Robust Anomaly Clue Localization of Multi-Dimensional Derived Measure for Online Video Services

Yongqian Sun, *Member, IEEE*, Daguo Cheng, Pengxiang Jin, Quan Ding, Shenglin Zhang, *Member, IEEE*, Xu Chen, Yuzhi Zhang, Minghan Liang, Dan Pei, *Senior Member, IEEE*, Jianyan Zheng, Sen Luo, and Xinyu Tang

Abstract—Anomaly clue localization of multi-dimensional derived measure is vitally important for the reliability of online video services. In this paper, we propose RobustSpot, an end-to-end framework for localizing the clues to anomalous multi-dimensional derived measures. RobustSpot integrates two novel indicators, i.e., "Anomaly Degree" and "Contribution Ability", with a simple yet effective method, weighted association rule mining (WARM), to automatically mine the hidden relationships across data dimensions for localizing the most likely clues to the root cause. Using 135 real-world cases collected from a top-tier global online video service provider H with 170+ million monthly active users, we demonstrate that RobustSpot achieves high accuracy (Top-5 accuracy of 98%), significantly outperforming state-of-the-art methods. The average localization time of RobustSpot is 1.83s, which is satisfying in our scenario. We have open-sourced the implementation of RobustSpot as well as the data used in the evaluation experiments.

Index Terms—Multi dimension, anomaly localization, AIOps

1 Introduction

Today's online video services provide application services for a significant number of customers. The performance of online video services is vitally important to customers' quality of experience (QoE) because even a short period of failure can bring poor experience to many customers. Therefore, operators carefully monitor these services' performance using some measures, e.g., the ratio of viewers suffering from stalling (stalling ratio) [1], [2]. When these measures become anomalous, it usually demonstrates that a high ratio of users suffers from QoE problems, and operators need to localize the clue to the root cause for mitigating the anomaly quickly.

- *Yongqian Sun, Daguo Cheng, Pengxiang Jin, Shenglin Zhang, Xu Chen, Yuzhi Zhang, and Minghan Liang are with Nankai University, Tianjin 300071, China. E-mail: {sunyongqian, zhangsl, zyz}@nankai.edu.cn, {chengdaguo, jinpengxiang, cnarutox, liangminghan}@mail.nankai.edu.cn.*
- *Quan Ding is with the Institute of Computing Technology, University of Chinese Academy of Sciences, Beijing 101408, China. E-mail: dingquan20g@ict.ac.cn.*
- *Dan Pei is with the Department of Computer Science, Tsinghua University, Beijing 100190, China, and also with the Beijing National Research Center for Information Science and Technology, Beijing 100190, China. E-mail: peidan@tsinghua.edu.cn.*
- *Jianyan Zheng, Sen Luo, and Xinyu Tang are with HUYA, Inc., Guangzhou 511446, China. E-mail: {zhengjianyan, luosen, tangxinyu}@huya.com.*

Manuscript received 15 Apr. 2021; revised 24 Jan. 2022; accepted 25 Feb. 2022. Date of publication 1 Mar. 2022; date of current version 10 Apr. 2023.
This work was supported in part by the National Key R&D Program of China under Grant 2021YFB0300104, in part by the National Science Foundation of Tianjin under Grant 21JCQNJC00180, in part by the National Natural Science Foundation of China under Grants 61902200, and 62072264, in part by the China Postdoctoral Science Foundation under Grant 2019M651015, in part by the Beijing National Research Center for Information Science and Technology (BNRist), and in part by the Tianjin Key Laboratory of Operating System.
(Corresponding author: Shenglin Zhang.)
Digital Object Identifier no. 10.1109/TSC.2022.3155500

Usually, these measures are derived from *fundamental measures* [3], [4], and they have several attributes, each of which has multiple attribute values. Taking stalling ratio (SR) as an example, it is a *derived measure* [3], [4] generated from two fundamental measures: the number of viewers suffering from stalling (SV) and the number of online viewers (OV). Specifically, $SR = SV/OV$. Table 1 lists three real-world examples of the measurement records of SV, OV and SR. Usually, a measurement record has several attributes, e.g., CDN (the Content Delivery Network provider that delivers the video), Bitrate (the number of bits transmitted per second needed to play the video smoothly), and Device Type (the type of devices viewers used to play the video). Each attribute has multiple attribute values, e.g., the device used by viewers can be an iOS phone, an Android phone, or a PC. Typically, these measurement records are collected in every time interval (e.g., every minute). Moreover, the most fine-grained records can be naturally combined to form more coarse-grained measures. For example, all the measurement records with C=CDN1 and B=500 can be combined into $(C = CDN1, B = 500, D = *)$, where $*$ is a wildcard.

