# LEMMA-RCA: A Large Multi-modal Multi-domain Dataset for Root Cause Analysis

Lecheng Zheng*
lecheng4@illinois.edu
University of Illinois Urbana-Champaign
Champaign, IL, USA

Zhengzhang Chen*†
zchen@nec-labs.com
NEC Laboratories America
Princeton, NJ, USA

Dongjie Wang*
wangdongjie100@gmail.com
University of Kansas
Lawrence, KS, USA

Chengyuan Deng*
cd751@rutgers.edu
Rutgers University
New Brunswick, NJ, USA

Reon Matsuoka
r-matsuoka@nec.com
NEC Laboratories Japan
Tokyo, Japan

Haifeng Chen
Haifeng@nec-labs.com
NEC Laboratories America
Princeton, NJ, USA

## Abstract

Root cause analysis (RCA) is crucial for enhancing the reliability and performance of complex systems. However, progress in this field has been hindered by the lack of large-scale, open-source datasets tailored for RCA. To bridge this gap, we introduce LEMMA-RCA, a large dataset designed for diverse RCA tasks across multiple domains and modalities. LEMMA-RCA features various real-world fault scenarios from Information Technology (IT) and Operational Technology (OT) systems, including microservices, water distribution, and water treatment systems, with hundreds of system entities involved. We evaluate the performance of six baseline methods on LEMMA-RCA across various settings, including offline and online models, as well as single and multi-modal settings. Our study demonstrates the utility of LEMMA-RCA in facilitating fair evaluation and promoting the development of more robust RCA techniques. The dataset is publicly available at https://www.lemmarca.info. **The online supplementary materials can be found at https://arxiv.org/pdf/2406.05375**.

## CCS Concepts

• **Security and privacy** → **Web application security**; • **Computing methodologies** → **Artificial intelligence**.

## Keywords

Root Cause Analysis, Multimodal Data, Microservice System

**ACM Reference Format:**

Lecheng Zheng, Zhengzhang Chen*, Dongjie Wang*, Chengyuan Deng*, Reon Matsuoka, and Haifeng Chen. 2026. LEMMA-RCA: A Large Multi-modal Multi-domain Dataset for Root Cause Analysis. In *Proceedings of the 35th ACM International Conference on Information and Knowledge Management (CIKM '26), November 7–11, 2026, Rome, Italy*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3799682.3840176

*Equal Contribution.
†Corresponding Author.



This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.
*CIKM '26, Rome, Italy.*
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2539-5/2026/11
https://doi.org/10.1145/3799682.3840176

## 1 Introduction

Root cause analysis (RCA) is essential for identifying the underlying causes of system failures and ensuring the reliability of real-world systems [9, 10, 26, 27]. As modern systems grow in complexity and interdependence, faults arising from interactions among modular services can disrupt user experiences and incur significant financial losses. Manual RCA is labor-intensive, costly, and error-prone, making efficient data-driven RCA methods crucial for fault localization and loss mitigation.

RCA has been extensively studied across various domains and settings [3, 5, 8], and can be conducted in an *offline/online* fashion with *single/multi-modal* data. Existing approaches leverage techniques such as Bayesian methods [2] and decision trees [6]. In particular, causal structure learning [4, 35, 37, 38, 40, 43] has proven effective for constructing causal or dependency graphs between system entities and key performance indicators. Despite progress in RCA techniques, large-scale public datasets remain scarce due to confidentiality concerns [12], hindering fair comparison. Available datasets typically contain manually injected faults and cover only a single domain, limiting their utility for real-world scenarios and potentially leading to regulatory and ethical consequences.

To address these limitations, we introduce **LEMMA-RCA**, a collection of LargE-scale Multi-ModAl datasets with real system faults to facilitate future Root Cause Analysis research. LEMMA-RCA spans real-world applications including IT operations and water treatment systems with **hundreds of system entities**, and accommodates **multi-modal** data including textual system logs with millions of event records and time series metrics with over 100,000 timestamps. We annotate ground-truth labels marking the precise timestamps of **real** system faults and their root-cause entities.

Table 1 compares LEMMA-RCA with existing datasets: *NeZha* [40] is limited in size, has missing monitoring data, and focuses only on microservices; *PetShop* [30] contains only 41 components/pods; *Sock-Shop* [14] includes only 13 microservices with synthetic faults, is non-public, and single-modal; *ITOps* [20] is non-public and provides only structured logs; and *Murphy* [12] is collected from a simple, non-public system. In contrast, LEMMA-RCA provides greater accessibility, authenticity, and diversity. It raises no direct ethical concerns, as it contains system monitoring data rather than human-subject or personal user data. LEMMA-RCA also enables

