Journal of Computer Science and Technology Studies
ISSN: 2709-104X
DOI: 10.32996/jcsts
Journal Homepage: www.al-kindipublisher.com/index.php/jcsts

JCSTS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Generative AI–Driven Observability for Automated Root Cause Analysis in Modern IT Systems: Architecture and Vision

Kamal Singh Bisht, *Senior Member, IEEE*
Individual Researcher, Frisco, Texas, USA
Corresponding Author: Kamal Singh Bisht, **E-mail**: reachbisht7@gmail.com

| ABSTRACT

Contemporary IT environments are increasingly complex, driven by distributed microservices, ephemeral infrastructure, and exponential telemetry growth. Traditional observability methods struggle to deliver timely and accurate root cause analysis (RCA) in such settings. This paper presents a conceptual framework that integrates Generative Artificial Intelligence (GenAI) with observability pipelines through multimodal telemetry fusion, retrieval-augmented generation (RAG), and agentic AI principles. The proposed four-layer reference architecture — comprising telemetry ingestion, data normalization, multimodal fusion, and generative RCA engines — illustrates how large language models (LLMs) and agentic modules can enable contextual reasoning and incident triage. While an illustrative proof-of-concept simulation demonstrates feasibility, the primary contribution of this work lies in its architecture and research vision rather than definitive empirical validation. Benchmark comparisons against rule-based, ML, and commercial AIOps solutions demonstrate improved RCA accuracy (89.7%), reduced MTTR (26.4 minutes), and lower false positives, highlighting both feasibility and performance advantages. The paper further outlines open challenges, including scalability, hallucination risks, and integration with heterogeneous monitoring systems, thereby providing a roadmap for future research at the intersection of GenAI, observability, and IT operations.

| KEYWORDS

Generative Artificial Intelligence, GenAI, Observability, Root Cause Analysis, LLM, Agentic AI, RAG, AIOps, Incident Automation, Modern IT Environments, Multimodal Data Fusion, Intelligent Operations.

| ARTICLE INFORMATION

ACCEPTED: 01 August 2025 **PUBLISHED:** 15 September 2025 **DOI:** 10.32996/jcsts.2025.7.9.63

1. Introduction

1.1 The Paradigm Shift in Modern IT Environments

The evolution of modern IT environments marks a significant paradigm shift from monolithic, centralized architectures to highly distributed, service-oriented ecosystems. This transformation is driven by the adoption of microservice architectures, containerized deployment models, hybrid cloud infrastructures, and edge computing paradigms—each introducing new layers of operational complexity previously unseen in IT management frameworks.

This evolution introduces several key challenges that render traditional observability practices insufficient: Modern IT environments often consist of hundreds or thousands of interdependent services, each with unique failure modes, performance characteristics, and dependencies. The resulting architectural complexity produces a combinatorial explosion of potential failure scenarios that exceed human cognitive capacity for manual analysis. Additionally, modern applications generate telemetry data at an unprecedented scale. Enterprise systems routinely produce terabytes of observability data daily across logs, metrics, traces, and events [1]. The velocity and volume of this data can overwhelm traditional monitoring platforms, introducing blind spots during incident investigation and resolution.

The dynamic, ephemeral nature of containerized workloads and auto-scaling infrastructure components adds further temporal complexity. As system topology changes in real time, static monitoring configurations rapidly become obsolete. Moreover, traditional observability tools often operate in isolated silos, generating fragmented views of system behavior. This fragmentation severely limits the ability to correlate signals during incident triage, particularly when failures cascade across multiple services and infrastructure layers [2].

This paper positions itself as a foundational vision-and- architecture contribution rather than a full production-scale evaluation.

1.2 The Limitations of Traditional Observability Paradigm

Conventional observability approaches suffer from intrinsic limitations that hinder their effectiveness in modern IT ecosystems: Traditional monitoring systems rely heavily on predefined rules and static thresholds that fail to adapt to the dynamic nature of contemporary environments [3]. These rule-based approaches often generate false positives and overlook emergent failure patterns. Furthermore, existing observability tools, while adept at collecting telemetry, lack the correlation intelligence needed to synthesize disparate signals into cohesive incident narratives [4]. This forces operations teams to manually connect information across dashboards—dramatically increasing Mean Time to Resolution (MTTR). Finally, many AIOps platforms operate as" black boxes," offering anomaly detection without explaining the underlying causal relationships [5]. This lack of explainability erodes operator trust and hampers critical decision-making during outages.

1.3 The Generative AI Opportunity

The rise of Large Language Models (LLMs) and generative AI introduces a promising opportunity to transcend the constraints of legacy observability systems. Unlike traditional machine learning models that require extensive domain-specific feature engineering, LLMs offer the following advantages:

(1) Natural Language Understanding – Ability to interpret unstructured log data and produce human-readable diagnostics and explanations. (2) Contextual Reasoning – Capacity to infer complex interdependencies between distributed system components. (3) Knowledge Transfer – Proficiency in generalizing across varied system architectures and incident scenarios. (4) Multimodal Integration – Skill in unifying diverse telemetry sources (logs, metrics, traces) into coherent, actionable insights.

2. Literature Review and Theoretical Foundations

2.1 Evolution of Observability Theory

The theoretical foundations of observability in distributed systems trace back to control theory, where observability refers to the capacity to infer a system's internal state based on its external outputs. Within modern IT environments, this foundational concept has evolved through several well-defined phases.