When the total measure (i.e., $(C = *, B = *, D = *)$) is anomalous, e.g., experiencing a sudden spike or level shift, it is crucial but challenging to localize the clue to the root cause, i.e., a combination of some attribute values (called single-clue hereinafter, e.g., $(C = CDN2, B = *, D = iOS)$), or two or more such combinations (called multi-clue hereinafter, e.g., $(C = CDN3, B = *, D = iOS)$ and $(C = CDN2, B = *, D = PC)$), that have the most potential to have caused the anomalous total measure. According to the clue, operators can expertise root cause localization and take action to quickly mitigate the anomaly. However, the localization of potential clues, e.g., $(C = CDN2, B = *, D = iOS)$, is quite difficult,

TABLE 1
Examples of SV, OV, SR Records

Attribute	CDN (C)	Bitrate (B)	Device Type (D)	Measure	SV	OV	SR
Attribute Value	CDN1	500	PC	Measurement record	5	90	0.056
	CDN2	500	iOS		8	100	0.08
	CDN3	2000	Android		5	110	0.045

because a sudden change (anomalous behavior) in the clue combination, $(C = CDN2, B = *, D = iOS)$, can propagate to more coarse-grained combinations (cause them to change unexpectedly), e.g., $(C = *, B = *, D = iOS)$, and more fine-grained combinations, e.g., $(C = CDN2, B = 2000, D = iOS)$, or even other related combinations, e.g., $(C = *, B = 2000, D = *)$ [4], [5] (more details can be seen in Section 2.2). Consequently, it is infeasible to *manually* localize the anomaly clue of multi-dimensional derived measures. In this paper, we aim to automatically, accurately, and rapidly localize the clue when the total multi-dimensional derived measure of online video services becomes anomalous. Note that we only deal with the case where the total derived measure is anomalous in this paper.

Although several works have been proposed for anomaly clue localization of multi-dimensional measures, including iDice [6], HotSpot [5], Squeeze [4] and ImpAPTr [7], the former two works cannot localize anomalies for derived measures. Additionally, ImpAPTr cannot be applied for the multi-clue scenario, and neither Squeeze nor ImpAPTr can address the following two challenges well (more details can be seen in Section 2.2).

1) *Derived measures have more complex patterns than fundamental measures.* More specifically, the magnitude of the difference between forecast and actual measurement, on which iDice, HotSpot, ImpAPTr, and Squeeze relied, cannot fully capture the anomaly patterns of derived measures. To accurately localize clues, we should consider, for both derived measures and fundamental measures, the differences between their forecast measurement and actual measurement in terms of magnitude. In addition, we should also pay attention to the direction of the change from the actual measurement to the forecast measurement, i.e., an increase or a decrease.
2) *The propagation pattern and distribution of attribute values are complex*. After extensive investigations on real-world cases of multi-dimensional derived measures, we find that the propagation pattern from the clue combinations to other combinations, which is the key to infer the clue combinations, does not always comply with the "Ripple Effect" rule serving as the base of HotSpot and Squeeze. This causes neither HotSpot nor Squeeze to perform well in our scenario (more details can be seen in 5.2). Moreover, the fundamental measures with different attribute values in normal situations can be very imbalanced. For example, the SV with $(D = iOS)$ or that with $(D = Android)$ are much larger than that with $(D = PC)$ when the total SR is normal. This can easily cause that an anomaly localization method, e.g., iDice, HotSpot, ImpAPTr, and Squeeze, mistakenly regard the combinations which contain attribute values occupying a large portion in normal situation as the clue.

