on seasonal, high-velocity baselines (flash sales, peak shopping days), must connect platform health to revenue, and must retain governed escalation for high-impact actions — exactly the properties IRIS-SC encodes.

F. Evaluation of LLM ops systems

The evaluation problem is now the field's rate limiter. OpsEval, a comprehensive benchmark with 7,334 multiple-choice and 1,736 QA questions across 24 LLMs under self-consistency, chain-of-thought, and in-context settings, proposes an Ops-specific QA evaluation method with a 0.9185 agreement coefficient with human experts, improving over BLEU and ROUGE by 0.4471 and 1.366 respectively (Liu et al., 2025). The pedagogical critique goes further: LLMs are not a silver bullet for operational challenges, and metrics such as pass@k conceal how failures compound across multi-step reasoning and tool use — motivating better-grounded evaluation metrics (Chen et al., 2025). Benchmarking must also be continual: a modular suite for networking operations argues that the absence of standardized benchmarks challenges progress tracking and comparison, evaluating GPT-4.1, Gemini 2.5-Pro, and Claude 3.7 Sonnet across over 100 test cases and two pipeline variants (Ioannis & Laurent, 2025; Protogeros & Vanbever, 2025). The adjacent supply chain side mirrors this: SupChain-Bench exposes substantial execution-reliability gaps in long-horizon tool orchestration (Guan et al., 2026), and CSCBench shows substantial degradation on the Variety axis of commodity reasoning (Cui et al., 2026).

G. Governance, explainability, and integration

Automation is licensed by governance, not by accuracy. Guardrail-centric fine-tuning aligns a quantized Qwen2.5-Coder-14B model to approximately fifty generalized behavioral guardrails with a deterministic Python enforcement layer, separating probabilistic reasoning from exact execution (Davenport, 2026); governed LLM optimization cuts unsafe decision outputs by 45% while sustaining drift-detection accuracy above 92% (Jingar,

2023); and agentic AI in adjacent security domains documents actions in a blockchain ledger for integrity and auditing, achieving better detection accuracy and shorter mitigation latency than rule-based, provenance-only, and RL-only baselines (Syed et al., 2025). In commerce-adjacent decision settings, human-specified optimization achieves statistically optimal and stable outcomes relative to fully automated pipelines (Wu, 2026), and OR-augmented agents outperform either approach alone, with human–AI teams outperforming humans or agents alone across more than 1,000 benchmark instances (Baek et al., 2026). Explainability requirements are established: predictive analytics must be interpretable or it cannot be integrated into risk decision processes (Baryannis et al., 2019; Iruku, 2025; R. et al., 2024); the same discipline applies to incident explanations, where directional grounding of generated narratives (99.6% consistency in the portfolio's maritime study) is the pattern IRIS-SC generalizes (Xue et al., 2026). At the system level, seven-agent frameworks detect, analyze, and respond to disruptions with F1 0.962–0.991 in 3.83 minutes at $0.0836 per disruption (AlMahri et al., 2026), and Agentic Control Towers integrate LLM agents, multi-agent RL, and digital twins into closed-loop autonomy with guardrails for safety, cost, and service (Awad & Alahmari, 2025).

H. Research gap

The literature separates diagnosis (RCA frameworks (Chen et al., 2024; Tian et al., 2025; Wang et al., 2025)), response (self-healing and playbook systems (Laheri, 2025; Sharma & Jain, 2026; Singh, 2025)), and evaluation (benchmarks (Liu et al., 2025; Protogeros & Vanbever, 2025)), and the strongest systems optimize one stage in isolation. What is missing is a single governed loop for commerce platforms in which detection, alert-type-matched evidence generation, LLM diagnosis, automated response, and continual benchmarking are jointly designed — where the explanation is bound to evidence by construction, the response is executed by deterministic components, and the whole is measured against Ops-domain standards rather than NLP metrics. This is the integration IRIS-SC proposes.

III. PROBLEM FORMULATION

A. System and telemetry model

Let the commerce platform be a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ of microservices with call-dependency edges. Each node emits multi-source telemetry — metrics, logs, and traces — the data sources whose joint modeling is documented as decisive for troubleshooting quality (Lee et al., 2023; Scheinert et al., 2026). Telemetry baselines are highly seasonal in commerce (peak shopping days, flash sales), so anomaly decisions must be time-aware (Jayakumar & Chan, 2026).

B. Incident and cost model

An incident manifests as anomalies across telemetry, propagates along dependency edges, and is resolved by an action $a \in \mathcal{A}$. Its lifecycle cost decomposes into detection time, diagnosis time, remediation time, and recurrence. Following the compound-Poisson accounting of enterprise downtime, the annual downtime cost S has mean $E[S] = \lambda \cdot c$ and variance $Var[S] = \lambda \cdot c^2$, where λ is the outage rate and c the mean per-outage cost (Wang & Franke, 2020); recovery times follow a log-normal distribution, so expected MTTR is a function of both detection and repair intervals (Franke et al., 2014).

C. Root cause analysis task

Given a telemetry window and an alerted set, the system must identify the root-cause set

$$\hat{c} = \arg\max_{c \subseteq \mathcal{V}} P(c|X_w, \mathcal{G}) \text{s.t.} |c| \leq k,$$

using multi-source evidence, alert-type matching, and causal inference over the dependency graph (Chen et al., 2024; Meyer et al., 2025), with the constraint that every element of $\hat{c}$ is supported by observed evidence — no hallucinated services (Cornacchia et al., 2025).

D. Automated response task

The response policy maps diagnosis to action: playbook execution where the failure pattern matches known cases — the 30-second fast path of integrated self-healing (Laheri, 2025) — and a learned recovery policy otherwise, the predictive-repair pattern pairing forecasting with reinforcement learning (Singh, 2026), subject to guardrails and human escalation (Jayakumar & Chan, 2026). The response quality metric is the autonomous success rate: the share of incidents resolved without human intervention (Sharma & Jain, 2026; Wang et al., 2025).

E. Objective

The framework optimizes

$$\min \underbrace{\mathbb{E}[\text{MTTD}]}_{\text{detection}} + \underbrace{\mathbb{E}[\text{MTTR}]}_{\text{diagnosis+remediation}} + \lambda(1 - \text{ASR}) + \mu \text{Risk}(a),$$

where ASR is the autonomous success rate and Risk(a) penalizes unsafe or non-compliant actions. This is the reward structure that makes automation maximally valuable ($100M-per-hour economics (Ahmed et al., 2023)) and maximally safe (guardrail-bounded actions (Davenport, 2026; Jingar, 2023)).

F. Metrics

Detection: accuracy/F1 of anomaly models (Wang et al., 2025). Diagnosis: root-cause-identification accuracy and time-to-root-cause (Jingar, 2023). Response: MTTR, autonomous success rate, downtime reduction (Damarched, 2026; Laheri, 2025). Economics: per-incident cost and latency (Gupta et al., 2026; Nguyễn et al., 2018). Quality-of-explanation: evidence-boundness and Ops-domain evaluation scores (Liu et al., 2025). Governance: unsafe-output rate and drift-detection accuracy (Jingar, 2023).

IV. PROPOSED FRAMEWORK: IRIS-SC

A. Overview

IRIS-SC couples six layers — detection, diagnosis, response, governance, evaluation, and learning — around the commerce-platform graph. Its outputs feed the portfolio's upstream layers (Paper 7's retrieval intelligence, Paper 8's multi-agent decision loop, Paper 9's explanation layer) with resolved-incident facts. Fig. 1 shows the architecture.