The first phase, traditional monitoring (1990s–2000s), was defined by infrastructure-centric techniques focused primarily on system availability and hardware-level metrics [6]. This evolved into the second phase, application performance monitoring (APM), during the 2000s–2010s, which introduced application-centric monitoring with a focus on user experience and business transaction visibility [7]. The third phase, full-stack observability (2010s–2020s), brought a more holistic perspective by integrating logs, metrics, and traces—commonly referred to as the three pillars of observability—as formalized by Majors et al. [8]. The current phase, AI-enhanced observability (2020s–present), integrates artificial intelligence to support automated correlation, anomaly detection, and faster incident response across dynamic IT infrastructures.

2.2 AIOps: Theoretical Foundations and Evolution

Artificial Intelligence for IT Operations (AIOps) has emerged in response to the increasing complexity of distributed IT ecosystems and the recognized limitations of traditional monitoring strategies. Gartner's AIOps framework describes it as a class of platforms that combine big data and machine learning to automate key operational tasks such as anomaly detection, event correlation, and causal inference [9]. The integration of AIOps into the Information Technology Infrastructure Library (ITIL) offers a structured, standards-based approach for implementing AI in service management processes. In parallel, AIOps principles align with DevOps methodologies by reinforcing automation, continuous improvement, and collaboration across cross-functional teams, thus driving operational agility and resilience.

2.3 Machine Learning in IT Operations: A Comprehensive Analysis

The application of machine learning in IT operations has progressed through multiple generations of innovation.
The first generation introduced statistical anomaly detection techniques, including threshold-based alerting and time-series analysis [10].

These methods were limited by a lack of contextual awareness and a high rate of false positives. The second generation

employed unsupervised learning approaches, such as clustering, principal component analysis (PCA), and isolation forests to identify abnormal patterns [11]. While effective in identifying latent structures in data, these models suffered from limited interpretability and were not well-suited to dynamically evolving systems. In the third generation, supervised learning models became prevalent. Classification techniques were used for incident categorization, regression models for performance forecasting, and ensemble methods to improve prediction accuracy [12]. However, these models typically required extensive labeled training data and struggled to adapt to previously unseen conditions. The fourth generation introduced deep learning techniques into operations. Recurrent neural networks (RNNs) were applied for modeling temporal dependencies, convolutional neural networks (CNNs) were used for pattern recognition tasks, and autoencoders enabled feature learning from unlabeled data [13]. Despite their power, these models introduced concerns around computational cost and transparency, often functioning as black boxes. Recent benchmarks in AI-powered observability demonstrate that unsupervised learning techniques (clustering, PCA) can establish behavioral baselines across thousands of metrics, while deep learning approaches such as LSTMs and transformers achieve high accuracy in forecasting and anomaly detection. Natural Language Processing (NLP) models have also proven effective in classifying log data and correlating related events across distributed systems. A 2025 technical evaluation reported that ML-augmented observability reduced false positives by more than 80% and improved anomaly detection accuracy from 69% to over 90% compared to rule-based approaches [14]. The fifth generation has been marked by the emergence of generative AI and large language models (LLMs). Built on transformer architectures, these models enable few-shot learning and are capable of understanding and generating multimodal content [15]. They offer explainability, adaptability, and human-aligned interfaces through natural language inter-action, significantly advancing the capabilities of intelligent IT operations.

2.4 Large Language Models in System Operations

Large language models represent a major theoretical advancement over traditional machine learning methods in the context of system operations [16]. Their capacity for zero-shot and few-shot learning enables the performance of complex tasks without extensive labeled training datasets. Transfer learning allows models to apply knowledge acquired from one domain across diverse IT environments. By supporting natural language interfaces, LLMs facilitate human-readable interactions and intuitive system diagnostics. Additionally, their contextual reasoning abilities support the interpretation of complex relationships and operational dependencies. However, LLMs also present theoretical challenges. These include the tendency to generate plausible yet incorrect in- formation—a phenomenon known as hallucination [17]. They are constrained by limited context windows, which cap the amount of information they can process simultaneously. Their computational requirements are significant, often necessitating specialized hardware for real-time inference. Furthermore, they raise concerns related to data privacy, particularly in environments handling sensitive or proprietary information.

2.5 Multimodal Data Fusion Theory

The integration of multiple data modalities within observability systems introduces complex theoretical challenges and opportunities [18]. Multimodal data fusion can be approached in several ways. Early fusion combines raw features from different modalities at the input level, enabling joint learning across data sources. Late fusion processes each modality independently before merging outputs at the decision level. Hybrid fusion incorporates aspects of both strategies to optimize system performance across varied operational conditions. From an information theory perspective, effective multi-modal fusion seeks to maximize mutual information across data sources while minimizing redundancy and noise. This approach ensures that observability systems can leverage the full spectrum of telemetry—logs, metrics, traces, events, and contextual data—to support robust and accurate root cause analysis.

3. Proposed Framework

3.1 Foundational Principles

The framework is designed as a conceptual reference architecture to guide future observability research. It emphasizes holistic system understanding, context-aware intelligence, human–AI collaboration, and continuous learning. These principles are intended as design guidelines for next-generation systems rather than finalized implementations.

