Journal of Information Technology and Applications Research
Volume 1 Nomor 2 November 2025
e-ISSN: 3089-8013; p-ISSN:3089-8021, Page 183-194
DOI: https://doi.org/10.63956/jitar.v1i2.66
Available online at: https://jitar.org/index.php/jitar/article/view/66

LLM-Driven Root Cause Analysis for Distributed System Incidents: A Reproducible Framework

Akhil Reddy Mandadi
Independent, India

Abstract
In Distributed Systems, Root Cause Analysis (RCA) is a complex and time-sensitive task that demands correlating a variety of telemetry data such as logs, metrics, and distributed traces. In traditional contexts these activities are fundamentally dependent on the skills of Site Reliability Engineers (SREs) and incident diagnosis can be time-consuming and challenging, and becomes increasingly difficult as applications move to increasingly complex cloud-native contexts. The opportunities for automated reasoning by recent advances in Large Language Models (LLMs) are also accompanied by methods for automated RCA based on LLMs, which are often limited to unstructured summaries and have the potential for hallucination and unreliable conclusions. VerifiedRCA proposes an extensible and reproducible approach combining LLM reasoning, structured telemetry retrieval and tool-call verification to ensure trustworthy automated incident diagnosis. The proposed architecture consists of four layers of a pipeline: telemetry ingestion, retrieval-augmented context construction, hypothesis generation and evidence-based verification. Synthetic scenarios were injected (120 in total) into a realistic micro services environment built using Google Hipster Shop, Istio service mesh and Chaos Mesh fault injection for evaluation. Using the accuracy, time-to-diagnosis and explanation quality metrics, and performance was compared to rule-based systems, classical machine learning approaches, and ungrounded LLM baselines. The results show that grounding the LLM's reasoning with telemetry verification greatly enhances the diagnostic accuracy, with higher precision, more rapid convergence, and a decrease in unsupported reasoning paths. This study provides a clear RCA framework and an open benchmark that can facilitate future research and development in operational intelligence with the help of LLMs and support reproducible research.

Keywords : *Root Cause Analysis; Large Language Models; Distributed Systems; Observability; AIOps; Site Reliability Engineering; Chaos Engineering; Retrieval-Augmented Generation*

INTRODUCTION

The world is now full of software systems that operate in distributed architectures and massive systems, making the management of these systems more complex. The rapid adoption of cloud-native architectures and large-scale distributed systems has greatly complicated the operational model of the software environment. Modern enterprise environments typically contain dozens of tightly coupled micro services that generate a tremendous amount of telemetry data, such as logs, metrics, traces, alerts, and system events, which are often diverse in type. While these observability signals can shed light on the condition of the system, determining the root cause of service failures is a challenging and time-critical problem [5, 20].

In the context of Site Reliability Engineering (SRE) and AIOps, Root Cause Analysis (RCA) is a crucial step for engineers who need to pinpoint the root cause of failures and suggest fixes. RCA is a task that involves multiple telemetry modalities, dependency analysis of the services, and operational testing of alternate diagnoses [5, 12]. In traditional manual RCA, an increasing number of alerts, overloaded telemetry, and long

Received: October 15, 2025; Accepted: November 22, 2025; Published: February 05, 2026
*Corresponding author : akhilreddymandadi95@gmail.com

incident investigation cycles make it increasingly difficult to maintain. As distributed systems grow larger and more dynamic, traditional manual RCA becomes increasingly challenging because of alert fatigue, telemetry overload, and long incident investigation cycles.

Most of the existing RCA systems are based on rules or machine learning techniques. Rule based methods use rules with predefined thresholds and with handcrafted detection patterns for known faults. These approaches are transparent and deterministic, but may have some drawbacks in handling previous unknown incident patterns and changes in system configuration [7]. Machine learning techniques, such as supervised classifiers and anomaly detection models, are able to learn the patterns of past incidents and increase adaptability, but often they make predictions without providing interpretable explanations or reasoning paths, or even evidence-based explanations [9].