Therefore, we propose a novel anomaly localization framework, RobustSpot, to accurately and rapidly perform anomaly localization for multi-dimensional derived measures. The main contributions of this work are summarized as follows:

- We propose two simple yet effective indicators, namely "Anomaly Degree" and "Contribution Ability", through which the possible clues are narrowed down to a small set. The combination of these two indicators accurately captures the complex patterns of anomalous derived measures and thus addresses the first challenge.
- We propose a new algorithm, Weighted Association Rule Mining (WARM), to capture the hidden relationship across dimensions and effectively learn the propagation pattern from clue combinations to other combinations. In addition, for each attribute, WARM successfully balances the biased distribution of its different attribute values. WARM addresses the second challenge.
- Based on 135 real-world cases collected from a top-tier global online video service provider H with 170+ million monthly active users, we conduct extensive experiments to demonstrate the performance of RobustSpot. RobustSpot achieves 98% $ACC@Top5$, which is greatly higher (by 26%) than the best baseline method. In addition, the average anomaly localization time for each case is 1.83s, satisfying the requirement of online deployment.
- We have implemented and deployed RobustSpot in H for 10+ months, and share some successful stories in Section 6. Moreover, to get readers more easily understand our work, we have made the data and code publicly available [8].

2 Preliminaries

2.1 Problem Definition

Table 2 lists the important notations of this paper. A measure is an indicator reflecting the QoE. It is dynamically monitored and collected at the server-side using application performance management (APM) systems such as CAT, pinpoint, skywalking, and zipkin [7]. It can be classified into two types: *fundamental measure* and *derived measure*. A fundamental measure usually counts the number of viewers with predefined characteristics, e.g., setting video's Bitrate = 500, using Android devices, suffering from stalling. Typically, it is additive, e.g., for the SV in Table 1, $SV(B = 500) = 5 + 8 = 13$. A derived measure (e.g., SR), however, is usually a ratio derived from fundamental measures[1], and it is not additive, e.g.,

1. We mainly focus on ratio based derived measures in this work.

TABLE 2
List of Important Notations

Notation	Definition	Notation	Definition
Fundamental measure	The additive measure, e.g., SV, OV	$f_d(T)$	The forecast measurement value of the total SR
Derived measure	The non-additive measure derived from fundamental measures	$\mathbb{P}$	The set of preserved ε
Attribute (dimension)	A feature of an online viewer, e.g., video bitrate, CDN, device type	$\mathbb{D}_p$, $\mathbb{D}_o$	The databases weighting ε in $\mathbb{P}$ and $\mathbb{E}$, respectively
Attribute value	The potential value of an attribute, e.g., the device type can be iOS, Android or PC	$Support_p$, $Support_o$	The $Support$ of ε in $\mathbb{D}_p$ and $\mathbb{D}_o$, respectively
η	An attribute combination	$Support_b$	The difference between $Support_p$ and $Support_o$
ε	A leaf attribute combination	τ_a, τ_c	The thresholds of AD and CA, respectively
$\mathbb{E}$	The set of ε	$\mathbb{C}$	The set of $\eta \in \mathbb{D}_p$
$m_d(\varepsilon)$, $m_s(\varepsilon)$, $m_o(\varepsilon)$	The actual measurement value of ε's SR, SV and OV, respectively	$\mathbb{S}$	The set constituted by the subsets of $\mathbb{C}$ where there are no two subsets share common attribute value
$f_d(\varepsilon)$, $f_s(\varepsilon)$, $f_o(\varepsilon)$	The forecast measurement value of ε's SR, SV and OV, respectively	$\mathbb{R}$	The candidate clue set

$SR(B = 500) \neq 0.056 + 0.08$. It can only be calculated by aggregating its fundamental measures, e.g., $SR(B = 500) = \frac{5+8}{90+100} = 0.068$. Typically, operators pay more attention to derived measures than fundamental measures, because the former one more comprehensively reflects the performance of services.

A measure usually contains multi-dimensional *attributes*. An attribute, e.g., CDN, Bitrate, Device Type, Internet service provider (ISP), defines a type of configuration information *at the viewer-side* and it is recorded by the APM systems *at the server-side*. It usually has different values. For example, a viewer can set the bitrate to be 500, 1200, or 2000 (kbps), choose a CDN as CDN1, CDN2, or CDN3, and use a device like an iOS phone, an Android phone, or a PC. A viewer usually falls into a specific *attribute combination*, e.g., $(C = CDN2, B = 500, D = iOS)$ in Table 1. This most-fine-grained attribute combination is called a *leaf attribute combination* (notated as ε). There are also aggregated attribute combinations, *e.g.*, $(C = CDN2, B = *, D = iOS)$, $(C = *, B = 2000, D = *)$. Fig. 1 shows all possible attribute combinations of the measures in Table 1, and the most-fine-grained attribute combination set in Layer 3 is called *leaf set* (notated as $\mathbb{E}$).

