

Article

# Hi-RCA: A Hierarchy Anomaly Diagnosis Framework Based on Causality and Correlation Analysis

Jingjing Yang \*, Yuchun Guo, Yishuai Chen and Yongxiang Zhao

School of Electronic Information and Engineering, Beijing Jiaotong University,  
Beijing 100044, China; ychguo@bjtu.edu.cn (Y.G.); yschen@bjtu.edu.cn (Y.C.); yxzhaob@bjtu.edu.cn (Y.Z.)  
\* Correspondence: 19111044@bjtu.edu.cn

**Abstract:** Microservice architecture has been widely adopted by large-scale applications. Due to the huge amount of data and complex microservice dependency, it also poses new challenges in ensuring reliable performance and maintenance. Existing approaches still suffer from limitations of anomaly data, over-simplification of metric relationships, and lack of diagnosing interpretability. To solve these issues, this paper builds a hierarchy root cause diagnosis framework, named Hi-RCA. We propose a global perspective to characterize different abnormal symptoms, which focuses on changes in metrics' causation and correlation. We decompose the diagnosis task into two phases: anomalous microservice location and anomalous reason diagnosis. In the first phase, we use Kalman filtering to quantify microservice abnormality based on the estimation error. In the second phase, we use causation analysis to identify anomalous metrics and generate anomaly knowledge graphs; by correlation analysis, we construct an anomaly propagation graph and explain the anomaly symptoms via graph comparison. Our experimental evaluation on an open dataset shows that Hi-RCA can effectively locate root causes with 90% mean average precision, outperforming state-of-the-art methods.

**Keywords:** microservice architecture; intervention recognition; root cause location



**Citation:** Yang, J.; Guo, Y.; Chen, Y.; Zhao, Y. Hi-RCA: A Hierarchy Anomaly Diagnosis Framework Based on Causality and Correlation Analysis. *Appl. Sci.* **2023**, *13*, 12126. <https://doi.org/10.3390/app132212126>

Academic Editor: Benoit Lung

Received: 19 October 2023

Revised: 2 November 2023

Accepted: 3 November 2023

Published: 8 November 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Microservice architecture (MSA) has been widely adopted in domains such as the Internet of Things [1] and mobile and cloud [2] services for its scalability, flexibility, and resilience. MSA decomposes an application into small-scale and single-function microservices which cooperate through lightweight intercommunication [3]. Hence, an MSA-based system contains numerous components and processes, with complex structures and dynamic interactions, where diagnosing anomalies is particularly important.

With the aid of monitoring tools and anomaly detection techniques, a large number of system measurements, such as hardware resource consumption, can be observed. Based on such measurements (i.e., metric data), several meaningful methods have been proposed to pinpoint the anomaly culprit. A common method is to construct the relationship graph of metrics, and pinpoint the root cause metric through the random walking algorithm on the dependency graph [4–7].

However, automatic diagnosis of root causes from the observed data is difficult owing to the following challenges.

- *Limitation by anomaly data volume.* Most of the existing research adopts causal inference techniques to obtain the variables' causality, which requires sufficient length of the anomaly data. However, in real situations, the anomaly duration is an uncontrollable variable. When anomaly data do not satisfy the requirements of causal inference methods, untrusted pseudo-causality will arise, which limits, or even worsens, the diagnosis performance.
- *Over-simplification of metric relationships.* Since the relationship between diverse monitored metrics is complex, causation and correlation exist simultaneously. Existing

research oversimplifies the metric relationship either as causation or correlation. In fact, correlation does not imply causation.

- *Lack of diagnosing interpretability.* Under multiple anomaly types and anomaly cascading, anomaly symptoms are diverse; hence, interpretability for the diagnosis results is important. Anomaly diagnosis requires not only locating the root cause, but also explaining the logic behind anomaly symptoms. For example, a CPU hog occurs at a microservice, causing both CPU metrics and memory metrics to increase. Without the explanation of anomaly symptoms, i.e., the original CPU anomaly propagates to memory, engineers may not trust the diagnosis results, because memory can also be a possible cause.

To address the above challenges, we propose a hierarchy framework, named Hi-RCA. Hi-RCA consists of two diagnosis phases. In the first phase, it pinpoints the anomalous microservice, using the Kalman filtering approach to quantify the microservice abnormality based on the self-contrastive evaluation. In the second phase, it designs a hierarchy-diagnosing method that analyzes the metric causation and correlation separately. Firstly, Hi-RCA transfers the anomaly detection into the causal inference, i.e., the intervention recognition task. Secondly, it constructs the directed anomaly propagation graph based on the metric correlation and anomaly propagation characteristics. The reason for the anomaly is located based on the propagation graph comparison. Therefore, Hi-RCA not only pinpoints the reason for the anomaly, but also infers a directed anomaly propagation graph, to explain the anomaly symptoms. Experimental results show that Hi-RCA has high diagnostic precision in 164 anomaly cases involving different anomaly types. For top three precision, compared to the best baseline, its accuracy improvement ranges from 5%~38% in different datasets.

Our contribution can be summarized as follows.

- We propose a general anomaly evaluator using Kalman filtering, which requires no expert knowledge and is general for various metrics with different characteristics.
- We use the intervention recognition task to identify anomalous metrics, to avoid the causal inference performance being limited by anomaly data distribution, which utilizes the causation change in metrics for accurate anomaly diagnosis.
- We analyze the causation and correlation separately. We characterize anomaly symptoms from the global perspective where all metrics are considered jointly. Based on the correlation-based anomaly knowledge graph, diverse anomaly symptoms can be explained.

The remainder of this paper is organized as follows. Related work is discussed in Section 2. Section 3 gives the problem statement, an overview of Hi-RCA, and detailed localization modules. Section 4 describes the experimental evaluation, ablation study, and the discussion. Section 5 concludes the paper.

## 2. Related Work

Compared to logs and traces, metric data can manifest abnormality in the system and require no instrumentation to the source code. Many approaches in the literature employ observational metrics to infer the root causes of performance issues. We classify the related work based on metric data into two types: non-causal methods and causality-based approaches.

Non-causal methods can be classified into unsupervised models and supervised machine learning-based approaches. For unsupervised methods, it is common to train a baseline model based on normal data, and detect anomalies when KPIs diverge from the baseline model [8–11]. To distinguish the normal fluctuations and anomalous situations, CoFlux [12] determines whether two KPIs are correlated by fluctuations automatically, considering the temporal order and fluctuating direction. Shang et al. [13] analyze time series anomalies based on correlation analysis and the Hidden Markov Model (HMM [14]) which is used to find close relationships between abnormal KPIs.  $\epsilon$ -diagnosis [15] infers the possible root causes based on the similarity computation of monitored samples; KPIs

are anomalous if the similarity is below a given tolerance threshold. Based on Robust Principal Component Analysis (RPCA [16]), CloudDiag [17] identifies the services that most contributed to the anomaly. Root causes are ranked based on the number of times they appeared in anomalous categories. PAL [18] and FChain [19] exploit offline detection of anomalous KPIs to generate a ranked list. To reduce false positives, they filter external reasons by checking whether anomalies affected all application services. The earliest anomalies are indeed considered the most probable root causes.