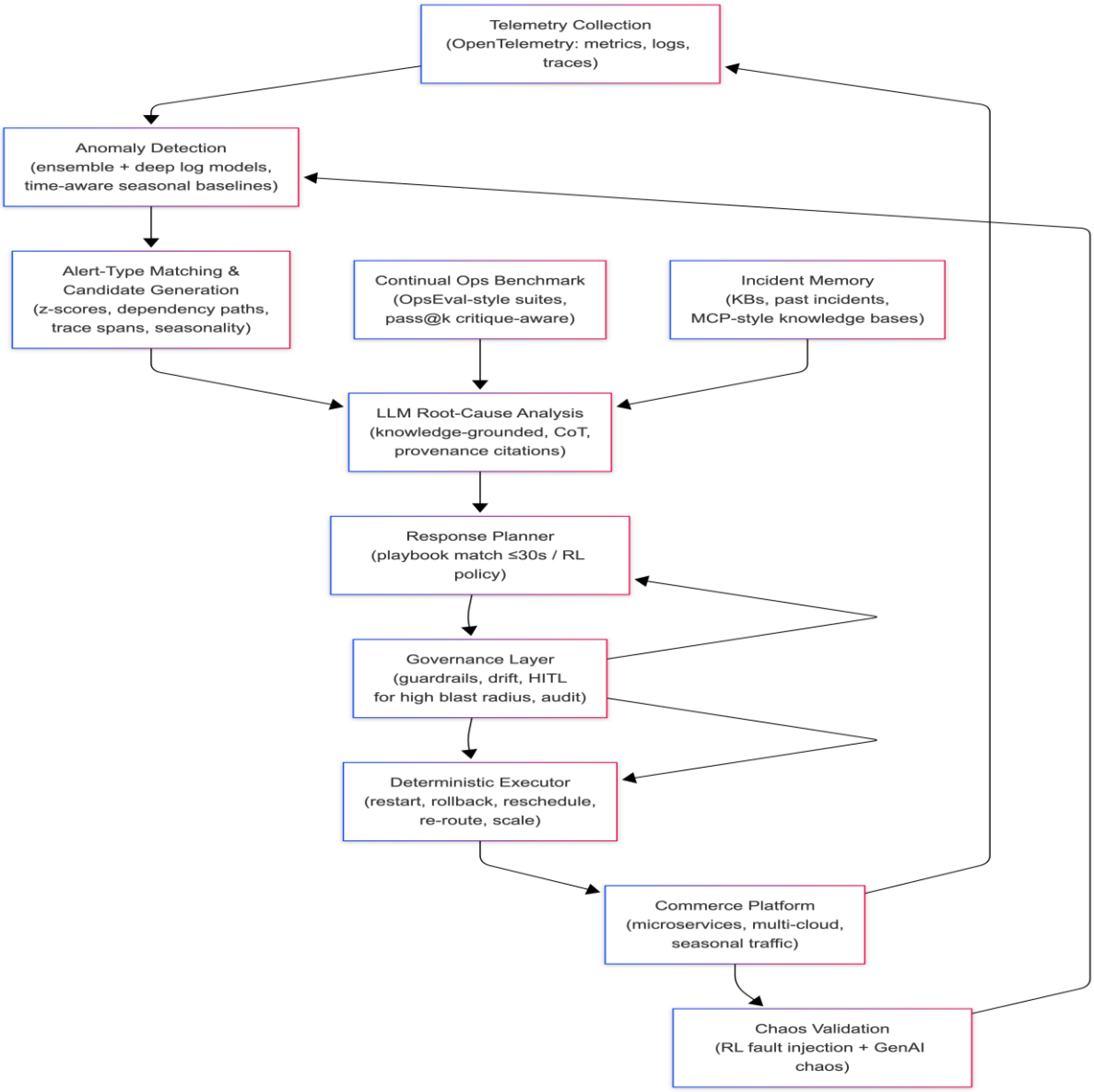

Fig. 1. IRIS-SC architecture

Telemetry collection feeds ensemble anomaly detection on time-aware seasonal baselines; alert-type matching generates grounded candidates; knowledge-grounded LLM RCA produces provenance-cited root causes; the response planner blends playbooks with learned policies; governance applies guardrails, drift detection, human escalation for high blast-radius actions, and audit; a deterministic executor acts on the platform; incident memory and continual Ops benchmarks condition every diagnosis; a chaos-validation loop retrains the system.

B. Detection layer: multi-source anomaly detection

An ensemble of statistical and deep models over metrics, logs, and traces, following the multi-source evidence base of Eadro (Lee et al., 2023) and the 24-method comparison that found multi-source approaches most accurate (Scheinert et al., 2026): voting-classifier and multilayer ensembles with documented accuracies of 97.97% (Padhy et al., 2026) and 99.83%/99.97% (Elkaradwy et al., 2025); LSTM-sequence log models in the DeepLog tradition (Du et al., 2017); Transformer heads in the Loader tradition (Xiao et al., 2023); and the LLM-based detectors LogReasoner (Ma et al., 2026), ScalaLog (Zhang et al., 2025), and LoFI (Huang et al., 2024). In commerce deployments, detection runs against two-stage thresholds and time-aware seasonal baselines so that Black-Friday-scale surges do not trigger false alarms (Jayakumar & Chan, 2026). The chaos-validation loop continuously refreshes the training distribution, the mechanism behind the reported 28% detection-accuracy gains under AI-driven fault injection (Gunawat et al., 2025).

C. Diagnosis layer: alert-type-matched, evidence-grounded llm rca

The RCA module first matches the incoming incident to an alert type — the RCACopilot pattern that routes incidents to appropriate handlers and aggregates critical runtime diagnostic information (Chen et al., 2024) — then transforms model-internal evidence (feature z-scores, dependency paths, trace spans) into structured prompts that constrain LLM reasoning to verifiable facts, the pattern demonstrated at 84.2% MTTR reduction (Wang et al., 2025) and up to 42.22% accuracy gains over state-of-the-art methods (Tian et al., 2025). Generation proceeds with chain-of-thought over causal-inference results and cites provenance for every named entity, preventing the hallucinated reasoning paths that systematic agent evaluation documents as the field's primary failure mode (Cornacchia et al., 2025). Incident memory — schemas, queries, architecture, and operational patterns organized as knowledge bases in the Model Context Protocol pattern (Gujarathi, 2026) — grounds diagnosis in organizational expertise and enables less-experienced engineers to execute expert tasks (Baswareddy, 2025). The portfolio's incident-intelligence results bound expectations: agentic root-cause reasoning delivered a 30% mean-time-to-root-cause reduction and a 22% incident-resolution-accuracy improvement (Jingar, 2023), and grounded explanation generation achieved 99.6% directional consistency (Xue et al., 2026).

D. Response layer: automated incident response

The response planner selects actions following Section III-D: playbooks provide the fast path for known patterns (within 30 seconds in the integrated design) (Laheri, 2025), while a learned policy handles novel patterns — the predictive-repair pattern pairing forecasting with reinforcement learning (Singh, 2026). Execution is deterministic: restart, rollback, reschedule, re-route, or scale, through the executor patterns validated in self-healing systems (Nunavath, 2025; Singh, 2024; USA & Sannareddy, 2025). The quantitative anchors for the response layer are the 91% autonomous-resolution rate of the observability-LLM system (Wang et al., 2025), the 89.5% autonomous success rate of the RAG-healing architecture (Sharma & Jain, 2026), the 68%/54%/12% MTTD/MTTR/availability improvements of the production AIOps framework (Singh, 2025), and the greater-than-80% resolution-time reduction of the e-commerce zero-touch case study (Allam, 2024).

E. Governance layer

All actions pass guardrails before execution: unsafe or non-compliant actions are blocked, high blast-radius scenarios escalate to humans — the oversight posture of the e-commerce pre-detection program (Jayakumar & Chan, 2026) and the human-oversight thresholds the AIOps literature mandates (Singh, 2026) — and drift detectors monitor telemetry distributions with the >92% accuracy demonstrated in governed LLM optimization (Jingar, 2023). An audit ledger records every diagnosis step and action for traceability (Syed et al., 2025). Guardrail-centric fine-tuning keeps probabilistic reasoning and deterministic enforcement separate (Davenport, 2026).

F. Evaluation layer: continual ops benchmarking

IRIS-SC evaluates against Ops-domain standards rather than NLP metrics: the OpsEval-style protocol with its 0.9185 human-agreement coefficient, which improves over BLEU and ROUGE by 0.4471 and 1.366 respectively (Liu et al., 2025), and the continual-benchmarking discipline of the networking-operations suite

(Protogeros & Vanbever, 2025). Because pass@k conceals compounding multi-step failures (Chen et al., 2025), evaluation tracks end-to-end resolution correctness per incident class.

