



Constructing Large-Scale Real-World Benchmark Datasets for AIOps

Li Zeyan¹, Han Jing², Zhao Nengwen¹,
Zhang Shenglin³, Sun Yongqian³, Chen Pengfei⁴,
Wen Xidao¹, Ma Minghua⁵, Pei Dan¹

(1. Tsinghua University, Beijing 100084, China;
2. ZTE Corporation, Shenzhen 518057, China;
3. Nankai University, Tianjin 300071, China;
4. Sun Yat-sen University, Guangzhou 510275, China;
5. Microsoft Research, Redmond, WA 98052, USA)

DOI: 10.12142/ZTECOM.202603008

<https://kns.cnki.net/kcms/detail/34.1294.TN.20260903.1213.002.html>,
published online September 3, 2026

Manuscript received: 2025-02-21

Abstract: Artificial Intelligence for IT Operations (AIOps) has emerged as a critical approach for managing complex IT systems, yet its advancement has been hindered by the lack of comprehensive public datasets for evaluation. This study addresses this challenge by introducing three large-scale, real-world benchmark datasets specifically designed for AIOps research and development. We present datasets covering key AIOps scenarios: key performance indicator (KPI) anomaly detection with diverse patterns from multiple Internet companies, multi-dimensional root cause localization with 400 labeled cases, and failure discovery and diagnosis containing 169 injected failures across various system components. These datasets are distinguished by their scale, authenticity, and detailed ground-truth labels, enabling rigorous evaluation of AIOps algorithms. The practical value of our datasets has been validated through three annual AIOps algorithm competitions (2018–2020), which attracted over 400 participating teams and led to numerous research publications. By providing these comprehensive benchmarks, our work establishes a foundation for reproducible AIOps research and facilitates fair comparison of different approaches, serving a role for AIOps analogous to that of ImageNet in computer vision.

Keywords: AIOps; benchmark datasets; KPI anomaly detection; root cause localization; failure discovery and diagnosis

Citation (Format 1): Li Z Y, Han J, Zhao N W, et. al. Constructing large-scale real-world benchmark datasets for AIOps [J]. ZTE Communications, 2026, 24(3): 59 – 69. DOI: 10.12142/ZTECOM.202603008

Citation (Format 2): Z. Y. Li, J. Han, N. W. Zhao, et. al., “Constructing large-scale real-world benchmark datasets for AIOps,” *ZTE Communications*, vol. 24, no. 3, pp. 59 – 69, Sept. 2026. doi: 10.12142/ZTECOM.202603008.

1 Introduction

In recent years, the growing complexity of modern IT service architectures has made traditional manual operations and maintenance increasingly time-consuming and prone to errors. To address this challenge, the concept of Artificial Intelligence for IT Operations (AIOps) has emerged to reduce manual intervention. According to Gartner^[1], AIOps refers to the utilization of machine learning (ML) and data mining techniques in IT operations. By combining the power of big data and ML, AIOps aims to enhance and, to some extent, replace conventional manual tasks such as performance monitoring, anomaly detection, event correlation and analysis, and incident management.

Extensive research efforts have been dedicated to various

aspects of AIOps in the literature, such as anomaly detection^[2–5], root cause localization^[6–17], failure prediction^[18–23], incident management^[24–27], and resource management^[28–29]. However, a major challenge arises from the absence of publicly available large-scale real-world datasets, which makes it difficult to conduct fair comparisons of these approaches.

While these works demonstrate promising performance on their respective private datasets, their real-world robustness remains uncertain. Moreover, the diversity of target AIOps systems makes it challenging to generalize approaches to real-world applications. Consequently, AIOps represents a convergence point between artificial intelligence research and operations engineering, necessitating collaboration between academia and industry to access authentic data and scenarios for research purposes.

In practice, researchers often face challenges in finding industrial partners due to the highly confidential nature of the monitoring data used in AIOps. Consequently, the lack of

This work was supported by ZTE Industry-University-Institute Cooperation Funds under Grant No. IA20231009003.

public large-scale real-world datasets remains a significant limitation in both AIOps research and practical application.

While there are a few public datasets available for AIOps research, these datasets suffer from several limitations. First, most existing datasets focus on a single AIOps scenario, such as Yahoo S5^[30] for key performance indicator (KPI) anomaly detection and the system-trace corpora generated from the Sock Shop and TrainTicket microservice benchmarks to evaluate MEPFL^[31]. However, AIOps encompasses a wide range of scenarios, including KPI anomaly detection, log anomaly detection, capacity planning, root cause localization, and more. Evaluating multi-scenario AIOps approaches on scenario-specific datasets individually is inadequate to capture the holistic nature of AIOps. Second, the scale of these datasets is limited. For instance, Yahoo Benchmark for KPI anomaly detection consists of only hundreds of data points. In reality, service systems often support numerous users and operate in complex architectures. Therefore, to approximate real-world conditions, datasets should be large and diverse. Third, some datasets are synthetic rather than real-world. Such synthetic datasets fail to accurately reflect the behavior of actual systems, which poses risks for deploying AIOps techniques in practice. Consequently, there is a pressing need for rich, large-scale, and real-world datasets to facilitate the advancement of AIOps research.

To foster the development of AIOps, we have released three large-scale, real-world datasets on GitHub^[32–34]. These datasets are sourced from the production systems of our industrial partners, including Sogou, eBay, Tencent, Suning, and China Mobile Zhejiang. Each dataset caters to a specific aspect of AIOps: KPI anomaly detection, multi-dimensional root cause localization, and comprehensive failure discovery and diagnosis. Numerous studies^[6, 10, 35–41] have already utilized these datasets. Furthermore, based on these datasets, we have organized an annual AIOps algorithm competition since 2018. We organized three AIOps algorithm competitions in 2018, 2019, and 2020. These editions attracted 125, 141, and 141 teams, respectively, totaling 407 team registrations and more than 1 000 participant registrations.¹ To encourage collaboration and knowledge sharing, we have publicly released the solutions from the top teams^[42]. Moving forward, we remain committed to creating and publishing new datasets, as well as organizing the AIOps algorithm competition annually.

The main contributions of our work are summarized as follows.

- **Publication of large-scale real-world datasets:** We have released three large-scale real-world datasets^[32–34], covering essential aspects of AIOps such as KPI anomaly detection, root cause localization for multi-dimensional data, and fail-

ure discovery and diagnosis. These datasets have served as valuable resources and have been utilized by many research works in the field.

- **Systematic dataset design and validation:** Our datasets are constructed through a well-defined methodology ensuring data quality and comprehensive coverage of real-world scenarios. We have implemented a rigorous failure injection process and provided expert labeling from domain practitioners. The practical value of these datasets has been validated through their use in numerous studies and annual AIOps algorithm competitions.