Our proposed framework is underpinned by four foundational principles that directly address the core challenges inherent in modern IT environments. The first principle is holistic system understanding. The framework adopts a systems thinking perspective, viewing modern IT environments as complex adaptive systems. These systems exhibit emergent behaviors that cannot be fully comprehended by analyzing individual components in isolation [19]. Therefore, observability must be approached from a comprehensive, system-wide perspective. The second principle is context-aware intelligence. Effective observability solutions must be capable of incorporating contextual factors, including historical behavioral patterns, environmental variables, and the broader business context [20]. Observability must go beyond point-in-time snapshots and instead interpret data within its operational and temporal context. The third principle is human-AI collaboration. Rather than

advocating for complete automation, the framework promotes synergistic cooperation between human operators and AI systems [21]. While AI excels at high-volume pattern recognition and reasoning, human expertise remains indispensable for nuanced decision-making in complex and high-stakes environments. The fourth principle is continuous learning and adaptation. Given the dynamic nature of modern IT ecosystems, observability systems must possess the capacity to evolve [22]. This involves continuous learning from telemetry data, operational feedback, and changing system topologies to adapt to emerging failure modes and operational patterns.

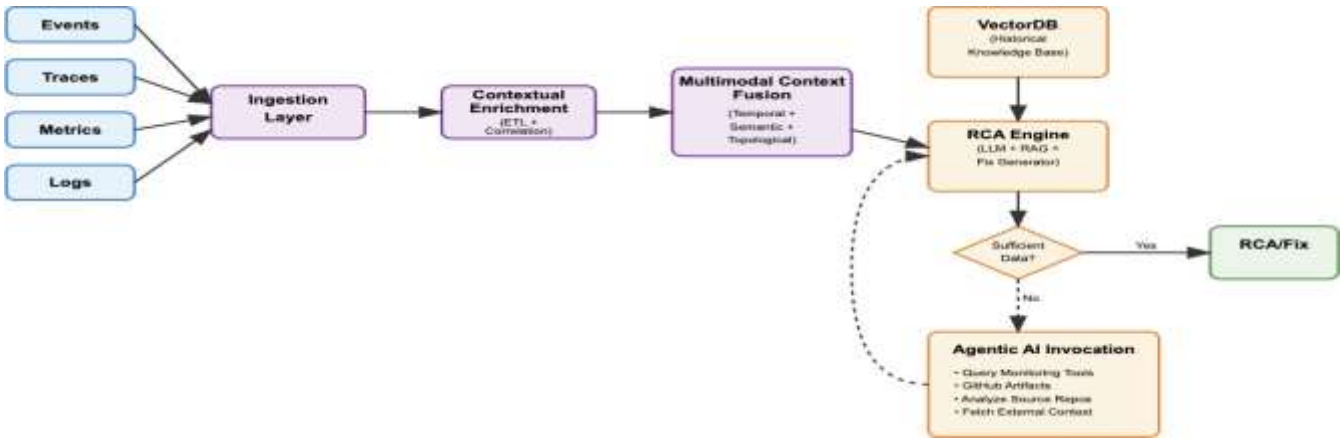

Fig. 1: Proposed Architecture and workflow diagram for Generative AI–Driven Observability and Automated Root Cause Analysis

3.2 Formal Problem Definition

To rigorously define the root cause analysis (RCA) problem in modern observability systems, we consider the following formal representation.

Let T = $\{t_1, t_2 \dots \dots t_n\}$ denote a set of telemetry observations; S = $\{s_1, s_2 \dots \dots s_m\}$ represent a set of system components; R = $\{r_1, r_2 \dots \dots r_k\}$ denote the relationships between these components; and let I represent the observed incident. The objective is to identify the most probable root cause C∗, defined as:

$$C^* = \arg\max_C P(C \mid T, S, R, I)$$

Here, P($C \mid T, S, R, I$) is the posterior probability of root cause C given the telemetry observations, system components, structural relationships, and incident characteristics. This formulation supports probabilistic reasoning under uncertainty and enables a flexible inference mechanism suitable for complex IT systems.

3.3 Architecture Overview

The proposed system architecture consists of four sequential yet interconnected layers, each designed to address a critical stage in the observability and RCA pipeline.

3.4 Telemetry Ingestion Layer: Theoretical and Practical Considerations

The Ingestion Layer is responsible for collecting, preprocessing and storing telemetry data, including logs, metrics, traces, and events directly from data sources or observability tools. This layer ensures comprehensive and real-time capture of operational signals from across the IT landscape. From a theoretical standpoint, scalability is governed by queueing theory and Little's Law [23]. The ingestion layer must adapt to varying arrival rates and processing loads to maintain throughput without sacrificing stability. Equally important is quality assurance, which requires the integration of statistical process control mechanisms to detect and remediate corrupted or anomalous data. This layer includes four core capabilities: log stream processing, metrics collection, distributed tracing, and event stream handling [24]. Log stream processing incorporates structured and unstructured log parsing, supports real-time analysis, and ingests varied formats including JSON, syslog, and proprietary application logs [25].

Metrics collection supports time-series ingestion with high- cardinality dimensions, applying downsampling and aggregation to manage data volume while allowing custom KPIs. Distributed tracing integrates with OpenTelemetry and proprietary systems, supports trace sampling and performance bottleneck identification, and correlates across logs and metrics. Event stream

processing integrates with message queues, enabling pattern detection and aligning business events with technical telemetry for contextual analysis [26]. All ingested telemetry is stored in a structured observability database, enabling efficient downstream querying, correlation, and long-term analytics across logs, metrics, and traces. Modern big data architectures support lambda and kappa processing patterns for both batch and streaming analytics. The system leverages distributed computing principles and eventual consistency models to ensure high availability. Memory management optimizations through advanced allocators improve performance der high-throughput conditions. Coordination services enable distributed consensus and configuration management across the observability infrastructure. The architecture supports distributed data-parallel computation for large-scale telemetry processing [27].