For supervised methods, labeled anomaly data are required. Nedelkoski et al. [20] utilize a variational autoencoder [21] to model normal behavior and recognize anomalies based on reconstruction error. They also train a convolutional neural network [22] on the anomalous traces, to recognize which failures caused a performance anomaly in a service. Seer [23] trained deep learning algorithms on massive amounts of data to identify root causes, but its performance may degrade with system updates. Scheinert proposed Arvalus and its improved algorithm D-Arvalus [24]. In the two algorithms, the system components are regarded as microservices, and the dependencies between components are regarded as connections, to identify the root cause in a graph. Sage [25] leverages unsupervised learning to identify causes of performance degradation in complex microservice dependency graphs. It examines the impact of microservices on end-to-end performance based on microservice dependency graphs via graph variational autoencoders. GDN [26] detects anomalies based on multivariate time series. It uses an implicit embedding feature construction to learn the relationship between monitored objects, uses a graph attention-based method to predict the metric values, and anomalies are detected based on the comparison of the results between real values and the predicted values. UBL [27] leverages self-organizing maps to capture emergent system behaviors and predict unknown anomalies for cloud infrastructures.

Machine learning-based models provide meaningful perspectives for root cause location. However, the limitations come from two aspects. The first is labeled data limitation; in the actual situation, it is hard to obtain sufficient anomaly cases for model training. The second is interpretability, which is vital for the result of root cause location. Hence, engineers need to understand the mechanism behind how the root cause is located; otherwise, the outcome may not be trusted.

For causation-based analysis, one of the dominant methods for anomaly diagnosis is to derive an automatic causality graph [5–7,28], then determine the root cause based on the graph-based analysis. In these causality graphs, vertices are typically modeled as the services, and oriented arcs indicate the microservices' dependency. To capture the causality, several causal inference methods are used. Refs. [5,6] exploit the PC algorithm [29] to build a causality graph while considering the service availability and resource consumption.

Ref. [30] uses two causal inference models to derive a metric causality graph, where the DirectLiNGAM method [31] is adopted to model the causality between resource metrics and the Granger causality test [32] is used to infer the causality between resource and service metrics. MicroCause [33] applies a variant of the PC algorithm to capture the time order of metrics to metrics to construct the causality among metrics. Qiu et al. [34] propose a root cause analysis method based on a knowledge graph and a causal search algorithm. Furthermore, they describe how to construct a knowledge graph and improve the PC algorithm. Nie et al. [35] propose an automatic diagnosis system based on a causality graph and supervised learning, which require no expert knowledge and implementation details.

Based on the causality graph, different root cause determination methods are proposed, and one of the common methods is the random walking algorithm. Refs. [4–6] perform a random walk to pinpoint the possible root cause. The random walk starts from the application frontend and makes a fixed number of iterations. At each iteration, the probability of visiting a service in the neighborhood is proportional to the correlation of its KPIs with those of the application frontend. The root cause is located based on the visiting time of services, under the assumption that most visited services constitute the most probable root causes for the anomaly. MS-Rank [5] proposes a hybrid influence graph construction algorithm and introduces a new metric concept, which uses multiple types of

metrics to discover the causal relationship between microservices. FacGraph [28] searches for anomaly propagation subgraphs in the obtained causality graphs, each subgraph having a tree-like structure, and the tree root is the frontend. Subgraphs' anomaly scores are based on the appearance frequency in the causality graphs. With a threshold mechanism, root causes are returned as a set of leaf services of the kept subgraphs. ServiceRank [36] uses a second-order random walk to identify the root causes. Loud [8] uses the causality graph of KPI metrics from the anomaly detection system directly and locates the faulty components using different graph centrality algorithms. CauseInfer [37] uses the depth-first search method to traverse the metric causal dependency graph of each microservice to obtain the cause of the failure.

In different causality formulations, some researchers build dependent graphs from the correlation analysis. MonitorRank [38] uses the historical and current time series metrics, considering internal and external factors. It proposes a pseudo-anomaly clustering algorithm to classify external factors and identifies anomalous services with a random walk algorithm. MicroHECL [39] analyzes possible anomaly propagation chains and ranks candidate root causes. FluxInfer [40] constructs a weighted undirected dependency graph to represent the dependency relationships of anomalous KPIs, then applies a weighted PageRank algorithm to localize root cause-related KPIs.

However, since a microservice system consists of diverse microservices with a large amount of monitoring metrics, it is hard to capture the accurate dependency based on the microservices' invocation relationship and deployment relationship. Furthermore, the dependency graph is not equivalent to the anomaly propagation graph. Hence, under diverse anomaly types with different propagation situations, it is challenging to pinpoint the root cause based on a static graph. For causality models, sufficient anomaly data volume is required, but unfortunately, due to the actual situation, it is hard to collect enough anomaly data to satisfy the requirements. Some methods require independent tests for all gathered metrics, which may cause a significant overhead.

### 3. Model

#### 3.1. Problem Formulation

The data analyzed in this paper are the metric data of microservice systems. Suppose there are  $M$  microservices; for each microservice,  $N$  indicators (metrics) have been monitored. When an anomaly occurs, given metric data in a certain time window  $T$ , our goal is to locate which microservice is anomalous and what is the reason for the anomaly, i.e., the root cause microservice  $m_{rc}$  and root cause metric  $e_{rc}$ .

#### 3.2. System Framework

In this paper, we propose an anomaly diagnosis framework based on a hierarchy inference mechanism, named Hi-RCA, characterizing anomaly symptoms from a global perspective. As shown in Figure 1, Hi-RCA mainly consists of two modules: *Anomalous Microservice Locator* and *Anomalous Reason Diagnoser*. In the first module, based on the monitored metric data of MSA, *Anomalous Microservice Locator* assesses all of the microservices' abnormality based on the estimation error of the evaluated microservice's metrics. In the second module, given the anomalous microservice  $m_{rc}$ , *Anomalous Reason Diagnoser* locates the root cause of the anomaly based on graph comparison. Firstly, we transfer the anomaly detection into an intervention recognition task, which not only detects anomalous metric type, but also generates the anomaly knowledge graph. We adopt a structured causal model to formulate the causal relationship between four meta-metrics and recognize the anomalous metric types based on regression-based hypothesis testing. Then, *Anomalous Reason Diagnoser* constructs the correlation-based anomaly propagation graph (CPG) based on the metric correlation analysis, which regards all anomalous metrics as a whole for anomaly analysis.

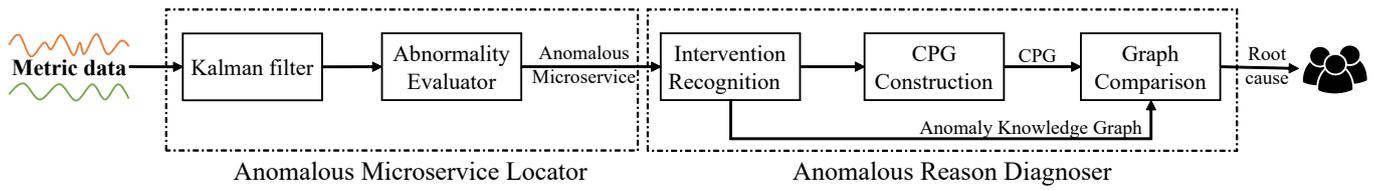


Figure 1. Hi-RCA anomaly diagnosis procedures.