G. Learning layer

The chaos-validation loop continuously injects faults via AI-driven and generative-AI mechanisms, improving fault-detection accuracy by 28% and reducing recovery time by 35% over static methods (Gunawat et al., 2025; Pandey et al., 2025), generating fresh failure labels, and updating the RCA prompts, playbooks, and policies. Algorithm 1 summarizes the loop.

Algorithm 1: IRIS-SC Incident Loop
Input: platform G, telemetry X, guardrails Gv, playbooks P, incident memory M, benchmark suite B
Output: resolved incidents, audit log, updated models
Loop on alert:
1. DETECT
alerts ← EnsembleAD(X) # envelope: ACC 97.97% / 99.83% (Elkaradwy et al., 2025; Padhy et al., 2026)
if no alert: continue
2. MATCH + GENERATE EVIDENCE
handler ← MatchAlertType(alerts) # RCACopilot pattern (Chen et al., 2024)
cands ← extract_evidence(alerts, G, traces) # z-scores, paths, spans
3. DIAGNOSE
rc, expl ← LLM_RCA(cands, M) # CoT, provenance, guardrail-checked
if explainability < τ: escalate to human
4. PLAN
if pattern ∈ P: a ← playbook (≤30 s) # fast path (Laheri, 2025)
else: a ← RL_policy(b_t) # predictive repair (Singh, 2026)
if not Guardrails.ok(a) or blast_radius(a) high: a ← HITL(a) (Jayakumar & Chan, 2026)
5. EXECUTE (deterministic)
exe(a) → outcome; audit.log(evidence, rc, a, outcome) (Syed et al., 2025)
6. LEARN
chaos ← AI-driven + GenAI injection (Gunawat et al., 2025; Pandey et al., 2025); retrain AD; refresh M;
benchmark(B, rc, a) (Liu et al., 2025; Protogeros & Vanbever, 2025); monitor drift (Jingar, 2023)

H. Implementation

Detection: ensemble classifiers (Elkaradwy et al., 2025; Padhy et al., 2026) + LSTM/Transformer log models (Du et al., 2017; Xiao et al., 2023) + LLM detectors (Huang et al., 2024; Ma et al., 2026; Zhang et al., 2025); diagnosis: alert-type matching (Chen et al., 2024), fine-tuned domain LLM with chain-of-thought and OpenTelemetry multimodal conditioning (Wang et al., 2025), MCP-style knowledge bases (Gujarathi, 2026); response: rule engine + learned recovery policy (Laheri, 2025; Singh, 2026); governance: guardrail checker, drift detector, audit ledger (Davenport, 2026; Jingar, 2023; Syed et al., 2025); evaluation: OpsEval-style suite (Liu et al., 2025); learning: AI-driven fault injection + generative-AI chaos (Gunawat et al., 2025; Pandey et al., 2025).

V. EXPERIMENTAL SETUP

A. Environments and data

Evaluation is anchored to the documented settings of the component literature: the public microservice datasets of the observability-LLM study (Wang et al., 2025); the year of Microsoft incidents and 44,340-incident/1,759-service corpora of RCACopilot and the Microsoft LLM evaluation (Ahmed et al., 2023; Chen et al., 2024); the open RCA benchmark of the GALA evaluation (Tian et al., 2025); the 44.3-million-log-line, 80-failure production dataset of LogRCA (Wittkopp et al., 2024); the benchmark microservices of Eadro (Lee et al., 2023); the GPU-driven LLM-inference deployment and 24-method comparison of the RCA generalization study (Scheinert et al., 2026); the OpsEval suites (7,334 MCQ + 1,736 QA) (Liu et al., 2025); the 220-fault-scenario protocol of integrated self-healing (Laheri, 2025); and the commerce-platform evidence of the retail and e-commerce case studies (Allam, 2024; Baswareddy, 2025; Jayakumar & Chan, 2026; Sachdeva, 2024).

B. Baselines

Rule-based monitoring and remediation (Wang et al., 2025); static thresholds and manual RCA (Damarched, 2026); statistical and single-modality detectors (Scheinert et al., 2026); state-of-the-art RCA methods of the GALA comparison (Tian et al., 2025); deep-learning and statistical log baselines of LogRCA (Wittkopp et al., 2024); playbook-only and learned-policy-only response (the 60%/50% MTTR-reduction ablation results of the integrated design) (Laheri, 2025); ungoverned LLM prompting (the hallucinated-reasoning baseline)

(Cornacchia et al., 2025); ChatGPT-class extraction baselines (Huang et al., 2024); and non-continual evaluation against NLP metrics (Chen et al., 2025; Liu et al., 2025).

C. Scenarios

(S1) payment-API pod crash with cascading degradation; (S2) transaction latency spike with multi-service propagation; (S3) memory leak with slow-burn anomaly signature; (S4) configuration regression requiring trace-span localization; (S5) peak-shopping-day traffic surge testing seasonal baselines and predictive repair (Jayakumar & Chan, 2026); (S6) compound incident under chaos injection; (S7) adversarial/out-of-distribution incident testing the assisted-to-governed autonomy boundary.

D. Protocol

Temporal splits prevent leakage; detection is scored by accuracy/F1 including newly injected chaos faults; diagnosis is scored by root-cause-identification accuracy and time-to-root-cause; response is scored by MTTR, autonomous success rate, and downtime reduction; explanations are scored by evidence-boundness and the OpsEval evaluation method (Liu et al., 2025); economics follow the per-minute cost accounting of Section II-A. All quantitative entries below are literature-reported values from the cited studies.

VI. RESULTS AND ANALYSIS

A. Task–module mapping

Table I maps each framework layer to its task and evaluation.

TABLE I: IRIS-SC MODULES, TASKS, AND EVALUATION

Layer	Task	Output	Evaluation
Detection	Multi-source anomaly detection	Ranked alerts	Accuracy, F1 (Padhy et al., 2026; Wang et al., 2025)
Diagnosis	Alert-type-matched, evidence-grounded LLM RCA	Root cause + explanation	RC accuracy (Chen et al., 2024), time-to-RC (Jingar, 2023)
Response	Playbook + learned-policy action	Executed action	MTTR (Laheri, 2025), ASR (Wang et al., 2025), downtime (Allam, 2024)
Governance	Guardrails, drift, audit	Approved/escalated actions	Unsafe-output rate, drift accuracy (Jingar, 2023)
Evaluation	Continual Ops benchmarking	Model/inference scores	OpsEval-style scores (Liu et al., 2025)
Learning	Chaos-driven retraining	Updated models	Detection gain, recovery-time gain (Gunawat et al., 2025)

B. Reported performance envelope

Table II consolidates reported results across the LLM-RCA and incident-response literature.

TABLE II: REPORTED PERFORMANCE IN LLM-BASED RCA AND INCIDENT RESPONSE

Method	Task	Reported result	Source
LLM + OpenTelemetry observability	Automated RCA + remediation	MTTR −84.2% vs rule-based; F1 95%; 91% of incidents automated	(Wang et al., 2025)
ARCH LLM + RAG self-healing	Autonomous cloud healing	MTTR −82% (up to); ASR 89.5%	(Sharma & Jain, 2026)
RCACopilot	Production RCA at Microsoft	Accuracy up to 0.766; 1 year of incidents; diagnostics in use 4+ years	(Chen et al., 2024)
Microsoft LLM evaluation	Root-cause + mitigation recommendation	44,340 incidents, 1,759 services; human validation by incident owners	(Ahmed et al., 2023)
AIOps observability framework	Production incident management	MTTD −68%; MTTR −54%; availability +12%	(Singh, 2025)
Zero-touch self-healing (e-commerce)	Platform incident resolution	Resolution time −80%+; downtime slashed	(Allam, 2024)
LLM incident assistants (industry synthesis)	Incident management	MTTR −40–60%; response time −30–70%; $400B potential	(Damarched, 2026)
GALA graph-augmented LLM RCA	Microservice RCA	Accuracy +42.22% over SOTA	(Tian et al., 2025)
LLM agent architectures (systematic evaluation)	Automated failure analysis	Promise and limitations; hallucinated reasoning paths identified	(Cornacchia et al., 2025)
LLM + MCP diagnostics	Distributed-system diagnosis	KB-encoded expertise; MTTR improved in production	(Gujarathi, 2026)
LoFI	Fault-indicating log extraction	F1 +25.8–37.9 vs ChatGPT baseline; deployed at CloudA	(Huang et al., 2024)
LLM-ID	Log-based fault localization	Fault-location accuracy +16.2%	(Ji & Luo, 2025)
LogReasoner	LLM log analysis reasoning	SOTA on 4 tasks; Qwen-2.5, Llama-3	(Ma et al., 2026)
ScalaLog	RAG-based failure diagnosis	Accuracy ↑ without training/log parsing	(Zhang et al., 2025)
Production LLM log analytics	IT support ticket diagnosis	70 products, 2000+ tickets; 300+ man-hours; $15,444/month	(Gupta et al., 2026)
RCA method comparison (24 methods)	RCA generalization on LLM workloads	Multi-source highest accuracy; trace-based largely fail	(Scheinert et al., 2026)