**Table 1: Existing datasets for root cause analysis. The top row corresponds to our dataset. The symbols ✓ and ✗ indicate that the dataset has or does not have the corresponding feature, respectively.**

| Dataset | Public | Real Faults | Large-scale | Multi-domain | Modality | |
|---|---|---|---|---|---|---|
| | | | | | Single | Multiple |
| LEMMA-RCA | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| NeZha | ✓ | ✗ | ✗ | ✗ | ✓ | ✓ |
| PetShop | ✓ | ✗ | ✗ | ✗ | ✓ | ✗ |
| Sock-Shop | ✗ | ✗ | ✗ | ✗ | ✓ | ✗ |
| ITOps | ✗ | ✓ | ✓ | ✗ | ✓ | ✗ |
| Murphy | ✗ | ✓ | ✗ | ✗ | ✓ | ✗ |

fair comparison across RCA methods. We evaluate six baselines and data modality quality in offline settings, then adapt these methods to a standardized online formulation, enabling rigorous comparison of online RCA methods on common ground. Experimental results demonstrate LEMMA-RCA's effectiveness and broad utility for advancing RCA research.

## 2 Preliminaries and Related Work

**Key Performance Indicator (KPI)** is a monitoring time series that captures system-level status, such as latency or response time, where anomalies indicate potential degradation or failure.

**Entity Metrics** are multivariate time series collected from system components, such as machines, containers, or pods, and include CPU, memory, and disk I/O utilization. Abnormal entities are treated as candidate root causes.

**Data-driven Root Cause Analysis Problem.** Given monitoring data (metrics and logs) of system entities and KPIs, RCA aims to identify the top-K entities most relevant to KPIs when a fault occurs. RCA settings are commonly categorized along two dimensions: offline (retrospective, historical data) vs. online (real-time streams), and single-modal vs. multi-modal. Figure 1 illustrates the single-modal offline and multi-modal online procedures.

**Single-modal Offline RCA** identifies fault causes retrospectively from a single data type [19, 23, 33, 34, 38]. Representative works include sequential and causal-temporal modeling of metrics [23], random walks on interdependent causal networks [38], and trace-based localization via abnormal/normal ratios [19]. LLM-based approaches have also emerged as a promising direction for learning causal relations [7, 11, 29, 32, 39, 44]. However, reliance on a single modality may yield biased or suboptimal results.

**Multi-modal Offline RCA** follows two main paradigms [13, 16, 40, 43]: (i) extracting information per modality and then integrating it, as in Nezha [40] and PDiagnose [13]; and (ii) modeling cross-modal interactions, such as MULAN [43], which learns inter-modal correlations to build a unified causal graph, and MM-DAG [16], which jointly learns multiple DAGs.

**Online RCA.** Most RCA methods operate offline, requiring full retraining for new faults and delaying response. Wang *et al.* [37] decouple state-invariant and state-dependent information to incrementally update the causal graph, while Li *et al.* [17] leverage system architecture in a causal Bayesian network to reduce bias toward new data. Yet these methods remain single-modal, leaving online multi-modal RCA an open challenge.

## 3 LEMMA-RCA Data

This section outlines the data resources, preprocessing steps, and key characteristics of the released data.

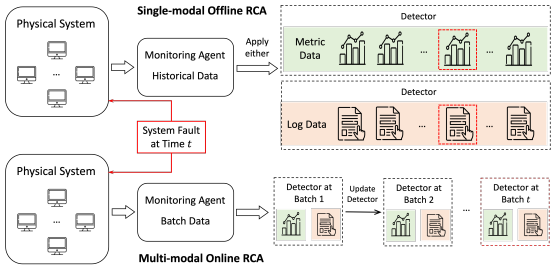



**Figure 1: Illustration of RCA workflow in the single-modal offline setting (top) and the multi-modal online setting (bottom). The other two settings can be viewed as an ensemble of corresponding components (data collection, detector, modality) and follow the same systematic procedure.**

**Table 2: Data statistics of IT and OT operation sub-datasets.**

| Microservice System (IT) | Product Review | Cloud Computing |
|---|---|---|
| Original Dataset Size | 765 GB | 540 GB |
| Number of (#) fault types | 4 | 6 |
| Average # entities per fault | 216.0 | 167.71 |
| Average # metrics per fault | 11 (node) + 6 (pod) | 6 (node) + 7 (pod) |
| Average # timestamps per fault | 131,329.25 | 109,350.57 |
| Average max log events per fault | 153,081,219.0 | 63,768,587.25 |
| **Water Treatment/Distribution (OT)** | **SWaT** | **WADI** |
| Original Dataset Size | 4.47G | 5.67G |
| Number of (#) fault types | 16 | 9 |
| Average # entities per fault | 51.0 | 123.0 |
| Average # metrics per fault | 7 (node) + 7 (pod) | 7 (node) + 7 (pod) |
| Average # timestamps per fault | 56,239.88 | 85,248.47 |