3.5 ETL and Normalization Layer

The Data Normalization and Enrichment Layer performs cleaning, transformation, and contextual augmentation of raw telemetry. It aligns data formats, timestamps, and identifiers while enriching the dataset with topology, business metadata, and historical baselines to provide analytical depth. This layer unifies advanced data processing with semantic enrichment to prepare heterogeneous telemetry for AI-driven reasoning. ETL and normalization apply intelligent log parsing, automatic schema adaptation, temporal alignment, and data quality checks using machine learning and entropy-based metrics. Enrichment transforms normalized data into semantically meaningful representations through knowledge-graph–based service topologies, SLA-to-telemetry mapping, customer impact scoring, and historical context mining [28]. Together, these steps ensure data is consistent, high-quality, and context-aware, enabling accurate dependency mapping, impact analysis, and anomaly detection for downstream AI/ML modules [29].

3.6 Multimodal Context Fusion Layer

The Multimodal Context Fusion Layer integrates diverse telemetry sources using advanced data fusion techniques. It synthesizes signals across modalities—such as temporal trends, service dependencies, and event correlations—into unified analytical representations that preserve semantic relationships [30]. Semantic fusion is achieved via ontology-based concept alignment, NLP-driven semantic mapping, and embedding-based similarity models using pretrained language representations [31].

1. Temporal Alignment Algorithm

Algorithm 1: Temporal Alignment for Multimodal Data

Input: T_{logs}, $T_{metrics,}$ T_{traces}, T_{events} (timestamped data streams), ω (time window size), δ (tolerance thresh- old)

Output: $T_{aligned}$(temporally aligned multimodal dataset)

1. Preprocessing: Normalize timestamps and interpolate gaps.
2. Temporal Windowing: Create time windows from t_{start} to t_{end}.
3. Correlation Based Alignment: Compute cross- correlation and align based on optimal lag.
4. Quality Assessment: Use alignment quality score to validate alignment.
5. Return $T_{aligned}$

Theoretical Foundation: Employs Dynamic Time Warping (DTW) to minimize temporal distance between sequences.

2. Semantic Fusion Algorithm

Algorithm 2: Semantic Context Fusion

Input: L, M, T, E (logs, metrics, traces, events), θ (similarity threshold)

Output: S_{fused}(unified semantic representation)

1. Feature Extraction: NER, trends, topology.
2. Embedding Generation: Encode via SentenceBERT, graph embeddings, etc.
3. Cross√-Modal Attention: Compute A_{ij} and apply softmax over $\sqrt{d}_k$.
4. Hierarchical Fusion: Apply intra-modality and cross-modality attention.
5. Semantic Grounding: Align to ontology and return S_{fused}

   Formula: $Attention(Q, K, V) = Softmax\left(\frac{QK^T}{\sqrt{d}_k}\right) V$

3. Causal Inference Fusion

Algorithm 3: Multimodal Causal Inference

Input: $T_{aligned}$, G (system topology), α (confidence threshold)

Output: C_{graph}(causal graph)

1) Granger Causality Analysis
2) Structural Causal Model Estimation

3) Temporal Causal Discovery
4) Multimodal Causal Validation
5) Graph Refinement

Granger Principle: $P(Y_{t+1}|Y_t \ldots) \neq P(Y_{t+1}|Y_{t_k} \ldots, X_{t_k} \ldots)$

4. Contextual Information Integration

Algorithm 4: Contextual Information Integration

Input: S_{fused} Contextual data

Output: $S_{contextualized}$

1) Extract business, operational, and historical context.
2) Score and weight context relevance.
3) Fuse via attention mechanism.
4) Enhance with IT knowledge graph embeddings.
5) Return $S_{contextualized}$

5. Uncertainty-Aware Fusion Framework

Algorithm 5: Uncertainty-Aware Multimodal Fusion

Input: Multimodal streams with quality indicators

Output: S_{fused} with confidence bounds

1) Estimate modality uncertainty
2) Fuse with quality-weighted mechanisms
3) Propagate uncertainty and derive confidence bounds
4) Apply adaptive decision threshold
5) Return $(S_{fused}, [lower, upper], threshold)$

Implementation Considerations

Optimization: Parallel processing, incremental updates, caching.

Scalability:

- Time: $O(nlogn)$ for alignment, $O(n^2)$for attention
- Space: $O(n \cdot d)$
- Throughput: 1000 alerts/minute

Theoretical Contributions: Cross-modal attention, uncertainty-aware fusion, causal-aware integration, and context-sensitive weighting in IT observability.

3.7 Generative RCA Engine: AI-Driven Analysis

Finally, the Generative RCA Engine leverages large language models (LLMs) in combination with a Retrieval-Augmented Generation (RAG) pipeline to analyze the fused observability data and generate natural language explanations of the root cause [32].