LogRCA	Minimal-set log RCA	44.3M log lines, 80 failures; beats DL/statistical baselines	(Wittkopp et al., 2024)
Eadro	End-to-end multi-source troubleshooting	Outperforms SOTA on 2 benchmark microservices	(Lee et al., 2023)
Integrated predictive + RL + playbooks	Self-healing	MTTR −85% (90 → 13.5 min); uptime > 98%	(Laheri, 2025)
AutoPilot-K8s	DRL cluster orchestration	Latency −42%; SLA adherence +60%	(Kripa, 2025)
OpsEval	Ops-LLM benchmark	7,334 MCQ + 1,736 QA; 24 LLMs; 0.9185 human agreement	(Liu et al., 2025)
Governed LLM optimization	Safe optimization	Unsafe outputs −45%; drift detection > 92%	(Jingar, 2023)
AI-driven chaos	Resilience validation	Detection +28%; recovery −35%	(Gunawat et al., 2025)
GenAI chaos (serverless)	Chaos automation	Setup −80%; scenario diversity +71%; diagnosis −67%	(Pandey et al., 2025)

C. Worked calculations

1) Downtime economics (the headline arithmetic). At the reported $100 million per hour for Amazon on major shopping days (Ahmed et al., 2023), the per-minute cost is $100M / 60 ≈ $1.67 million. Deducting one minute from diagnosis or remediation therefore saves on the order of $1.67M at peak — consistent with the independent 2016 datum of $66,240 per minute of Amazon server downtime (Nguyễn et al., 2018), whose 25× gap to the peak figure reflects the peak-season multiplier. At data-center scale, the average outage of $740,357 per event at ~$7,900/minute (Nguyễn et al., 2018), or ~$8,000/minute with a 95-minute average incident length (Xie & Li, 2023), implies that a system cutting MTTR from 95 to, say, 20 minutes saves roughly 75 × $8,000 ≈ $600,000 per average incident. At the enterprise level, 14–18 hours of annual downtime at up to $14,056 per minute (Damarched, 2026) implies an annual exposure of 14 × 60 × $14,056 ≈ $11.8M to 18 × 60 × $14,056 ≈ $15.2M per enterprise; the industry-reported 30–70% response-time reductions (Damarched, 2026) applied to these figures translate into millions of dollars of annual exposure reduction.

2) MTTR reduction ladder. Residual-MTTR arithmetic across systems, all reductions relative to reactive/rule-based baselines: rule-based baseline 100%; LLM assistant class 40–60% reduction → 40–60% residual (Damarched, 2026); production AIOps observability 54% reduction → 46% residual (Singh, 2025); ARCH up to 82% reduction → ≤18% residual (Sharma & Jain, 2026); LLM observability 84.2% reduction → 15.8% residual (Wang et al., 2025). The integrated self-healing benchmark — 85% reduction, from 90 to 13.5 minutes (Laheri, 2025) — sits at the top of the ladder; the diagnosis-focused systems approach it on the diagnosis-plus-repair interval alone, and the e-commerce zero-touch deployment exceeds an 80% resolution-time reduction (Allam, 2024). Fig. 2 visualizes the ladder.

3) Automation-rate arithmetic. A 91% autonomous-resolution rate (Wang et al., 2025) and an 89.5% autonomous success rate (Sharma & Jain, 2026) mean roughly nine in ten common incidents resolve without human intervention; the residual ~9–10% — novel, high-severity, or guardrail-flagged incidents — routes to human oversight, the operating point the AIOps literature prescribes (Singh, 2026) and the e-commerce program implements for high blast-radius actions (Jayakumar & Chan, 2026). In a commerce platform operating at 1,000 incidents per year, this implies approximately 90–105 human-handled incidents per year — a bounded operator load rather than an unbounded one.

4) Diagnosis-accuracy gains. RCACopilot's accuracy up to 0.766 on a year of real Microsoft incidents (Chen et al., 2024) and GALA's up-to-42.22% improvement over state-of-the-art RCA methods (Tian et al., 2025) quantify what alert-type matching plus multimodal causal reasoning plus guided LLM inference is worth. The supporting evidence is granular: LoFI's absolute F1 improvement of 25.8–37.9 over ChatGPT for fault-indicating extraction (Huang et al., 2024) and LLM-ID's 16.2% fault-location-accuracy improvement (Ji & Luo, 2025) show that the extraction-and-localization sub-steps — not the final narrative — drive diagnosis quality. Combined with the 30% time-to-root-cause reduction of agentic incident intelligence (Jingar, 2023), the composite effect is a diagnosis stage that is both faster and more often correct.

5) Operator-load economics of LLM log analysis. A production LLM log-analytics tool processing over 2,000 tickets across 70 software products saved over 300 man-hours and approximately $15,444 per month in manpower costs (Gupta et al., 2026). Annualized, this is ~$185K per year for a single product portfolio — and in incident terms, every man-hour returned from log triage to engineering is time an on-call engineer does not spend in the diagnosis loop that the Microsoft study documents as the manual-toil bottleneck (Ahmed et al., 2023).

6) Method-selection evidence. The 24-method evaluation on GPU-driven LLM workloads — 20 metric-based, two trace-based, two multi-source — found multi-source approaches most accurate, metric-based methods fault-type-dependent, and trace-based methods largely failing (Scheinert et al., 2026). The design implication is structural: IRIS-SC's multi-source detection and diagnosis layers are not a convenience but a validated choice, and the framework's evaluation layer must include per-method ablation on each incident class.

7) End-to-end economics of the portfolio loop. Combining the seven-agent disruption-analysis cadence — 3.83 minutes at $0.0836 per disruption (AlMahri et al., 2026) — with platform-incident resolutions at automation rates

near 90% (Sharma & Jain, 2026; Wang et al., 2025) implies a monitoring-and-response posture that is both minutes-fast and effectively free per event — the marginal economics (tens of cents per incident) that make continuous, always-on diagnosis affordable at hyperscale.

D. Mttr reduction ladder

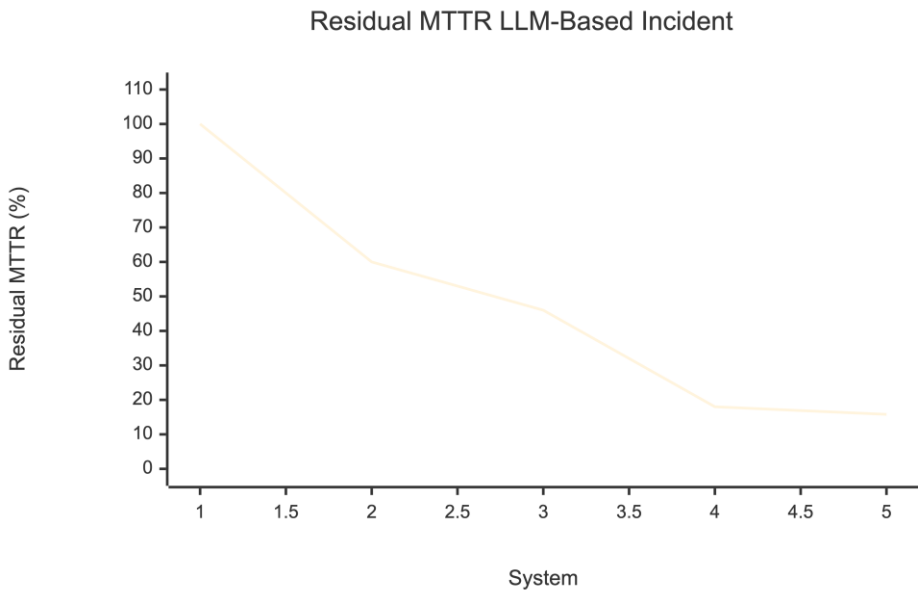

Fig. 2 plots the residual-MTTR comparison across reported systems.