2 Our Work

Building upon existing work^[43], we categorize AIOps research into two main areas: failure management and resource provisioning, as illustrated in Fig. 1. Failure management focuses on minimizing the negative impact of failures, with the ultimate goal of ensuring service availability and optimal user experience. This area encompasses various techniques and approaches to mitigate failure impact. Meanwhile, resource provisioning pertains to the allocation and optimization of diverse resources, such as power and virtual machines (VMs), in the context of IT service management. This aspect of AIOps aims to effectively manage and utilize resources to meet service demands while optimizing resource consumption and costs. At present, our primary focus lies within the domain of failure management. However, we acknowledge that resource provisioning is an important aspect that warrants attention and could potentially be a focus area for our

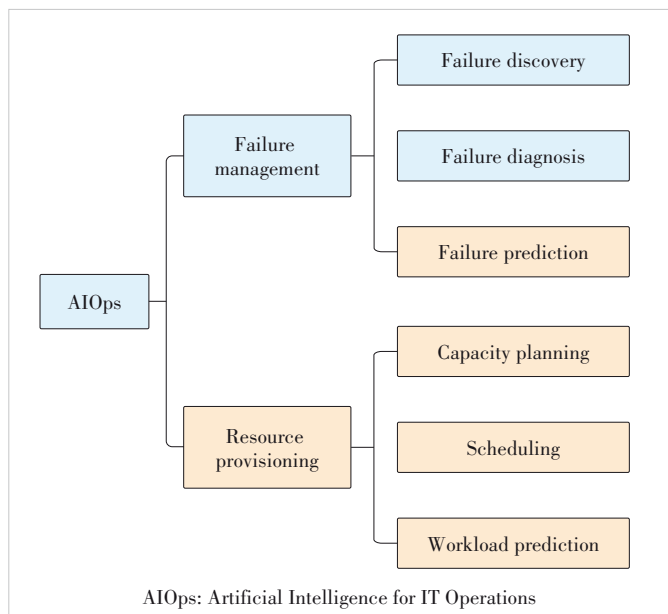


Figure 1. Taxonomy of AIOps research

¹ <https://competition.aiops-challenge.com/home/competition>

future work.

As depicted in Fig. 2, mean time to detect (MTTD) measures failure-discovery latency, mean time to restore (MTTR) measures the time required to diagnose and restore services after a failure, and mean time between failures (MTBF) measures service reliability between successive failures. To address these limitations, we draw inspiration from ImageNet^[44] and aim to develop large-scale real-world benchmarks for AIOps. Our initial focus is on failure management, and we have published three related datasets. These datasets encompass KPI anomaly detection (Section 3), root cause localization for multi-dimensional data (Section 4), and failure discovery and diagnosis (Section 5). In the subsequent sections, we provide an overview of each dataset and its background.

3 KPI Anomaly Detection

In this section, we provide an overview of the background of KPI anomaly detection, followed by a detailed description of the dataset we have published to facilitate research in this area.

3.1 Background

To monitor the operational status of services and promptly identify system failures, engineers often collect a multitude of KPIs and analyze them to detect anomalies. Accurate KPI anomaly detection plays a crucial role in triggering timely troubleshooting, minimizing economic losses, and maintaining a satisfactory user experience^[3-4, 36, 45].

Over the years, substantial efforts have been devoted to KPI anomaly detection. Existing approaches can be broadly categorized into three main types: supervised methods (e.g., Opprentice^[4] and EGADS^[30]), unsupervised methods (e.g., Donut^[3]), and semi-supervised methods (e.g., ADS^[2]). Each method offers unique techniques for KPI anomaly detection.

Despite these advancements, several significant challenges still impede research progress in KPI anomaly detection^[36]. These challenges need to be addressed to enhance the effectiveness and efficiency of anomaly detection methods in real-world scenarios.

- Various KPI patterns: KPIs exhibit diverse patterns, which can be classified into three main categories: seasonal (e.g., transaction volume), stable (e.g., success rate), and un-

stable (e.g., CPU utilization). A successful KPI anomaly detection approach should generalize well across different pattern types, ensuring its applicability in various scenarios.

- Diverse KPI anomaly patterns: KPI anomalies can manifest in different forms, such as spikes, dips, and jitters, representing deviations from normal behavior. Moreover, different scenarios and components may have varying tolerance levels towards anomalies. Therefore, an effective approach should be capable of detecting diverse anomaly patterns and adapting to different contexts.

- Limited availability of labeled data: Obtaining accurately labeled KPI data is challenging. Large-scale systems often generate a vast number of KPIs, making it difficult to label anomalies manually. Moreover, labeling anomalies requires domain experts, further increasing the time and effort required. Consequently, the scarcity of labeled data hampers the construction of supervised models and the evaluation of detection performance.

Despite the existence of numerous KPI anomaly detection approaches, their evaluation is often limited to private datasets, and the lack of large-scale real-world datasets hinders their practical deployment in industry. Public benchmarks, such as Yahoo S5^[30] and the Numenta Anomaly Benchmark^[46], contain mixtures of real-world and synthetic or artificial series and have substantially shorter series than Dataset A.

In reality, the KPIs in large-scale systems are numerous and diverse. Therefore, there is a pressing need for a comprehensive, large-scale, and diverse KPI dataset for anomaly detection. Such a dataset would enable researchers and practitioners to develop and evaluate their approaches on realistic scenarios, bridging the gap between promising research results and real-world performance.

By addressing these limitations and providing a rich dataset for KPI anomaly detection, we aim to facilitate the development of more effective and practical anomaly detection techniques for real-world deployment.

3.2 Dataset

To address the aforementioned challenges, we have developed a large-scale real-world KPI anomaly detection dataset, referred to as Dataset A. This dataset has been constructed to encompass diverse KPI and anomaly patterns, and it provides sufficient labeled data for model training and evaluation. Dataset A was collected from Sogou, eBay, Baidu, Tencent, and Alibaba; Dataset B was derived from production data at Suning; and Dataset C was collected from a production system at China Mobile Zhejiang.

Dataset A consists of 7 KPIs, each of which has been labeled for anomalies by experienced engineers from these five companies. This ensures the accuracy and reliability of the labeled data. In Table 1, we provide some key statistics about Dataset A, including the total number of data points, anomaly ratio, and time span. Notably, the time span of each

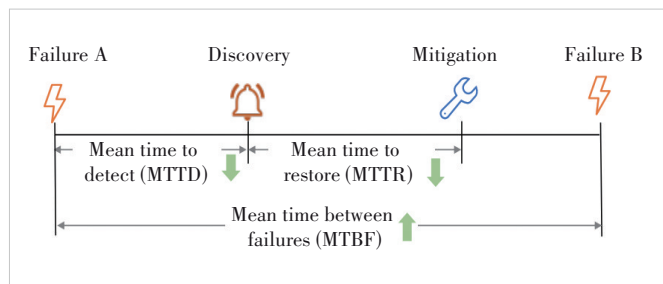


Figure 2. Pipeline and key measurements in failure management

KPI in the dataset ranges from two months to seven months, which demonstrates its large-scale nature compared with existing datasets like Yahoo^[30]. Fig. 3 illustrates five selected KPIs within a one-week time span from Dataset A, reflecting the diverse nature of the dataset.

By providing such a comprehensive dataset, we aim to facilitate research and development in KPI anomaly detection, enabling the exploration and evaluation of novel approaches in real-world scenarios.