With the latest developments of Large Language Models (LLMs), new possibilities for intelligent incident diagnosis have emerged. LLMs are well-equipped with the features of multi-step reasoning, natural language explanation generation, and contextual understanding, which make them promising candidates for automated RCA systems [1], [8], [19]. Recent research has shown that LLMs can be used to solve the incident triage, failure diagnosis, and mitigation recommendation problems in cloud infrastructures [1] [19]. Ahmed et al. compared LLM-based systems with over 44,000 cloud incidents, and found that the LLM-based systems outperformed other models in both the root cause analysis and the mitigation recommendation tasks [1]. Likewise, Wang et al. showed that LLMs can offer reliable and comprehensible assistance for operational incident triaging [19].

The current RCA systems based on LLM have significant reliability issues, however. The majority of the methods work on the principle of summarizing the telemetry data and sending a brief context to an LLM for reasoning. This can lead to plausible, but incorrect, results, since conclusions generated from the system may not be supported by a verifiable system evidence [15]. This can be especially important in operational environments where incorrect diagnoses can add to the mean time to resolution (MTTR) and result in poor remediation decisions.

Recently, diagnostic architectures using tools and agents have thus been studied. To enhance factual grounding and incident reasoning, RCAgent proposed autonomous reasoning agents which can retrieve telemetry data [14]. For autonomous remediation frameworks, intelligent operational management and adaptive decision-making, on the other hand, have been demonstrated through LLM-based argentic systems [2, 16]. With the development of structured reasoning and evidence retrieval, more and more RCA frameworks are emerging in graph-enhanced and multi-agent formats, which further illustrate the necessity of structured reasoning and evidence retrieval in complex fault environment [3] and [18] respectively.

Research Gap

While recent LLM-based RCA frameworks show promising results, there are key challenges that need to be addressed. Current methods have often relied on a summary of telemetry data and are often lacking in clear means of checking hypothesized events against retrieved operational data [14, 15]. Moreover, the available reproducible benchmarking environments are still quite limited and it is difficult to compare the proposed approaches directly [7]. Existing systems also fail to perform well in multi-

cause incidents, noisy telemetry environments and in dynamic fault scenarios frequently found in cloud-based environments.

Proposed Framework

To overcome these challenges, this paper proposes a framework, VerifiedRCA, to enable evidence-based and reproducible LLM-based distributed system incident RCA. VerifiedRCA builds on previous solutions that just summarize and generate hypotheses, adding a verification layer to allow the LLM to verify generated hypotheses by interacting with real-time telemetry retrieval systems, such as querying metrics, searching logs, inspecting traces, or analyzing topology.

This proposed framework is built on four layers including the ingestion and retrieval of the telemetry, the building of context with the help of the retrieval, the reasoning with the structured LLMs, and the verification of the results of the tool calls. The framework is designed to increase the reliability of diagnostics, minimize unsupported reasoning paths and increase transparency of investigation when using automated tools.

Main Contributions

This study has a number of important contributions that are summarized as follows:

1. A four-layer LLM-based system that leverages explicit telemetry verification mechanisms to ground RCA reasoning.
2. 120 labeled incident scenarios and a reproducible synthetic incident generation environment, created using Google Hipster Shop, Istio service mesh and Chaos Mesh fault injection.
3. A thorough comparative assessment, based on RCA accuracy, time-to-diagnosis, and explanation quality, with rule-based systems, classical machine learning models, and ungrounded LLM baselines.
4. An empirical study of the limitations of the framework such as multi-fault scenarios, stale telemetry conditions and verification failures.

BACKGROUND AND RELATED WORK

Root cause analysis for distributed systems can be difficult due to the fact that for systems with multiple components, multiple failures may yield similar outward effects of the system failure, but completely different remedies. Thus, cross-log, cross-trace, cross-metric and cross-service dependency correlation is required for effective diagnosis. [5]

There are two extremely popular approaches to traditional RCA systems, which are the rule-based approach and the machine learning approach. In rule-based systems, the expert's knowledge is captured in the form of manually created detection patterns. Such systems are efficient in computation, but are hard to deal with when dealing with new fault patterns and dynamically changing environments [7]. By learning patterns, machine learning techniques enhance adaptability but often, they lack interpretability and diagnostic transparency [9].

