

CMMD: Cross-Metric Multi-Dimensional Root Cause Analysis

Shifu Yan^{1*}, Caihua Shan², Wenyi Yang², Bixiong Xu², Dongsheng Li², Lili Qiu², Jie Tong², Qi Zhang²

¹East China University of Science and Technology, ²Microsoft

shifuyan677@hotmail.com, {caihuashan, wenya, bix, dongsheng.li, liliqu, jietong, qizhang}@microsoft.com

ABSTRACT

In large-scale online services, crucial metrics, a.k.a., key performance indicators (KPIs), are monitored periodically to check their running statuses. Generally, KPIs are aggregated along multiple dimensions and derived by complex calculations among fundamental metrics from the raw data. Once abnormal KPI values are observed, root cause analysis (RCA) can be applied to identify the reasons for anomalies, so that we can troubleshoot quickly. Recently, several automatic RCA techniques were proposed to localize the related dimensions (or a combination of dimensions) to explain the anomalies. However, their analyses are limited to the data on the abnormal metric and ignore the data of other metrics which may be also related to the anomalies, leading to imprecise or even incorrect root causes. To this end, we propose a cross-metric multi-dimensional root cause analysis method, named CMMD, which consists of two key components: 1) relationship modeling, which utilizes graph neural network (GNN) to model the unknown complex calculation among metrics and aggregation function among dimensions from historical data; 2) root cause localization, which adopts the genetic algorithm to efficiently and effectively dive into the raw data and localize the abnormal dimension(s) once the KPI anomalies are detected. Experiments on synthetic datasets, public datasets and online production environment demonstrate the superiority of our proposed CMMD method compared with baselines. Currently, CMMD is running as an online service in Microsoft Azure.

KEYWORDS

Root cause analysis, graph neural network, genetic algorithm

ACM Reference Format:

Shifu Yan^{1*}, Caihua Shan², Wenyi Yang², Bixiong Xu², Dongsheng Li², Lili Qiu², Jie Tong², Qi Zhang². 2018. CMMD: Cross-Metric Multi-Dimensional Root Cause Analysis. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 12 pages. <https://doi.org/XXXXXXX.XXXXXXX>

*The work was done during the author's internship with Microsoft.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

© 2018 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00
<https://doi.org/XXXXXXX.XXXXXXX>

1 INTRODUCTION

In the digital era, many companies and organizations monitor the status of their products, services, and business through data intelligence. A general pipeline is shown in Fig. 1: 1) collect the huge volume of data streams; 2) integrate the raw data into KPI metrics related to the performance and reliability issues; 3) monitor KPI metrics periodically and detect the anomalies intelligently; 4) diagnose root causes to prevent anomalies becoming customer-impacting incidents. In this paper, we focus on the last but the essential part, root cause analysis (RCA). Formally, given the multi-dimensional streaming data as input, RCA localizes which dimension(s) of the data as the reason to the anomaly detected in KPIs.

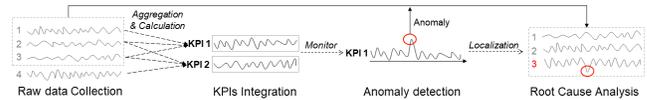


Figure 1: General pipeline of KPI monitoring.

There are two main challenges because the raw data contains many multi-dimensional metrics, while the monitored KPI metric is highly aggregated and often derived. For a certain metric, there could be multiple dimensions, e.g., region and channel. There exist numerous values in each dimension, such as Region={US, UK,...}. Given L dimensions and V values in each dimension, we have V^L dimension value combinations (e.g., Region=US \cap Channel=Search). It is too resource-consuming to monitor each of the combinations independently when L and V are large. Thus, the best practice is to aggregate the value along the dimension. Fig. 2(a) shows a dimension tree, and “AGG” indicates we conduct an aggregation function (e.g., SUM or MEAN) on all values in that dimension. As long as the root node is monitored, we can check the status of all the dimensions. Overall, the first challenge for RCA is how to accurately and quickly find a set of dimension value combinations (leaf nodes) to resolve the anomaly detected in the root node.

Besides, the monitored KPI metric often requires the complex calculation of the raw data. Some applications include the total error rate to check whether a large-scale software is running normally, profitability index to decide the performance of a company product, and overall equipment effectiveness to evaluate how well a manufacturing operation is utilized. Take a business system as a detailed instance: the system monitors the conversion rate of advertisements as its KPI to measure the success of advertisements. The raw streaming data contain the fundamental metrics (e.g., #views, #conversions), and the derived KPI metrics (e.g., conversion rate = $\frac{\#conversions}{\#views}$). It is obvious that the deviations of derived KPI metrics are caused by the changes of fundamental metrics. Thus, the second challenge is how to explain the anomaly occurred in the derived

Table 1: A Snapshot of business metrics at timestamp t . We show the real value (forecast value in parentheses) for each metric in each possible dimension. The monitored metric Conversion Rate is abnormal because of a 27% decrease (colored in red). Previous methods found two root causes colored in yellow. CMMD believed the real root cause is Search|US colored in green.

Dimensions		Fundamental Metrics			Derived Metrics	
Channel	Region	#Views	#Conversions	Cost	Conversion Rate	Cost per Conversion
Search	US	51949(57328)	14651(25741)	219765(249067)	0.28(0.45)	15(17)
Search	Norway	3152(2627)	783(1228)	13311(12528)	0.25(0.47)	17(16)
Search	Brazil	3125(2981)	341(980)	6820(7502)	0.11(0.33)	20(22)
Search	Others	64351(59721)	19321(25931)	618272(579630)	0.30(0.43)	32(30)
Social Media	US	43949(39312)	21525(24057)	344400(322875)	0.49(0.59)	16(15)
Social Media	Norway	20453(18327)	8731(9068)	139696(148427)	0.43(0.50)	16(17)
Social Media	Brazil	1957(1512)	1023(1001)	17391(16368)	0.52(0.66)	17(16)
Social Media	Others	70384(60413)	32253(35912)	903084(838578)	0.46(0.59)	28(26)
AGG	US	95898(96640)	36176(49798)	564165(614992)	0.38(0.50)	16(17)
AGG	Norway	23605(20954)	9514(10296)	153007(161738)	0.40(0.49)	16(17)
AGG	Brazil	5082(4493)	1364(1981)	24211(25916)	0.27(0.44)	18(19)
AGG	Others	134735(120134)	51574(61843)	1521356(1598794)	0.38(0.50)	29(31)
Search	AGG	122577(122657)	35096(53880)	858168(877400)	0.28(0.43)	24(25)
Social Media	AGG	136743(119564)	63532(70038)	1404571(1397704)	0.46(0.59)	22(22)
AGG	AGG	259320(242221)	98628(123918)	2262739(2268444)	0.38(0.52)	22(23)

metric by considering all the metric data, even if we do not know the exact calculation among metrics.

Recently, several works were proposed to automatically identify root causes by regarding each root cause as the combination of values from different dimensions, which cannot address the above challenges. For instance, iDice [9] and HotSpot [14] only works when the monitored metric is fundamental. Adtributor [1] and Squeeze [8] considered the derived metrics. But Adtributor [1] assumes the root cause is only in one dimension. Squeeze [8] supports multiple dimension value combinations, but it needs to know the number of root causes as a prior.