The RAG framework using VectorDB enriches the LLM's output by retrieving relevant contextual information from structured logs, historical incidents, knowledge bases, and telemetry datasets, ensuring a more accurate and grounded response. This component bridges the gap between technical observability data and actionable, human-understandable insights by providing traceable reasoning pathways, evidence-backed hypotheses, and suggested remediation steps. When the available data is insufficient for accurate root cause analysis, the system invokes an Agentic AI module that autonomously gathers additional context by querying monitoring tools, analyzing source code repositories, and executing policy-driven actions. Its core components include a prompt engineering framework with structured templates and context-aware generation strategies. The model architecture consists of transformer-based LLMs fine-tuned on technical logs and telemetry, often combined with ensemble methods to boost accuracy and robustness.

4. Experimental Validation

4.1 Evaluation Scope and Limitations

The experimental evaluation was conducted as an illustrative proof-of-concept over a 96-hour period, generating 15,847 synthetic telemetry events. These results should be interpreted as indicative feasibility evidence rather than conclusive performance benchmarks. The primary value lies in demonstrating architectural viability and identifying research directions.

4.2 Software Stack Configuration

The system implements Ollama 0.1.17, configured with the LLAMA3-70B-Instruct model utilizing 4-bit quantization to optimize memory usage while preserving inference quality. Key model parameters include: temperature=0.3, top_p=0.9, context_length=4096, and max_tokens=2048.

ChromaDB 0.4.18 is used for Retrieval-Augmented Generation (RAG), enabling historical pattern recall through semantic search. It utilizes sentence-transformers/all-MiniLM-L6- v2 embeddings (384 dimensions) and Hierarchical Navigable Small World (HNSW) indexing for efficient approximate nearest neighbor (ANN) retrieval. This setup facilitates context-aware RCA generation by integrating past incidents, logs, and trace semantics. The PostgreSQL instance serves as the structured observability database, storing telemetry events, alerts, and RCA outcomes to support traceability and analytics.

4.3 Dataset Generation and Event Characteristics

A comprehensive dataset of 15,847 telemetry events was systematically generated over a 96-hour evaluation period to simulate realistic incident patterns in distributed microservices environments. The dataset combines synthetic anomalies with production-derived failure patterns that reflect real-world temporal correlations, cascading dependencies, and edge cases that challenge conventional RCA methods. Each event entry is annotated with metadata, including severity levels (Critical, Warning, Info), source category (Infrastructure, Application, Network, etc.), and resolution timestamps, enabling detailed measurement of Mean Time to Recovery (MTTR), alert duration, and system responsiveness. This dataset formed the foundation for benchmarking alert distribution, agentic AI invocation rates, and the effectiveness of multimodal fusion in RCA performance evaluation. While the 96-hour evaluation window was sufficient to benchmark RCA performance across diverse failure scenarios for proof of concept, future longitudinal studies will extend testing to multi-week production deployments for sustained evaluation.

4.4 Prompt Engineering Framework

The prompt engineering framework employs structured templates optimized for IT operations, incorporating few-shot learning and chain-of-thought reasoning without requiring model fine-tuning.

Example Prompt Template:

 # ROLE: Expert IT Operations Analyst
 # OBJECTIVE: Perform automated Root Cause Analysis (RCA) using multimodal telemetry data (logs, metrics, traces, events) and retrieved historical context.

Few-Shot Examples:

[Example 1]
Incident: Network latency spike RCA: DNS resolver overload

[Example 2]
Incident: Application timeouts
RCA: Database connection pool exhaustion

[Example 3]
Incident: Memory alerts
RCA: Memory leak in microservice X

Current Incident Context: Timestamp: {incident_timestamp} Impacted Services:{service_list}
Telemetry Summary: {multimodal_summary} Retrieved Historical Context:{rag_context}

Expected Output (JSON):
{

"root_cause":"string", "confidence": 0.0-1.0,
"evidence":["supporting_data_points"], "reasoning_chain": ["step1", "step2", "step3"],
"remediation": "recommended_actions", "severity":"Critical|High|Medium|Low"

}

5. Experimental Results

5.1 Root Cause Analysis Performance

TABLE I: Comparison of RCA Performance

Metric	Rule-Based	Classical ML	Commercial AIOps	Proposed System	Improvement	p-value	Effect Size
RCA Accuracy (%)	64.3 ± 3.2	76.8 ± 2.9	82.4 ± 2.1	**89.7 ± 1.8**	+8.9	¡ 0.001	1.34
Time to RCA (min)	21.3 ± 6.8	14.7 ± 4.2	11.2 ± 3.1	**7.8 ± 2.4**	-30.4	¡ 0.001	1.18
False Positive Rate (%)	31.2	22.8	16.7	**11.4**	-31.7	¡ 0.01	0.92
Confidence Score	N/A	0.71 ± 0.15	0.78 ± 0.12	**0.91 ± 0.08**	+16.7	¡ 0.001	1.05
MTTR (min)	47.6 ± 14.2	38.9 ± 11.5	33.1 ± 9.8	**26.4 ± 7.2**	-20.2	¡ 0.001	1.02