LLM-based RCA systems have been the subject of growing research interest recently. To address the factual grounding problem, RCAgent used tool-augmented autonomous agents to assist cloud incident analysis, achieving better factual grounding than generative approaches [14]. Roy also demonstrated that agent-based architectures are more effective than retrieval only architectures for out of distribution incident scenarios [15].

Large-scale studies by Ahmed et al. showed that LLMs can effectively be used for the management and mitigation of cloud incidents and the creation of recommendations, respectively [1]. Likewise, Wang et al. demonstrated that outputs of incident triage generated by the LLM can be interpretable and useful [19].

In recent years, several studies have investigated the application of graph-based RCA systems [18], multi-agent collaborative RCA systems [3] and reliability-oriented LLM ecosystems [6]. The studies have shown that future RCA systems should be more grounded in evidence, offer better verification systems, and have more reproducible evaluation systems.

Building on these observations, VerifiedRCA uses retrieval augmentation and tool-based verification in an evidence-centered reasoning framework that aims to enhance trustworthiness and reproducibility.

THE VERIFIEDRCA FRAMEWORK

Ultimately, the core idea behind VerifiedRCA was that language models should use the evidence they see to confirm the diagnosis as well as come up with one. Previous research shows that LLM can provide plausible yet inaccurate explanations during operational incident analysis [14, 15, and 19]. Thus, instead of being an after-thought, VerifiedRCA makes evidence verification a fundamental building block of the architecture.

The structure is a four-layer design that is expected to be deployed in a distributed environment, such as Kubernetes, with standard cloud observability infrastructure. The architecture enables the collection and retrieval of telemetry, generation of structured reasoning and retrieval-augmented context, and tool-assisted verification in order to facilitate trustworthy automated RCA.

Telemetry Ingestion and Normalization Layer

The first architectural layer is for acquiring and normalizing the telemetry. Logs from multiple services are gathered using Loki and, if possible, structured. The Prometheus metrics are scraped at regular intervals, and request rates, service latency statistics, and error metric are automatically added by Istio sidecar proxies. Traces are captured with Traceability in Tempo using service-mesh level context propagation.

It is a multi-signal observability model, which has been advocated by recent studies highlighting the need for correlating telemetry analysis for incident understanding and RCA performance [5], [8], [20].

Telemetry signals from several layers of the system are brought together to a common incident representation, which enables downstream retrieval and reasoning, before diagnosis is started. The overall architecture and information flow of the VerifiedRCA is shown in **Figure 1.**