Most importantly, all the existing methods focused on the single-metric instead of cross-metric scenario, i.e., they only consider the data related to the abnormal metric and ignore the data of other metrics which are also related to the anomaly. We illustrate it with the following example. Tab. 1 shows a snapshot of a business system. We sum up all the values in a dimension as a tree based on Fig. 2(a), and calculate two derived metrics Conversion Rate and Cost per Conversion based on Fig. 2(b). To see whether the system is running normally, we monitor two KPIs, Conversion Rate of AGG|AGG and Cost per Conversion of AGG|AGG. At timestamp t , an anomaly is detected in Conversion Rate of AGG|AGG due to a 27% decrease (0.52 \rightarrow 0.38). Traditional methods only focus on the changes of conversion rate (the data in the column of conversion rate), so they regard the root cause as Search|Norway and Search|Brazil since 47% and 67% decline are detected in their conversion rate (0.47 \rightarrow 0.25, 0.33 \rightarrow 0.11). However, from the absolute value of #Views and #Conversions, we find that the decrease of #Conversions of Search|US contributes most to the anomaly. If we recover #Conversions of Search|US (14651 \rightarrow 25741), #Conversions of AGG|AGG becomes 109718, and then Conversion Rate of AGG|AGG is changed from 0.38 to 0.42. If we recover Search|Norway and Search|Brazil, Conversion Rate of AGG|AGG is almost not changed. Search|US has a higher recovery ratio. In summary, this example shows that

it is hard to find the accurate root cause when fundamental metrics are not considered.

To this end, we propose a **cross-metric multi-dimensional** root cause analysis approach, named CMMD, which can identify precise root causes with fewer assumptions. Firstly, we do not assume that we know the exact calculation among metrics and aggregation function among dimensions. We automatically model the relationship among metrics of different dimensions by graph neural network from historical data. Secondly, there is no assumption of the number or range of root causes. We adopt the genetic algorithm and design a novel fitness score based on the trained GNN to compute the probability of a candidate to be a real root cause. The candidate could be any possible dimension value combinations of all the metrics related to the anomaly. Besides, we also utilize the attention weights in the trained GNN to speed up. Thirdly, CMMD could do fine-grained analyses for the raw data. Since we aggregate all the dimensions as a tree, we can monitor the status of any node at any level, not only the root node. Further, after we obtain the root causes searched by the genetic algorithm, we backtrack and aggregate these root causes in the dimension tree to make the result succinct and fine-grained. Finally, CMMD is evaluated by three types of dataset. For online production datasets where anomalies are detected in the derived metric while root causes are in the fundamental metric, we achieve at least 9% improvement compared with baselines. For public datasets where anomalies and root causes are both in the same metric, a competitive performance is also exhibited. The major contributions are summarized as follows:

- To our best knowledge, CMMD is the first RCA approach for the cross-metric scenario and can analyze the root causes for both fundamental and derived metrics.
- We model the unknown relationship among metrics and dimensions based on graph neural networks.
- We utilize the genetic algorithm with designed fitness score to search out the precise root cause.

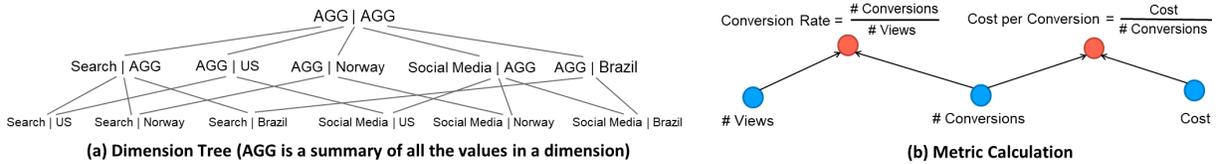


Figure 2: Dimensions and metrics of a business system: an example.

Table 2: Notations and corresponding examples used in this paper.

Notation	Definition	Example
a_{ij}	a value of each dimension	US, Norway for Region; Social Media, Search for Channel
\mathbb{T}	dimension tree	An aggregated tree among dimensions
a node in \mathbb{T}	a dimension value combination	Social Media US, Search Norway, AGG US
root cause	a set of dimension value combinations	{Search US, Social Media AGG}
M^F / M^D	fundamental metric / derived metric	#Views, #Conversions, Cost / Conversion Rate, Cost per Conversion
$v_{i,m} / f_{i,m}$	real/expected values of the metric m in the node i	In Tab. 1, $v = 341$ and $f = 980$ of the node Search Brazil for the metric #Conversions

- We demonstrate the industrial practice in Microsoft Azure.

2 RELATED WORKS

In this section, we introduce existing approaches for root cause analysis and multivariate time-series tasks, and summarize the differences between these approaches and CMMD.

2.1 Root Cause Analysis

The goal of root cause analysis in multi-dimensional data is to localize a combination of dimension values to explain the anomaly that happened in the monitored metric. Adtributor [1] is firstly proposed under the assumption that the root cause lies in only one dimension. Based on the Jensen-Shannon divergence between real and expected values of abnormal metrics, Adtributor selected the dimension value with the highest divergence as the root cause. Adtributor can be applied recursively, named RAdtributor, to find several dimension values to be the root cause. iDice [9] removed the assumption and considered the root cause is a combination of dimension values, i.e., the anomaly occurs with the coaction of many dimensions. Thus, iDice utilized a tree-based method to evaluate the dimension value combinations by Fisher distance between real and expected values of abnormal metrics. At the same time, the pruning strategy is also adopted to filter the most effective dimension combination. ImAPTr [13] used a similar tree-based method to analyze root causes for declining success rate and replaced Fisher distance with Jensen-Shannon divergence. HotSpot [14] defined *ripple effect* to propagate the anomaly into different dimensions and proposed *potential score* to evaluate a set of dimension value combinations by replacing their real values with corresponding expected values and calculating the difference with or without the replacement. HotSpot also adopted the Monte Carlo Tree Search and a hierarchical pruning strategy to improve search efficiency. According to HotSpot, Squeeze [8] proposed *generalized ripple effect* and *generalized potential score* for both fundamental and derived metric. Differently, Squeeze first grouped potential abnormal dimension value combinations into clusters based on generalized ripple effect, and then searched for the root causes based on generalized potential score within those clusters. To improve the clustering performance in Squeeze, AutoRoot [7] used the kernel density estimation to cluster

adaptively and proposed a new method to calculate generalized potential score for insignificant deviation magnitude. However, Squeeze and AutoRoot need to assume the number of root causes in a cluster as a prior. Given the dependency among dimensions, HALO [18] formed a hierarchy structure of dimensions instead of the dimension tree based on conditional entropy and searched root causes by walking in this structure. MID [5] improved the efficiency of iDice by an evolution-based search framework. HALO and MID are specially designed for the cloud system.

iDice and HotSpot work when the monitored metric is fundamental, while Adtributor, Squeeze and AutoRoot support the analysis of derived metrics. However, all the existing methods are focused on the single-metric situation. They only consider the multi-dimensional data of the abnormal metric and ignore the data of other metrics which are also related to the anomalies. CMMD takes all the multi-dimensional data among various metrics into consideration and thus achieves more precise root cause analysis.

2.2 Multivariate Time-series Tasks

Forecasting. Time-series forecasting aims to better understand the data by predicting future values in the time series. Traditional univariate time-series methods, such as auto regression (AR) [6], moving average (MA) [11] and auto regressive integrated moving average (ARIMA) [2] predict the data behavior based on the historical data in a linear way. Considering nonlinearity, neural networks including CNN [10] and LSTM [12] have been successfully applied. Recently, some GNN-based methods [3, 17] were proposed to capture both intra-series and inter-series correlations for multivariate time-series forecasting, which obtained outstanding performance.

Different from multivariate time-series data, CMMD is focused on multi-dimensional streaming data. If there are millions or billions of dimension value combinations, it is impossible to train a multivariate model to learn the relationship in practice. Thus, based on the observation that multi-dimensional data is conducted by the same aggregation function and calculation at each timestamp, we simplify the problem and propose a single-layer GNN model. Note that this observation is universal in many industrial applications.