### 3.1 Data Collection

We collect real-world data from two domains: IT and OT operations, including **Product Review**, **Cloud Computing**, **SWaT**, and **WADI**. Dataset statistics are summarized in Table 2. The **Product Review** Platform comprises six OpenShift nodes and 216 system pods, collecting four fault types (out-of-memory, high-CPU-usage, external-storage-full, and DDoS attack), each lasting at least 49 hours. Node- and pod-level metrics were recorded by Prometheus [36] at 1-second granularity, while logs were collected by ElasticSearch [41]. Prometheus [36] records node- and pod-level metrics at 1-second intervals, ElasticSearch [41] collects logs, and JMeter [25] monitors latency as the KPI. The **Cloud Computing** Platform captures six fault types (e.g., cryptojacking, GitOps mistakes, configuration change failure), with metrics from CloudWatch Metrics and logs (general, API debug, MySQL) from CloudWatch Logs; latency, error rate, and utilization rate serve as KPIs. For OT operations, **SWaT** and **WADI** [15] provide sensor/actuator time series from iTrust testbeds. SWaT [22] contains 11 days of data from 51 sensors with 16 faults, while WADI [1] contains 16 days from 123 sensors/actuators with 15 faults. Figure 2 illustrates KPIs for eight failure cases, showing characteristic spikes or drops during system failures.

### 3.2 Data Preprocessing

After collection, we exclude non-stationary pods to retain only those suitable for modeling. For **Product Review** and **Cloud Computing**, following [42], we transform unstructured logs into time-series via three feature types: (i) $X_1^L \in \mathbb{R}^T$, the occurrence frequency of structured log templates extracted by Drain over 10-minute windows with 30-second intervals; (ii) $X_2^L \in \mathbb{R}^T$, the frequency of abnormal templates containing golden-signal keywords

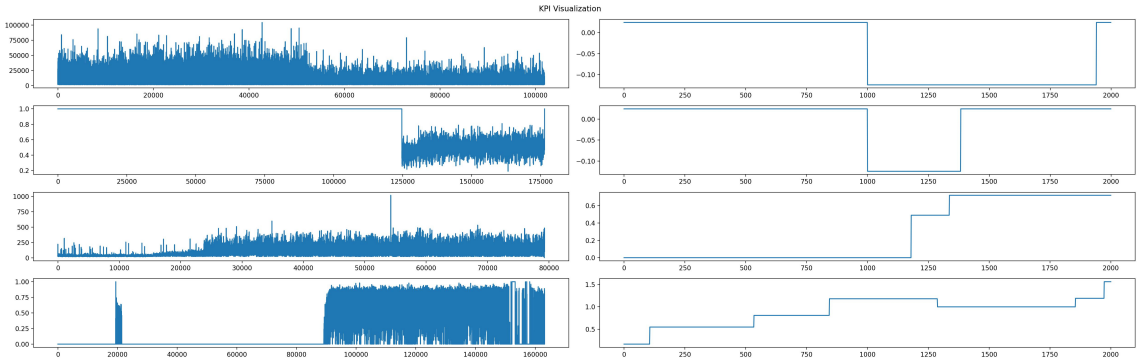



**Figure 2: KPI visualization for system failures. Left: Product Review (first two) and Cloud Computing (last two); Right: SWaT (first two) and WADI (last two).**

(e.g., 'error', 'exception', 'critical'); and (iii) $X_3^L \in \mathbb{R}^T$, the leading PCA component of TF-IDF features. The final feature matrix is $X^L = [X_1^L; X_2^L; X_3^L] \in \mathbb{R}^{3\times T}$. For **SWaT** and **WADI**, since the original label column is discrete, we apply anomaly detection algorithms (e.g., Support Vector Data Description, Isolation Forest) to derive continuous anomaly scores serving as KPIs.

# 4 Experiments

## 4.1 Experimental Setup

**Evaluation Metrics**. To assess baseline RCA method on LEMMA-RCA, we choose three widely-used metrics [24, 38, 43] and introduce them below.

(1) **Precision@K (PR@K)**: It measures the probability that the top $K$ predicted root causes are real, defined as:

$$\text{PR@K} = \frac{1}{|\mathbb{A}|}\sum_{a\in\mathbb{A}}\frac{\sum_{i<k} R_a(i) \in V_a}{\min(K, |v_a|)} \tag{1}$$

where $\mathbb{A}$ is the set of system faults, $a$ is one fault in $\mathbb{A}$, $V_a$ is the real root causes of $a$, $R_a$ is the predicted root causes of $a$, and $i$ is the $i$-th predicted cause of $R_a$.