### 3.3. Anomalous Microservice Locator

Since a microservice system consists of several microservices with various monitored metrics, we propose a step-by-step diagnosis strategy to decrease the computational burden. In the first module, the *Anomalous Microservice Locator* needs to find the anomalous microservice among all microservices in the system. However, valuable information is submerged in numerous monitoring data. A microservice has many monitored metrics with various magnitudes, periodicity, etc. Customizing an anomaly evaluator for diverse metrics is unrealistic. In this paper, we design an *Anomalous Microservice Locator*, adopting Kalman filtering [41] and calculating the estimation error to evaluate metrics' abnormality.

Kalman filtering is used for the following reasons. (1) There exists system noise in the data collection process, causing burrs and fluctuations in normal data, which can be smoothed with Kalman filtering. (2) When the system is in a normal condition, system noise can be regarded as white noise; when an anomaly occurs, anomaly data will exhibit a significant difference from normal data. Therefore, the optimal estimation based on normal data will not be able to fit abnormal points, so we use the estimation error to quantify the metric abnormality.

**Kalman filtering** [41]. Kalman filtering is an optimal state estimation process applied to a dynamic linear system that involves random perturbations, and it has been widely used in many areas of industrial, even complicated real-time applications.

Consider a linear system with a state space description

$$x_k = Ax_{k-1} + Bu_{k-1} + \omega_{k-1} \tag{1}$$

$$Z_k = Hx_k + v_k \tag{2}$$

where  $A$  and  $B$  are known constant matrices.  $x_k$  is the system state, i.e., the estimation variable.  $\omega_k$  and  $v_k$  are, respectively, unknown system and observation noise sequences.  $Z_k$  is the observed sequence.  $H$  is the state observation matrix. Then, the a priori estimation  $\hat{x}_{k-}$  can be derived as

$$\hat{x}_{k-} = A\hat{x}_{k-1-} + Bu_{k-1} \tag{3}$$

$$P_{k-} = AP_{k-1}A^T + Q \tag{4}$$

The optimal estimation (i.e., a posteriori state estimation) of  $x$  is  $\hat{x}_k$ :

$$\hat{x}_k = \hat{x}_{k-} + K_k(z_k - H\hat{x}_{k-}) \tag{5}$$

$$P_k = (I - K_kH)P_{k-} \tag{6}$$

$$K_k = \frac{P_{k-}H^T}{HP_{k-}H^T + R} \tag{7}$$

where  $P_k$  is the estimate (error) covariance matrix,  $K_k$  is the Kalman gain, and it represents the ratio of model predicted error and measurement error in the optimal estimation process.

**Estimation error-based anomaly evaluation.** Given the time window  $T$ , for a microservice  $i$ , the metric data of its  $j$ th metric (i.e., KPI) is a time series,  $\mathbf{X}_{i,j}^T$ .

At time  $t$ , the monitored (observed) value is  $x_{i,j}^t$ . At each time  $t$ , we compute the optimal estimation  $\hat{x}_{i,j}^t$ . The anomaly severity of  $x_{i,j}^t$  is computed as the estimation error between the observed value  $x_{i,j}^t$  and the estimated value  $\hat{x}_{i,j}^t$ :

$$s_{i,j}^t = \left| \frac{x_{i,j}^t - \hat{x}_{i,j}^t}{\hat{x}_{i,j}^t} \right| \tag{8}$$

For a microservice  $i$ , its anomaly severity of  $x_i^t$  is denoted as the summarization of all its KPIs:

$$s_i^t = \sum_j s_{i,j}^t \tag{9}$$

Then, based on the anomaly score, root cause microservice  $m_{rc}$  is pinpointed as the microservice with the highest anomaly score:

$$m_{rc} = \arg \max_i (s_i^t) \tag{10}$$

### 3.4. Anomalous Reason Diagnoser

Given the anomalous microservice  $m_{rc}$ , the target of the *Anomalous Reason Diagnoser* is to diagnose the real reason for the anomaly (i.e.,  $e_{rc}$ ) among several anomalous metrics caused by the anomaly propagation. Under multiple anomalous metrics, checking each metric’s variation in isolation is insufficient, because the metric with the highest degree of anomaly may not be the root cause, but one of the affected indicators. In Hi-RCA, we focus on complex associations among various metrics, including causation and correlation. To fully utilize their relationships, we propose a hierarchy approach to diagnose the root cause gradually.

In the first phase, the *Anomalous Reason Diagnoser* identifies anomaly metric type based on the metric causality analysis, which adopts the structural causal model to formulate the causality relationship and recognizes anomalous metrics by the intervention recognition. Furthermore, based on the recognition results, anomaly knowledge graphs are generated. In the second phase, based on the correlation analysis, the *Anomalous Reason Diagnoser* pinpoints the root cause metric and explains anomaly symptoms. Specifically, based on the anomalous type of intervention recognition, we classify one type of metric into the utilization group and the failure group, and compute the anomaly propagation graph among different groups. Finally, the root cause is located based on the graph similarity between CPG and the anomaly knowledge graph.

**Intervention recognition.** The invention recognition problem using Judea Pearl’s “Ladder of Causation” [42]. The first layer of the causal ladder encodes the observation knowledge  $L_1(X) = P(X)$ , where  $P(X)$  is the joint probability distribution. The second layer encodes the interventional knowledge  $L_2(m) = P_m$ , where  $P_m(X) = P(X|do(m))$  and  $M \subseteq X$ . The *do-operator*  $do(m)$  means fixing variable  $M$  to the given values  $m$ , which is defined as an intervention [43].  $P(X|do(m))$  denotes the probability distribution over  $X$  under the intervention to  $M$ .

If we want to answer the question layer  $i$ , we need knowledge at layer  $i$  or higher.

We model the causal relations among meta-metrics using the structural causal model (SCM) [43]. An SCM model consists of a set of structural equations

$$x_k = f_k(pa(X_k), u_k) \tag{11}$$

where  $x_k \in X$  and  $pa(X_k) \in X$ ,  $u_k$  denotes unobserved variables  $\mathbf{U}_k, U_i \cap X = \emptyset$ . We define the graph encoded by the SCM as  $G = (V, E)$ , where  $E = \{X_j \rightarrow X_k | X_v \in pa(X_k)\}$  is the set of directed causal edges.  $Ch(X_k) = \{X_v | X_k \in pa(X_v)\}$  is the children of  $X_k$ .

In this paper, the root cause analysis problem is mapping to the intervention recognition problem. Normal data obey the observation distribution. The anomaly occurring in the

microservice system is mapped into an unexpected intervention. Data with anomalies come from the intervention distribution. Hence, the intervention recognition task is formulated as [44].