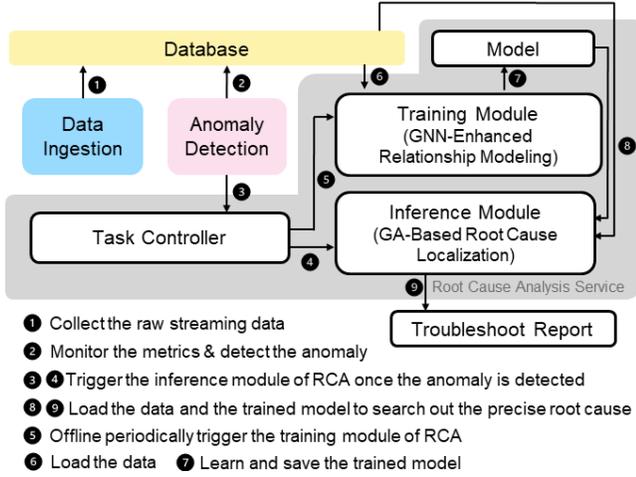


Figure 3: Framework of the proposed method. The main algorithm is described by the components colored in grey.

Anomaly Detection. Many multivariate time-series anomaly detection approaches [4, 12] have been studied recently. However, we regard the module of anomaly detection as a black box in the paper. We consider the situation where the anomaly is already detected.

3 PROBLEM FORMULATION

Here we formulate the problem for cross-metric multi-dimensional root cause analysis. Tab. 2 lists the notions and examples. In the raw streaming data, there are multiple dimensions and each dimension contains numerous values. We denote a_{ij} as the j -th value in the i -th dimension. Besides, the raw data also contain two kinds of metrics, fundamental metrics $M^F = \{m_1^F, m_2^F, \dots, m_p^F\}$ and derived metrics $M^D = \{m_1^D, m_2^D, \dots, m_q^D\}$. At each timestamp, we have the real numerical value of each a_{ij} for every metric m as the input.

Dimension values are often aggregated as a dimension tree \mathbb{T} , as shown in Fig. 2. AGG represents a summary of all the values in a dimension. Each node in \mathbb{T} is a combination with specific values in the dimension, i.e., a vector $[a_{11}, a_{23}, \dots, a_{ij}, \dots]$. For non-leaf nodes, a_{ij} in some positions could be AGG, which represents that the i -th dimension is already aggregated. For the root node, all the positions in the vector are AGG that means all the dimensions are aggregated.

After that, the most representative status, the derived KPI metric of the root node is monitored periodically. When the anomaly is detected by Anomaly Detection module, the main objective of Root Cause Analysis is to analyze which set of dimension value combinations is the root cause of the anomaly. In other words, we need to identify a set of nodes in \mathbb{T} as the root cause, such as $\{[a_{11}, a_{21}, \dots], [a_{14}, a_{27}, \dots], \dots\}$.

4 METHODOLOGY

In the section, we first present the framework of CMMD, and then introduce the two important modules in CMMD, training and inference modules. Finally, we describe two extra functions for fine-grained analysis and speed-up.

4.1 Framework

The overall framework of CMMD is presented in Fig. 3. In the stream processing, we collect and store the raw data in the database. Anomaly detection is used to monitor KPI metrics integrated from the streaming data. Once it detects the anomaly, it sends the signal to the root cause analysis service. RCA is batch-processing where tasks are managed in the task controller. There are two kinds of tasks: 1) we offline periodically trigger the training module to train/update the model for the relationship between the monitored metric and the fundamental metrics obtained from the raw streaming data; 2) once there exists the anomaly signal, we trigger the inference module to search out the precise root cause and output a troubleshooting report.

In detail, we construct a dimension tree by aggregating dimensions in the offline training module, and a graph attention network [15] is used to learn the tree-based relationship. In the online inference module, a bottom-up search strategy is adopted based on the genetic algorithm [16]. We design a fitness function based on the graph attention model to evaluate the probability of the candidate dimension value combination to be a root cause.

4.2 GNN-Enhanced Relationship Modeling

We often monitor the root node in the dimension tree for derived metrics. However, we need to localize the root cause in the raw streaming data, i.e., the leaf nodes for fundamental metrics. Thus, the first step is to model the complex relationship. In this section, we introduce how to utilize the graph attention network (GAT) to learn the relationship from historical data automatically.

Although we do not know the detailed calculation used to get the value of derived metrics from fundamental metrics and the exact aggregation function to aggregate the value along the dimension, we find the calculation and aggregation are the same in any two-layer subtree. As shown in Fig. 4(a), we extract several two-layer subtrees from the dimension tree in Fig. 2. The data processing in each subtree is the same: at each timestamp, the raw data are collected in child nodes for fundamental metrics. Then these data are aggregated along the dimensions to generate the value of the parent node. Meanwhile, the value of fundamental metrics is calculated to obtain the value of derived metrics in each node. Since the number of child nodes is uncertain, graph neural network is suitable to learn the pattern in the subtree.

Formally, we define $v_{t,i,m}$ as the values of node i for each metric m at timestamp t . Suppose that we have L dimensions, P fundamental metrics and Q derived metrics. For each node i , the value of fundamental metrics is the input $x_i = [v_{t,i,m_1^F}, v_{t,i,m_2^F}, \dots, v_{t,i,m_P^F}]$, and the value of derived metrics is the output $y_i = [v_{t,i,m_1^D}, v_{t,i,m_2^D}, \dots, v_{t,i,m_Q^D}]$. Since the relationship is the same at each timestamp, we omit the subscript t . For each subtree, we utilize the **single-layer graph attention** \mathcal{G} to model the relationship between the child nodes and the parent node: $y_i = \mathcal{G}(\{x_j | \text{node } i \text{ is the parent of node } j\})$. If we conduct L graph convolutions, we iteratively learn the relationship between the leaf nodes and the root node in the dimension tree, i.e., $y_{\text{root}} = \mathcal{G}^L(\{x_j | \text{node } j \text{ is the leaf node}\})$.

Single-layer graph attention. According to the tree structure, we specify the neighbors \mathcal{N} as the child nodes of each parent node. As shown in Fig. 4(b), the representation z_i of node i after message

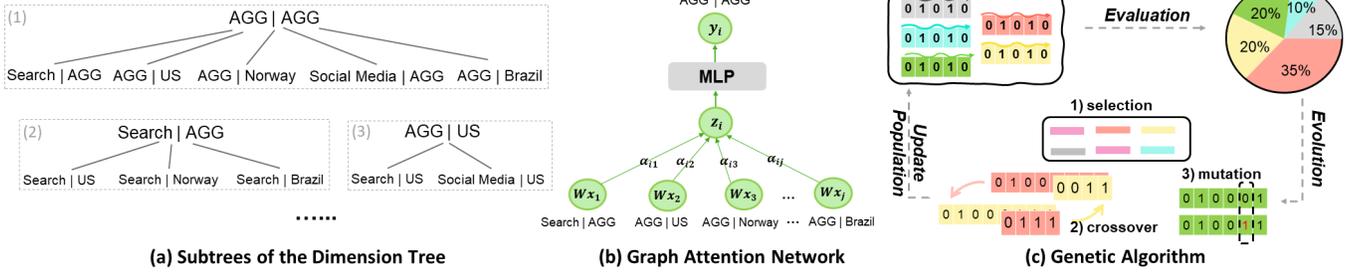


Figure 4: The key components of CMMD.

passing is calculated as follows:

$$z_i = \sigma \left(\sum_{j \in N_i} \alpha_{ij} Wx_j \right), \quad (1)$$

where W and σ are the parameters and activation function in GAT. Please note that we add the depth of node j into the input feature x_j to enrich the information. The attention coefficients α_{ij} between node i and node j can be computed as follows:

$$\alpha_{ij} = \frac{\exp(\text{LeakyReLU}(\theta^T [Wx_i \| Wx_j]))}{\sum_{k \in N_i} \exp(\text{LeakyReLU}(\theta^T [Wx_i \| Wx_k]))}, \quad (2)$$

where θ is the learnable parameter of a single-layer feedforward neural network indicating the attention mechanism and $\|$ means concatenation of vectors. We also implement the multi-head attention to learn more useful information:

$$z_i = \parallel_{k=1}^K \sigma \left(\sum_{j \in N_i} \alpha_{ij}^k W^k x_j \right). \quad (3)$$

Objective function. After aggregating information from child nodes and obtain the representations z_i , we finally apply a multi-layer perceptron to fit the function between the representations and the real value of derived metrics $\hat{y} \in \mathbb{R}^q$. Here we use the Mean Squared Error (MSE) to guide the GAT training:

$$L_{MSE} = \sum_{t=1..T} \sum_{i=1..N'} \|\hat{y}_{t,i} - MLP(z_{t,i})\|_2^2, \quad (4)$$

where T is the total number of timestamps and N' is the number of non-leaf nodes in the dimension tree.