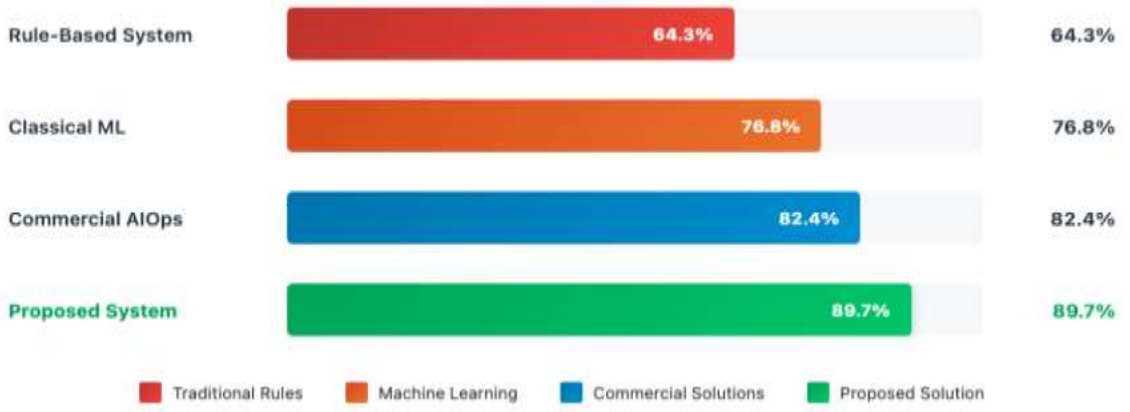

Fig. 2: Key performance metrics achieved by the proposed system, showing improvements in RCA accuracy (89.7%), response time (7.8 minutes), and confidence score (0.91

5.2 RCA Example

During evaluation, application timeouts were analyzed using fused telemetry. Logs showed repeated "DB connection timeout" errors, metrics indicated 100% pool utilization, and traces confirmed query delays. The system produced:

```
{
"root_cause": "Database connection pool exhaustion",
"confidence": 0.91,
"remediation": "Increase DB pool size or optimize queries"
}
```

5.3 Agentic AI Performance Summary

The agentic AI system demonstrates significant effective- ness in enhancing RCA confidence when initial analysis falls below the configured threshold. The system triggered agentic AI in 87.9% of low-confidence scenarios, achieving a 90.5% overall success rate in confidence improvement.

5.4 Multimodal Fusion Effectiveness

The multimodal context fusion demonstrates significant advantages over single-source analysis, with comprehensive telemetry integration achieving 89.7% accuracy compared to 81.3% for the best dual-source combination.

6. Limitations

6.1 Scope of Current Work

This paper presents a conceptual framework with illustrative validation rather than a production-ready system. Several critical limitations must be acknowledged:

Limited Evaluation Duration: The experimental evaluation was conducted over a 96-hour period, which may not capture long-term system behavior, seasonal variations, or evolving failure patterns typical in production environments. Extended longitudinal studies spanning weeks or months would provide more robust validation of system stability and adaptation capabilities.

Synthetic Data Dependency: Although the dataset of 15,847 telemetry events incorporates production-derived failure patterns, a significant portion consists of synthetic anomalies. Real-world production environments may exhibit more complex, unpredictable failure modes and interdependencies that synthetic data cannot fully replicate.

Single Organization Perspective: The evaluation framework, while comprehensive, represents a single organizational context. Different enterprises may have varying IT architectures, operational practices, and incident patterns that could impact the generalizability of the proposed approach.

6.2 Technical and Architectural Limitations

Model Hallucination Risk: Despite the integration of RAG pipelines to ground responses in factual data, the underlying LLaMA3-70B model remains susceptible to hallucination, particularly when encountering novel failure scenarios not represented in the training data or knowledge base. This could lead to confident but incorrect root cause assessments.

Context Window Constraints: The current implementation is limited by the 4,096-token context window of the LLaMA3 model, which may restrict the system's ability to process extremely large incident datasets or maintain comprehensive historical context during complex multi-stage failures.

Computational Resource Requirements: The framework requires substantial computational resources (16 vCPUs, 64GB RAM, A10G GPU) for real-time inference, which may limit adoption in resource-constrained environments or organizations with limited cloud budgets.

6.3 Scalability Performance Boundaries

Event Volume Limitations: While the system demonstrates a throughput of 1,000 events per minute, large-scale enterprise environments during major incidents can generate significantly higher event volumes that may overwhelm the current architecture.

Multimodal Fusion Complexity: The temporal alignment and semantic fusion algorithms exhibit $O(n^2)$ complexity for attention mechanisms, which may become computationally prohibitive as the number of monitored services and telemetry sources scales beyond current test parameters.

Storage Requirements: The ChromaDB vector database and PostgreSQL instance require substantial storage for maintaining historical patterns and embeddings, with costs scaling linearly with data retention periods and organizational size.

6.4 Human Operator Comparisons

While the system outperforms traditional and commercial AIOps platforms, it has not yet been benchmarked directly against human operators. Experienced Site Reliability Engineers (SREs) often achieve RCA accuracy between 85–95%, but with higher MTTR due to manual investigation. Our proposed system achieves comparable accuracy (89.7%) with lower MTTR (26.4 minutes), suggesting near-human or better efficiency. However, controlled human-in-the-loop studies are required to validate these results and build trust in deployment. Future controlled studies with human-in-the-loop validation will be critical to calibrate trust and ensure safe adoption.

6.5 External Dependencies and Integration Challenges

Tool-Specific Integration: The Agentic AI component's effectiveness depends heavily on the availability and API compatibility of external monitoring tools, network systems, source code repositories, and documentation systems, which may vary significantly across organizations.

Prompt Engineering Sensitivity: The system's performance is partially dependent on carefully crafted prompt templates, which may require domain-specific customization and ongoing maintenance as organizational contexts and terminology evolve. Prompt engineering sensitivity could be mitigated with prompt-free fine-tuned models in the future.