Utilizing Dataset A, we organized the inaugural AIOps algorithm challenge in 2018, which garnered significant participation from both academia and industry. The challenge attracted 125 teams and over 300 participants, highlighting the strong interest in the AIOps community. Detailed information about the competition can be found on the competition website.²

To evaluate the performance of KPI anomaly detection algorithms effectively, we introduced a novel evaluation method inspired by the alarm delay restriction proposed in Donut^[3]. This evaluation method, which takes into account alarm delay constraints, was specifically designed for the competition. To facilitate further research and development, we have made the evaluation method available on GitHub,³ allowing researchers to assess the performance of their algorithms using the same metrics and constraints.

Since its release, Dataset A and the associated evaluation method have been widely adopted and referenced in various research works, including papers such as Refs. [35 – 37, 41]. This demonstrates the utility and impact of the dataset and evaluation method for KPI anomaly detection.

During the competition, the best F1-score achieved by the participating teams reached 0.821 6, showcasing the high-quality solutions for KPI anomaly detection. Moreover, the solutions proposed by the top teams have been publicly shared,⁴ allowing for further analysis and collaboration among researchers and practitioners.

As shown in Table 2, Dataset A demonstrates several advantages compared to existing benchmarks in KPI anomaly

Table 1. Summary of the preliminary-stage release of Dataset A

Statistic	Value	Range	Mean
KPI series	26	-	-
Data points	4 821 526	16 495 – 295 379	185 443.31
Reported anomaly ratio	-	0.01% – 4.37%	0.96%
Time span/d	-	60 – 207	148.31

KPI: key performance indicator

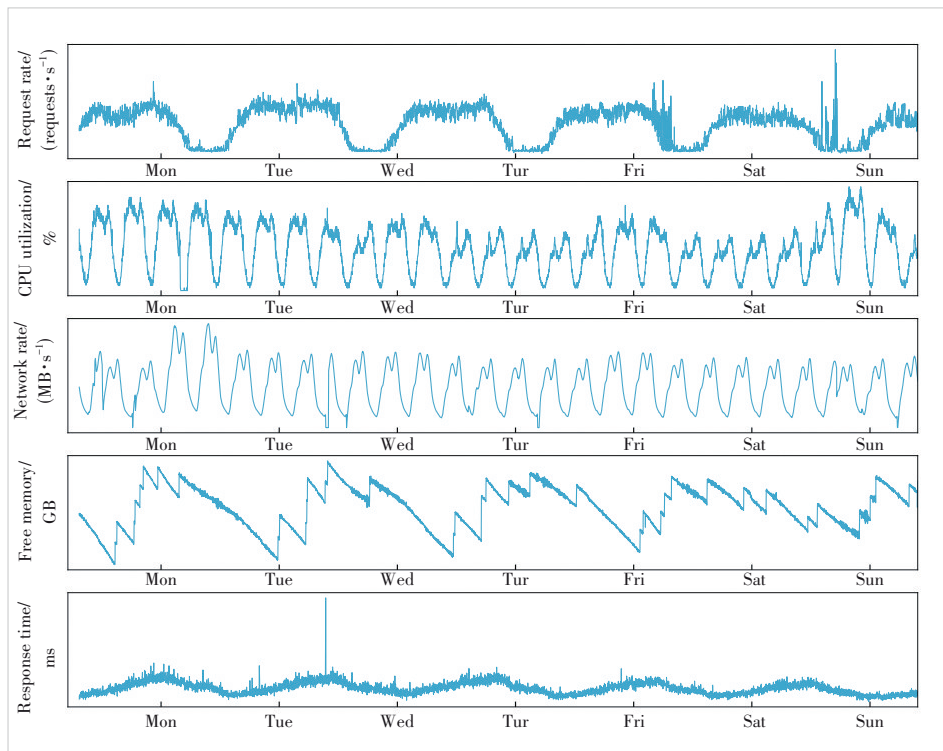


Figure 3. Some KPI examples of Dataset A

detection. With an average length of 211 969.75 data points, Dataset A is significantly larger than other widely-used datasets such as WSD (36 471.00) and UCR (55 935.25), providing a more robust foundation for large-scale evaluation. Furthermore, like other datasets in our comparison, Dataset A originates from real-world data, ensuring representative evaluation under real-world conditions.

4 Root-Cause Localization for Multi-Dimensional Data

In this section, we first introduce the background of root cause localization for multi-dimensional data. Then we present our published large-scale dataset for this problem.

² <https://competition.aiops-challenge.com/home/competition/1484452272200032281>

³ <https://github.com/iopsai/iops/tree/master/evaluation>

⁴ <https://workshop.aiops.org>

Table 2. Comparison of KPI anomaly detection datasets

Dataset	Origin	Average Length	Average Anomaly Ratio
A	Real	211 969.75	0.022 7
WSD	Real	36 471.00	0.016
UCR	Real	55 935.25	0.003
TODS	Real	10 000.00	0.058
Yahoo	Real	1 561.22	0.007
NAB	Real	15 863.10	0.098

KPI: key performance indicator

NAB: Numanta Anomaly Benchmark

TODS: Time-series Outlier Detection System

WSD: Web Service Dataset

4.1 Background

To ensure close monitoring of system events, a wide range of structured logs are generated, which contain crucial information for diagnosing failures. These structured logs document events and encompass multiple fields that provide descriptive details. For instance, consider the example of an online shopping platform that generates structured logs to record orders. In the provided example displayed in Table 3, the Timestamp field indicates the occurrence time of the order, the Amount field specifies the monetary value associated with the order, the Region field denotes the customer's location, and the internet service provider (ISP) field identifies the network utilized by the customer.

Engineers aim to gather aggregated information from specific fields, such as the total dollar amount, to monitor the overall status of the system. These fields, which encompass aggregated values, are referred to as measures^[6,9]. On the other hand, the remaining fields that characterize events are referred to as attributes^[6-7,9,11].

By grouping event-level records within a monitoring interval by Region and ISP and aggregating Order Amount, engineers obtain multi-dimensional data^[6]. Table 4 presents one resulting snapshot, including actual and expected order amounts, which serves as direct input for root cause localization.

A failure in the system is indicated when the overall value of a measure deviates from the expected value, such as a decrease in the total dollar amount compared to the expected value. However, in practical scenarios, failures typically affect only specific attribute combinations, resulting in abnormal measure values^[7,47]. An attribute combination refers to a subset of the entire multi-dimensional data obtained by selecting records with specified attribute values, and its formal definition can be found in existing literature^[7,47]. In Table 4, for example, the measure values associated with {Region=Beijing} are substantially below their expected values. Such combinations indicate the scope of a failure and provide clues for diagnosis.

Table 3. Example of raw structured transaction logs

Order ID	Timestamp	Amount (CNY)	Region	ISP
T001	2021.11.11 10:00:01	102	Beijing	China Mobile
T002	2021.11.11 10:00:05	134	Beijing	China Unicom

ISP: Internet service provider

Table 4. Aggregated data grouped by Region and ISP for root-cause localization

Region	ISP	Amount (CNY)	Expected Amount
Beijing	China Mobile	32	64
Beijing	China Unicom	64	128
Shanghai	China Unicom	192	198
Guangdong	China Mobile	64	62
Zhejiang	China Unicom	13	13
Guangdong	China Unicom	1 283	1 347
Shanxi	China Unicom	128	141
Jiangsu	China Unicom	1 283	1 302
Tianjin	China Mobile	263	276

ISP: Internet service provider

4.2 Dataset

The majority of existing research studies on multi-dimensional data root cause localization^[7-16] utilize proprietary datasets for evaluation. Consequently, to advance the field, we have developed and made publicly available a dataset^[33] (referred to as Dataset B) based on real-world data obtained from Suning Ltd., a large online shopping platform. This dataset aims to facilitate research in this area.