4.3 GA-Based Root Cause Localization

Based on the above model, we are able to get the effect of fundamental metrics in the leaf nodes on a specific monitored metric. Therefore, we can analyze the causes for the abnormal metrics once detected. However, it is time-consuming because of a large scale of dimension value combinations in practice. In this way, tree-based search methods are popular in root cause analysis. If we utilize the top-down search strategy, there may exist many assumptions and pruning thresholds, and some root causes will be missing. Thus, we introduce a heuristic search framework based on the genetic algorithm for this combinatorial optimization problem and the bottom-up search strategy is applied with few assumptions about the root cause.

Chromosome	1	0	1	0	0	1
Dimension value combination	Search US	Search Norway	Search Brazil	Social Media US	Social Media Norway	Social Media Brazil

Figure 5: Chromosome Encoding.

Anomaly detection in a metric is based on the deviation between the real value v and the predicted value f . Such prediction can be implemented by most of the time-series algorithms (e.g., AR, MA, ARIMA). Considering the efficiency and accuracy in root cause analysis, AR is adopted to obtain the forecast value of each node in this work. When an anomaly is detected, we conduct the GA-based search which is shown in Fig. 4(c).

Encoding. First of all, we encode root cause candidates, i.e., all the leaf nodes in the dimension tree, into a vector (chromosome). As shown in Fig. 5, the chromosome is a binary bit string, which can be represented as:

$$s = [s_1, \dots, s_i, \dots, s_n] \quad s_i \in \{0, 1\}, 1 \leq i \leq n. \quad (5)$$

Each bit s_i indicates the status of a dimension value combination, such as Search|US. $s_i = 1$ means that this dimension value combination is chosen as the root cause and $s_i = 0$ means that it is excluded. Obviously, the number of root causes can be more than one, which means that the abnormality is due to the coaction of multiple abnormal dimension value combinations. Note that we do not assume the number of root causes, and it is decided in the search process automatically.

Evaluation. Based on the encoding mechanism, we randomly sample multiple chromosomes as an initial population. The next step is to compute the probability of each chromosome to be an accurate root cause set. We design the fitness score, whose motivation is to evaluate whether the abnormality still exists if we 'troubleshoot' the root cause set indicated by the chromosome.

We denote the deviation as $\Delta_{root,m} = |v_{root,m} - f_{root,m}|$ where $v_{root,m}$ and $f_{root,m}$ are the real value and forecast value of the root node for the abnormal metric m . Since the chromosome s indicates abnormal dimension value combinations, we replace the values of abnormal dimension value combinations with the predicted values, maintain the values of normal dimension value combination, and obtain the new input $X(s)$. Next, we could obtain the new output value $\mathcal{G}^L(X(s))$ based on the graph model \mathcal{G}^L learned in Sec. 4.2. To evaluate whether eliminating the abnormal dimension value combinations indicated by s is enough to troubleshoot the target

metric, we define the fitness score as:

$$\text{Fitness score} = \frac{|\mathcal{G}^L(X(s)) - f_{\text{root},m}|}{|v_{\text{root},m} - f_{\text{root},m}|} + \beta \|s\|_1. \quad (6)$$

Obviously, chromosome with smaller score is considered to be the root cause set with higher probability. Besides, we need to consider the succinctness of the root causes to avoid selecting all of them. β is a trade-off parameter between the deviation and succinctness.

Evolution. The core of the genetic algorithm is the iterative evaluation and evolution which is made up of three behaviors, selection, crossover and mutation.

Given a population with the fitness score of each chromosome, the **selection** operator aims to choose chromosomes for the next iteration based on the roulette and a chromosome with a smaller fitness score is more likely to be selected obviously. After selection, we perform **crossover** and **mutation** to enrich chromosome diversity. The crossover operation is controlled by the cross rate t_c . For a pair of chromosomes, if the probability we randomly generated exceeds t_c , we swap part of these two chromosomes at a random bit. t_c determines the frequency of new chromosomes. A small t_c leads to the slow generation of new chromosomes and affects the diversity. Meanwhile, a large t_c results in the unstable heredity of chromosomes and makes GA similar to random algorithm. Therefore, we set $t_c = 0.5$. The mutation operation is controlled by another mutation rate t_f . Similarly for each chromosome, if the probability exceeds t_f , then we randomly pick a bit in this chromosome and reverse 0 and 1. Since mutation is a small probability event and the influence of mutation should be much less than crossover, t_f is usually recommended to be less than 0.2 and we set $t_f = 0.1$. We summarized the whole search process in Alg. 1 in Appendix.

4.4 Extra Functions

4.4.1 Backtrack for Fine-grained Analysis. In the previous sections, we find out the root causes based on the optimal chromosome s^* with the smallest fitness score. Furthermore, we could obtain the fine-grained result by aggregating these chosen leaf nodes into their parent nodes. For example, if the leaf nodes Social Media|US and Search|US are both chosen as the root cause, the result can be compacted as their parent node, AGG|US. To achieve this goal, we backtrack in the dimension tree.

We summarized the whole process in Alg. 2 in Appendix. Assume that we have a set of chosen leaf nodes \mathcal{S} and a backtrack threshold t_γ as input. Then, we conduct a bottom-up aggregation layer by layer. For a parent node, if the percentage of its child nodes in \mathcal{S} is more than t_γ , we remove these child nodes from \mathcal{S} and add the parent node into \mathcal{S} . t_γ is a user-defined which determines the succinctness of results and large t_γ will make root causes include more leaf nodes. $t_\gamma \geq 0.5$ is usually preferred and we set $t_\gamma = 0.6$ in the experiments. We stop when there is no parent node added in a layer. Finally, the new root cause set \mathcal{S}^* is succinct.

4.4.2 Filtering for Speed-up. The efficiency of root causes analysis depends largely on the number of dimension value combinations, i.e., the chromosome length. Given the fact that not all dimension value combinations are abnormal when the anomaly is detected, we filter out the normal dimension value combinations to obtain the candidates for the genetic algorithm. Considering the deviation

between real value $v_{i,m}$ and expected value $f_{i,m}$ of the leaf node i of the abnormal metric m , and the importance of leaf node i to the root node, we design the following score to measure whether it can be a candidate:

$$\text{Filtering score}_i = \frac{|v_{i,m} - f_{i,m}|}{|v_{i,m}|} * \frac{\alpha_{i,\text{root}}}{\sum_j \alpha_{j,\text{root}}}. \quad (7)$$

The importance to the root node $\alpha_{i,\text{root}}$ is computed iteratively by the average k -head attention score $\frac{1}{K} \sum_k \alpha_{ij}^k$ in the graph model. If filtering score _{i} is smaller than the filtering threshold t_δ , we filter the node i .

5 EXPERIMENT

In the section, we conduct extensive experiments to verify both the effectiveness and efficiency of the proposed CMMD on multiple datasets. Due to the space limitation, we move the implementation detail (including hyper-parameter setting of CMMD and baselines) and parameter study to Appendix.