Fig. 2. Computed residual MTTR from the reported reductions: rule-based baseline 100%; LLM assistant class (upper bound) 60% (40–60% reduction) (Damarched, 2026); production AIOps observability 46% (54% reduction) (Singh, 2025); ARCH ≤18% (up to 82% reduction) (Sharma & Jain, 2026); LLM observability 15.8% (84.2% reduction) (Wang et al., 2025). Values derived arithmetically from the reported percentages.

E. Automation success rates

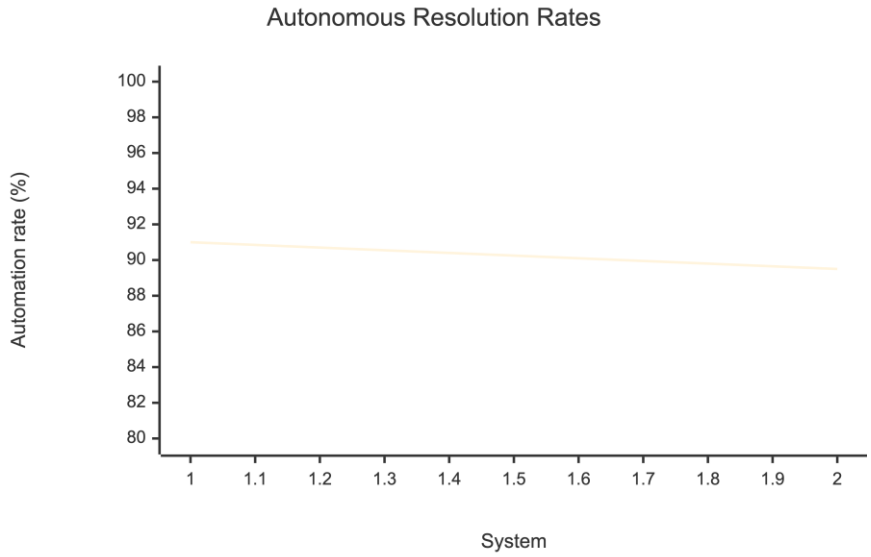

Fig. 3 plots the two reported autonomous-resolution rates framing the response layer.

Fig. 3. Reported autonomous resolution rates: 91% of common incidents automated without human intervention by the LLM observability framework (Wang et al., 2025); 89.5% autonomous success rate of the LLM+RAG ARCH framework (Sharma & Jain, 2026).

F. Ablation logic

Removing the multi-source detection ensemble reverts to single-modality alerts that miss anomalies invisible in any one source — the pattern the 24-method comparison showed for trace-only and metric-only designs (Scheinert et al., 2026); removing alert-type matching forfeits the routing-and-context advantage behind RCACopilot's 0.766 accuracy (Chen et al., 2024); removing evidence extraction from the RCA prompt reintroduces the hallucinated reasoning paths documented in the systematic agent evaluation (Cornacchia et al., 2025) and forfeits the up-to-42.22% accuracy gain of grounded inference (Tian et al., 2025); removing the guardrails reintroduces the 45% unsafe-decision exposure of ungoverned generation (Jingar, 2023); removing continual Ops benchmarking reverts evaluation to BLEU/ROUGE-style scores shown to disagree with human assessment by up to 1.366 points (Liu et al., 2025); and removing the chaos-learning loop removes the drift countermeasure — 28% detection and 35% recovery gains (Gunawat et al., 2025) — that keeps automation rates near 90% over time.

VII. DISCUSSION

A. Diagnosis, not generation, is the differentiator

The systematic evaluation evidence is blunt: LLM agents hallucinate reasoning paths and use context inefficiently without grounding (Cornacchia et al., 2025). The systems that define the envelope — 84.2% MTTR reduction (Wang et al., 2025), up-to-42.22% accuracy gains (Tian et al., 2025), 91%/89.5% automation (Sharma & Jain, 2026; Wang et al., 2025), 0.766 production RCA accuracy (Chen et al., 2024) — all constrain generation with evidence: alert types, model-internal features, dependency paths, trace spans, and organizational knowledge bases (Chen et al., 2024; Gujarathi, 2026). IRIS-SC encodes this as a construction-time constraint: every diagnosis must cite its evidence, and every explanation is scored for evidence-boundness.

B. Automation is valuable in direct proportion to governance

The ~$1.67M-per-minute peak-cost arithmetic of Section VI-C makes automation maximally valuable — and the guardrail evidence (45% unsafe-output reduction, >92% drift detection (Jingar, 2023); approximately fifty guarded behaviors with deterministic enforcement (Davenport, 2026); human approvals for high blast-radius actions (Jayakumar & Chan, 2026); human thresholds for high-risk scenarios (Singh, 2026)) makes it deployable. The asymptotic operating point — roughly 90% autonomous resolution with the residual deliberately routed to humans — is not a limitation; it is the design's risk posture.

C. Mttr is a log-normal random variable, not a number

The empirical finding that enterprise recovery times fit a log-normal distribution over more than 1,800 incidents (Franke et al., 2014) has a practical consequence: single-point MTTR comparisons understate variance, and SLA exposure is driven by the tail. IRIS-SC therefore evaluates MTTR as a distribution — reporting percentiles, not just means — and the compound-Poisson cost model $E[S] = \lambda \cdot c$, $Var[S] = \lambda \cdot c^2$ (Wang & Franke, 2020) formalizes why variance reduction (shorter, more numerous outages) is preferable when costs are duration-proportional.

D. Evaluation defines progress

The field's rate limiter is measurement: BLEU and ROUGE mis-rate Ops answers by margins (0.4471 and 1.366) larger than typical model differences (Liu et al., 2025), and pass@k conceals compounding multi-step failures (Chen et al., 2025). The continual-benchmarking discipline — modular test suites, multiple pipeline variants, evolving task complexity (Protogeros & Vanbever, 2025) — is the only evaluation posture that can track a self-improving IR loop. IRIS-SC therefore treats evaluation as a module, not a report, and extends it with commerce-specific incident corpora (Jayakumar & Chan, 2026; Sachdeva, 2024).

E. Integration with the portfolio

IRIS-SC is the diagnosis-and-response engine of the ten-paper program: it is the platform twin of Paper 6's prediction-and-healing framework (prediction says when; IRIS-SC says why and how); it consumes Paper 7's retrieval-grounded evidence and Paper 9's knowledge-graph structure to ground its diagnoses (Xue et al., 2026); it operationalizes Paper 8's multi-agent decisions at the platform layer; and its resolved-incident facts feed the supply-chain early-warning layers of Papers 1–5 with operational ground truth. Where Papers 6–9 established the platform's intelligence, IRIS-SC establishes its reflexes.

F. Limitations

Reported results span heterogeneous tasks, datasets, and systems rather than a single controlled evaluation; the strongest MTTR numbers come from single-framework evaluations (Sharma & Jain, 2026; Singh, 2025; Wang et al., 2025); production-scale LLM-RCA evaluations are rare (RCACopilot (Chen et al., 2024) and

LogRCA's 44.3M-line dataset (Wittkopp et al., 2024) are outliers); the 24-method comparison shows RCA tools do not generalize to LLM workloads without tailoring (Scheinert et al., 2026); hallucination risk persists for out-of-distribution incidents (Cornacchia et al., 2025); and Ops-domain benchmark coverage remains partial relative to the task space (Liu et al., 2025). These define the agenda below.