Dataset B contains five attributes labeled as i, e, c, p, and l for confidentiality reasons. The measure is defined as the average number of orders per minute. The dataset comprises a total of 400 synthetic failures, each accompanied by ground-truth root cause labels, spanning eight weeks. Synthetic failures were used instead of real-world failures due to the difficulty in obtaining a substantial number of production failures with known ground-truth root causes.

The process of synthesizing failures involves six steps:

1) Anomaly injection: A time point is selected where the overall measure values are smooth.

2) Attribute selection: A random set of m attributes (where $m \in \{1, 2, 3, 4\}$) is chosen from the five available attributes.

3) Ground-truth root cause selection: n attribute combinations (where $n \in \{1, 2, 3, 4\}$) are randomly selected to represent the ground-truth root cause. These combinations correspond to the m selected attributes.

4) Modification of measure values: The measure values of the rows that satisfy the selected attribute combinations at the chosen time point are modified. The modified values are calculated using the Generalized Ripple Effect (GRE)

method^[47] with a random magnitude.

5) Addition of noise: Gaussian-distributed noise is added to the modified rows and other rows to simulate cases where GRE is not perfectly accurate and forecast residuals exist.

6) Synthetic failure validation: The synthetic failure is checked against the following conditions. If any of these conditions are triggered, the process returns to Step 2 to regenerate a failure at the same time point.

- Existence of other attribute combinations that share almost the same rows as the selected ground-truth attribute combinations.

- Injection of noise that is so significant that the overall measure value of unmodified records (from Step 3) becomes abnormal.

Among the 400 synthetic faults, 144 faults contain 1 root-cause attribute combination, 131 contain 2 combinations, 89 contain 3 combinations, and 36 contain 4 combinations. Furthermore, 96 faults involve 1 attribute, 136 involve 2 attributes, 108 involve 3 attributes, and 60 involve 4 attributes.

Utilizing Dataset B, we organized the second AIOps Challenge in 2019.⁵ The competition attracted 141 teams and 547 participants from both academia (39%) and industry (61%). The best-performing team achieved an F1-score of 0.959 3. Detailed solutions proposed by the top teams have been published in the workshop proceedings^[42].

Several existing studies^[6, 10] have utilized the data provided by the AIOps Challenge 2019. The B0-B4 datasets released with Squeeze^[6] are derived from the same original data as Dataset B and use a similar synthesis process.

As shown in Table 5, Dataset B contains 400 labeled synthetic failure cases generated from production measurements, whereas the RobustSpot evaluation dataset^[48] contains 135 manually investigated real-world anomaly cases. Both datasets have five attributes. While numerous studies in this field rely on private datasets that cannot be independently verified or reproduced, we introduce this dataset as a significant contribution to the research community. The severe shortage of public datasets has historically hindered reproducible research and fair comparative evaluations in root cause localization.

5 Failure Discovery and Diagnosis

In Sections 3 and 4, we presented our datasets for KPI anomaly detection and root cause localization for multi-dimensional data, respectively. These datasets serve as valuable resources for evaluating failure discovery and diagnosis algorithms, respectively. However, there are two important considerations. Firstly, engineers often require end-to-end automatic failure discovery and diagnosis. Secondly, while root cause attribute combinations provide important clues,

Table 5. Comparison of multi-dimensional data root cause localization datasets

Dataset	Origin	Case	Dimension
B	Real	400	5
RobustSpot	Real	135	5

they still necessitate further diagnosis to obtain detailed and fine-grained root causes.

To address these considerations, we have constructed a dataset for benchmarking both failure discovery and fine-grained diagnosis approaches. In this section, we provide an overview of the problem's background, followed by a detailed description of our dataset.

5.1 Background

A distributed system comprises multiple cooperating components, such as load balancers, web servers, application servers, and databases. To illustrate, we present the architecture of a production distributed system called sysA at China Mobile Zhejiang, a major Internet service provider (see Table 4). Each vertex in the figure represents a component, and these components can be categorized into five classes: Oracle Service Bus (OSB), service, database (DB), Docker, and operating system (OS).

To monitor the status of these components, engineers collect various KPIs related to their performance, such as queries per minute, success rate, and average response time. By applying KPI anomaly detection techniques to these metrics, we can identify failures within the distributed system and determine which components are exhibiting abnormal behavior. However, due to complex dependencies, a failure occurring in a root-cause component can propagate to other components, resulting in abnormal KPIs for those as well^[40].

To diagnose the failure, it is necessary to distinguish the root cause components from the other abnormal components. Additionally, to localize the fine-grained root causes of a failure, we need to identify the specific root cause metrics associated with the root cause components. Each component has multiple metrics available for monitoring, such as CPU utilization and network throughput. By identifying the root cause metrics, such as CPU usage, we can determine that the failure is caused by CPU exhaustion in that particular component.

In summary, our objective is to detect failures through KPI anomaly detection, and subsequently identify the root cause components and the specific metrics associated with the failure.

As introduced in Section 3, KPI anomaly detection has been extensively studied. The existing approaches to identifying root cause components^[31, 39 - 40, 49 - 53] can be classified into two categories: trace-based^[31, 39 - 40, 51 - 52] and dependency graph-based^[49 - 50, 53]. In addition, other notable ap-

⁵ <https://competition.aiops-challenge.com/home/competition/1484446614851493956>

proaches aim to identify root cause metrics. FluxRank^[54] formulates the metric ranking task as a learning-to-rank problem, where a machine learning model is trained to rank metrics by their importance for anomaly detection. MicroDiag^[55], CauseRank^[56] and CIRCA^[57] discover causal graphs among metrics and rank them by their causal impact on the observed anomalies.

5.2 Dataset

In the field of failure diagnosis, many studies rely on private datasets for evaluation, limiting the reproducibility and comparability of results. While some existing works have published datasets, such as those by Zhou et al.^[31] and Li et al.^[40], these datasets primarily consist of traces and do not support fine-grained failure diagnosis. Although there are public datasets available, such as Azure and Alibaba datasets that contain fine-grained metrics, they lack ground-truth information regarding failure time points and root causes. This hinders the thorough evaluation and benchmarking of failure discovery and fine-grained diagnosis approaches.

To address this gap and provide a comprehensive benchmark, we have constructed and published a dataset called Dataset C^[34]. This dataset has already been utilized by several existing works^[38–40, 58]. By leveraging Dataset C, researchers can evaluate their failure discovery and fine-grained diagnosis techniques in a more comprehensive and standardized way.

Dataset C is based on the architecture of sysA, as depicted in Fig. 4. This dataset contains three types of monitoring data: traces, KPIs, and metrics. In the dataset, user requests are initially sent to one of the OSB components, which then invoke other components such as service and DB. The entire execution process of a user request is called a trace. Each trace is recorded using spans, which represent units of work performed by each component of the distributed system, following the OpenTracing specification.