5.1 Experimental setup

Datasets. To comprehensively assess the effectiveness of our method, we conduct experiments on three types of situations, production datasets (obtained from online services at Microsoft), public datasets (released by the previous work) and synthetic datasets.

Production datasets \mathcal{D}_1 . Here we have 5 production cases. The first 3 cases came from a business analysis scenario. Conversion rate, cost per conversion, and cost per view are three important metrics to track the marketing effort and analyze the results. They are generated by three fundamental metrics, the number of conversions, the number of views and cost. The calculations are conversion rate = $\frac{\#conversions}{\#views}$, cost per conversion = $\frac{\text{cost}}{\#conversions}$, and cost per view = $\frac{\text{cost}}{\#views}$. We collected 3-month data with 91 timestamps, consisting of two dimensions (Region and Channel). There are 249 countries or regions in Region and 3 values in Channel. Except the undefined combinations, there are 503 leaf dimension value combinations, and 270 aggregated dimension value combinations in the tree.

The remaining 2 cases are from an alerting service. We gathered 1-year metrics with 300 timestamps. There is one dimension notification channel which has 5 different values. Alert open rate and click-through rate are monitored to bring to light the quality and the engagement of the alerts, which is calculated based on the number of alerts, the number of opened alerts and the number of click-through alerts. The calculations are alert open rate = $\frac{\#opened\ alerts}{\#alerts}$, and click-through rate = $\frac{\#click-through\ alerts}{\#alerts}$.

Public datasets \mathcal{D}_2 . The public dataset is released by Squeeze [8]. \mathcal{D}_2 is a semi-synthetic dataset constructed based on the real-world dataset from an Internet Company. Here we select its derived sub-dataset as \mathcal{D}_2 . The derived metric is the success rate in transactions which is calculated by successful transaction number and total transaction number. There are 100 timestamps in each case with 4 dimensions and 21600 leaf dimension value combinations. In this way, anomalies are injected as (p,q) where p, q mean the number of abnormal dimensions and the number of root causes respectively. We set $p = 1,2,3$ and $q = 1,2,3$ to test 9 cases respectively.

Synthetic datasets \mathcal{D}_3 . Apart from the production and public datasets, we also conduct experiments on a synthetic dataset \mathcal{D}_3 . We

have two fundamental metrics b and c , and two derived metrics $a = g(c)$ and $d = f(a, b)$. For evaluation on more complex relationships among metrics, we set function f as

1) $\frac{a}{b}$, 2) $a \times b$, 3) $\frac{\log a}{\log b}$, 4) $a \times e^b$ and 5) $\frac{\log(a+1)}{\log(b+1)}$. The function g is randomly selected from $\{c, \sin(c), e^c, c^2, \sqrt{c}\}$. In each case, 200 timestamps are simulated. The detailed simulation process is described in Appendix.

Evaluation metrics. In the following experiments, F1-score is used to evaluate the performance in cross-metric multiple dimensional root cause analysis. Since current methods only focus on the dimensions, we only compare the result whether the root causes are the accurate set of dimension value combinations. The calculation of F1-score is mathematically described as follows.

$$F1 \text{ score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (8)$$

where Precision (P) = $\frac{TP}{TP+FP}$ and Recall (R) = $\frac{TP}{TP+FN}$. TP (true positive) is the number of dimension value combinations both reported by the algorithm and the ground truth. FP (false positive) is the number of dimension value combinations reported by algorithm but not in the ground truth. FN (false negative) means the number of root causes in ground truth but not reported by the algorithm.

The ground truth of \mathcal{D}_2 is provided in [8]. In other two datasets, since we know the exact calculation among metrics and aggregation function among dimensions, the ground truth is determined by the recovery ratio of specific dimension value combinations. Please see the detailed description in Appendix.

Comparative methods. We compare the proposed CMMD with four baselines, Adtributor [1], HotSpot [14], Squeeze [8] and AutoRoot [7]. The detailed information for baselines is introduced in Appendix.

5.2 Overall Performance

Firstly, we conduct experiments on the production datasets \mathcal{D}_1 and present the results of the compared methods in Tab. 3. In the results, CMMD outperforms other methods in these five cases and achieves nearly 9% higher than the second-best performance on average F1 score. In these methods, Squeeze obtains the highest Recall, especially in the case of cost per conversion. HotSpot, Squeeze and AutoRoot exhibit similar F1 scores with competitive Recall in industrial practice, but the Precision of the detected root causes needs to be further improved especially for Squeeze. These methods do not consider the cross metrics which will result in multiple false positives since many abnormal dimension value combination does not contribute to the monitored metrics. Although Adtributor provides a better Precision than HotSpot, Squeeze and AutoRoot, the Recall of root causes cannot meet the requirements of practical application which means some root causes are undetected by Adtributor. Overall, the proposed CMMD provides the best performance and demonstrates that fundamental metrics can improve the accuracy of root cause analysis when monitoring the derived metrics.

Next, we evaluate these five methods on the public datasets \mathcal{D}_2 . The results in Tab. 4 suggest that CMMD can still exhibit better F1 scores on average. It is worth noting that the anomalies are injected directly into the derived metrics rather than the fundamental metrics and this mechanism leads to a single-metric scenario.

Therefore, HotSpot, Squeeze and AutoRoot for the single-metric task can also exhibit competitive performance. Similarly, Adtributor is not effective in Recall. With the increasing number of dimensions in a root cause p and the increasing number of root causes q , the performance of all methods gradually deteriorates. The clustering algorithms in Squeeze and AutoRoot can help determine the number of root causes and thus they have better results compared with Adtributor and HotSpot. Although CMMD is not good as AutoRoot in some cases, it still obtains a competitive performance.

Finally, we provide the Precision, Recall and F1 score in Fig. 6(a-c) for the five synthetic datasets. Overall, CMMD can still achieve better F1 scores and is flexible for different calculations between fundamental and derived metrics. In these datasets, Adtributor exhibits a better Precision but a bad Recall since it tends to consider fewer root causes. The performance of Squeeze and AutoRoot is more balanced in Precision and Recall. HotSpot presents a closer performance to our method with higher Precision in a few cases. The good performance on synthetic datasets demonstrates the generalizability of our proposed CMMD.

Besides, the efficiency of root cause analysis is also important for industrial practice. We also conduct experiments on synthetic datasets with an increasing number of dimensions and set each dimension consisting of ten dimension values. Therefore, we evaluate these methods with up to 10^5 leaf dimension value combinations. The result is shown in Fig. 6(d). For the sake of analysis, the time in the figure is logarithmic. Obviously, all the methods will cost more inference time and exhibit the exponential trends to analyze the root causes when more dimensions are collected. In these methods, Adtributor is the most time-consuming method in all the situations which is followed by HotSpot. These two methods cost more than 500 seconds for 10^5 leaf dimension value combinations. Although Squeeze and AutoRoot may be quick in less dimension value combinations, our CMMD can also provide competitive performance and achieve the best efficiency in the largest case. This is mainly due to the effect of the genetic algorithm on such combinatorial explosion problems to save the tedious traversal time in the tree-based search.

6 INDUSTRIAL PRACTICE

CMMD has been released as a private preview feature on an online service in Microsoft Azure which provides users to detect the anomaly and analyze root causes. Currently, it has been used by several customers from different backgrounds, including business marketing, AIOps(artificial intelligence for IT operations), and networking. In this section, we will analyze two real cases and share some lessons we learned from practice.

6.1 Case 1: Business marketing

Conversion rate is one of the most commonly used and valuable KPIs to understand and effectively analyze the results of the marketing campaigns, which is derived by $\frac{\#Conversions}{\#Views}$. In this case, we monitor the Conversion Rate of a business system. There are two dimensions (Region and Channel) aggregated.