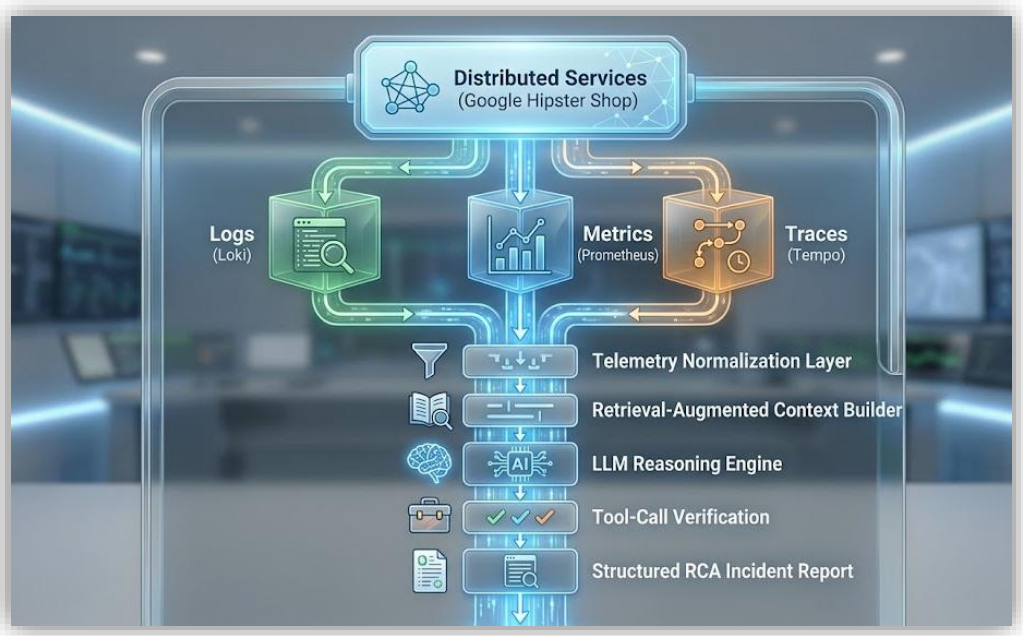

Figure 1. Overall VerifiedRCA architecture and telemetry processing workflow.

Figure 1 shows the flow of the telemetry information to the four-layer reasoning architecture from distributed services. Logs, metrics and traces are rolled-up, normalized and then retrieved and reasoned. The structured evidence then goes through verification stages and finally, a final incident diagnosis is generated.

Retrieval-Augmented Context Construction

The framework (after the alerting) creates an enriched diagnostic context for the LLM. Retrieve context includes alert definition, trends for metrics in past 30 minutes, log snippets from affected services, and semantically like previous incidents retrieved using vector similarity search.

Retrieval augmentation has been shown to be very effective for improving contextual grounding and avoiding reasoning drift in operational systems augmented by LLMs [14, 19]. Logically sound support for reasoning under unusual fault conditions comes from similar historic incidents.

Structured LLM Reasoning Engine

The third layer is a structured reasoning engine that is used to generate ranked diagnostic hypotheses.

The model is trained to recognize:

1. Candidate root causes
2. Supporting evidence
3. Contradictory evidence
4. Information needed for validation by telemetry.

This is explicit reasoning structure that eliminates the assumption and promotes diagnostic action that is interpretable. In incident triage and cloud RCA systems, similar agentic reasoning strategies have been proven to be more effective [2, 14].

Verification Layer for Tool Calls

The verification layer is the main contribution of VerifiedRCA.

After generation of hypotheses, the framework allows to access seven external verification tools:

1. Prometheus metric queries
2. Loki structured log retrieval
3. Tempo trace retrieval
4. Service dependency lookup
5. Historical incident similarity search
6. Run book retrieval
7. Isolation Forest anomaly scoring

Recently, systems with tools have been shown to achieve better grounding performance than generation-only systems [14], [2] and [18].

Theories that fail to be supported by retrieved evidence are lowered in status; theories that are supported are progressed through the diagnostic stages which follow.

In order to assess the role of verification, an ablation experiment was performed, which eliminated tool assisted validation without changing any of the other architectural components. The difference in performance is summarized in **Table 1.**

Table 1. Ablation Study Of Tool-Call Verification

Configuration	Top-1 Accuracy	Top-3 Accuracy
VerifiedRCA	79.2%	91.4%
Without Verification Layer	61.0%	78.3%
Performance Difference	+18.2%	+13.1%

The results show that the use of evidence verification significantly enhances diagnostic accuracy. Removing verification leads to a drop in accuracy, indicating that the main advantage is not increased language generation but does not necessarily lie in more systematic grounding of evidence.

EXPERIMENTAL SETUP AND BASELINES

Evaluation of RCA frameworks is still difficult, as production incident data sets are frequently proprietary and hard to replicate [7]. With this limitation in mind, the following synthetic benchmark environment was created: Google Hipster Shop, Istio service mesh, and Chaos Mesh fault injection.

There are 120 scenarios with labels for each of six fault categories in the benchmark:

1. Service Unavailability
2. Latency Injection
3. Resource Exhaustion

4. Dependency Failures
5. Network Partition
6. Configuration Failures

Reproducible experimentation and comparison of diagnostic systems are facilitated by the architecture.