(2) **Mean Average Precision@K (MAP@K)**: It assesses the top $K$ predicted causes from the overall perspective, defined as:

$$\text{MAP@K} = \frac{1}{K|\mathbb{A}|}\sum_{a\in\mathbb{A}}\sum_{i\le j\le K}\text{PR@j} \tag{2}$$

where a higher value indicates better performance.

(3) **Mean Reciprocal Rank (MRR)**: It evaluates the ranking capability of models, defined as:

$$\text{MRR@K} = \frac{1}{|\mathbb{A}|}\sum_{a\in\mathbb{A}}\frac{1}{\text{rank}_{R_a}} \tag{3}$$

where $\text{rank}_{R_a}$ is the rank number of the first correctly predicted root cause for system fault $a$.

**Baselines**. We evaluate the performance of the following RCA baselines on the benchmark sub-datasets, selecting only those with publicly available code to ensure fair and reproducible comparisons: (1) **PC-based** [21]: This approach first employs a Peter-Clark (PC) algorithm [4] to construct the anomaly behavior graph, and then applies a random walk algorithm to rank the root causes based on the estimated graph structure; (2) **CIRCA** [18]: This model utilizes structural graph construction, regression-based hypothesis testing, and descendant adjustment to identify root cause metrics; (3) **$\epsilon$-Diagnosis** [31]: This model diagnoses small-window, long-tail latency in large-scale microservice platforms using a two-sample test and $\epsilon$-statistics; (4) **RCD** [14]: This technique hierarchically localizes the root cause of failures by focusing on relevant sections of the dependency graph; (5) **BARO** [28]: It is an end-to-end approach integrating Bayesian change point detection and nonparametric hypothesis testing to detect anomalies and identify root causes in microservice systems; (6) **Nezha** [40]: A multi-modal method designed to identify root causes by detecting abnormal patterns.

When using only metric data (*e.g.*, CPU usage, memory usage) for root cause analysis, we first identify the potential root cause associated with each metric. We then compute the final score for each candidate by averaging its ranking scores across all metrics. This procedure is similarly applied in log-only and multi-modal scenarios to rank and identify the root causes. For the hyperparameters, we use the default parameter values for all baselines to ensure a fair comparison.

**Table 3: Results for RCA baselines with multiple modalities on the Product Review dataset.**

| Modality | Model | PR@1 | PR@5 | PR@10 | MRR | MAP@3 | MAP@5 | MAP@10 |
|---|---|---|---|---|---|---|---|---|
| Metric Only | PC | 0 | 0 | 0.250 | 0.053 | 0 | 0 | 0.050 |
| | RCD | 0 | 0.250 | 0.750 | 0.185 | 0.167 | 0.200 | 0.350 |
| | $\epsilon$-Diagnosis | 0 | 0 | 0.250 | 0.038 | 0 | 0 | 0.050 |
| | CIRCA | 0 | 0.750 | 0.750 | 0.283 | 0.250 | 0.450 | 0.600 |
| | BARO | 0.250 | 0.250 | 0.250 | 0.286 | 0.250 | 0.250 | 0.250 |
| Log Only | PC | 0 | 0 | 0.250 | 0.069 | 0 | 0 | 0.125 |
| | RCD | 0 | 0 | 0.250 | 0.056 | 0 | 0 | 0.025 |
| | $\epsilon$-Diagnosis | 0 | 0 | 0.250 | 0.038 | 0 | 0 | 0.050 |
| | CIRCA | 0 | 0 | 0.500 | 0.080 | 0 | 0 | 0.125 |
| | BARO | 0 | 0.250 | 0.250 | 0.139 | 0.167 | 0.200 | 0.225 |
| Multi-Modality | PC | 0 | 0 | 0.250 | 0.064 | 0 | 0 | 0.125 |
| | RCD | 0 | 0.500 | 0.750 | 0.231 | 0.167 | 0.300 | 0.525 |
| | $\epsilon$-Diagnosis | 0 | 0 | 0.250 | 0.041 | 0 | 0 | 0.075 |
| | CIRCA | 0 | **0.750** | **1.000** | 0.299 | 0.250 | 0.450 | 0.650 |
| | BARO | **0.750** | **0.750** | **1.000** | **0.775** | **0.750** | **0.750** | **0.775** |
| | Nezha | 0 | 0.500 | 0.750 | 0.193 | 0.083 | 0.250 | 0.475 |

## 4.2 Root Cause Analysis Results

**Product Review and Cloud Computing**. We evaluate six RCA methods including both single-modal and multi-modal methods on

Product Review and Cloud Computing sub-datasets. The experimental results are presented in Table 3 with respect to Precision at K (PR@K), Mean Reciprocal Rank (MRR), and Mean Average Precision at K (MAP@K).