Usually, the root cause of an anomalous total SR can be a failed CDN service provider, an erroneous software upgrade, a misconfiguration at the server-side, a network disconnection of an ISP, *etc*. Localizing such root causes is quite challenging due to the large scale, frequent update, and high complexity of today's online video services[9]. When operators observe that the QoE is degraded (e.g., a sudden increase in the total SR), they usually try to identify the clue attribute combination(s) at first, which represents the right direction to explore the root cause of anomalous total SR without misleading information [7]. For example, after the service provider released a new version of the Android application, which did not effectively support high bitrate (e.g., $B = 2000$) video, most of the viewers with the attribute combination $(C = *, B = 2000, D = Android)$ suffered from stalling, leading to an anomalous total SR. The combination of $(C = *, B = 2000, D = Android)$ was a valid clue attribute combination for the anomalous total SR. However, neither $(C = *, B = 500, D = iOS)$ nor $(C = *, B = *, D = Android)$ was a valid clue attribute combination because the former showed the wrong direction for operators to explore the root cause, and the latter provided minimal information. After the clue attribute combination, i.e., $(C = *, B = 2000, D = Android)$, was localized, operators investigated what happened to the viewers using Android devices with Bitrate=2000. They found that the newly released Android application did not support a high bitrate well, which caused the viewers having upgraded Android applications to suffer from stalling. Therefore, they quickly fixed the bugs in the Android application and released a new version. Similarly, when operators found that $(C = CDN3, B = *, D = *)$ was the clue attribute combination, they investigated what happened to the CDN provider CDN3, and found that CDN3 experienced a network outage because of a router failure. Consequently, they switched the content delivery services provided by CDN3 to other CDN providers, successfully addressing the problem.

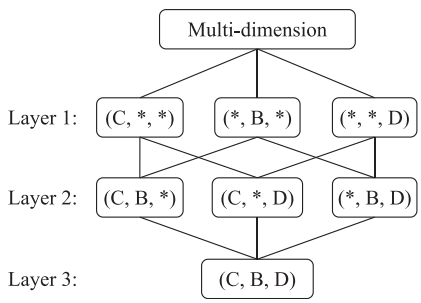


Fig. 1. The possible combinations for the measures in Table 1.

The search space of all possible clues is quite large. For example, when the number of attribute values of C, B and D attributes is 13, 24 and 9, respectively, there will be 3499 combinations. In this case, the search space becomes $2^{3499} - 1$ [5], making the brute-force search algorithm inapplicable in our scenario. Therefore, instead of brute-force search, RobustSpot integrates two novel indicators, i.e., "Anomaly Degree" and "Contribution Ability", with a simple yet effective method, WARM, to efficiently mine the hidden relationships across data dimensions (see Section 5.4 for more details).

Now we apply a toy example to show how to manually localize anomaly clues for multi-dimensional derived measures. As shown in Table 3, suppose that SR has two attributes – C and B, and their attribute value sets are {CDN1, CDN2, CDN3} and {1200, 500}, respectively. Apparently, when the

TABLE 3
A Toy Example of Clue Localization for Anomalous Total SR ($= \frac{SV}{OV}$) (Clue: C=CDN1)

f(C,B)→m(C,B)		CDN(C)			
		CDN1	CDN2	CDN3	*
Bitrate (B)	1200	$\frac{5}{110}\rightarrow\frac{75}{85}$	$\frac{5}{80}\rightarrow\frac{7}{80}$	$\frac{3}{100}\rightarrow\frac{2}{110}$	$\frac{13}{290}\rightarrow\frac{84}{275}$
	500	$\frac{3}{90}\rightarrow\frac{12}{65}$	$\frac{1}{20}\rightarrow\frac{1}{30}$	$\frac{3}{100}\rightarrow\frac{3}{110}$	$\frac{7}{210}\rightarrow\frac{16}{205}$
	*	$\frac{8}{200}\rightarrow\frac{87}{150}$	$\frac{6}{100}\rightarrow\frac{8}{110}$	$\frac{6}{200}\rightarrow\frac{5}{220}$	$\frac{20}{500}\rightarrow\frac{100}{480}$