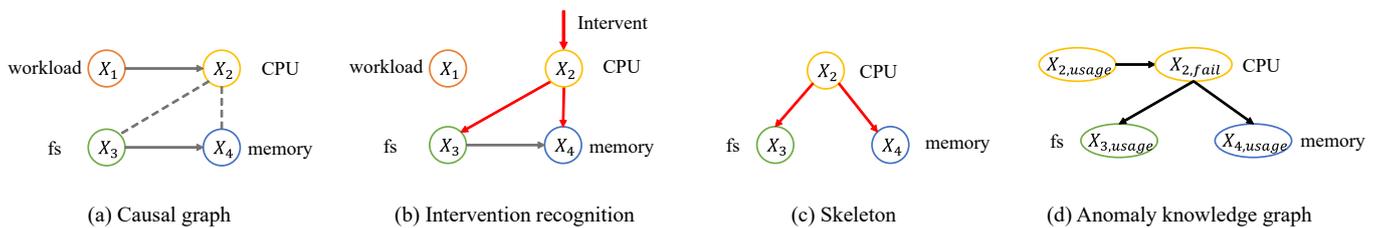
**Definition 1.** (Intervention Recognition, IR). For a given SCM  $M$ , let  $L_1$  be the observational distribution of  $M$  and  $P_m = P(X|do(m))$  be the interventional distribution of a certain intervention  $do(m)$ . Intervention recognition is to find  $m$  based on  $L_1$  and  $P_m$ .

**Theorem 1.** (Intervention Recognition Criterion) Let  $G$  be a CBN and  $pa(X_k)$  be the parents of  $X_k$  in  $G$ . Under the Faithfulness assumption,  $X_k$  is intervened iff  $X_k$  no longer follows the distribution defined by  $pa(X_k)$ , i.e.,  $X_k \in M \Leftrightarrow P_m(x_k|pa(x_k)) \neq L_1(x_k|pa(x_k))$ .

**Intervention recognition-based anomaly detection.** Since the observation data and the intervention data come from two different distributions, instead of comparing two distributions directly, the intervention recognition criterion is reformulated as hypothesis testing, because it is difficult to obtain the complete distribution of the anomaly data.

Similar to [44], we use regression-based hypothesis testing to recognize the abnormal metrics. Those metrics that no longer obey the normal distribution are identified as anomalous metrics, i.e.,  $x_k^t \approx L_1(x_k^t|pa^{(t)}(x_k))$ . The regression model is used to calculate the expected distribution  $L_1(x_k^t|pa^{(t)}(x_k))$ . We train a regression model for each variable using the normal data, as the proxy fitting of the structural equation. We calculate the residual of the regression value and observed value  $x_k^t$ . We assume the residual follows the normal distribution  $N(\mu_{\epsilon,k}, \sigma_{\epsilon,k})$ . Metric  $x_k$  is abnormal if its residual is out of the normal distribution.

In order to perform efficient dependency construction from complex relationships, Hi-RCA selects four meta-metrics to model and monitor dependent changes in causality between metrics: workload, CPU utilization, memory utilization, and file system (fs) utilization. The procedures of intervention are presented in Figure 2. As shown in Figure 2a, we model the causal relationship between four meta-metrics; directed arrows denote the causal relationships, and the dotted line indicates potential causation. In Figure 2b, we present an example where the CPU node is intervened, then its causation changes, i.e., it does not obey the distribution given its parent (workload) and generates new causations to the fs node and memory node. Then, the CPU node is recognized as the intervened node, i.e., anomalous node. Based on the intervened nodes, we construct the anomaly knowledge graph for root cause localization.



**Figure 2.** The processes of intervention recognition and anomaly knowledge graph construction. (a,b) show the process of intervention recognition. (c,d) present the construction of an anomaly knowledge graph.

**Construction of anomaly knowledge graph (AKG).** Based on the intervention recognition task, we expand the causal graph  $G$  into the anomaly knowledge graph (AKG). Assume the anomalous set  $O$  is recognized as the intervened nodes. For each node  $o$ , the graph skeleton  $H$  is the intervened results, as shown in Figure 2c. Then, we expand  $H$  into an AKG based on two rules. (1) For the same metric type, anomaly propagates from the utilization group into the failure group. (2) For different metric types, anomaly

propagates from the parent node's failure group into the child node's utilization group. Figure 2d presents an example of AKG.

**CPG-based anomaly diagnosis.** Since there exist various anomalies with different anomaly symptoms, we diagnose the root cause based on CPG, not the causality graph owing to the following reasons. (1) The anomaly period is uncertain; under a short anomaly period, it is unable to obtain the anomaly distribution, or use the existing causal inference techniques to discover the metric causation. (2) Correlation is another type of metric association that can characterize diverse anomaly symptoms.

Specifically, we divide one type of metric into two groups: resource utilization and the failure group. For example, CPU-related metrics are divided into CPU usage groups and CPU failure groups. Based on the anomalous data, we generate CPG, and the correlation between groups is denoted as the maximum average correlation. In two different groups  $G_U$  and  $G_V$ , for  $x_u \in G_U, x_v \in G_V$ , we compute the maximum correlation of different groups, which represent the anomaly propagation direction, and the directed edges in CPG demonstrate the probable process of anomaly propagation between metrics:

$$C_{U,V} = \max_u \left( \frac{\sum_v \text{Pearson}(x_u, x_v)}{|G_V|} \right) \quad (12)$$

Here, we choose a threshold-based method; edges with correlation  $C_{U,V}$  higher than the threshold  $\alpha$  are selected. Then, we generate the correlation-based propagation graph, which denotes the anomaly propagation among different metric groups. Based on CPG, we compute the graph similarity between the CPG and anomaly knowledge graphs.

Finally, we compute the graph similarity between the real-time CPG and AKG to pinpoint the root cause. The graph similarity of two graphs  $G_i$  and  $G_j$  is defined as

$$\text{Sim}_{G_i, G_j} = \frac{|E_{G_i} \cap E_{G_j}|}{|E_{G_j}|} \quad (13)$$

where  $E_{G_i}$  denotes the edge set of graph  $G_i$ . The root cause  $e_{rc}$  is pinpointed as the metric that has the highest similarity score of its anomaly knowledge graph and CPG.

## 4. Experiment

### 4.1. Experiment Setup

**Dataset Description.** In this paper, we focus on resource-type failures, such as CPU hog, memory leak, etc. We used three datasets A, B, and C, published by the AIOps Challenge 2022 [45], containing 164 resource failures in total. Dataset A contains 59 failures, dataset B contains 50 failures, and dataset C contains 55 failures. The main difference between these datasets is their deployment relationship. All of the datasets are collected from a widely used open microservice system, Hipster-Shop [46], a web-based e-commerce app that consists of 10 microservices.

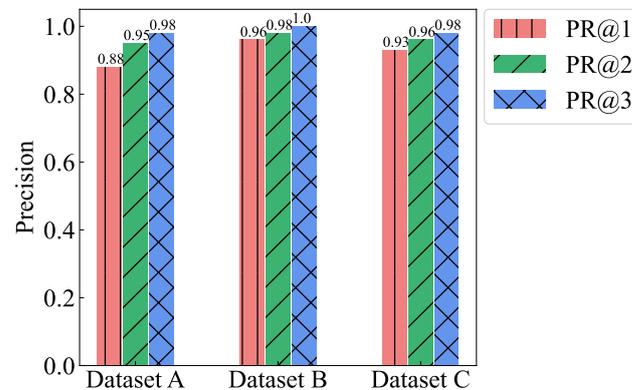
**Evaluation Metrics.** To quantify the performance of each system, we use the following performance metric. Precision at top  $k$  denotes the probability that the top  $k$  results given by a system include the real root cause, denoted as  $PR@k$ . A higher  $PR@k$  score, especially for small values of  $k$ , represents that the system correctly identifies the root cause. Let  $R[i]$  be the rank of each cause and  $e_{rc}$  be the set of root causes. More formally,  $PR@k$  is defined on a set of given anomalies  $A$  as

$$PR@k = \frac{1}{|A|} \sum_{a \in A} \frac{\sum_{i < k} (R[i] \in e_{rc})}{(\min(k, |e_{rc}|))} \quad (14)$$

### 4.2. Evaluation of Anomalous Microservice Locator

In this subsection, we discuss the effectiveness of *Anomaly Microservice Locator*. In Figure 3, we present the performance of Hi-RCA in anomalous microservice location in different datasets. It shows that Hi-RCA can achieve almost 90% in terms of  $PR@1$ , it can

effectively locate all root causes in the top three anomaly services, and its performance achieves almost 100% in terms of  $PR@3$ .