Our observations reveal the following insights: (1) The PC algorithm and $\epsilon$-Diagnosis perform worst on both the Product Review and Cloud Computing sub-datasets. We conjecture that PC, RCD, and $\epsilon$-Diagnosis struggle to capture long-term dependencies in large-scale datasets, making it difficult to detect abnormal temporal patterns. (2) CIRCA outperforms RCD and $\epsilon$-Diagnosis, consistent with findings from the Petshop study [30], where CIRCA's regression-based hypothesis testing and adjustment mechanisms led to higher diagnostic accuracy. (3) Multi-modal input—combining both metric and log data—significantly enhances the performance of RCA methods compared to using either modality alone. For example, BARO achieves only 25% PR@1 with metric data and 0% with log data on the Product Review sub-dataset, but reaches 75% PR@1 when both are used, correctly identifying the root cause in 75% of fault scenarios. This highlights the complementary nature of log and metric data and the importance of integrating both to improve diagnostic accuracy and overall performance, particularly in terms of MRR.

**Table 4: Results for RCA baselines on the SWaT sub-dataset.**

| Dataset | Model | PR@1 | PR@5 | PR@10 | MRR | MAP@3 | MAP@5 | MAP@10 |
|---|---|---|---|---|---|---|---|---|
| SWaT | PC | 0.125 | **0.344** | 0.583 | 0.262 | 0.129 | **0.204** | 0.350 |
| | RCD | 0.125 | 0.125 | 0.625 | 0.228 | 0.125 | 0.125 | 0.344 |
| | $\epsilon$-Diagnosis | 0.125 | 0.125 | 0.563 | 0.217 | 0.125 | 0.125 | 0.294 |
| | CIRCA | **0.188** | 0.250 | **0.688** | **0.287** | **0.188** | 0.200 | **0.394** |
| | BARO | 0 | 0.208 | 0.208 | 0.124 | 0.083 | 0.133 | 0.171 |

**Table 5: Results for RCA baselines on the WADI sub-dataset.**

| Dataset | Model | PR@1 | PR@5 | PR@10 | MRR | MAP@3 | MAP@5 | MAP@10 |
|---|---|---|---|---|---|---|---|---|
| WADI | PC | 0.071 | 0.350 | 0.500 | 0.277 | 0.163 | 0.239 | 0.346 |
| | RCD | 0 | 0 .071 | 0.119 | 0.054 | 0.048 | 0.057 | 0.076 |
| | $\epsilon$-Diagnosis | 0 | 0 | 0.022 | 0.020 | 0 | 0 | 0.009 |
| | CIRCA | **0.143** | **0.550** | **0.714** | **0.350** | **0.301** | **0.400** | **0.529** |
| | BARO | 0 | 0.143 | 0.143 | 0.085 | 0.071 | 0.100 | 0.121 |

**Water Treatment/Distribution**. We evaluate five single-modal RCA methods on the SWaT and WADI sub-datasets using the same set of evaluation metrics. Table 4 presents the results for SWaT, while Table 5 shows the results for WADI. We observe that CIRCA consistently achieves the best overall performance on both datasets, although the PC algorithm occasionally outperforms it on SWaT by a small margin. This trend aligns with the observations from the Product Review and Cloud Computing sub-datasets. Notably, the results also indicate considerable room for improvement, likely due to the characteristics of the SWaT and WADI datasets—faults are short-lived, and the intervals between them are brief. These fleeting events are easily overlooked by most RCA methods, posing a significant challenge for accurate root cause identification.

## 4.3 Online Root Cause Analysis Results

We evaluate three RCA methods on the Product Review sub-dataset to demonstrate the utility of the LEMMA-RCA in an online setting in Table 6. Because the runtime of CIRCA, PC, and Nezha exceed 24 hours per case, they are excluded from the online evaluation. Notably, LEMMA-RCA is a large-scale real-world dataset, consisting of more than 100,000 timestamps across several days with various system fault scenarios, which can be naturally transformed to the online setting, compared with small datasets with limited timestamps for online RCA. Although RCD, $\epsilon$-Diagnosis, and BARO are originally designed for offline analysis, we adapt them to the online setting by formatting the data into a sequence of streaming snapshots. Each baseline method is then evaluated on consecutive snapshots until similar results appear three times in succession or the data stream reaches its final snapshot. Empirically, we observe that all three methods suffer a significant drop in performance under the online setting compared to their offline results. We conjecture that this degradation arises from their inability to consistently capture temporal dependencies across snapshots. This observation highlights the need for developing dedicated online root cause analysis methods.