Table 2. Comparison Of Evaluated Systems

System	Telemetry Input	Grounding Mechanism	Output
Rule-Based	Metrics + alerts	Threshold rules	Label
Random Forest	Engineered features	Supervised learning	Label + feature importance
GPT-4 Ungrounded	Telemetry summaries	None	Free-text
VerifiedRCA	Logs + Metrics + Traces	Tool verification	Structured report

This baseline selection is used as practical alternatives to the operational teams and can be directly compared with the conventional, machine learning, and LLM driven approaches.

RESULTS AND ANALYSIS

The study of VerifiedRCA applied to 120 synthetic incident scenarios shows significant gains in diagnostics, response efficiency and explanation quality over conventional RCA processes. The accuracy of Top-1 RCA, Top-3 RCA, Time-to-Diagnosis (TTD), and blind human evaluation with experienced SRE reviewers were used to assess performance.

Performance across Fault Categories

The variability of performance was observed across fault categories due to differences in telemetry ambiguity of different types of incidents. The diagnostic accuracy from resource exhaustion and service unavailability were the highest because their signatures on the telemetry were relatively obvious and led to good evidence patterns in the logs and metrics. The results for the multi-hop dependency reasoning task were, however, lower than the other faults.

We found that misconfiguration scenarios had the lowest performance in all cases. The symptoms of configuration faults often occur long after the configuration has changed, and so are often not detected as resource or availability faults. As a result, it was often impossible to determine directly what kind of event was causing recent anomalies for the creation of corresponding telemetry windows. Table III shows the performance of VerifiedRCA per category.

Table 4. Verifiedrca Top-1 Accuracy Across Fault Categories

Fault Category	Top-1 Accuracy
Resource Exhaustion	87%
Service Unavailability	84%
Dependency Failure	82%
Latency Injection	77%
Network Partition	73%
Misconfiguration	68%

The findings show that diagnostic accuracy falls as the incident causality becomes more abstract and remote from the observable evidence from the telemetry. Even more difficult retrieval scenarios will be those involving misconfiguration, where a recent operational state is privileged over a long time history of configuration states.

Figure 2 shows diagnostic accuracy at category level for the systems evaluated before a cross-system comparison.