An example trace is presented in Table 6, where each row

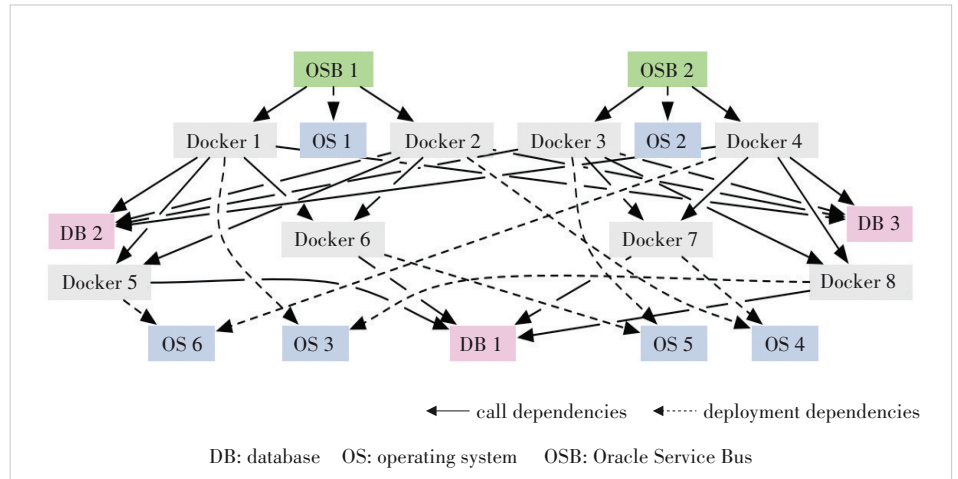


Figure 4. Components and their relationships, including call dependencies (solid) and deployment dependencies (dashed), in the distributed sysA

corresponds to a span. Each span has a unique identifier (id), a parent-span identifier (pid), and a trace identifier (traceId), which together describe the parent-child relationships within a trace. The remaining fields record the Unix start timestamp in milliseconds (startTime), duration in milliseconds (elapsedTime), response status (success), call type, and component identifier (cmdbId). In addition to traces and KPIs, Dataset C includes fine-grained metrics for each component.

By aggregating the attributes of spans within each component, KPIs can be derived to reflect the overall status of each component. The KPIs of a component, such as success rate and average response time, can be obtained by grouping spans based on the cmdbId field and aggregating relevant fields (e.g., success, elapsedTime). The system-level KPIs (i.e., KPIs of the OSB components) are already provided in the published dataset.

In addition to traces and KPIs, the dataset also includes fine-grained metrics for each component. These metrics can be categorized into four categories, as summarized in Table 7. These metrics provide detailed information about the performance and behavior of each component in the distributed system.

Using the traces, KPIs, and metrics provided in Dataset C, researchers can evaluate and develop techniques for failure discovery and fine-grained diagnosis realistically and comprehensively.

In Dataset C, we have included 169 injected failures with their corresponding ground-truth time points and root causes.

Table 6. An example trace in the dataset

callType	startTime(Unix)/ms	elapsedTime/ms	success	traceId	id	pid	cmdbId	serviceName
OSB	1590249600016	274	True	568385015	919979385015	None	os_021	osb_001
CSF	1590249600025	257	True	568385015	4eba31485915	919979385015	os_021	csf_001
RemoteProcess	1590249600026	254	True	568385015	82dad5063a15	4eba31485915	docke_003	csf_001

CSF: Client Service Framework

OSB: Oracle Service Bus

RemoteProcess: remote process call

These failures span one month. Injected failures are used instead of real-world failures due to the difficulty of obtaining a large number of real-world failures with accurately labeled root causes. In the sysA system, we have injected a total of 7 types of failures. These failure types and their descriptions are summarized in Table 8.

In 2020, the third AIOps Challenge⁶ was conducted based on Dataset C, which aimed at online failure detection and diagnosis. The competition attracted 141 teams and 517 participants from both academia and industry.

The challenge required participants to detect and diagnose the injected failures in Dataset C. For each failure, teams were allowed to submit at most two potential root-cause metrics. In cases of container network failures (loss and delay), as container-level network metrics were not collected, participants were asked to localize the root cause containers only.

To evaluate the submitted results, a precision score threshold of 0.5 was used. A submission for a failure was

Table 7. Metrics collected in Dataset C

Category	Metrics List
Docker	container_{thread_total, fgct, thread_idle, thread_used_pct, session_used, thread_running, fg, cpu_used, mem_used, fail_percent}
Linux	Agent_ping, Buffers_used, CPU_{idle_pct, user_time, system_time, iowait_time, util_pct}, Cache_used, Disk_{rd_ios, io_util, await, avgqu_sz, svctm, wr_kbs, rd_kbs, wr_ios}, FS_{used_space, used_pct, total_space, max_util, max_avail}, ICMP_ping, Incoming_network_traffic, Memory_{used_pct, used, available, free, available_pct, total}, Num_{of_processes, of_running_processes}, Outgoing_network_traffic, Page_{po, pi}, Processor_{load_5_min, load_1_min, load_15_min}, Received_{packets, errors_packets, queue}, Recv_total, Send_total, Sent_{errors_packets, queue, packets}, Shared_memory, Swap_used_pct, System_{wait_queue_length, block_queue_length}, Zombie_Process, ss_total
Oracle	ACS, AIOS, AWS, Asm_Free_Tb, CPU_{free_pct, Used_Pct}, Call_Per_Sec, DFParaWrite_Per_Sec, DbFile_Used_Pct, DbTime, Exec_Per_Sec, Hang, LFPParaWrite_Per_Sec, LFSync_Per_Sec, Logic_Read_Per_Sec, Login_Per_Sec, MEM_{real_util, Used, Total, Used_Pct}, New_{Tbs_Free_Gb, Tbs_Used_Pct}, On_Off_State, PGA_{Used_Pct, used_total}, Physical_Read_Per_Sec, Proc_{User_Used_Pct, Used_Pct}, Redo_Per_Sec, Row_Lock, SEQ_Used_Pct, SctRead_Per_Sec, SeqRead_Per_Sec, Sess_{Used_Undo, Used_Temp, Connect, Active}, Session_pct, TPS_Per_Sec, Tbs_{Used_Pct, Free_Gb}, TempTbs_Pct, Total_Tbs_Size, UndoTbs_Pct, Used_Tbs_Size, User_Commit, tnspring_result_time
Redis	Redis_key_count, blocked_clients, connected_clients, evicted_keys, expired_keys, instantaneous_{output_kbps, input_kbps, ops_per_sec}, keyspace_{hits, misses}, maxmemory, mem_fragmentation_ratio, redis_{ping, load}, rejected_connections, total_{connections_received, commands_processed}, used_{memory_peak, cpu_sys, memory_rss, cpu_user, memory}

considered valid only if met or exceeded this threshold. For each failure, valid submissions were ranked in ascending order of time to diagnose divided by the $F_{0.5}$ score (lower is better), and a result at rank i received $\max(10 - i + , 0)$ points. The best competition score was 755 over 169 scored failures. The public description does not specify how these 169 scored failures map to the 169 injected failures reported for Dataset C.