**Table 6: Results for root cause analysis baselines on Product Review sub-dataset in the online setting.**

| Modality | Model | PR@1 | PR@5 | PR@10 | MRR | MAP@3 | MAP@5 | MAP@10 |
|---|---|---|---|---|---|---|---|---|
| Metric Only | RCD | 0 | **0.250** | **0.250** | 0.054 | 0 | 0.050 | 0.150 |
| | $\epsilon$-Diagnosis | 0 | 0 | 0 | 0.019 | 0 | 0 | 0 |
| | BARO | **0.250** | **0.250** | **0.250** | 0.269 | **0.250** | **0.250** | **0.250** |
| Log Only | RCD | 0 | 0 | 0 | 0.012 | 0 | 0 | 0 |
| | $\epsilon$-Diagnosis | 0 | 0 | 0 | 0.025 | 0 | 0.100 | 0.175 |
| | BARO | 0 | 0 | 0 | 0.023 | 0 | 0.100 | 0.175 |
| Multi-Modality | RCD | 0 | **0.250** | **0.250** | 0.054 | 0 | 0.050 | 0.150 |
| | $\epsilon$-Diagnosis | 0 | 0 | 0 | 0.027 | 0 | 0 | 0 |
| | BARO | **0.250** | **0.250** | **0.250** | **0.270** | **0.250** | **0.250** | **0.250** |

## 4.4 Online Supplementary Materials

Due to space limitations, we defer extensive materials to the full online version[1]. Appendix B provides detailed descriptions of all fault scenarios, additional preprocessing steps for SWaT/WADI, and extended visualizations. In particular, Appendix B.3 describes the 10 distinct real-world fault types covered by LEMMA-RCA across Product Review and Cloud Computing. Detailed implementation is provided in Appendix C, while Appendix D describes the segmentation procedure for SWaT and WADI. Additional experimental results on Cloud Computing (Tables 7 and 8) are reported in Appendix E. Reproducibility details, including hardware and software environments, are provided in Appendix G, and descriptions of the evaluated baseline methods are included in Appendix H.

## 5 Conclusion

In this work, we introduce LEMMA-RCA, the first large-scale, open-source dataset featuring real system faults across multiple application domains and data modalities. We conduct a comprehensive empirical study using six baseline RCA methods, evaluating their performance on both single-modal and multi-modal data under offline and online settings. The experimental results highlight the utility of LEMMA-RCA as a benchmarking resource. By releasing this dataset publicly, we aim to advance research in root cause analysis for complex systems and support the development of more robust and reliable methodologies, particularly for mission-critical applications.

[1]Supplementary materials are available at https://arxiv.org/pdf/2406.05375.

## Generative AI Usage Disclosure

In this work, generative AI tools, including large language models (LLMs), were used solely for writing, editing, and formatting purposes.

## References


[1] Chuadhry Mujeeb Ahmed, Venkata Reddy Palleti, and Aditya P Mathur. 2017. WADI: a water distribution testbed for research in the design of secure cyber physical systems. In *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*. 25–28.

[2] Adel Alaeddini and Ibrahim Dogan. 2011. Using Bayesian networks for root cause analysis in statistical process control. *Expert Systems with Applications* 38, 9 (2011), 11230–11243.

[3] Álvaro Brandón, Marc Solé, Alberto Huélamo, David Solans, María S Pérez, and Victor Muntés-Mulero. 2020. Graph-based root cause analysis for service-oriented and microservice architectures. *Journal of Systems and Software* 159 (2020), 110432.

[4] Tom Burr. 2003. Causation, Prediction, and Search. *Technometrics* 45, 3 (2003), 272–273.

[5] Alfonso Capozzoli, Fiorella Lauro, and Imran Khan. 2015. Fault detection analysis using data mining techniques for a cluster of smart office buildings. *Expert Systems with Applications* 42, 9 (2015), 4324–4338.

[6] Mike Chen, Alice X Zheng, Jim Lloyd, Michael I Jordan, and Eric Brewer. 2004. Failure diagnosis using decision trees. In *International Conference on Autonomic Computing, 2004. Proceedings.* IEEE, 36–43.

[7] Yinfang Chen, Huaibing Xie, Minghua Ma, Yu Kang, Xin Gao, Liu Shi, Yunjie Cao, Xuedong Gao, Hao Fan, Ming Wen, et al. 2024. Automatic root cause analysis via large language models for cloud incidents. In *Proceedings of the Nineteenth European Conference on Computer Systems*. 674–688.

[8] Ailin Deng and Bryan Hooi. 2021. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4027–4035.

[9] George K Fourlas and George C Karras. 2021. A survey on fault diagnosis methods for UAVs. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 394–403.

[10] Zhiwei Gao, Carlo Cecati, and Steven X. Ding. 2015. A Survey of Fault Diagnosis and Fault-Tolerant Techniques—Part I: Fault Diagnosis With Model-Based and Signal-Based Approaches. *IEEE Transactions on Industrial Electronics* 62, 6 (2015), 3757–3767. doi:10.1109/TIE.2015.2417501

[11] Drishti Goel, Fiza Husain, Aditya Singh, Supriyo Ghosh, Anjaly Parayil, Chetan Bansal, Xuchao Zhang, and Saravan Rajmohan. 2024. X-lifecycle Learning for Cloud Incident Management using LLMs. In *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*. 417–428.