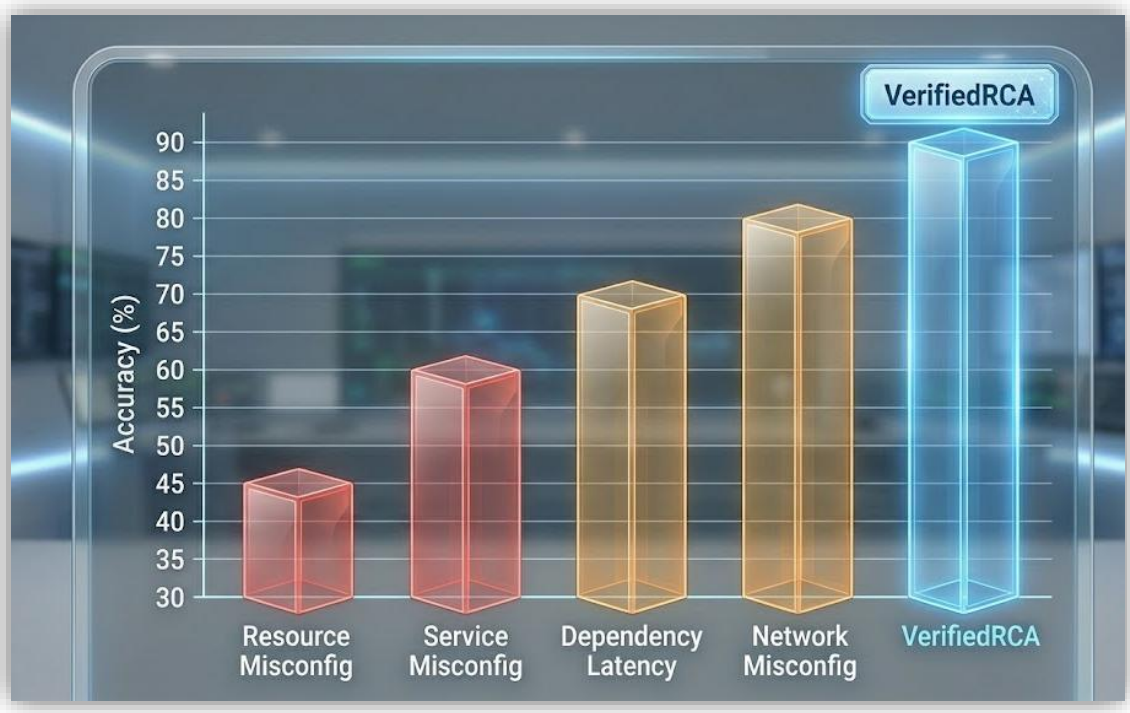

Figure 2. Best-1 accuracy for 6 fault categories in systems evaluated. Under the complex dependency failures, VerifiedRCA shows better performance in 5 of 6 categories and is stable under complex dependency failures. The most difficult scenario in all the approaches that have been tested is misconfiguration.

As can be seen in **Figure 2,** VerifiedRCA continues to demonstrate performance benefits in almost all areas. Traditional rule-based methods were only able to function well with very repetitive service failures and suffered from lack of authored coverage when tested under conditions of latency injection and network partitions.

Evaluation of the quality of explanation by human beings

Diagnostic usefulness is more than just predictive accuracy; it is also related to quality of explanation. For the purpose of assessment of its interpretability, five senior SREs conducted blind reviews on the outputs generated and rated the outputs on factual correctness, actionability, completeness, and readability on a five-point scale.

The most significant differences were found for explanation readability and factual reliability. The ungrounded GPT-4 baseline generated very readable explanations, but reviewers often noted unsupported causal stories and strong (but wrong) conclusions. This behavior indicates a risk of hallucination in a real-world context where it is possible for the language to be accurate and yet contain factual inaccuracies [14, 19].

The highest ratings, due to evidence retrieval and verification, were given to VerifiedRCA for factual correctness and action ability, which generated structured diagnostic reports directly related to the observations seen in the telemetry.

Table 4. Blind Sre Explanation Quality Evaluation (1–5 Scale)

System	Factual Correctness	Actionability	Readability
Rule-Based	2.1	2.7	1.2
Random Forest	2.9	3.0	2.5
GPT-4 Ungrounded	2.8	3.3	4.3
VerifiedRCA	4.4	4.2	3.6

The ability to read is not sufficient to mean that something will be useful to operate. Narratives generated by GPT-4 were fluent, but reports grounded in evidence by VerifiedRCA had higher practical decision support ratings.

DISCUSSION

The findings indicate that the grounding of evidence is among the key factors in effective LLM-driven RCA. The model scored an 18.2 percentage point higher for top-1 accuracy compared to the ungrounded GPT-4 baseline using the same language model and prompt structure. This result suggests that the improvements are, to a large extent, from the validation of the evidence and not from the language-generating capacity.

The proposed framework reduces unnecessary exploration and shortens convergence time through intelligent guidance. Therefore, verification served as a means of reliability as well as an optimization method in performing computations.

Some of these benefits were noted, but there were also some drawbacks. Most common failure mode was premature convergence, choosing a hypothesis before the contradictory evidence was retrieved. Further, the letter of the law was occasionally

overcome by telemetry inconsistencies that resulted in stale observations being fed into the verification pipeline, and therefore less diagnostic value.

The most severe degradation occurred when two or more faults occurred simultaneously. The Top-1 accuracy suffered by about 22% when several Chaos Mesh fault injections happened simultaneously. The results point to the need to explore adaptive verification budgets and multi-cause attribution techniques in future studies.

Last but not least, the work provides an important contribution in terms of reproducibility. Previous LLM based RCA research often uses proprietary data sets and non-standard evaluation methods [7, 14]. An open benchmark is proposed in this work with the objective of providing reproducible experiments and facilitating easy comparison across future approaches.