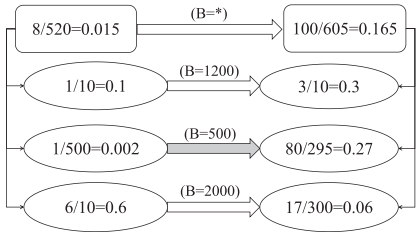


Fig. 2. An example of anomaly localization. The left is the forecast measurement of SR, and the right is the actual measurement of SR. (Actual Clue: B=500)

total SR becomes anomalous, the clue is an element or a non-empty subset of $\{(C = CDN1, B = *), (C = CDN2, B = *) \ldots, (C = CDN3, B = 500)\}$. For each cell in Table 3, the value before the arrow is the forecast (normal) measurement of $SR = SV/OV$, and the one after arrow is the actual value of SR. Obviously, the bottom-right cell denotes the total value of SR, which has a large difference between the forecast measurement, i.e., 0.04 ($\frac{20}{500}$), and the actual measurement, i.e., 0.21 ($\frac{100}{480}$), and thus it is anomalous. In this case, we can see that $(C = CDN1, B = 1200)$, $(C = CDN1, B = 500)$ and $(C = CDN1, B = *)$ all have large differences between their forecast and actual measurement, respectively. On the contrary, other attribute combinations have relatively small differences. That is, the ratios of all attribute combinations with $C = CDN1$ suffer from anomalies. Therefore, we can conclude that the clue of the anomalous total SR is $(C = CDN1, B = *)$, or $(C = CDN1)$ for short[2]. Although it seems straightforward to deduce the clue in this case, as the number of different attributes, and/or that of different attribute values of each attribute increase, the search space will increase exponentially.

2.2 Challenges

There are two main challenges in anomaly localization of multi-dimensional derived measures as follows.

Challenge 1: Derived Measures Have More Complex Patterns Than Fundamental Measures. As aforementioned, the clue combinations are those contribute the most to the anomalous total measure. In anomaly localization for fundamental measure, only considering the magnitude of the difference between the forecast and actual measurement of fundamental measures is sufficient [4], [5], [6]. For example, when conducting anomaly localization for the fundamental measure SV, we only need to pay attention to its forecast and actual measurement. However, because the fundamental measures of a derived measure can change (increase or decrease) simultaneously, the difference between the forecast and actual measurement of derived measures has a far more complex pattern than fundamental measures. To conduct anomaly localization for a derived measure, we should consider not only the direction and magnitude of the difference between its forecast and actual measurement, but also those of the difference between the forecast and actual measurement of its fundamental measures, respectively. For instance, when localizing anomalies for the derived measure SR, we need to pay attention to the forecast and actual measurement of SV, OV, and SR, respectively. In addition, compared to fundamental measures, the magnitude and variance of the derived measures are smaller, so it is more difficult to find significant anomalous values in the search space.

For example, as shown in Fig. 2, suppose that there are three candidate attribute values contributing to this anomaly, i.e., $(B = 1200)$, $(B = 500)$, and $(B = 2000)$. The total SR is suffering from an anomaly because its actual measurement (0.165) is much larger than its forecast measurement (0.015). If we only pay attention to the magnitude of the difference between the forecast and actual measurement of SR, $(B = 1200)$, $(B = 500)$, and $(B = 2000)$ all have a huge difference between their forecast and actual measurement of SR, and they are all supposed to be candidate clues. However, $(B = 2000)$ has a negative contribution to the increase of the total SR, because its SR decreases from 0.6 to 0.06. Moreover, although $(B = 1200)$ contributes positively to the increase of the total SR, neither of its fundamental measures, i.e., SV and OV, has a large difference between the forecast and actual measurement. That is, $(B = 1200)$ contributes little to the large difference between the forecast and actual measurement of the total SR. Therefore, neither $(B = 2000)$ nor $(B = 1200)$ is the clue. $(B = 500)$, on the other hand, not only has a positive contribution to the increase of the total SR, but also has large differences between its forecast and actual measurement of fundamental measures. Consequently, it is the actual clue of the anomalous total SR.

Challenge 2: The Propagation Pattern and Distribution of Attribute Values are Complex. The propagation pattern from the clue combination to other combinations is complex. As shown in Table 3, when $(C = CDN1, B = *)$ is anomalous, it will propagate to other related combinations (cause them to change unexpectedly), e.g., $(C = CDN1, B = 1200)$, $(C = CDN1, B = 500)$, $(C = *, B = 1200)$, $(C = *, B = 500)$. Because both $(C = CDN1, B = 1200)$ and $(C = CDN1, B = 500)$ are the "child" combinations of $(C = CDN1, B = *)$, they change as $(C = CDN1, B = *)$ changes[3]. Additionally, $(C = *, B = 1200)$ and $(C = *, B = 500)$ are the "parent" combinations of $(C = CDN1, B = 1200)$ and $(C = CDN1, B = 500)$, respectively, and thus they change accordingly.