[12] Vipul Harsh, Wenxuan Zhou, Sachin Ashok, Radhika Niranjan Mysore, Brighten Godfrey, and Sujata Banerjee. 2023. Murphy: Performance Diagnosis of Distributed Cloud Applications. In *Proceedings of the ACM SIGCOMM 2023 Conference*. 438–451.

[13] Chuanjia Hou, Tong Jia, Yifan Wu, Ying Li, and Jing Han. 2021. Diagnosing Performance Issues in Microservices with Heterogeneous Data Source. In *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), New York City, NY, USA, September 30 - Oct. 3, 2021*. IEEE, 493–500.

[14] Azam Ikram, Sarthak Chakraborty, Subrata Mitra, Shiv Saini, Saurabh Bagchi, and Murat Kocaoglu. 2022. Root cause analysis of failures in microservices through causal discovery. *Advances in Neural Information Processing Systems* 35 (2022), 31158–31170.

[15] iTrust. 2022. The Website of iTrust Lab. [EB/OL]. https://itrust.sutd.edu.sg/itrust-labs_datasets/dataset_info/.

[16] Tian Lan, Ziyue Li, Zhishuai Li, Lei Bai, Man Li, Fugee Tsung, Wolfgang Ketter, Rui Zhao, and Chen Zhang. 2023. Mm-dag: Multi-task dag learning for multi-modal data-with application for traffic congestion analysis. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1188–1199.

[17] Mingjie Li, Zeyan Li, Kanglin Yin, Xiaohui Nie, Wenchi Zhang, Kaixin Sui, and Dan Pei. 2022. Causal Inference-Based Root Cause Analysis for Online Service Systems with Intervention Recognition. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*, Aidong Zhang and Huzefa Rangwala (Eds.). ACM, 3230–3240.

[18] Mingjie Li, Zeyan Li, Kanglin Yin, Xiaohui Nie, Wenchi Zhang, Kaixin Sui, and Dan Pei. 2022. Causal inference-based root cause analysis for online service systems with intervention recognition. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3230–3240.

[19] Zeyan Li, Junjie Chen, Rui Jiao, Nengwen Zhao, Zhijun Wang, Shuwei Zhang, Yanjun Wu, Long Jiang, Leiqin Yan, Zikai Wang, et al. 2021. Practical root cause localization for microservice systems via trace analysis. In *2021 IEEE/ACM 29th International Symposium on Quality of Service (IWQOS)*. IEEE, 1–10.

[20] Zeyan Li, Nengwen Zhao, Shenglin Zhang, Yongqian Sun, Pengfei Chen, Xidao Wen, Minghua Ma, and Dan Pei. 2022. Constructing large-scale real-world benchmark datasets for aiops. *arXiv preprint arXiv:2208.03938* (2022).

[21] Meng Ma, Jingmin Xu, Yuan Wang, Pengfei Chen, Zonghua Zhang, and Ping Wang. 2020. Automap: Diagnose your microservice-based web applications automatically. In *Proceedings of The Web Conference 2020*. 246–258.

[22] Aditya P Mathur and Nils Ole Tippenhauer. 2016. SWaT: A water treatment testbed for research and training on ICS security. In *2016 international workshop on cyber-physical systems for smart water networks (CySWater)*. IEEE, 31–36.

[23] Yuan Meng, Shenglin Zhang, Yongqian Sun, Ruru Zhang, Zhilong Hu, Yiyin Zhang, Chenyang Jia, Zhaogang Wang, and Dan Pei. 2020. Localizing failure root causes in a microservice through causality inference. In *2020 IEEE/ACM 28th International Symposium on Quality of Service (IWQoS)*. IEEE, 1–10.

[24] Yuan Meng, Shenglin Zhang, Yongqian Sun, Ruru Zhang, Zhilong Hu, Yiyin Zhang, Chenyang Jia, Zhaogang Wang, and Dan Pei. 2020. Localizing Failure Root Causes in a Microservice through Causality Inference. In *28th IEEE/ACM International Symposium on Quality of Service, IWQoS 2020, Hangzhou, China, June 15-17, 2020*. IEEE, 1–10.

[25] Dmitri Nevedrov. 2006. Using jmeter to performance test web services. *Published on dev2dev* (2006), 1–11.

[26] Ignavier Ng, AmirEmad Ghassami, and Kun Zhang. 2020. On the Role of Sparsity and DAG Constraints for Learning Linear DAGs. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (Eds.).