VIII. CONCLUSION AND FUTURE WORK

This paper formalized LLM-based root cause analysis and automated incident response for large-scale distributed commerce systems as a grounded diagnosis-and-response loop and proposed IRIS-SC, a framework coupling multi-source anomaly detection, alert-type-matched evidence generation, LLM diagnosis with provenance, automated response blending playbooks and learned policies, guardrail governance, continual Ops benchmarking, and chaos-driven learning. It consolidated the reported evidence envelope — $100M-per-hour peak downtime cost (Ahmed et al., 2023), $66,240/minute and $740,357-per-event outage economics (Nguyễn et al., 2018), $99.75M Facebook outage loss (Xie & Li, 2023), and $14,056-per-minute enterprise exposure (Damarched, 2026); MTTR reductions of 84.2% (Wang et al., 2025), up to 82% (Sharma & Jain, 2026), 54% (Singh, 2025), and 40–60% for the assistant class (Damarched, 2026), with >80% resolution-time reduction in e-commerce (Allam, 2024); automation rates of 91% and 89.5% (Sharma & Jain, 2026; Wang et al., 2025); production RCA accuracy of 0.766 (Chen et al., 2024) and diagnosis gains up to 42.22% (Tian et al., 2025); operational savings of 300+ man-hours and ~$15,444/month in LLM log analytics (Gupta et al., 2026); the multi-source superiority result across 24 RCA methods (Scheinert et al., 2026); and Ops evaluation achieving 0.9185 human agreement (Liu et al., 2025) — and argued that grounded diagnosis, deterministic execution, and governed automation are what convert LLMs from incident assistants into incident operators.

Future work will (i) implement IRIS-SC end-to-end on production-scale commerce platforms with unified evaluation under the Section V protocol, including MTTR-distribution and cost-variance reporting per Section VII-C; (ii) harden diagnosis against hallucination through knowledge-graph grounding (Xue et al., 2026), provenance checking, decomposed reasoning, and alert-type-matched context aggregation in the RCACopilot pattern (Chen et al., 2024); (iii) extend continual benchmarking from OpsEval-style suites (Liu et al., 2025) to commerce-specific incident corpora spanning seasonal peak traffic (Jayakumar & Chan, 2026); (iv) deepen the learning loop by coupling chaos-injected failures (Gunawat et al., 2025; Pandey et al., 2025) with resolved-incident memory so that every resolution retrains the system; and (v) build the human-AI operating model — bounded operator load, selective human specification, and audit-complete records (Baek et al., 2026; Syed et al., 2025; Wu, 2026) — so that IRIS-SC's ~90% automation becomes a licensed production posture, moving incident management from reactive response to self-diagnosing, self-healing commerce infrastructure.

References

[1]. Ahmed, T., Ghosh, S., Bansal, C., Zimmermann, T., Zhang, X., & Rajmohan, S. (2023). Recommending Root-Cause and Mitigation Steps for Cloud Incidents using Large Language Models. 1737–1749. https://doi.org/10.1109/icse48619.2023.00149

[2]. Allam, H. (2024). Zero-Touch Reliability: The Next Generation of Self-Healing Systems. International Journal of Artificial Intelligence Data Science and Machine Learning, 5, 59–71. https://doi.org/10.63282/3050-9262.ijaidsml-v5i4p107

[3]. AlMahri, S., Xu, L., & Brintrup, A. (2026). Automating Supply Chain Disruption Monitoring via an Agentic AI Approach. ArXiv.Org. http://arxiv.org/abs/2601.09680

[4]. Awad, A., & Alahmari, D. (2025). Agentic Control Towers. In Advances in computational intelligence and robotics book series (pp. 161–182). IGI Global. https://doi.org/10.4018/979-8-3373-7006-4.ch006

[5]. Babatope, O. M., Oyewole, T., Ogbole, J. I., & Okoruwa, P. O. (2023). Developing an AI-Based Incident Response Automation Framework to Minimize Downtime in IT Service Operations. International Journal of Advanced Multidisciplinary Research and Studies, 3(6), 2406–2423. https://doi.org/10.62225/2583049x.2023.3.6.5415

[6]. Baek, J., Fu, Y., Ma, W., & Peng, T. (2026). AI Agents for Inventory Control: Human-LLM-OR Complementarity. In arXiv (Cornell University). Cornell University. https://doi.org/10.48550/arxiv.2602.12631

[7]. Baryannis, G., Dani, S., & Antoniou, G. (2019). Predicting supply chain risks using machine learning: The trade-off between performance and interpretability. Future Generation Computer Systems, 101, 993–1004. https://doi.org/10.1016/j.future.2019.07.059

[8]. Basiri, A., Hochstein, L., Jones, N. L., & Tucker, H. (2019). Automating Chaos Experiments in Production. In arXiv (Cornell University) (pp. 31–40). Cornell University. https://doi.org/10.1109/icse-seip.2019.00012

[9]. Baswareddy, J. P. R. (2025). Intelligent CI/CD Pipelines: Leveraging AI for Predictive Maintenance and Incident Management. European Journal of Computer Science and Information Technology, 13(7), 50–65. https://doi.org/10.37745/ejcsit.2013/vol13n75065

[10]. Chen, X., Wanigarathna, Y. R., Chattopadhyay, A., Tarkoma, S., & Rao, A. (2025). A Pedagogical Approach for Evaluating AIOps. 40–49. https://doi.org/10.1109/aimlsystems67835.2025.11330228

[11]. Chen, Y., Xie, H., Ma, M., Kang, Y., Gao, X., Shi, L., Cao, Y., Gao, X. Y., Fan, H., Wen, M., Zeng, J., Ghosh, S., Zhang, X., Zhang, C., Lin, Q., Rajmohan, S., Zhang, D., & Xu, T. (2024). Automatic Root Cause Analysis via Large Language Models for Cloud Incidents. 674–688. https://doi.org/10.1145/3627703.3629553

[12]. Cornacchia, A., Alabdulaal, I., Saghier, I., Mirdad, A., Fayoumi, O., & Canini, M. (2025). Between Promise and Pain: The Reality of Automating Failure Analysis in Microservices with LLMs. 155–167. https://doi.org/10.1145/3725783.3764388

[13]. Cui, Y., Zeng, Y., Yan, J., Lin, K. L., Ji, K. Y., Zeng, J., Zhang, S., Luo, X., Su, B., Shen, C., & Yu, J. (2026). CSCBench: A PVC Diagnostic Benchmark for Commodity Supply Chain Reasoning. In arXiv (Cornell University). Cornell University. https://doi.org/10.48550/arxiv.2601.01825

[14]. Damarched, M. K. (2026). Harnessing Large Language Models and Agentic AI for Transformative Cloud Reliability and Incident Management: A Comprehensive Suggestive Review. Journal of Computer Science and Technology Studies, 8(5), 43–81. https://doi.org/10.32996/jcsts.2026.8.5.4

[15]. Davenport. (2026). GUARDRAIL-CENTRIC FINE-TUNING FOR DETERMINISTIC DECISION SYSTEMS. Zenodo (CERN European Organization for Nuclear Research). https://doi.org/10.5281/zenodo.18305825

[16]. Du, M., Fei-Fei, L., Zheng, G., & Srikumar, V. (2017). DeepLog. 1285–1298. https://doi.org/10.1145/3133956.3134015

[17]. Elkaradwy, A., Elshenawy, A., & Harb, H. (2025). Enhancing Cloud Job Failure Prediction With a Novel Multilayer Voting-Based Framework. IEEE Access, 13, 140600–140613. https://doi.org/10.1109/access.2025.3593808

[18]. Franke, U., Holm, H., & König, J. (2014). The Distribution of Time to Recovery of Enterprise IT Services. IEEE Transactions on Reliability, 63(4), 858–867. https://doi.org/10.1109/tr.2014.2336051

[19]. Fu, N., Cheng, G., Teng, Y., Dai, G., Yu, S., & Chen, Z. (2025). Intelligent Root Cause Localization in MicroService Systems: A Survey and New Perspectives. ACM Computing Surveys, 57(12), 1–37. https://doi.org/10.1145/3736755