**Figure 3.** Precision of anomaly microservice location in different datasets.

Since the anomalous microservice location is the middle process in Hi-RCA, we compare its performance with other baseline methods to illustrate its effectiveness. We choose three different anomaly detectors; for each microservice, its abnormality is quantified as the amount of its anomalous metrics. The root cause microservice is defined as the microservice with the highest abnormality severity.

#### Baseline anomaly detectors.

- *Quantile anomaly detector* [47] compares each time series value with historical quantiles. In our experiment, we detect anomalies if their value is above 99% or below 1%.
- *Level-shift anomaly detector* [47] detects a shift of value level by tracking the difference between median values at two sliding time windows next to each other.
- *Cauchy detector* [48] detects an anomaly by comparing the current value and the history-smoothed value in a sliding window.

As presented in Table 1, it can be observed that our *Anomaly Microservice Locator* designed based on Kalman filtering achieves the best accuracy. The reason is that a single anomaly detector can only obtain one type of anomaly and fails to capture various anomalies with different symptoms. Without the consideration of noise in data collection, the detection precision is more vulnerable to false alarms.

**Table 1.** Effectiveness of Kalman filtering with ablation experiments.

Method	Dataset A			Dataset B			Dataset C		
	PR@1	PR@2	PR@3	PR@1	PR@2	PR@3	PR@1	PR@2	PR@3
Quantile	0.25	0.29	0.37	0.26	0.42	0.48	0.27	0.31	0.35
Level-shift	0.66	0.76	0.78	0.82	0.84	0.90	0.71	0.80	0.85
Cauchy	0.73	0.86	0.90	0.84	0.96	0.96	0.82	0.87	0.89
Kalman filtering	0.88	0.95	0.98	0.96	0.98	1.0	0.93	0.96	0.98

#### 4.3. Evaluation of Hi-RCA

In this subsection, we evaluate the effectiveness of our method by comparing it with five baseline methods (RS, Loud, Cauchy, MicroDiag, MicroDiag-V1) and ablation experiments.

##### Baseline methods.

- *MicroDiag* [30]: MicroDiag locates the culprit metric based on the metric's causality graph. It identifies the potential propagation among components first, considers different methods of anomaly propagation, and uses two types of causal inference methods to construct the propagation graph among different components and metrics. The root cause metric is localized based on the PageRank algorithm of the causality graph.

- *MicroDiag-V1*: Since MicroDiag directly locates the root cause metric from diverse components, to eliminate the distractions of diverse microservices, we implement it in a simplified scenario where the anomalous microservice has been located, to compare the effectiveness of MicroDiag.
- *Loud* [8]: Loud localizes the culprit metrics based on a propagation graph of anomalous metrics. The graph is constructed by the Granger causality test of metrics. To implement Loud, the anomalous metrics are chosen by the Cauchy Detector [48], the propagation graph is built using the Granger test, and the root cause is located based on the PageRank result of the propagated graph with weighted edges.
- *Cauchy* [48]: An anomaly detector that can quantify the time series abnormality and detects an anomaly by comparing the current value and the history-smoothed value in a sliding window. We compute all metrics' Cauchy anomaly scores and locate the metric with the highest anomaly score as the root cause. Cauchy is chosen as a baseline to show the result where metrics are observed in isolation, regardless of the association between them.
- *Random Selection (RS)*: Random selection is a method engineers use when lacking domain-specific knowledge of the system. Every time, they randomly select an unchecked metric to investigate until the root cause is found.

We apply Hi-RCA and five baseline methods to all anomaly cases and obtain their performance in terms of  $PR@1$ ,  $PR@2$ , and  $PR@3$ , as shown in Table 2. We can see that none of these methods can pinpoint the culprit metric in the top one of the ranked list. However, compared to the best baseline method MicroDiag-V1, our Hi-RCA achieves 48%~64% precision improvement in  $PR@1$ .

**Table 2.** Performance of Hi-RCA with baseline and ablation experiments.

Method	Dataset A			Dataset B			Dataset C		
	PR@1	PR@2	PR@3	PR@1	PR@2	PR@3	PR@1	PR@2	PR@3
RS	0.03	0.08	0.19	0.14	0.14	0.18	0.05	0.15	0.18
Loud	0.29	0.34	0.39	0.18	0.34	0.38	0.31	0.36	0.38
Cauchy	0.15	0.2	0.24	0.14	0.16	0.18	0.11	0.16	0.24
MicroDiag	0.22	0.46	0.64	0.24	0.48	0.66	0.22	0.53	0.76
MicroDiag-V1	0.27	0.54	0.71	0.22	0.38	0.6	0.22	0.45	0.84
<b>Hi-RCA</b>	<b>0.75</b>	<b>0.85</b>	<b>0.88</b>	<b>0.86</b>	<b>0.89</b>	<b>0.98</b>	<b>0.76</b>	<b>0.8</b>	<b>0.89</b>
Hi-RCA-w/o-1	0.75	0.75	0.85	0.82	0.90	0.92	0.73	0.78	0.82
Hi-RCA-w/o-2	0.07	0.25	0.32	0.2	0.32	0.38	0.11	0.25	0.29

**Comparisons.** Results of MicroDiag show the challenge of root cause localization. Different from Hi-RCA, which pinpoints the anomalous microservice first, MicroDiag directly finds the root cause metric from diverse components. It is difficult to construct complete and accurate causal relationships from the original time series data, owing to the following reasons. (1) Metric data are collected with the background noise. (2) The anomaly not only propagates to related metrics, but also dependent components, and hence, it is challenging to construct an accurate propagation graph under the interference both from affected microservices and metrics. Even given the anomalous microservice, MicroDiag-V1 cannot accurately pinpoint the root cause metric. The reason is that the relationships among metrics are complex, and consist of causal relationships and correlation relationships. If the length of anomaly data is a short window, such as 5 min, causal inference techniques are unable to construct an accurate metric causality graph for effective root cause localization.

Results of Loud show inefficiencies in locating anomalies, because Loud constructs the metric graph among all metrics, but given various monitored metrics with complex relationships, only focusing on the causal relationship is insufficient, since there may not be causality between some metrics, but only correlation. The Granger causality test model used in Loud only analyzes the causal relationship between two metrics, ignoring the complete causal structure. Loud only constructs the metric-propagated graph by relying on

the Granger test result, which may add spurious causal relations among metrics, reducing its efficiency in root cause localization.