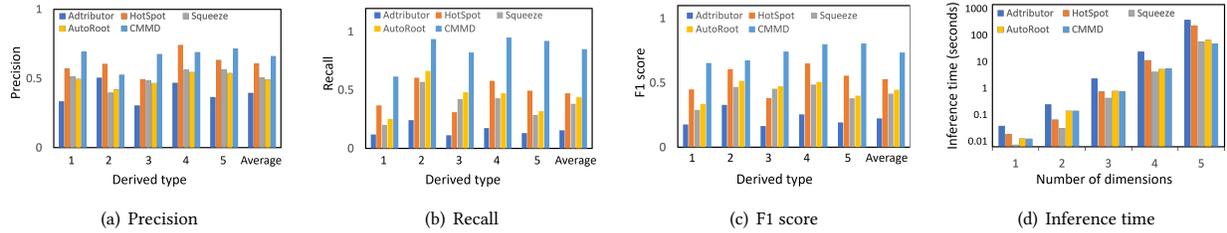
Fig. 7 shows the interface of our service based on CMMD. At the top of Fig. 7, Conversion Rate of AGG|AGG was monitored with a 26.92% decrease on January 10, 2020. The module of root cause

Table 3: Experimental results on production datasets.

Datasets	Adtributor			HotSpot			Squeeze			AutoRoot			CMMD		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Conversion rate	0.7679	0.5939	0.6698	0.7036	0.8591	0.7736	0.6136	0.7459	0.6733	0.6005	0.7182	0.6541	0.7339	0.9144	0.8143
Cost per conversion	0.7202	0.6205	0.6667	0.6572	0.7733	0.7105	0.6844	0.8902	0.7739	0.6693	0.8019	0.7296	0.8054	0.8496	0.8269
Cost per view	1.0000	0.5000	0.6667	1.0000	0.6154	0.7619	1.0000	0.8077	0.8936	0.9524	0.7693	0.8511	1.0000	0.8462	0.9167
Alert open rate	0.4400	0.3200	0.3721	0.4800	0.9600	0.6400	0.4386	1.0000	0.6098	0.4310	1.0000	0.6024	0.8333	0.8000	0.8164
Click-through rate	1.0000	0.3143	0.4783	1.0000	0.7714	0.8525	1.0000	0.8571	0.9231	1.0000	0.8286	0.9063	1.0000	0.8857	0.9394
Average	0.7856	0.4697	0.5707	0.7682	0.7958	0.7477	0.7473	0.8602	0.7747	0.7306	0.8236	0.7487	0.8745	0.8592	0.8627

Table 4: Experimental results on public real-world datasets.

Root cause type	Adtributor			HotSpot			Squeeze			AutoRoot			CMMD		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
(1,1)	0.8000	0.9500	0.8686	0.7882	1.0000	0.8815	0.9200	0.9200	0.9200	0.9000	0.9000	0.9000	0.9000	0.9500	0.9243
(1,2)	0.9855	0.7083	0.8240	0.5394	0.7500	0.6275	0.9540	0.8300	0.8877	0.9253	0.8050	0.8610	0.9242	0.9150	0.9196
(1,3)	0.9171	0.6700	0.7743	0.3322	0.6575	0.4414	0.9336	0.7226	0.8146	0.9115	0.7055	0.7954	0.8793	0.8500	0.8644
(2,1)	0.4826	0.8384	0.6125	0.6899	0.7200	0.7042	0.8700	0.8200	0.8443	0.8700	0.8700	0.8700	0.8519	0.9200	0.8846
(2,2)	0.5472	0.7323	0.6264	0.4462	0.5200	0.4803	0.8955	0.9000	0.8977	0.8955	0.9000	0.8977	0.8974	0.8750	0.8861
(2,3)	0.6735	0.5556	0.6089	0.4181	0.4000	0.4088	0.8930	0.8930	0.8930	0.8930	0.8930	0.8930	0.8970	0.8700	0.8833
(3,1)	0.2908	0.8200	0.4294	0.4502	0.5300	0.4869	0.8700	0.8900	0.8799	0.8911	0.9000	0.8955	0.8762	0.8500	0.8629
(3,2)	0.4255	0.6061	0.5000	0.3500	0.4100	0.3776	0.9250	0.8900	0.9059	0.9200	0.9200	0.9200	0.9333	0.9200	0.9266
(3,3)	0.4580	0.4511	0.4545	0.3550	0.3533	0.3541	0.8896	0.8867	0.8881	0.8896	0.8867	0.8881	0.8616	0.8800	0.8707
Average	0.6200	0.7035	0.6332	0.4855	0.5934	0.5291	0.9056	0.8614	0.8812	0.8996	0.8645	0.8801	0.8912	0.8922	0.8914

**Figure 6: Experimental results on synthetic datasets.**

analysis is triggered to find out the abnormal dimensions. Traditional methods solved this task only based on Conversion Rate data. Squeeze found two root causes: Search|Brazil with a 66.67% decrease and Search|Norway with a 46.81% decline both in Conversion Rate. HotSpot and Adtributor also regarded Search|Brazil as the root cause. AutoRoot searched out an aggregated result Search|AGG with a 34.88% drop. However, Search|Brazil or Search|AGG are not the real root cause because: 1) Although Search|Brazil changed much in Conversion Rate, its changes in #Conversions have less impact (2.53%) on #Conversions of AGG|AGG. Thus, it also has a lower proportion to influence Conversion Rate of AGG|AGG; 2) 220 leaf dimension value combinations are involved in Search|AGG and only 3.2% of them were abnormal at that timestamp.

Different from previous works, CMMD believed that Search|US is the real root cause. The 43.08% decrease in #Conversions of Search|US caused a 20.41% drop in #Conversions of AGG|AGG, and then led to a 26.92% decrease in Conversion Rate of AGG|AGG. Therefore, we show that Search|US is the root cause in the bottom left of Fig. 7. Besides, we also show the trends of corresponding metrics in the bottom right. With CMMD, our service can point

customers to the real root cause in one step. It was also confirmed by the data owners that the drop was caused by a conversion issue from one of the Search channels.

6.2 Case 2: AIOps

AIOps (Artificial Intelligence for IT Operations) is to enhance IT operations and guarantee the performance and reliability of services via AI techniques. CMMD also works well to monitor the status of services and help to restore the service performance when the anomaly occurs. In this case, we applied CMMD into an alerting service. We monitor the metric, Alert Open Rate, which is calculated by $\frac{\#OpenAlerts}{\#Alerts}$. There is one dimension notification channel = {Email, Azure DevOps, Microsoft Teams,...}.

A typical case is root cause analysis for abnormal Alert Open Rate of AGG with a 35.7% drop (0.28 \rightarrow 0.18). Other methods found DevOps as the root cause since DevOps has the largest deviation of Alert Open Rate. Although the change of Alert Open Rate of Email is only a half of DevOps, CMMD identified Email as a real abnormal channel because its deviation of #Alerts contributes over 67.2% to the anomaly compared with 2.3% in DevOps. It also has

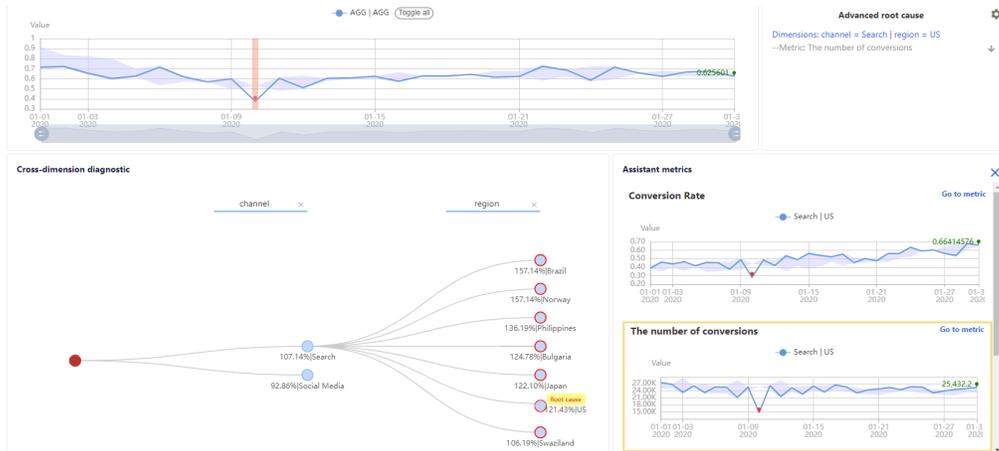


Figure 7: Interface of our root cause service. (x% is equal to $\frac{\text{the value of node} - \text{the expected value of node}}{\text{the value of root node} - \text{the expected value of root node}}$. Previous methods found Brazil and Norway with their higher x%. CMMD believes US is the root cause by the absolute value of #Conversions.)

been confirmed by our customers. At that timestamp, they sent many duplicate alerts in Email channel due to an updated version, which leads to the drop of Alert Open Rate.