Conclusion And Future Work

This study introduces VerifiedRCA, a framework for automated root cause analysis in distributed systems via Large Language Models (LLMs), structured telemetry retrieval and tool-call verification using evidence. The framework was designed to tackle an important challenge that has been identified in current LLM-based RCA systems such as generating plausible, but unverifiable, conclusions during the analysis of operational incidents [14, 19]. Instead of just generative reasoning, VerifiedRCA has incorporated a verification mechanism, where a diagnostic hypothesis must be verified within the context of what is observed before the final diagnosis is generated.

The results from the experimental evaluation, leveraging a set of 120 synthetic incident scenarios, showed that VerifiedRCA outperformed rule-based systems, classical machine learning techniques, and ungrounded LLM baselines across the scenarios. This framework outperformed the baseline overall, with an accuracy of 79.2% for Top-1 RCA, 91.4% for Top-3 accuracy and a median time to diagnosis of 2.3 minutes, which represents significant improvement regarding both the diagnostic results and the operational efficiency. Human evaluation also revealed that reports generated with evidence got higher factual correctness and action ability scores compared to generation-only reports.

Overall, the findings suggest that evidence grounding is not only beneficial in terms of the reliability of diagnoses but also in alleviating reasoning burden. The framework helped the reasoning process narrow the search for incident explanations by either confirming or disproving hypotheses early on, thus minimizing unsanctionable diagnostic investigation.

However, there are still a number of restrictions. Misconfiguration incidents remain challenging due to the fact that events are often outside of the short telemetry window. Likewise, performance was seriously degraded in multi-cause fault scenarios, as the signatures of the various faults occurred simultaneously. The quality of the telemetry was also found to affect the diagnostic performance, as it sometimes resulted in the observability data being stale or incomplete, making validation less reliable.

Future Work

There are a number of avenues for expanding the current system. Further studies are needed to explore adaptive reasoning budgets that have the potential of allocating extra verification steps when model confidence is ambiguous. There is also a need to explore multi-cause attribution mechanisms to help support complex incident environments in which multiple independent failures are occurring simultaneously.

Further enhancements might involve adding configuration change history, deployment events and infrastructure state changes as first-class telemetry signals. Multi-agent and graph enhanced reasoning strategies that have been recently investigated in RCA research [3, 18] could further enhance reasoning performance under complex dependency structure.

Finally, future research should be expanded to include more diverse environments such as Google Hipster Shop, as well as different types of deployments and other fault taxonomies. Additional evaluation settings would enhance the external validity, and provide more powerful comparative analysis of future LLM based RCA systems.

Reproducibility Statement

The VerifiedRCA framework, synthetic incident generator, all 120 incident scenarios with labels, evaluation scripts, and Chaos Mesh fault injection manifests will be published under the Apache 2.0 License, once the publication is accepted. The release package will contain deployment manifests for Google Hipster Shop, Istio service mesh configurations and full observability stack configurations (Prometheus, Loki, and Tempo).

The benchmark was developed to facilitate repeatable experiments and comparison across different future RCA systems. It takes about an hour to deploy it within a 6 node Kubernetes environment based on the provided quick-start configuration guide.

REFERENCES

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. In Proceedings of the 45th International Conference on Software Engineering (ICSE 2023).

Avgerinos, V., Ramantas, K., Alonso, L., & Verikoukis, C. (2025). ARM: Autonomous Remediation & Management with LLM Agents for Intent-Driven Control. IEEE Internet of Things Journal. https://doi.org/10.1109/JIOT.2025.3648858

Bocanet, V. I., Muntean, M. H., & Fleseriu, C. (2025, August). Multi-agent Framework for AI-Supported Collaborative Root Cause Analysis in Quality Assurance. In IFIP International Conference on Advances in Production Management Systems (pp. 202-216). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-032-03538-7_15

Chowdhury, A. (2025). Design and Evaluation of a Synthetic Data-Driven Hybrid ML and LLM Pipeline for Critical Infrastructure Security. Available at SSRN 5429995. https://dx.doi.org/10.2139/ssrn.5429995

Ding, R., Zhang, C., Wang, L., Xu, Y., Ma, M., Wu, X., et al. (2023). TraceDiag: Adaptive, interpretable, and efficient root cause analysis on large-scale microservice systems. In Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2023) (pp. 1762–1773).