This challenge provided a platform for participants to demonstrate their skills in failure detection and diagnosis using Dataset C, further advancing research and development in AIOps.

As shown in Table 9, Dataset C differs from existing benchmarks in provenance and monitoring modalities. It provides aligned traces and metrics from a production system with injected-failure ground truth. DejaVu^[56] reports 601 failures across three production systems and one open-source benchmark, whereas Eadro^[58] reports 234 injection operations on two open-source testbeds. These counts are not directly comparable because the studies use different systems

Table 8. Failure injection in Dataset C

Component	Injection type	Description
Database	Close	Close the listening port of the target instance
Database	Session limit	Decrease the session limit of the target instance
Container	CPU stress (*)	Stress the CPU of the target container
Container	Network delay (\$)	Randomly delay packets on the target container
Container	Network loss (\$)	Randomly drop packets on the target container
Physical node	Network delay (\$)	Randomly delay packets on the target node
Physical node	Network loss (\$)	Randomly drop packets on the target node

Notes: stress-ng (*) and tc (\$) are injection tools.

Table 9. Comparison of failure discovery and diagnosis datasets

Dataset	Origin	Cases	Modality
Dataset C	Real	169	Trace/Metric
DejaVu	Testbed	601	Trace/Metric
Eadro	Testbed	117	Log/Metric/Trace
Nezha	Testbed	21	Log/Metric/Trace
SMAP, MSL	Real	105	Metric
TraceRCA	Testbed	33	Trace

MSL: Mars Science Laboratory

SMAP: Soil Moisture Active and Passive

⁶ <https://competition.aiops-challenge.com/home/competition/1484441527290765368>

and counting units.

6 Conclusions

AIOps has attracted significant attention from both academia and industry due to its potential. However, the lack of publicly available real-world and large-scale datasets remains a major obstacle to research progress. To address this gap, we have published three datasets covering KPI anomaly detection, multi-dimensional root cause localization, and failure discovery and diagnosis.

In addition, we have organized an annual algorithm competition based on these datasets, attracting numerous teams from around the world. This competition provides a platform for practitioners and researchers to advance AIOps techniques and enhance service reliability.

Moving forward, we will continue releasing datasets covering a wider range of AIOps scenarios and plan to persist in organizing the competition.

References

- [1] Moore S. Gartner: how to get started with AIOps [EB/OL]. [2024-04-15]. <https://www.gartner.com/smarterwithgartner/how-to-get-started-with-aioops>
- [2] Bu J, Liu Y, Zhang S, et al. Rapid deployment of anomaly detection models for large number of emerging KPI streams [C]//Proceedings of the 2018 IEEE International Performance, Computing and Communications Conference (IPCCC). IEEE, 2018: 1 – 8. DOI: 10.1109/PCCC.2018.8711315
- [3] Xu H, Chen W, Zhao N, et al. Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications [C]//Proceedings of the 2018 World Wide Web Conference (WWW). ACM, 2018: 187 – 196. DOI: 10.1145/3178876.3185996
- [4] Liu D P, Zhao Y J, Xu H W, et al. Opprentice: towards practical and automatic anomaly detection through machine learning [C]//Proceedings of the 2015 Internet Measurement Conference. ACM, 2015: 211 – 224. DOI: 10.1145/2815675.2815679
- [5] Ma M, Zhang S, Chen J, et al. Jump-starting multivariate time series anomaly detection for online service systems [C]//Proceedings of the 2021 USENIX Annual Technical Conference. USENIX Association, 2021: 413 – 426. DOI: 10.5555/3472716.3472752
- [6] Li Z Y, Luo C Y, Zhao Y W, et al. Generic and robust localization of multi-dimensional root causes [C]//Proceedings of IEEE 30th International Symposium on Software Reliability Engineering (ISSRE). IEEE, 2019: 47 – 57. DOI: 10.1109/ISSRE.2019.00015
- [7] Gu J, Luo C, Qin S, et al. Efficient incident identification from multi-dimensional issue reports via metaheuristic search [C]//Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2020). ACM, 2020: 1234 – 1245. DOI: 10.1145/3368089.3409742
- [8] Ahmed F, Erman J, Ge Z H, et al. Detecting and localizing end-to-end performance degradation for cellular data services based on TCP loss ratio and round trip time [J]. IEEE/ACM transactions on networking, 2017, 25(6): 3709 – 3722. DOI: 10.1109/TNET.2017.2761758
- [9] Bhagwan R, Kumar R, Ramjee R, et al. Adtributor: revenue debugging in advertising systems [C]//Proceedings of the 11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14). USENIX Association, 2014: 43 – 55
- [10] Jing P K, Han Y N, Sun J Y, et al. AutoRoot: a novel fault localization schema of multi-dimensional root causes [C]//Proceedings of IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2021: 1 – 7. DOI: 10.1109/wcnc49053.2021.9417302
- [11] Lin Q, Lou J-G, Zhang H, et al. iDice: problem identification for emerging issues [C]//Proceedings of the 38th International Conference on Software Engineering (ICSE 2016). ACM, 2016: 214 – 224. DOI: 10.1145/2884781.2884824
- [12] Lin F, Muzumdar K, Laptev N P, et al. Fast dimensional analysis for root cause investigation in a large-scale service environment [J]. Proceedings of the ACM on measurement and analysis of computing systems, 2020, 4(2): 1 – 23. DOI: 10.1145/3392149
- [13] Qureshi M A, Qiu L L, Mahimkar A, et al. Multi-dimensional impact detection and diagnosis in cellular networks [C]//Proceedings of 16th International Conference on Mobility, Sensing and Networking (MSN). IEEE, 2020: 561 – 568. DOI: 10.1109/msn50589.2020.00093
- [14] Sun Y Q, Zhao Y J, Su Y, et al. HotSpot: anomaly localization for additive KPIs with multi-dimensional attributes [J]. IEEE access, 2018, 6: 10909 – 10923. DOI: 10.1109/ACCESS.2018.2804764
- [15] Wang H, Rong G P, Xu Y C, et al. ImpAPTr: a tool for identifying the clues to online service anomalies [C]//Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering. ACM, 2021: 1307 – 1311. DOI: 10.1145/3324884.3415301
- [16] Zhang X, Du C, Li Y F, et al. HALO: hierarchy-aware fault localization for cloud systems [C]//Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. ACM, 2021: 3948 – 3958. DOI: 10.1145/3447548.3467190
- [17] Ma M H, Yin Z, Zhang S L, et al. Diagnosing root causes of intermittent slow queries in cloud databases [J]. Proceedings of the VLDB endowment, 2020, 13(8): 1176 – 1189. DOI: 10.14778/3389133.3389136
- [18] Zhao N, Chen J, Wang Z, et al. Real-time incident prediction for online service systems [C]//Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2020). ACM, 2020: 315 – 326. DOI: 10.1145/3368089.3409745
- [19] Lin Q, Hsieh K, Dang Y, et al. Predicting node failure in cloud service systems [C]//Proceedings of the 26th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2018). ACM, 2018: 858 – 868. DOI: 10.1145/3236024.3275511
- [20] Xu Y, Sui K, Yao R, et al. Improving service availability of cloud systems by predicting disk error [C]//Proceedings of the 2018 USENIX Annual Technical Conference (USENIX ATC 18). USENIX Association, 2018: 481 – 494. DOI: 10.5555/3277355.3277400
- [21] Chen Y J, Yang X, Lin Q W, et al. Outage prediction and diagnosis for cloud service systems [C]//Proceedings of World Wide Web Conference. ACM, 2019: 2659 – 2665. DOI: 10.1145/3308558.3313501
- [22] Shatnawi M, Hefeeda M. Real-time failure prediction in online services [C]//Proceedings of IEEE Conference on Computer Communications (INFOCOM). IEEE, 2015: 1391 – 1399. DOI: 10.1109/infocom.2015.7218516
- [23] Zhang K, Xu J, Min M R, et al. Automated IT system failure prediction: a deep learning approach [C]//Proceedings of the 2016 IEEE International Conference on Big Data (Big Data 2016). IEEE, 2016: 1291 – 1300. DOI: 10.1109/BigData.2016.7840760
- [24] Chen Z B, Kang Y, Li L Q, et al. Towards intelligent incident management: why we need it and how we make it [C]//Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. ACM, 2020: 1487 – 1497. DOI: 10.1145/3368089.3417055
- [25] Chen J, He X, Lin Q, et al. An empirical investigation of incident triage for online service systems [C]//Proceedings of the 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice. IEEE, 2019: 111 – 120. DOI: 10.1109/ICSE-SEIP.2019.00021
- [26] Chen J, He X, Lin Q, et al. Continuous incident triage for large-scale online service systems [C]//Proceedings of the 34th IEEE/ACM International