6.3 Lessons learned

Although a scenario can contain a large volume of leaf dimension value combinations, not all the combinations have values at every timestamp. In other words, sparsity usually exists in the real industry. Therefore, it is unnecessary to monitor every leaf dimension value combination. The best practice is to aggregate the dimensions, monitor the aggregated metrics and analyze the root causes once the anomaly is detected.

In general, most of the KPIs that customers care about are not fundamental metrics, especially in the large-scale scenario. It is more important to monitor the derived metrics since these metrics are often related to profit or production safety. As can be seen in our previous analysis, considering only derived metrics is not enough for root cause analysis and the cross-metric algorithm can help find more accurate and fine-grained root causes.

7 CONCLUSION

In this paper, we proposed CMMD for cross-metric multiple dimensional root causes analysis. To our best knowledge, it is the first time to study the cross-metric problems which can accurately explain the anomalies occurred in derived metrics and find the root causes. Firstly, CMMD models the relationship between fundamental and derived metrics based on graph attention network. Based on the well-trained GAT, we adopt the genetic algorithm and design a novel fitness function to evaluate the probability of a set of dimension value combinations to be the root cause. We also use the attention mechanism in GAT for speed-up. Experiments suggest that CMMD can obtain better performance over other methods on average F1 scores and further explain the changes of monitored metrics. Through the case studies in production datasets, CMMD can locate the root causes more accurately and meet the requirements of industrial practice.

REFERENCES

- [1] Ranjita Bhagwan, Rahul Kumar, Ramachandran Ramjee, George Varghese, Suryakanta Mohapatra, Hemanth Manoharan, and Piyush Shah. 2014. Adtributor: Revenue debugging in advertising systems. In *11th {USENIX} Symposium on Networked Systems Design and Implementation (NSDI 14)*. 43–55.
- [2] George EP Box and David A Pierce. 1970. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models. *Journal of the American statistical Association* 65, 332 (1970), 1509–1526.
- [3] Defu Cao, Yujing Wang, Juanyong Duan, Ce Zhang, Xia Zhu, Conguri Huang, Yunhai Tong, Bixiong Xu, Jing Bai, Jie Tong, et al. 2020. Spectral temporal graph neural network for multivariate time-series forecasting. *NeurIPS* (2020).
- [4] Ailin Deng and Bryan Hooi. 2021. Graph neural network-based anomaly detection in multivariate time series. In *AAAI*, Vol. 35. 4027–4035.
- [5] Jiazhen Gu, Chuan Luo, Si Qin, Bo Qiao, Qingwei Lin, Hongyu Zhang, Ze Li, Yingnong Dang, Shaowei Cai, Wei Wu, et al. 2020. Efficient incident identification from multi-dimensional issue reports via meta-heuristic search. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 292–303.
- [6] Nikou Gunnemann, Stephan Gunnemann, and Christos Faloutsos. 2014. Robust multivariate autoregression for anomaly detection in dynamic product ratings. In *Proceedings of the 23rd international conference on World wide web*. 361–372.
- [7] Pengkun Jing, Yanni Han, Jiyan Sun, Tao Lin, and Yanjie Hu. 2021. AutoRoot: A Novel Fault Localization Schema of Multi-dimensional Root Causes. In *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 1–7.
- [8] Zeyan Li, Chengyang Luo, Yiwei Zhao, Yongqian Sun, Kaixin Sui, Xiping Wang, Dapeng Liu, Xing Jin, Qi Wang, and Dan Pei. 2019. Generic and robust localization of multi-dimensional root causes. In *2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 47–57.
- [9] Qingwei Lin, Jian-Guang Lou, Hongyu Zhang, and Dongmei Zhang. 2016. iDice: problem identification for emerging issues. In *Proceedings of the 38th International Conference on Software Engineering*. 214–224.
- [10] Mohsin Munir, Shoaib Ahmed Siddiqui, Andreas Dengel, and Sheraz Ahmed. 2018. DeepAnT: A deep learning approach for unsupervised anomaly detection in time series. *Ieee Access* 7 (2018), 1991–2005.
- [11] Masafumi Nakano, Akihiko Takahashi, and Soichiro Takahashi. 2017. Generalized exponential moving average (EMA) model with particle filtering and anomaly detection. *Expert Systems with Applications* 73 (2017), 187–200.
- [12] Daehyung Park, Yuuna Hoshi, and Charles C Kemp. 2018. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters* 3, 3 (2018), 1544–1551.
- [13] Guoping Rong, Hao Wang, Yong You, He Zhang, Jialin Sun, Dong Shao, and Yangchen Xu. 2020. Locating the Clues of Declining Success Rate of Service Calls. In *2020 IEEE 31st International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 335–345.
- [14] Yongqian Sun, Youjian Zhao, Ya Su, Dapeng Liu, Xiaohui Nie, Yuan Meng, Shiwen Cheng, Dan Pei, Shenglin Zhang, Xianping Qu, et al. 2018. Hotspot: Anomaly localization for additive kpis with multi-dimensional attributes. *IEEE Access* 6 (2018), 10909–10923.

- [15] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. *ICLR (2018)*.
- [16] Darrell Whitley. 1994. A genetic algorithm tutorial. *Statistics and computing* 4, 2 (1994), 65–85.
- [17] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. 2020. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *SIGKDD*. 753–763.
- [18] Xu Zhang, Chao Du, Yifan Li, Yong Xu, Hongyu Zhang, Si Qin, Ze Li, Qingwei Lin, Yingnong Dang, Andrew Zhou, et al. 2021. HALO: Hierarchy-aware Fault Localization for Cloud Systems. In *SIGKDD*.

A SUPPLEMENTARY MATERIALS

A.1 Pseudocode

Algorithm 1 GA-based root cause localization

Input: Candidate dimension combinations, Graph attention model \mathcal{G} , population size N_p , iterations N , crossover rate t_c , mutation rate t_f

Output: root cause set

- 1: Initialize a population S with N_p binary chromosomes;
 - 2: **for** $i = 1 \rightarrow N$ **do**
 - 3: **for** $j = 1 \rightarrow N_p$ **do**
 - 4: Compute the fitness score for each s_j in S by Eq. 6
 - 5: **end for**
 - 6: Record the optimal s^* with the smallest fitness score;
 - 7: Select new population S' based on roulette where the probability is consistent with fitness score;
 - 8: Cross random part of each two chromosomes in S' with probability t_c ;
 - 9: Mutate each chromosome in S' with probability t_f ;
 - 10: Replace the population S with S'
 - 11: **end for**
 - 12: Generate the root cause set \mathcal{S} based on s^*
-

Algorithm 2 Root cause backtrack

Input: Root cause set \mathcal{S} , number of layers L in dimension tree \mathbb{T} , backtrack threshold t_γ

Output: A succinct set of root causes \mathcal{S}^*

- 1: $\mathcal{S}^* = \mathcal{S}$
 - 2: **for** $l = L - 1$ to 1 **do**
 - 3: Search the parent node set PS of \mathcal{S}^* in layer l of \mathbb{T} ;
 - 4: **for** $p \in PS$ **do**
 - 5: Search leaf nodes belonging to p as a ;
 - 6: Search the leaf nodes belonging to p and also in \mathcal{S}^* as b ;
 - 7: **if** $\frac{\text{Count}(b)}{\text{Count}(a)} \geq t_\gamma$ **then**
 - 8: $\mathcal{S}^*.add(p)$
 - 9: $\mathcal{S}^*.remove(b)$
 - 10: **end if**
 - 11: **end for**
 - 12: **if** None of PS is added to \mathcal{S}^* **then**
 - 13: Break
 - 14: **end if**
 - 15: **end for**
 - 16: **return** \mathcal{S}^*
-

A.2 Implementation details

All the experiments are implemented by Python 3.8.8 in a Laptop with Intel(R) Core(TM) i7-8665U CPU @2.11GHz and 16G RAM.