Results of Cauchy show that directly pinpointing the root cause metric based on the metric value variation is ineffective, because Cauchy only detects the metrics' value variation, ignoring the metrics' inter-relationship. Owing to the dependent nature of microservices, the anomaly from the root cause metric will propagate to related metrics and dependent microservices, causing many anomalous metrics and microservices. One of the challenges of root cause localization is that there exist different symptoms for different anomaly types; hence, only depending on the metric value variation is one-sided, as it cannot pinpoint the root cause from diverse anomaly symptoms.

**Ablation Study.** We study the main modules of Hi-RCA by removing each of them or using a variant version. We compare Hi-RCA with its two ablation experiments as follows.

- *Hi-RCA-w/o-1*: Hi-RCA algorithm without the causal inference, i.e., the anomaly knowledge graph is constructed as a static knowledge graph manually, not from the intervention recognition task used in Hi-RCA. Then, the reason for the anomaly is inferred based on the graph comparison between CPG and the static knowledge graph.
- *Hi-RCA-w/o-2*: Hi-RCA algorithm with a variant version of the Anomalous Reason Diagnoser, instead of the graph comparison method. The root cause is located based on the PageRank algorithm [49] in the CPG where nodes are the anomaly metrics and edges denote their dependencies.

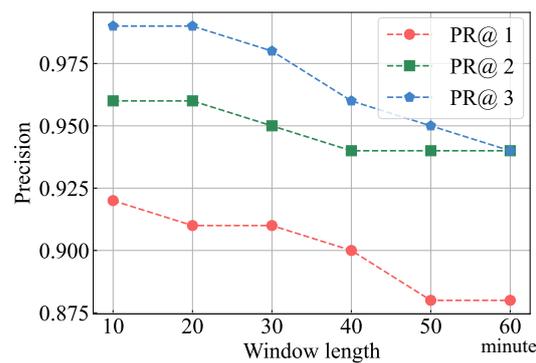
The performance of the ablation study, in terms of  $PR@1$ ,  $PR@2$ , and  $PR@3$ , is presented in Table 2. From the performance comparison between Hi-RCA and *Hi-RCA-w/o-1*, it is observed that the causal inference that recognizes anomalous metrics is better than the isolated anomaly evaluation methods, such as the 3-sigma method. The performance degradation of the *Hi-RCA-w/o-2* demonstrates the efficacy of CPG, which demonstrates the necessity for characterizing the anomaly from a multiple-metric perspective. The root cause metric based on the PageRank algorithm cannot pinpoint the real reason accurately, because the anomaly propagates to dependent metrics, causing several anomalous metrics. Without anomaly analysis from a global view, it is hard to diagnose the reason for anomalies.

#### 4.4. Discussion

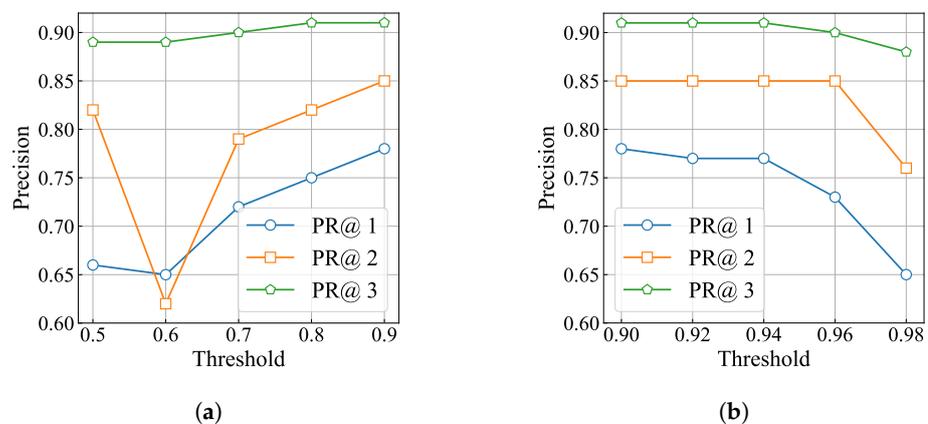
In this subsection, we first discuss the performance impact of experiment parameters in Hi-RCA, and then analyze the overhead and limitations of Hi-RCA.

**Parameters.** In Hi-RCA, there are two main parameters: time window length  $l$  of Kalman filtering in the *Anomalous Microservice Locator* and anomaly threshold  $\alpha$  in the CPG construction process. For time window length  $l$ , which denotes the data length used in Kalman filtering, we observe the  $PR@1$ ,  $PR@2$ , and  $PR@3$  precision of all datasets with increasing  $l$  from 10 min to 60 min. As shown in Figure 4, the longer the data length we use, the lower the precision of the root cause microservice localization. The reason is that with the data length increasing, Kalman filtering tends to fit the data distribution on long-time data, which may contain bigger noise, which decreases the diagnosis performance. In Hi-RCA, we set  $l$  as 10, i.e., data length of 10 min, which is easy to realize in practical systems.

For the threshold of anomaly propagation, i.e.,  $\alpha$ , we observe the  $PR@1$ ,  $PR@2$ , and  $PR@3$  precision of all datasets with increasing  $\alpha$  from 0.5 to 0.98. As shown in Figure 5a, we present location performance with  $\alpha$  ranging from 0.5 to 0.9. It can be observed that the precision of diagnosis does not increase with the threshold increasing, because a small threshold will select many normal relations as the anomaly propagation, which decreases the location performance. As shown in Figure 5b, we present location performance with  $\alpha$  ranging from 0.9 to 0.98. It can be observed that the precision of diagnosis decreases as the threshold increases, because a larger threshold denotes a stricter definition of anomaly propagation, which may miss the true anomaly propagation. In Hi-RCA, we set  $\alpha$  as 0.9, which may not be suitable for other microservice systems. In other systems, the threshold can be set as the maximum score obtained from normal data.



**Figure 4.** Precision variation with different length of time window in Anomaly Microservice Location.



**Figure 5.** Precision variation with different value of anomaly propagation threshold  $\alpha$ . (a) Performance with  $\alpha$  ranges from 0.5 to 0.9; (b) performance with  $\alpha$  ranges from 0.9 to 0.98.

**Overhead.** Since the *Anomalous Microservice Locator* processes all of the metric data, while the *Anomalous Reason Diagnoser* only analyzes anomalous metrics of the root cause microservice, the overhead of Hi-RCA is mostly caused by Kalman filtering, which is the main algorithm in the *Anomalous Microservice Locator*. From [50], the complexity of a single application of Kalman filtering is denoted as  $O(p^{2.4} + q^2)$ , where  $p$  is the dimension of observation, which is 10 in Hi-RCA;  $q$  is the number of states, which is 1 in the one-dimensional time series. The factor 2.4 comes from matrix inversion. In the *Anomalous Microservice Locator*, all of the microservices' monitored metrics are processed. Therefore, the total complexity is  $M * N * O(p^{2.4} + q^2)$ , where  $M$  is the number of microservices in the system, and  $N$  is the number of monitored metrics. For large-scale systems, Hi-RCA can be implemented with other anomaly detection methods to narrow the anomaly scope and shorten the evaluation time.