- tional Conference on Automated Software Engineering (ASE 2019). IEEE, 2019: 1126 – 1129. DOI: 10.1109/ASE.2019.00128
- [27] Lou J G, Lin Q W, Ding R, et al. Experience report on applying software analytics in incident management of online service [J]. Automated software engineering, 2017, 24(4): 905 – 941. DOI: 10.1007/s10515-017-0218-1
- [28] Delimitrou C, Kozyrakis C. Quasar: resource-efficient and QoS-aware cluster management [J]. ACM SIGPLAN notices, 2014, 49(4): 127 – 144. DOI: 10.1145/2644865.2541941
- [29] Cortez E, Bonde A, Muzio A, et al. Resource central: understanding and predicting workloads for improved resource management in large cloud platforms [C]//Proceedings of the 26th Symposium on Operating Systems Principles. ACM, 2017: 153 – 167. DOI: 10.1145/3132747.3132772
- [30] Laptev N, Amizadeh S, Flint I. Generic and scalable framework for automated time-series anomaly detection [C]//Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM, 2015: 1939 – 1947. DOI: 10.1145/2783258.2788611
- [31] Zhou X, Peng X, Xie T, et al. Latent error prediction and fault localization for microservice applications by learning from system trace logs [C]//Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering. ACM, 2019: 683 – 694. DOI: 10.1145/3338906.3338961
- [32] NetManAIOps. KPI-Anomaly-Detection: 2018 AIOps Challenge dataset [EB/OL]. [2024-04-15]. <https://github.com/netmanaiops/kpi-anomaly-detection>
- [33] NetManAIOps. MultiDimension-Localization: 2019 AIOps Challenge dataset [EB/OL]. [2024-04-15]. <https://github.com/NetManAIOps/MultiDimension-Localization>
- [34] NetManAIOps. AIOps-Challenge-2020-Data [EB/OL]. [2024-04-15]. <https://github.com/NetManAIOps/AIOps-Challenge-2020-Data>
- [35] Wang X R, Yin K L, Ouyang Q Y, et al. Identifying erroneous software changes through self-supervised contrastive learning on time series data [C]//Proceedings of IEEE 33rd International Symposium on Software Reliability Engineering (ISSRE). IEEE, 2022: 366 – 377. DOI: 10.1109/issre55969.2022.00043
- [36] Ren H S, Xu B X, Wang Y J, et al. Time-series anomaly detection service at microsoft [C]//Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2019: 3009 – 3017. DOI: 10.1145/3292500.3330680
- [37] Zhao N W, Zhu J, Liu R, et al. Label-less: a semi-automatic labelling tool for KPI anomalies [C]//Proceedings of IEEE Conference on Computer Communications. IEEE, 2019: 1882 – 1890. DOI: 10.1109/INFOCOM.2019.8737429
- [38] Cai Y, Han B, Li J, et al. ModelCoder: a fault model based automatic root cause localization framework for microservice systems [C]//Proceedings of IEEE/ACM 29th International Symposium on Quality of Service (IWQOS). IEEE, 2021: 1 – 6. DOI: 10.1109/iwqos52092.2021.9521318
- [39] Yu G B, Chen P F, Chen H Y, et al. MicroRank: end-to-end latency issue localization with extended spectrum analysis in microservice environments [C]//Proceedings of the Web Conference 2021. ACM, 2021: 3087 – 3098. DOI: 10.1145/3442381.3449905
- [40] Li Z Y, Chen J J, Jiao R, et al. Practical root cause localization for microservice systems via trace analysis [C]//Proceedings of IEEE/ACM 29th International Symposium on Quality of Service (IWQOS). IEEE, 2021: 1 – 10. DOI: 10.1109/iwqos52092.2021.9521340
- [41] Chen Z, Liu J, Su Y, et al. Adaptive performance anomaly detection for online service systems via pattern sketching [C]//Proceedings of the 44th International Conference on Software Engineering (ICSE 2022). ACM, 2022: 2009 – 2020. DOI: 10.1145/3510003.3510087
- [42] AIOps Workshop [EB/OL]. [2024-04-15]. <https://workshop.aiops.org>
- [43] Deng J, Dong W, Socher R, et al. ImageNet: a large-scale hierarchical image database [C]//Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2009). IEEE, 2009: 248 – 255. DOI: 10.1109/CVPR.2009.5206848
- [44] Notaro P, Cardoso J, Gerndt M. A survey of AIOps methods for failure management [J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2021, 12(6): 1 – 45. DOI: 10.1145/3483423
- [45] Zhang X, Kim J, Lin Q, et al. Cross-dataset time series anomaly detection for cloud systems [C]//Proceedings of the 2019 USENIX Annual Technical Conference (USENIX ATC 19). USENIX Association, 2019: 1063 – 1076. DOI: 10.5555/3358807.3358903
- [46] Lavin A, Ahmad S. Evaluating real-time anomaly detection algorithms: the numenta anomaly benchmark [C]//Proceedings of IEEE 14th International Conference on Machine Learning and Applications (ICMLA). IEEE, 2015: 38 – 44. DOI: 10.1109/ICMLA.2015.141
- [47] Li Z, Chen J, Chen Y, et al. Generic and robust root cause localization for multi-dimensional data in online service systems [J]. Journal of systems and software, 2023, 203: 111748. DOI: 10.1016/j.jss.2023.111748
- [48] Li Z, Zhao N, Li M, et al. Actionable and interpretable fault localization for recurring failures in online service systems [C]//Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2022). ACM, 2022: 996 – 1008. DOI: 10.1145/3540250.3549162
- [49] Kim M, Sumbaly R, Shah S. Root cause detection in a service-oriented architecture [J]. ACM SIGMETRICS performance evaluation review, 2013, 41(1): 93 – 104. DOI: 10.1145/2494232.2465753
- [50] Ma M, Xu J, Wang Y, et al. AutoMAP: diagnose your microservice-based web applications automatically [C]//Proceedings of the 2020 World Wide Web Conference (WWW 2020). ACM, 2020: 320 – 330. DOI: 10.1145/3366423.3380115
- [51] Lin J, Chen P, Zheng Z. Microscope: pinpoint performance issues with causal graphs in micro-service environments [C]//Proceedings of the 16th International Conference on Service-Oriented Computing (ICSOC 2018). Springer, 2018: 3 – 20. DOI: 10.1007/978-3-030-03596-9_1
- [52] Gan Y, Liang M, Dev S, et al. Sage: practical and scalable ML-driven performance debugging in microservices [C]//Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS 2021). ACM, 2021: 135 – 151. DOI: 10.1145/3445814.3446700
- [53] Yu Q, Pei C, Hao B, et al. CMDiagnosor: an ambiguity-aware root cause localization approach based on call metric data [C]//Proceedings of the ACM Web Conference 2023 (WWW 2023). ACM, 2023: 2937 – 2947. DOI: 10.1145/3543507.3583347
- [54] Liu P, Chen Y, Nie X, et al. FluxRank: a widely-deployable framework to automatically localize root cause machines for software service failure mitigation [C]//Proceedings of the 2019 IEEE 30th International Symposium on Software Reliability Engineering. IEEE, 2019: 35 – 46. DOI: 10.1109/ISSRE.2019.00014
- [55] Wu L, Tordsson J, Bogatinovski J, et al. MicroDiag: fine-grained performance diagnosis for microservice systems [C]//Proceedings of IEEE/ACM International Workshop on Cloud Intelligence (CloudIntelligence). IEEE, 2021: 31 – 36. DOI: 10.1109/cloudintelligence52565.2021.00015
- [56] Lu X, Xie Z, Li Z, et al. Generic and robust performance diagnosis via causal inference for OLTP database systems [C]//Proceedings of the 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet Computing (CCGrid 2022). IEEE, 2022: 655 – 664. DOI: 10.1109/CCGrid54584.2022.00076
- [57] Li M, Li Z, Yin K, et al. Causal inference-based root cause analysis for online service systems with intervention recognition [C]//Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2022). ACM, 2022: 3230 – 3240. DOI: 10.1145/3534678.3539041
- [58] Lee C, Yang T, Chen Z, et al. Eadro: an end-to-end troubleshooting