Adtribtor. Adtribtor locates the root causes by dimension. Values in each dimension are sorted by Jansen-Shannon divergence and the stop searching criteria within dimension values and dimensions are both controlled by thresholds of cumulative explanatory power. The threshold for explanatory power to stop searching in a dimension

Table 5: Hyper-parameters in CMMD

Component	Hyper-parameter	Value
GAT	Number of epochs	1000
	Embedding dimension	8
	Number of attention heads	8
	Learning rate	5e-4
	Optimizer	Adam
	Early stop patience	50
GA	Cross rate t_c	0.5
	Mutation rate t_f	0.1
	β in Eq. 6	1.0
Others	Backtrack threshold t_γ	0.6
	Filtering threshold t_δ	Searched from 0.1, 0.2

is specified as $T_{EEP} = 0.3$ and threshold for cumulative explanatory power of dimension values is specified as $T_{EP} = 0.8$.

HotSpot. HotSpot proposes the **potential score** to evaluate the dimension value combination and is used to sort the candidate dimension value combinations. In HotSpot, Monte Carlo Tree search is utilized to locate the root causes in an efficient way. Refer to the settings in [14], the maximum iteration M and stop condition PT which control the **potential score** of dimension value combinations are specified as $M = 25$ and $PT = 0.99$.

Squeeze. Squeeze analyzes the root cause in two procedures. Firstly, deviations between real and expected values of metrics in dimension value combinations are filtered and clustered into several clusters. The threshold in Squeeze to filter abnormal dimension values is automatically determined by *knee-point* method and the *bins* in clustering is set as 30. In the search process, $\theta = 0.9$ is set to control **generalized potential score** of candidate dimension values in the top-bottom strategy. Finally, the constant C for succinctness is calculated based on the original paper.

AutoRoot. AutoRoot also adopts two procedures similar to Squeeze. Differently, kernel density estimation is used in clustering to determine the number of clusters based on the distribution of deviations of dimension values. The bandwidth = 0.2 is set in the kernel function to estimate the distribution. Secondly, $\theta = 0.9$ is set as the threshold of **new potential score**.

CMMD. The proposed CMMD mainly consists of two modules, i.e., relationship modeling and root cause localization. The hyper-parameters used in these two modules are summarized in Table 5. During GAT training stage, the batch size is equal to the number of non-leaf nodes in the dimension tree. In root cause localization, the number of population N_p and the number of iteration N are depended on the size of data. N_p can be set from 10 to 50 and N can be set from 5 to 10 in the experiments. The length of chromosomes in GA is equal to the number of dimension value combinations after filtering. t_δ determines the abnormal degree of nodes in candidate set and we search t_δ from 0.1,0.2 for different datasets in this work.

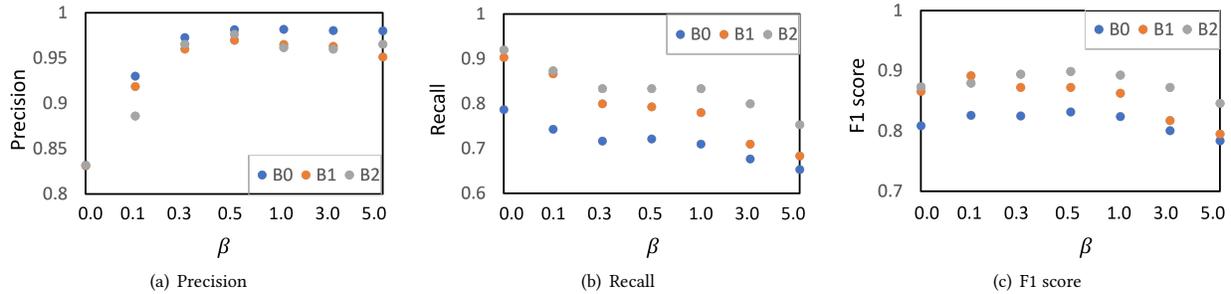


Figure 8: Performance over different β

A.3 Details in evaluation

In the experiments, evaluation metrics are calculated between the ground truth and the output of these methods. In the three types of datasets, the ground truth of \mathcal{D}_2 is provided in [8]. In other two datasets, we assume that we know the calculation and aggregation function. Thus, the ground truth is a set of dimension value combinations that has the highest recovery ratio based on the known calculation and aggregation function. In detail, we firstly replace the real values with expected values of fundamental metrics in the candidate set of dimension value combinations and then calculate the recovery values of the monitored metric by aggregation function. Further, the recovery ratio with and without replacement is also calculated. If the recovery value is closer to the expected value, the recovery ratio is higher which means that these dimension value combinations can explain the anomaly in the monitored metric better. Given the fact that every dimension value combination can contribute to the abnormal monitored metric more or less, we choose the dimension value combination(s) whose cumulative recovery ratio exceeds a certain threshold (80% in this paper) as the ground truth.

A.4 Synthetic datasets

The synthetic datasets \mathcal{D}_3 in the experiment are generated in the following steps:

- (1) Determine the number of dimensions and values in each dimension to form the dimension tree \mathbb{T}
- (2) Determine the metric relationship as follows. In \mathcal{D}_3 , we have two fundamental metrics b and c , and two derived metrics $a = g(c)$ and $d = f(a, b)$.
- (3) Generate the data of b and c in each dimension value combination. Here we firstly use the uniform distribution to generate the value of b and c at specific timestamps. Then we randomly choose an operator from $g = \{c, \sin(c), e^c, c^2, \sqrt{c}\}$ to transform c into a . Finally, we set function f as five derived functions: 1) $\frac{a}{b}$, 2) $a \times b$, 3) $\frac{\log a}{\log b}$, 4) $a \times e^b$ and 5) $\frac{\log(a+1)}{\log(b+1)}$.
- (4) Aggregate the fundamental metrics in the dimension tree \mathbb{T} and then calculate the corresponding derived metrics.
- (5) Inject the anomalies randomly in the fundamental metrics of selected dimension value combinations and then calculate the abnormal metrics of corresponding dimension values.

- (6) Calculate the expected values of each metric by the autoregressive models and detect the anomalies by the 3σ principal.

A.5 Parameter configuration

The parameter configurations are important for the existing root cause analysis approaches as well as the proposed CMMD. Here we consider the important trade-off parameter β in Eq. 6 for succinctness of root causes. To conduct experiments under different values of β , we select subdatasets B0, B1, B2 in [8]. The results under different parameter values are shown in Fig. 8. In general, with the increase of β , the strategy focuses more and more on the succinctness rather than the scores of root causes which will lead to the degradation of performance. Specifically, Precision and Recall show the opposite trend and we can find a better F1 score when $\beta \in [0.1, 1.0]$ and CMMD is somehow robust to this hyperparameter. Therefore it is not difficult to configure this parameter in practical.