**Limitations.** In the *Anomalous Microservice Locator*, we adopt a Kalman filtering to quantify microservice anomaly, based on two assumptions. (1) Metrics come from a linear system and (2) data noise obeys the Gaussian distribution. Although these are not strong assumptions, there may exist other systems that do not satisfy the above assumptions. In the *Anomalous Reason Diagnoser*, we transfer the anomaly metric classification as the intervention recognition task; owing to the complex relationship of metrics, we only choose four meta-metrics, and we consider expanding the causal graph in our future work.

## 5. Conclusions

In this paper, we propose a hierarchical anomaly diagnosis method, Hi-RCA. In Hi-RCA, the anomaly symptoms are analyzed based on a global perspective where the metric relationships are sufficiently utilized. In the first phase, we design a self-contrastive anomaly quantification method requiring no expert knowledge or handcraft parameters. Based on Kalman filtering, Hi-RCA locates the anomalous microservice based on the

estimation error of all its metrics. In the second phase, we focus on the change of interdependence of metrics by analyzing the causation and correlation. Firstly, we focus on the causality change to infer the anomalous nodes based on the intervention recognition. Secondly, we construct two types of graphs: (1) the anomaly knowledge graph based on the intervened nodes and (2) the anomaly propagation graph based on the correlation analysis. Finally, the root cause is located based on the comparison of two types of graphs, and the anomaly propagation graph is able to explain how the anomaly propagates in related metrics. Experimental evaluations in open datasets demonstrate that Hi-RCA can accurately localize root causes without labeled data or human intervention.

The indicator data analyzed by Hi-RCA are time series data, a common type of monitoring data. Therefore, Hi-RCA can be migrated to other similar systems, such as the Internet of Things and cloud-based systems. In future works, we will research the causal inference mechanism, and extend the causal graph to more types of metrics, to improve the anomaly diagnosis performance.

**Author Contributions:** Conceptualization, J.Y. and Y.C.; methodology, Y.Z. and J.Y.; validation, J.Y., Y.G. and Y.C.; writing—original draft preparation, J.Y.; writing—review and editing, Y.G., Y.C. and Y.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data used in this article were published by 2022 International AOps Challenge (<https://competition.aiops-challenge.com>), accessed on 1 September 2023.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Butzin, B.; Golasowski, F.; Timmermann, D. Microservices approach for the internet of things. In Proceedings of the IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), Berlin, Germany, 6–9 September 2016; pp. 1–6.
2. Di Francesco, P.; Malavolta, I.; Lago, P. Research on architecting microservices: Trends, focus, and potential for industrial adoption. In Proceedings of the IEEE International Conference on Software Architecture (ICSA), Gothenburg, Sweden, 3–7 April 2017; pp. 21–30.
3. Newman, S. *Building Microservices*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2021.
4. Wang, P.; Xu, J.; Ma, M.; Lin, W.; Pan, D.; Wang, Y.; Chen, P. Cloudranger: Root cause identification for cloud native systems. In Proceedings of the 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Washington, DC, USA, 1–4 May 2018; pp. 492–502.
5. Ma, M.; Lin, W.; Pan, D.; Wang, P. Ms-rank: Multi-metric and self-adaptive root cause diagnosis for microservice applications. In Proceedings of the IEEE International Conference on Web Services (ICWS), Milan, Italy, 8–13 July 2019; pp. 60–67.
6. Ma, M.; Xu, J.; Wang, Y.; Chen, P.; Zhang, Z.; Wang, P. Automap: Diagnose your microservice-based web applications automatically. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; pp. 246–258.
7. Lin, J.; Chen, P.; Zheng, Z. Microscope: Pinpoint performance issues with causal graphs in micro-service environments. In Proceedings of the Service-Oriented Computing: 16th International Conference—ICSOC 2018, Hangzhou, China, 12–15 November 2018; pp. 3–20.
8. Mariani, L.; Monni, C.; Pezzé, M.; Riganelli, O.; Xin, R. Localizing faults in cloud systems. In Proceedings of the IEEE 11th International Conference on Software Testing, Verification and Validation (ICST), Västerås, Sweden, 9–13 April 2018; pp. 262–273.
9. Wu, L.; Tordsson, J.; Elmroth, E.; Kao, O. Microrca: Root cause localization of performance issues in microservices. In Proceedings of the NOMS 2020–2020 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 20–24 April 2020; pp. 1–9.
10. Wu, L.; Bogatinovski, J.; Nedelkoski, S.; Tordsson, J.; Kao, O. Performance diagnosis in cloud microservices using deep learning. In Proceedings of the International Conference on Service-Oriented Computing, Dubai, United Arab Emirates, 14 December 2020; pp. 85–96.
11. Samir, A.; Pahl, C. DLA: Detecting and localizing anomalies in containerized microservice architectures using markov models. In Proceedings of the 7th International Conference on Future Internet of Things and Cloud (FiCloud), Istanbul, Turkey, 26–28 August 2019; pp. 205–213.