framework for microservices on multi-source data [C]//Proceedings of the 45th IEEE/ACM International Conference on Software Engineering (ICSE 2023). IEEE, 2023: 1750 – 1762. DOI: 10.1109/ICSE48619.2023.00150

Biographies

Li Zeyan received his PhD degree in computer science and technology from Tsinghua University, China in 2023, where he was a member of the Tsinghua NetMan Lab under the supervision of Prof. Pei Dan. He received his bachelor's degree in computer science and technology from Tsinghua University in 2018. His research interests focus on Artificial Intelligence for IT Operations (AIOps), particularly intelligent fault localization, root cause analysis, and reliability management for large-scale online service systems.

Han Jing (han.jing28@zte.com.cn) is a researcher and engineer at ZTE Corporation, where she develops next-generation intelligent network architectures, 6G wireless intelligence, and automated software maintenance systems.

Zhao Nengwen received her PhD degree in computer science and technology from Tsinghua University, China in 2022 under the supervision of Prof. Pei Dan, and her bachelor's degree in computer science and technology from Wuhan University, China in 2017. Her research interests include Artificial Intelligence for IT Operations (AIOps), software maintenance, and reliability management, with particular emphasis on anomaly detection, alert management, log anomaly detection, and software change analysis in large-scale online service systems.

Zhang Shenglin is a Professor at the College of Software, Nankai University, China and an affiliated faculty member of the Tsinghua NetMan Lab. He received his PhD degree in computer science and technology from Tsinghua University, China under the supervision of Prof. Pei Dan and Prof. Liu Ying, and his bachelor's degree in network engineering from Xidian University, China in 2012. During his doctoral studies, he was a visiting scholar at the School of Computer Science, Georgia Institute of Technology, USA. His research focuses on Artificial Intelligence for IT Operations (AIOps), including failure detection, diagnosis and prediction, root cause analysis, log and multi-modal observability analysis, and reliability management for large-scale service and cloud systems.

Sun Yongqian is an Associate Professor at Nankai University, China and an affiliated faculty member of the Tsinghua NetMan Lab. He received his PhD

degree from the Department of Computer Science and Technology at Tsinghua University, China under the supervision of Prof. Pei Dan and Prof. Zhao Youjian, and his bachelor's degree in statistics from Northwestern Polytechnical University, China in 2012. His research interests center on Artificial Intelligence for IT Operations (AIOps), including anomaly detection, failure diagnosis, root cause analysis, failure prediction, and intelligent operations for microservice and large-scale software systems.

Chen Pengfei is a Professor and doctoral supervisor at the School of Computer Science and Engineering, Sun Yat-sen University, China. He received his PhD degree in computer science from Xi'an Jiaotong University, China in 2016. His research interests include Artificial Intelligence for IT Operations (AIOps), distributed systems, cloud computing, microservices, software reliability, and computer networks, with particular focus on automated performance diagnosis, root cause analysis, anomaly detection, and intelligent management of cloud-native and microservice systems.

Wen Xidao was a postdoctoral researcher at Tsinghua University, China and a member of the Tsinghua NetMan Lab, completing his postdoctoral work in 2022. His research interests include Artificial Intelligence for IT Operations (AIOps), data mining, machine learning, time-series analysis, anomaly detection, fault diagnosis, and intelligent operations for large-scale service systems. He has collaborated extensively on research in multivariate time-series anomaly detection, software change analysis, root cause diagnosis, AIOps benchmarks, and large language models for IT operations.

Ma Minghua is a Senior Researcher at Microsoft Research in Redmond, USA. He received his PhD degree in computer science and technology from Tsinghua University, China in 2021 under the supervision of Prof. Pei Dan. From 2019 to 2020, he was a visiting scholar at the School of Computer Science, Georgia Institute of Technology, USA. His research interests lie at the intersection of software engineering, systems, and artificial intelligence, with a particular focus on Cloud Intelligence and Artificial Intelligence for IT Operations (AIOps).

Pei Dan is a Professor in the Department of Computer Science and Technology, Tsinghua University, China and the leader of the Tsinghua NetMan Lab. His current research focuses on Artificial Intelligence for IT Operations (AIOps) for critical infrastructure and embodied systems, as well as time-series intelligence. Before joining Tsinghua University, he was a researcher at AT&T Labs – Research in the United States. He is a Senior Member of both ACM and IEEE and has extensive research experience in networked systems, large-scale service reliability, intelligent operations, root cause analysis, anomaly detection, and time-series intelligence.