EBRAHIMI, S., & ASUDEH, A. (2025). A Survey on Reliability, Transparency, Accountability, and Fairness in LLM-based Multi-Agent Systems through the Responsibility Lens.

Fang, A., Yang, H., Dong, H., Lu, Q., Xu, J., & He, P. (2025). A Goal-Driven Survey on Root Cause Analysis. arXiv preprint arXiv:2510.19593. https://doi.org/10.48550/arXiv.2510.19593

Ghanta, S. (2023). From observability to understanding: Automated incident triage using large language model reasoning over logs, metrics, and traces. International Journal of Engineering & Extended Technologies Research (IJEETR), 5(5), 7242-7249. https://doi.org/10.15662/IJRPETM.2024.0701009

Huang, J., Liu, J., He, S., et al. (2024). FaultProFIT: Automated fault pattern profiling for microservice incident tickets. In Proceedings of the 46th International Conference on Software Engineering (ICSE 2024).

Karras, A., Theodorakopoulos, L., Karras, C., Theodoropoulou, A., Kalliampakou, I., & Kalogeratos, G. (2025). LLMs for Cybersecurity in the Big Data Era: A Comprehensive Review of Applications, Challenges, and Future Directions. Information, 16(11), 957. https://doi.org/10.3390/info16110957

Leesatapornwongsa, T., Faisal, F., & Nath, S. (2025). ReproCopilot: LLM-Driven Failure Reproduction with Dynamic Refinement. Proceedings of the ACM on Software Engineering, 2(FSE), 2920-2943. https://doi.org/10.1145/3729399

Mittamidi, V. K. R. (2025). AI/ML Powered Intelligent Root Cause Analysis and Automated Remediation for Multi System Data Integrity Issues. International Journal of AI, BigData, Computational and Management Studies, 6(4), 133-141. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V6I4P115

Nach, H. (2025, October). LLM-Based Analysis of the AI Incident Database: Insights for AI Governance. In 2025 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD) (pp. 1-8). IEEE. https://doi.org/10.1109/ICTMOD66732.2025.11372004

Wang, Z., Liu, Z., Zhang, Y., Zhong, A., Fan, L., Wu, L., & Wen, Q. (2023). RCAgent: Cloud root cause analysis by autonomous agents with tool-augmented large language models. arXiv. https://arxiv.org/abs/2310.16340

Roy, D. (2024). Exploring LLM-based agents for root cause analysis. arXiv. https://arxiv.org/abs/2403.04123

Sundar Ray, S. (2023). Autonomous Incident Response Using Generative AI and Agentic Systems in Distributed Enterprise Architectures. Available at SSRN 6647338. https://ssrn.com/abstract=6647338

Szandała, T. (2025, July). Aiops for reliability: Evaluating large language models for automated root cause analysis in chaos engineering. In International Conference on Computational Science (pp. 323-336). Cham: Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-97564-6_25

Tian, Y., Liu, Y., Chong, Z., Huang, Z., & Jacobsen, H. A. (2025). GALA: Can Graph-Augmented Large Language Model Agentic Workflows Elevate Root Cause Analysis?. arXiv preprint arXiv:2508.12472. https://doi.org/10.48550/arXiv.2508.12472

Wang, Z., Li, J., Ma, M., Li, Z., Kang, Y., Zhang, C., Bansal, C., Chintalapati, M., Rajmohan, S., Lin, Q., et al. (2024). Large language models can provide accurate and interpretable incident triage. In Proceedings of the 35th IEEE International Symposium on Software Reliability Engineering (ISSRE 2024) (pp. 523–534).

Zhang, H. (2024). A Unified AIOps Pipeline for Joint Log–KPI Anomaly Detection, Graph-Based Root Cause Localization, and LLM-Generated Runbooks. Journal of Advanced Computing Systems, 4(3), 57-73. https://doi.org/10.69987/JACS.2024.40305