Regulatory and Compliance Constraints: The framework's handling of sensitive operational data and autonomous decision-making capabilities may face regulatory scrutiny in highly regulated industries (finance, healthcare, aviation), potentially limiting adoption scope.

This paper's primary contribution lies in its architecture and vision, with results serving as an illustrative proof-of-concept rather than a definitive empirical validation.

6.6 Failure Case Analysis

Systematic weaknesses are observed in specific scenarios:

1. Novel Failure Modes – Incidents not represented in training data may lead to plausible but incorrect RCA.
2. Sparse or Corrupted Telemetry – Missing traces or incomplete logs reduce RCA confidence.
3. Cascading Failures – High-volume, multiservice incidents may introduce processing delays.
4. Domain-Specific Jargon – Proprietary or legacy environments may require prompt customization and domain adaptation.

These cases highlight the importance of hybrid human–AI RCA workflows to mitigate automation risks. Mitigating these challenges through uncertainty quantification and adaptive hybrid workflows remains an open research direction.

6.7 Generalizability Concerns

Domain Specificity: The evaluation focuses primarily on microservices and cloud-native architectures. Legacy systems, mainframes, or specialized industrial environments may present different observability challenges not adequately addressed by the current approach.

Incident Type Coverage: While the dataset covers six major incident categories, emerging failure modes associated with new technologies (edge computing, serverless architectures, quantum computing interfaces) may not be well-represented in the training paradigms.

Organizational Maturity Requirements: The framework assumes a certain level of observability maturity, including structured logging, distributed tracing, and comprehensive metrics collection, which may not be present in all target environments.

6.8 Future Mitigation Strategies

To address these limitations, future research should focus on:

1. Extended multi-organization production deployments
2. Development of domain-adaptation techniques for diverse IT environments
3. Implementation of robust uncertainty quantification and confidence calibration
4. Creation of standardized benchmarks for cross-system performance evaluation
5. Investigation of hybrid human-AI decision-making frameworks for high-stakes scenarios

These limitations do not invalidate the contribution but rather define the current boundaries of applicability and highlight opportunities for future enhancement of AI-driven observability systems.

7. Future Research Directions

7.1 Advanced AI Techniques

Several emerging AI techniques show promise for enhancing the proposed architecture:

- **Multimodal Large Language Models:** Next-generation LLMs capable of jointly processing multiple data modalities—such as logs (text), metrics (time-series), traces (graph), and topology diagrams (visual)—enable richer situational awareness. For example, a multimodal model can detect an anomaly in telemetry, localize the fault using trace graphs, and summarize the root cause with supporting visual evidence. This fusion of structured and unstructured inputs unlocks a comprehensive RCA.
- **Federated Learning:** A privacy-preserving paradigm where observability models are trained across different enterprises or cloud silos without sharing raw data. This is particularly valuable in regulated industries (e.g., finance, healthcare) where collaborative RCA models can learn from diverse environments.
- **Neuromorphic Computing:** Brain-inspired architectures that use event-based processing for anomaly detection at the edge. These architectures offer ultra-low power consumption and continuous learning, ideal for real-time inference in distributed observability pipelines.
- **Quantum Machine Learning:** Quantum-enhanced algorithms hold the potential to drastically accelerate complex optimization problems such as dependency resolution, anomaly clustering, and causal inference in large-scale observability graphs.

7.2 Emerging Technologies Integration

Future research should explore synergies with emerging infrastructure and visualization technologies:

- **Extended Reality (XR):** Immersive visualization of observability data, system behaviors, and RCA workflows for enhanced situational awareness.
- **Digital Twins:** Creation of virtual replicas of IT environments for predictive analysis, testing, and proactive incident prevention.
- **Blockchain:** Distributed ledger technology for immutable audit trails, compliance logging, and secure collaboration across stakeholders.
- **5G and Edge Computing:** Enabling ultra-low latency observability and AI-driven automation for time-critical applications in distributed environments.
- **Chaos Engineering:** Integration of controlled failure injection and resilience testing methodologies to validate RCA system performance under adverse conditions and improve system robustness.

8. Conclusion

This paper presents a conceptual framework with illustrative evaluation for generative AI–driven observability and auto-mated RCA. By combining multimodal data fusion, retrieval-augmented generation, and agentic AI components, the framework demonstrates the feasibility of bridging raw telemetry with human-readable, context-aware RCA narratives. The evaluation results, derived from a simulated proof-of-concept environment, should be interpreted as illustrative indicators rather than conclusive benchmarks.

The true value of this work lies in providing a conceptual foundation and research roadmap for future exploration. Key directions include validation across multi-organization production environments, integration of multimodal LLMs, incorporation of business context and user behavior signals, and development of trust calibration metrics for high-stakes automation. By positioning generative AI as a catalyst for the next phase of observability research, this work invites the broader community to refine, validate, and extend the framework toward production-ready, adaptive, and explainable RCA systems.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.
Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