[20]. Guan, S., Liu, Y., & Cao, L. (2026). SupChain-Bench: Benchmarking Large Language Models for Real-World Supply Chain Management. arXiv (Cornell University). http://arxiv.org/abs/2602.07342

[21]. Gujarathi, M. (2026). AI-Driven Production Support: Applying Large Language Models and Model Context Protocol to Distributed Systems Diagnostics. Journal of Computational Analysis and Applications, 35(2). https://doi.org/10.48047/jocaaa.2026.35.02.09

[22]. Gunawat, C., Gupta, R., & Nankani, J. S. (2025). AI-Driven Fault Injection Testing: Enhancing System Resilience with Automated Chaos Engineering. Asian Journal of Research in Computer Science, 18(6), 1–8. https://doi.org/10.9734/ajrcos/2025/v18i6675

[23]. Gupta, P., Bhukar, K., Kumar, H., Nagar, S., Mohapatra, P., & Kar, D. (2026). Scalable and Efficient Large-Scale Log Analysis with LLMs: An IT Software Support Case Study. Proceedings of the AAAI Conference on Artificial Intelligence, 40(47), 39958–39967. https://doi.org/10.1609/aaai.v40i47.41430

[24]. He, S., Deng, T., Chen, B., Sherratt, R. S., & Wang, J. (2023). Unsupervised Log Anomaly Detection Method Based on Multi-Feature. Computers, Materials & Continua/Computers, Materials & Continua (Print), 76(1), 517–541. https://doi.org/10.32604/cmc.2023.037392

[25]. Huang, J., Jiang, Z., Liu, J., Huo, Y., Gu, J., Chen, Z., Feng, C., Hui, D., Yang, Z., & Lyu, M. R. (2024). Demystifying and Extracting Fault-indicating Information from Logs for Failure Diagnosis. 511–522. https://doi.org/10.1109/issre62328.2024.00055

[26]. Hyderabad, J. N. T. U. (2025). LLM-Powered Supply Chain Dashboard: A Graph-based multi-module framework for Data-driven Decision making. In Zenodo (CERN European Organization for Nuclear Research). European Organization for Nuclear Research. https://doi.org/10.5281/zenodo.17695657

[27]. Ioannis, P., & Laurent, V. (2025). Continual Benchmarking of LLM-Based Systems on Networking Operations. In Repository for Publications and Research Data (ETH Zurich). ETH Zurich. https://doi.org/10.3929/ethz-c-000786602

[28]. Iruku, V. M. (2025). Multi-dimensional XAI Framework Revealing Critical Supply Chain Vulnerability Drivers. World Journal of Advanced Engineering Technology and Sciences, 15(3), 2141–2152. https://doi.org/10.30574/wjaets.2025.15.3.1154

[29]. Jagannathan, S., Sharma, Y., & Taheri, J. (2025). Towards Generic Failure-Prediction Models in Large-Scale Distributed Computing Systems. Electronics, 14(17), 3386. https://doi.org/10.3390/electronics14173386

[30]. Jaiswal, I. A. (2024). AI-Powered Observability and Incident Prediction in Distributed Enterprise Platforms. Scientific Journal of Artificial Intelligence and Blockchain Technologies, 1(1). https://doi.org/10.63345/sjaibt.v1.i1.201

[31]. Jayakumar, P., & Chan, D. (2026). Protecting Revenue at Scale: Pre-Detecting Anomalies in High Velocity ECommerce Systems using Artificial Intelligence (AI). International Scientific Journal of Engineering and Management, 5(1), 1–9. https://doi.org/10.55041/isjem05351

[32]. Ji, C., & Luo, H. (2025). Leveraging Large Language Model for Intelligent Log Processing and Autonomous Debugging in Cloud AI Platforms. 348–351. https://doi.org/10.1109/aemcse65292.2025.11042521

[33]. Jingar, N. K. (2023a). Ensuring Safety, Accountability, and Drift Resistance in LLM-Based Supply Chain Optimization. International Journal of Scientific Research in Science Engineering and Technology, 472. https://doi.org/10.32628/ijsrset2310372

[34]. Jingar, N. K. (2023b). Automated Incident Intelligence In Supply Chains Using Agentic AI And Root Cause Reasoning. Zenodo (CERN European Organization for Nuclear Research). https://doi.org/10.5281/zenodo.18628070

[35]. Kavlak, Y., & Sezen, A. (2026). A Comparative Study of Deep Learning Models for Log Anomaly Detection Using the HDFS Dataset. 306–309. https://doi.org/10.1109/iisec69317.2026.11418409

[36]. Kikuta, D., Ikeuchi, H., & Tajiri, K. (2025). LLM-Powered Fully Automated Chaos Engineering: Towards Enabling Anyone to Build Resilient Software Systems at Low Cost. ArXiv.Org, 3861–3865. https://doi.org/10.1109/ase63991.2025.00331

[37]. Kohyarnejadfard, I., Aloise, D., Azhari, S. V., & Dagenais, M. (2022). Anomaly detection in microservice environments using distributed tracing data analysis and NLP. Journal of Cloud Computing Advances Systems and Applications, 11(1), 25. https://doi.org/10.1186/s13677-022-00296-4

[38]. Konda, R. (2021). Machine Learning-Based Self-Healing Systems: Automated Failure Detection and Recovery in Microservices. International Journal For Multidisciplinary Research, 3(5). https://doi.org/10.36948/ijfmr.2021.v03i05.43946

[39]. Kripa, R. K. (2025). AutoPilot-K8s: An Autonomous Reinforcement Learning Framework for Self-Optimizing Kubernetes Clusters. 188–193. https://doi.org/10.1109/ised67359.2025.11405259

[40]. Laheri, R. (2025). Self-Healing Infrastructure: Leveraging Reinforcement Learning for Autonomous Cloud Recovery and Enhanced Resilience. Journal of Information Systems Engineering & Management, 10, 352–357. https://doi.org/10.52783/jisem.v10i49s.9888

[41]. Landauer, M., Onder, S., Skopik, F., & Wurzenberger, M. (2023). Deep learning for anomaly detection in log data: A survey. Machine Learning with Applications, 12, 100470. https://doi.org/10.1016/j.mlwa.2023.100470

[42]. Lee, C., Yang, T., Chen, Z., Su, Y., & Lyu, M. R. (2023). Eadro: An End-to-End Troubleshooting Framework for Microservices on Multi-source Data. 1750–1762. https://doi.org/10.1109/icse48619.2023.00150

[43]. Liu, Y., Pei, C., Xu, L., Chen, B., Sun, M., Zhang, Z. L., Sun, Y., Zhang, S., Wang, K., Zhang, H., Li, J., Xie, G., wen, xiaohui, Nie, X., Ma, M., & Pei, D. (2025). OpsEval: A Comprehensive Benchmark Suite for Evaluating Large Language Models' Capability in IT Operations Domain. 503–513. https://doi.org/10.1145/3696630.3728572

[44]. Lv, Y., & Min, S. (2023). Anomaly Detection from System Logs based on Deep Learning Techniques. 912–916. https://doi.org/10.1109/icemce60359.2023.10490680

[45]. Ma, L., Li, Y., Yang, W., Zhou, M., Liu, X., Fei, B., Li, S., Sun, X., Jiang, S., & Xiao, Y. (2026). LogReasoner: Empowering LLMs with Expert-like Coarse-to-Fine Reasoning for Automated Log Analysis. ACM Transactions on Software Engineering and Methodology. https://doi.org/10.1145/3801747

[46]. Mayo, W., Ogbole, J. I., Okoruwa, P. O., & Babatope, O. M. (2023). Designing an AI-Predictive Maintenance Model for E-Commerce Systems Using Machine Learning and Cloud Analytics. International Journal of Advanced Multidisciplinary Research and Studies, 3(6), 2454–2468. https://doi.org/10.62225/2583049x.2023.3.6.5418