[27] Roxana Pamfil, Nisara Sriwattanaworachai, Shaan Desai, Philip Pilgerstorfer, Konstantinos Georgatzis, Paul Beaumont, and Bryon Aragam. 2020. DYNOTEARS: Structure Learning from Time-Series Data. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy] (Proceedings of Machine Learning Research, Vol. 108)*, Silvia Chiappa and Roberto Calandra (Eds.). PMLR, 1595–1605.

[28] Luan Pham, Huong Ha, and Hongyu Zhang. 2024. Baro: Robust root cause analysis for microservices via multivariate bayesian online change point detection. *Proceedings of the ACM on Software Engineering* 1, FSE (2024), 2214–2237.

[29] Devjeet Roy, Xuchao Zhang, Rashi Bhave, Chetan Bansal, Pedro Las-Casas, Rodrigo Fonseca, and Saravan Rajmohan. 2024. Exploring llm-based agents for root cause analysis. In *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*. 208–219.

[30] Saurabh Garg, Imaya Kumar Jagannathan. 2024. Root cause analyses on PetShop Application. https://github.com/amazon-science/petshop-root-cause-analysis/tree/main?tab=readme-ov-file.

[31] Huasong Shan, Yuan Chen, Haifeng Liu, Yunpeng Zhang, Xiao Xiao, Xiaofeng He, Min Li, and Wei Ding. 2019. ?-diagnosis: Unsupervised and real-time diagnosis of small-window long-tail latency in large-scale microservice platforms. In *The World Wide Web Conference*. 3215–3222.

[32] Shiwen Shan, Yintong Huo, Yuxin Su, Yichen Li, Dan Li, and Zibin Zheng. 2024. Face it yourselves: An llm-based two-stage strategy to localize configuration errors via logs. In *Proceedings of the 33rd ACM SIGSOFT International Symposium on Software Testing and Analysis*. 13–25.

[33] Jacopo Soldani and Antonio Brogi. 2022. Anomaly detection and failure root cause analysis in (micro) service-based cloud applications: A survey. *ACM Computing Surveys (CSUR)* 55, 3 (2022), 1–39.

[34] LuAn Tang, Hengtong Zhang, Zhengzhang Chen, Bo Zong, LI Zhichun, Guofei Jiang, and Kenji Yoshihira. 2019. Graph-based attack chain discovery in enterprise security systems. US Patent 10,289,841.

[35] Alex Tank, Ian Covert, Nicholas J. Foti, Ali Shojaie, and Emily B. Fox. 2022. Neural Granger Causality. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 8 (2022), 4267–4279.

[36] James Turnbull. 2018. *Monitoring with Prometheus.* Turnbull Press.

[37] Dongjie Wang, Zhengzhang Chen, Yanjie Fu, Yanchi Liu, and Haifeng Chen. 2023. Incremental causal graph learning for online root cause analysis. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2269–2278.

[38] Dongjie Wang, Zhengzhang Chen, Jingchao Ni, Liang Tong, Zheng Wang, Yanjie Fu, and Haifeng Chen. 2023. Interdependent Causal Networks for Root Cause Localization. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2023, Long Beach, CA, USA, August 6-10, 2023*. ACM, 5051–5060.

[39] Zefan Wang, Zichuan Liu, Yingying Zhang, Aoxiao Zhong, Lunting Fan, Lingfei Wu, and Qingsong Wen. 2024. Rcagent: Cloud root cause analysis by autonomous agents with tool-augmented large language models. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*.

[40] Guangba Yu, Pengfei Chen, Yufeng Li, Hongyang Chen, Xiaoyun Li, and Zibin Zheng. 2023. Nezha: Interpretable Fine-Grained Root Causes Analysis for Microservices on Multi-modal Observability Data. (2023).

[41] Vlad-Andrei Zamfir, Mihai Carabas, Costin Carabas, and Nicolae Tapus. 2019. Systems monitoring and big data analysis using the elasticsearch system. In *2019 22nd International Conference on Control Systems and Computer Science (CSCS)*. IEEE, 188–193.

[42] Lecheng Zheng, Zhengzhang Chen, and Haifeng Chen. 2025. Online Multi-modal Root Cause Identification in Microservice Systems. In *2025 IEEE International Conference on Big Data (BigData)*. IEEE, 1661–1668.

[43] Lecheng Zheng, Zhengzhang Chen, Jingrui He, and Haifeng Chen. 2024. MULAN: Multi-modal Causal Structure Learning and Root Cause Analysis for Microservice Systems. In *Proceedings of the ACM on Web Conference 2024*. 4107–4116.

[44] Bin Zhou, Xinyu Li, Tianyuan Liu, Kaizhou Xu, Wei Liu, and Jinsong Bao. 2024. CausalKGPT: industrial structure causal knowledge-enhanced large language model for cause analysis of quality problems in aerospace product manufacturing. *Advanced Engineering Informatics* 59 (2024), 102333.