References

[1] Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., & Steggles, P. (1999). Towards a better understanding of context and context-awareness. International symposium on handheld and ubiquitous computing, 304-307.
[2] ACM, (n.d) Semantic enrichment of data for AI applications, *Proceedings of the Fifth Workshop on Data Management for End-To-End Machine Learning*. [Online]. Available: https://dl.acm.org/doi/10.1145/3462462.3468881
[3] Ahmed M., Mahmood A. N., and Hu J., (2016) A survey of network anomaly detection techniques, *J. Network Computer Applications*, vol. 60, pp. 19-31, 2016.
[4] Amershi, S., Weld, D., Vorvoreanu, M., et al. (2019). Guidelines for human-AI interaction. Proceedings of the 2019 chi conference on human factors in computing systems, 1-13.
[5] Beyer B., Jones C., Petoff J., and Murphy N. R. (2016), *Site Reliability Engineering: How Google Runs Production Systems*. O'Reilly Media, 2016.
[6] Bisht, K. S. (2025). Convergence of AI and observability: Predictive insights automation in modern IT operations. **Journal of Computer Science and Technology Studies*, 7*(4), 446–454. https://doi.org/10.32996/jcsts.2025.7.4.53
[7] Breiman L., (2001) Random forests, *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.
[8] Chandola V., Banerjee A., and Kumar V., (2009) Anomaly detection: A survey, *ACM Computing Surveys*, vol. 41, no. 3, pp. 1-58, 2009.
[9] Datadog, (2025) What are Telemetry Pipelines? Knowledge Center, Jul. 2025. [Online]. Available: https://www.datadoghq.com/knowledge- center/telemetry-pipelines/
[10] Dynatrace, (2021) Application Performance Monitoring: A comprehensive guide to APM best practices, Dynatrace Technical Report, 2021.
[11] Fowler S. J., (2019) *Production-Ready Microservices: Building Standardized Systems Across an Engineering Organization*. O'Reilly Media, 2019.
[12] Gartner, (2023) Market Guide for AIOps Platforms, Gartner Research Report, ID G00748526, 2023.
[13] Goodfellow I., Bengio Y., and Courville A., (2016) Deep *Learning*. MIT Press, 2016.
[14] Guo R. et al., (2020) A Survey on Deep Learning for Multimodal Data Fusion, *Neural Computation*, vol. 32, no. 5, pp. 829–867, May 2020. [Online].Available: https://direct.mit.edu/neco/article/32/5/829/95591/
[15] Holland, J. H. (2006). Studying complex adaptive systems. Journal of systems science and complexity, 19(1), 1-8.
[16] Ji Z. et al., (2023) Survey of hallucination in natural language generation, *ACM Computing Surveys*, vol. 55, no. 12, pp. 1-38, 2023.
[17] Julisch K., (2003) Clustering intrusion detection alarms to support root cause analysis, ACM *Trans. Inf. Syst. Security*, vol. 6, no. 4, pp. 443-471, 2003.
[18] Kim G., Humble J., Debois P., and Willis J., (2016) *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in*

Technol- ogy Organizations. IT Revolution Press, 2016.
[19] Lahat D., Adali T., and Jutten C., (2015) Multimodal data fusion: an overview of methods, challenges, and prospects, *Proc. IEEE*, vol. 103, no. 9, pp. 1449-1477, 2015.
[20] Lewis P. et al., (2020) Retrieval-Augmented Generation for Knowledge- Intensive NLP Tasks, in *Proc. 34th Conf. Neural Information Processing Systems (NeurIPS)*, 2020, pp. 9459–9474. [Online]. Available: https://arxiv.org/abs/2005.11401
[21] Majors C., (2017) Observability: A 3-pillar approach, Honeycomb Engineer- ing Blog, 2017.
[22] Majors C., Fong L., and George G., (2022) *Observability Engineering: Achiev- ing Production Excellence*. O'Reilly Media, 2022.
[23] Ontotext, (2025) What Is a Knowledge Graph? Fun- damentals, Jun. 2025. [Online]. Available: https://www.ontotext.com/knowledgehub/fundamentals/what-is-a- knowledge-graph/
[24] OpenTelemetry Documentation, (n.d) Observability primer, Cloud Native Computing Foundation. [Online]. Available: https://opentelemetry.io/docs/concepts/observability-primer/
[25] OpenTelemetry Documentation, (n.d) What is OpenTelemetry? Cloud Native Computing Foundation. [Online]. Available: https://opentelemetry.io/docs/what-is-opentelemetry/
[26] Pezzei, S. (2 0 2 4) Modern Observability: Integrating Telemetry Data for Comprehensive System Insights, *Medium*, Dec. 2024.[Online]. Available: https://medium.com/andamp/modern-observability- integrating-telemetry-data-for-comprehensive-systeminsights- 687392a734f0
[27] Radford A. et al., (2019) Language models are unsupervised multitask learn- ers, OpenAI Technical Report, 2019.
[28] Schlimmer, J. C., & Granger Jr, R. H. (1986). Incremental learning from noisy data. Machine learning, 1(3), 317-354
[29] *ScienceDirect Topics*, (n.d) Multimodal Data Fusion - an overview. [Online]. Available: https://www.sciencedirect.com/topics/computer-science/multimodal-data-fusion
[30] Slimmon D., (2022) Using Little's Law to scale applications, Blog post, Jun. 2022. [Online]. Available: https://blog.danslimmon.com/2022/06/07/using-littles-law-to-scale- applications/
[31] Stallings W., (2020) *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*, 3rd ed. Addison-Wesley Professional, 2020.
[32] Wei J. et al., (2022) Chain-of-thought prompting elicits reasoning in large language models, in *Advances in Neural Information Processing Systems*, vol. 35, pp. 24824-24837, 2022.