[47]. Meyer, O., Johnson, E. A., & Brown, J. D. (2025). A Unified Framework for Anomaly Detection and Root Cause Analysis in Microservice Systems. Jisuanji Shenghuojia., 13(2), 7–11. https://doi.org/10.54097/1gw77589

[48]. Nguyễn, T. A., Min, D., & Choi, E. (2018). Stochastic Reward Net-based Modeling Approach for Availability Quantification of Data Center Systems. In InTech eBooks. https://doi.org/10.5772/intechopen.74306

[49]. Nunavath, V. (2025). AI-enhanced self-healing Kubernetes for scalable cloud operations. World Journal of Advanced Engineering Technology and Sciences, 16(2), 21–29. https://doi.org/10.30574/wjaets.2025.16.2.1255

[50]. Padhy, A. K., Patel, T., Soni, V., Shivam, S., Thokala, G. B., & Vulugundam, B. (2026). Machine Learning-Based Fault Prediction in Large- Scale Distributed Systems. 1–6. https://doi.org/10.1109/icaic67076.2026.11395778

[51]. Pandey, R. K., Ànand, A., Soni, J. Y., & Pandey, P. (2025). Chaos Engineering Meets Generative AI: Towards Smarter and Safer Fault Injection. 841–846. https://doi.org/10.1109/cises66934.2025.11265516

[52]. Protogeros, I., & Vanbever, L. (2025). Continual Benchmarking of LLM-Based Systems on Networking Operations. 70–72. https://doi.org/10.1145/3744969.3748425

[53]. R., K. S., Ojha, D., Kaur, P., Mahto, R. V., & Dhir, A. (2024). Explainable artificial intelligence and agile decision-making in supply chain cyber resilience. University of North Texas Digital Library (University of North Texas), 180, 114194. https://doi.org/10.1016/j.dss.2024.114194

[54]. Sachdeva, M. S. (2024). AI-Driven Incident Management in Retail : A Case Study. International Journal of Scientific Research in Computer Science Engineering and Information Technology, 10(6), 355–363. https://doi.org/10.32628/cseit24106182

[55]. Scheinert, D., Acker, A., Wittkopp, T., Becker, S., Yous, H., Reddy, K. C., Farhat, I., Hacid, H., & Kao, O. (2026). Beyond Microservices: Testing Web-Scale RCA Methods on GPU-Driven LLM Workloads. In arXiv (Cornell University). Cornell University. https://doi.org/10.48550/arxiv.2603.02057

[56]. Sharma, R., & Jain, K. (2026). ARCH: An LLM-Driven Autonomous Self-Healing Framework for Cloud Operations Using Retrieval-Augmented Generation. Zenodo (CERN European Organization for Nuclear Research). https://doi.org/10.5281/zenodo.19204590

[57]. Shi, Q., Yang, Z., Haidar, A., Terry, B. J., Li, F., Zhang, J., & Shao, S. (2024). Anomaly Transformer-based System Log Anomaly Detection. 1–6. https://doi.org/10.1109/cars61786.2024.10778676

[58]. Singh, H. (2026). AI-Driven Incident Management for Distributed Cloud Systems: Detection, Mitigation, and Root Cause Automation. Journal of Information Systems Engineering & Management, 11, 977–985. https://doi.org/10.52783/jisem.v11i1s.14216

[59]. Singh, P. (2024). Self-Healing Architecture: Using ML to Enhance System Resilience in FinTech Platforms. Journal of Artificial Intelligence Machine Learning and Data Science, 2(1), 2719–2724. https://doi.org/10.51219/jaimld/prashant-singh/573

[60]. Singh, R. (2025). Optimizing Application Support Operations with LLMs- Insights from a Self-Healing Interface. International Journal of Scientific Research and Modern Technology., 4(8), 180. https://doi.org/10.38124/ijsrmt.v4i8.980

[61]. Singh, S. (2025). Advanced Observability and AIOps Framework for Intelligent IT Operations Management. Zenodo (CERN European Organization for Nuclear Research). https://doi.org/10.5281/zenodo.18030512

[62]. Syed, T. A., Belgaum, M. R., Jan, S., Khan, A. A., & Alqahtani, S. S. (2025). Agentic AI for Autonomous Defense in Software Supply Chain Security: Beyond Provenance to Vulnerability Mitigation. ArXiv.Org. http://arxiv.org/abs/2512.23480

[63]. Tian, Y., Liu, Y., Chong, Z. J., Huang, Z., & Jacobsen, H. (2025). GALA: Can Graph-Augmented Large Language Model Agentic Workflows Elevate Root Cause Analysis? In ArXiv.org. https://doi.org/10.48550/arxiv.2508.12472

[64]. USA, S. C. I. E., Illinois, & Sannareddy, S. B. (2025). Autonomous Kubernetes Cluster Healing using Machine Learnin. International Journal of Research Publications in Engineering Technology and Management, 7(5). https://doi.org/10.15662/ijrpetm.2024.0705006

[65]. Wang, C., Yuan, T., Hua, C., Lu, C., Yang, X., & Qiu, Z. (2025). Integrating Large Language Models with Cloud-Native Observability for Automated Root Cause Analysis and Remediation. In Preprints.org. https://doi.org/10.20944/preprints202512.1926.v1

[66]. Wang, H., Jiang, J., Hong, L. J., & Jiang, G. (2025). LLMs for Supply Chain Management. In ArXiv.org. https://doi.org/10.48550/arxiv.2505.18597

[67]. Wang, J. (2024). Multimodal Deep Learning Approach for Early Warning of Supply Chain Disruptions Using NLP and Anomaly Detection. Artificial Intelligence and Machine Learning Review, 5(3), 98–110. https://doi.org/10.69987/aimlr.2024.50308

[68]. Wang, S. S., & Franke, U. (2020). Enterprise IT service downtime cost and risk transfer in a supply chain. Operations Management Research, 13, 94–108. https://doi.org/10.1007/s12063-020-00148-x

[69]. Wichmann, P., Brintrup, A., Baker, S., Woodall, P., & McFarlane, D. (2020). Extracting supply chain maps from news articles using deep neural networks. International Journal of Production Research, 58(17), 5320–5336. https://doi.org/10.1080/00207543.2020.1720925

[70]. Wittkopp, T., Wiesner, P., & Kao, O. (2024). LogRCA: Log-based Root Cause Analysis for Distributed Services. In arXiv (Cornell University). Cornell University. https://doi.org/10.48550/arxiv.2405.13599

[71]. Wu, R. (2026). Inventory optimization under supply chain disruptions: Leveraging large language models for human-AI collaborative decision-making. Journal of King Saud University - Computer and Information Sciences, 38(2). https://doi.org/10.1007/s44443-025-00458-9

[72]. Xiao, T., Quan, Z., Wang, Z.-J., Le, Y., Du, Y., Liao, X., Li, K., & Li, K. (2023). Loader: A Log Anomaly Detector Based on Transformer. IEEE Transactions on Services Computing, 16(5), 3479–3492. https://doi.org/10.1109/tsc.2023.3280575

[73]. Xie, L., & Li, H. (2023). How to Integrate Digital Twin and Virtual Reality in Robotics Systems? Design and Implementation for Providing Robotics Maintenance Services in Data Centers. In arXiv (Cornell University). Cornell University. https://doi.org/10.48550/arxiv.2312.13076

[74]. Xue, Z., Wang, Y., & Huo, M. (2026). LLM-Grounded Explainable AI for Supply Chain Risk Early Warning via Temporal Graph Attention Networks. In arXiv (Cornell University). Cornell University. https://doi.org/10.48550/arxiv.2603.04818

[75]. Zhang, L., Jia, T., Jia, M., Wu, Y., Liu, A., Yang, Y., Wu, Z., Hu, X., Yu, P. S., & Li, Y. (2025). A Survey of AIOps in the Era of Large Language Models. ACM Computing Surveys, 58(2), 1–35. https://doi.org/10.1145/3746635

[76]. Zhang, L., Jia, T., Jia, M., Wu, Y., Liu, H., & Li, Y. (2025). ScalaLog: Scalable Log-Based Failure Diagnosis Using LLM. 1–5. https://doi.org/10.1109/icassp49660.2025.10888670