12. Su, Y.; Zhao, Y.; Xia, W.; Liu, R.; Bu, J.; Zhu, J.; Cao, Y.; Li, H.; Niu, C.; Zhang, Y.; et al. Coflux: Robustly correlating kpis by fluctuations for service troubleshooting. In Proceedings of the International Symposium on Quality of Service, Phoenix, AZ, USA, 24–25 June 2019; pp. 1–10.
13. Shang, Z.; Zhang, Y.; Zhang, X.; Zhao, Y.; Cao, Z.; Wang, X. Time series anomaly detection for kpis based on correlation analysis and hmm. *Appl. Sci.* **2021**, *11*, 11353. [[CrossRef](#)]
14. Eddy, S.R. Hidden markov models. *Curr. Opin. Struct. Biol.* **1996**, *6*, 361–365. [[CrossRef](#)] [[PubMed](#)]
15. Shan, H.; Chen, Y.; Liu, H.; Zhang, Y.; Xiao, X.; He, X.; Li, M.; Ding, W. ?-Diagnosis: Unsupervised and real-time diagnosis of small-window long-tail latency in large-scale microservice platforms. In Proceedings of the World Wide Web Conference, San Francisco, CA, USA, 13–17 May 2019; pp. 3215–3222.
16. Candès, E.J.; Li, X.; Ma, Y.; Wright, J. Robust principal component analysis? *J. ACM* **2011**, *58*, 1–37. [[CrossRef](#)]
17. Mi, H.; Wang, H.; Zhou, Y.; Lyu, M.R.; Cai, H. Toward fine-grained, unsupervised, scalable performance diagnosis for production cloud computing systems. *IEEE Trans. Parallel Distrib. Syst.* **2013**, *24*, 1245–1255. [[CrossRef](#)]
18. Nguyen, H.; Tan, Y.; Gu, X. Pal: P ropagation-aware a nomaly l ocalization for cloud hosted distributed applications. In *Managing Large-Scale Systems via the Analysis of System Logs and the Application of Machine Learning Techniques*; Association for Computing Machinery: New York, NY, USA, 2011; pp. 1–8.
19. Nguyen, H.; Shen, Z.; Tan, Y.; Gu, X. Fchain: Toward black-box online fault localization for cloud systems. In Proceedings of the IEEE 33rd International Conference on Distributed Computing Systems, Philadelphia, PA, USA, 8–11 July 2013; pp. 21–30.
20. Nedelkoski, S.; Cardoso, J.; Kao, O. Anomaly detection and classification using distributed tracing and deep learning. In Proceedings of the 19th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGRID), Larnaca, Cyprus, 14–17 May 2019; pp. 241–250.
21. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
22. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
23. Gan, Y.; Zhang, Y.; Hu, K.; Cheng, D.; He, Y.; Pancholi, M.; Delimitrou, C. Seer: Leveraging big data to navigate the complexity of performance debugging in cloud microservices. In Proceedings of the 24th International Conference on Architectural Support for Programming Languages and Operating Systems, Providence, RI, USA, 13–17 April 2019; pp. 19–33.
24. Scheinert, D.; Acker, A.; Thamsen, L.; Geldenhuys, M.K.; Kao, O. Learning dependencies in distributed cloud applications to identify and localize anomalies. In Proceedings of the IEEE/ACM International Workshop on Cloud Intelligence (CloudIntelligence), Madrid, Spain, 29–29 May 2021; pp. 7–12.
25. Gan, Y.; Liang, M.; Dev, S.; Lo, D.; Delimitrou, C. Sage: Practical and scalable ml-driven performance debugging in microservices. In Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Virtual, 19–23 April 2021; pp. 135–151.
26. Deng, A.; Hooi, B. Graph neural network-based anomaly detection in multivariate time series. *Proc. AAAI Conf. Artif. Intell.* **2021**, *35*, 4027–4035. [[CrossRef](#)]
27. Dean, D.J.; Nguyen, H.; Gu, X. UBL: Unsupervised behavior learning for predicting performance anomalies in virtualized cloud systems. In Proceedings of the 9th International Conference on Autonomic Computing, San Jose, CA, USA, 18–20 September 2012; pp. 191–200.
28. Lin, W.; Ma, M.; Pan, D.; Wang, P. Facgraph: Frequent anomaly correlation graph mining for root cause diagnose in micro-service architecture. In Proceedings of the IEEE 37th International Performance Computing and Communications Conference (IPCCC), Orlando, FL, USA, 17–19 November 2018; pp. 1–8.
29. Spirtes, P.; Glymour, C.N.; Scheines, R. *Causation, Prediction, and Search*; MIT Press: Cambridge, MA, USA, 2000.
30. Wu, L.; Tordsson, J.; Bogatinovski, J.; Elmroth, E.; Kao, O. Microdiag: Fine-grained performance diagnosis for microservice systems. In Proceedings of the IEEE/ACM International Workshop on Cloud Intelligence (CloudIntelligence), Madrid, Spain, 29 May 2021; pp. 31–36.
31. Shimizu, S.; Inazumi, T.; Sogawa, Y.; Hyvarinen, A.; Kawahara, Y.; Washio, T.; Hoyer, P.O.; Bollen, K.; Hoyer, P. Directlingam: A direct method for learning a linear non-gaussian structural equation model. *J. Mach. Learn. Res.-JMLR* **2011**, *12*, 1225–1248.
32. Granger, C.W.J. Investigating causal relations by econometric models and cross-spectral methods. *Econom. J. Econom. Soc.* **1969**, *37*, 424–438. [[CrossRef](#)]
33. Meng, Y.; Zhang, S.; Sun, Y.; Zhang, R.; Hu, Z.; Zhang, Y.; Jia, C.; Wang, Z.; Pei, D. Localizing failure root causes in a microservice through causality inference. In Proceedings of the IEEE/ACM 28th International Symposium on Quality of Service (IWQoS), Hangzhou, China, 15–17 June 2020; pp. 1–10.
34. Qiu, J.; Du, Q.; Yin, K.; Zhang, S.; Qian, C. A causality mining and knowledge graph based method of root cause diagnosis for performance anomaly in cloud applications. *Appl. Sci.* **2020**, *10*, 2166. [[CrossRef](#)]
35. Nie, X.; Zhao, Y.; Sui, K.; Pei, D.; Chen, Y.; Qu, X. Mining causality graph for automatic web-based service diagnosis. In Proceedings of the IEEE 35th International Performance Computing and Communications Conference (IPCCC), Las Vegas, NV, USA, 9–11 December 2016; pp. 1–8.
36. Ma, M.; Lin, W.; Pan, D.; Wang, P. Servicrank: Root cause identification of anomaly in large-scale microservice architectures. *IEEE Trans. Dependable Secur. Comput.* **2021**, *19*, 3087–3100. [[CrossRef](#)]

37. Chen, P.; Qi, Y.; Zheng, P.; Hou, D. Causeinfer: Automatic and distributed performance diagnosis with hierarchical causality graph in large distributed systems. In Proceedings of the IEEE INFOCOM 2014-IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014; pp. 1887–1895.
38. Kim, M.; Sumbaly, R.; Shah, S. Root cause detection in a service-oriented architecture. *ACM Sigmetrics Perform. Eval. Rev.* **2013**, *41*, 93–104. [[CrossRef](#)]
39. Liu, D.; He, C.; Peng, X.; Lin, F.; Zhang, C.; Gong, S.; Li, Z.; Ou, J.; Wu, Z. Microhecl: High-efficient root cause localization in large-scale microservice systems. In Proceedings of the IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Madrid, Spain, 25–28 May 2021; pp. 338–347.
40. Liu, P.; Zhang, S.; Sun, Y.; Meng, Y.; Yang, J.; Pei, D. Fluxinfer: Automatic diagnosis of performance anomaly for online database system. In Proceedings of the IEEE 39th International Performance Computing and Communications Conference (IPCCC), Austin, TX, USA, 6–8 November 2020; pp. 1–8.
41. Chui, C.K.; Chen, G. Kalman filtering with real time applications. *Appl. Opt.* **1989**, *28*, 1841.
42. Bareinboim, E.; Correa, J.D.; Ibeling, D.; Icard, T. On pearl’s hierarchy and the foundations of causal inference. In *Probabilistic and Causal Inference: The Works of Judea Pearl*; Association for Computing Machinery: New York, NY, USA, 2022; pp. 507–556.
43. Zelterman, D. *Causality: Models, Reasoning, and Inference*; Cambridge University Press: Cambridge, UK, 2001.
44. Li, M.; Li, Z.; Yin, K.; Nie, X.; Zhang, W.; Sui, K.; Pei, D. 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, Washington, DC, USA, 14–18 August 2022; pp. 3230–3240.
45. AIOps Challenge 2022. Available online: <https://competition.aiops-challenge.com/home/competition> (accessed on 1 September 2023).
46. Hipster-Shop with OpenTelemetry. Available online: <https://github.com/yuxiaoba/Hipster-Shop> (accessed on 1 September 2023).
47. ADTK. Available online: <https://adtk.readthedocs.io/en/stable> (accessed on 1 September 2023).
48. Cao, W.; Gao, Y.; Lin, B.; Feng, X.; Xie, Y.; Lou, X.; Wang, P. Tcprt: Instrument and diagnostic analysis system for service quality of cloud databases at massive scale in real-time. In Proceedings of the 2018 International Conference on Management of Data, Houston, TX, USA, 10–15 June 2018; pp. 615–627.
49. Page, L. The Pagerank Citation Ranking: Bringing Order to the Web. In Stanford Digital Library Technologies Project; Technical Report; University of Pennsylvania: Philadelphia, PA, USA, 1998.
50. Montella, C. The Kalman filter and related algorithms: A literature review. *Res. Gate* **2011**, 1–17.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.