In order to accurately localize the anomaly clue, both HotSpot [5] and Squeeze [4] are designed based on the assumption that anomalies propagate observing the Ripple Effect rule. This rule believes that the difference between

2. Hereinafter, we may omit an attribute in a combination when it is a wildcard. For example, $(C = CDN1, B = *)$ and $(C = CDN1)$ is interchangeable in this paper

3. After investigating hundreds of real-world cases, we find that, for all cases, when one of the clue combinations suffers from an unexpected change, all its child combinations will change accordingly.

the forecast and actual measurement of clue combination, i.e., $(C = CDN1, B = *)$, propagate to its "child" combinations, i.e., $(C = CDN1, B = 1200)$ and $(C = CDN1, B = 500)$, in proportion to the forecast measurement of each "child" combination, i.e., $\frac{5}{110}, \frac{3}{90}$, respectively. However, it does not hold in our scenario. For example, the larger the Bitrate is set, the better network transmission is needed. When the network conditions remain unchanged, we can infer that the larger the Bitrate is set, the more likely it causes the video to stall. Therefore, the proportional relationship between the measurement of the clue combination and its descendant combinations does not hold anymore. For example, if $(C = CDN1, B = 1200)$ complies with the Ripple Effect, the actual value of $SR(C = CDN1, B = 1200)$ should be

$$\left(\frac{87}{150} - \frac{8}{200}\right) \times \frac{\frac{5}{110}}{\frac{8}{200}} + \frac{5}{110} = \frac{\frac{87}{150} \times \frac{5}{110}}{\frac{8}{200}} = \frac{87}{132} \left(< \frac{75}{85}\right)$$

However, the actual measurement of $SR(C = CDN1, B = 1200)$ is $\frac{75}{85}$, rather than $\frac{87}{132}$, and $\frac{75}{85} > \frac{87}{132}$. This is mainly because the viewers with $B = 1200$ is more prone to stall than those with $B = 500$. Therefore, when $(C = CDN1)$ is the clue, viewers with $(C = CDN1, B = 1200)$ have a higher probability to suffer from stalling than those with $(C = CDN1, B = 500)$.

Additionally, the distribution of different attribute values is highly imbalanced. Usually, for an attribute, the fundamental measures with different attribute values are not balanced. For example, over 90% of viewers watch online videos using mobile devices including iOS devices and Android devices, leading to that in most cases, the SV with $(D = iOS)$ or $(D = Android)$ occupies a large portion of the total SV. It can easily cause an anomaly localization method, if it ignores the imbalanced distribution, to mistakenly consider that $(D = iOS)$ or $(D = Android)$ is always the clue.

3 Core Idea and Overview

In RobustSpot, we first propose two novel indicators, "Anomaly Degree (AD)" (Section 4.1) and "Contribution Ability (CA)" (Section 4.2), to comprehensively describe a leaf combination. Specifically, AD represents whether a leaf combination is anomalous, and if it is, how severe the anomaly is. In addition, CA denotes the contribution of an anomalous leaf combination to the anomaly of the total measure. In this way, only the leaf combinations that are significantly anomalous and contribute greatly to the anomalous total measures are preserved. This addresses the first challenge (more details can be seen in Section 2.2).

Furthermore, we propose a simple yet effective method, Weighted Association Rule Mining (WARM) (Section 4.3), to capture the propagation pattern from clue combination to other combinations. WARM does not assume that the propagation pattern observes the Ripple Effect rule (see Section 2.2 for more details). Considering that for a given attribute, the distribution of different attribute values can be very imbalanced, WARM also balances the biased distribution of each combination.

Fig. 3 shows the overview of RobustSpot. After an anomaly of the total measure, i.e., SR, is detected, we first collect the measurement data of the fundamental measures, i.e., SV and OV, at the most fine-grained level. In this way, we obtain the measurement data of all leaf combinations. After that, we obtain the significantly anomalous leaf combinations that contribute greatly to the anomaly of the total SR by combining the proposed AD and CA indicators. Using the novel WARM method, we finally obtain the candidate clue set.

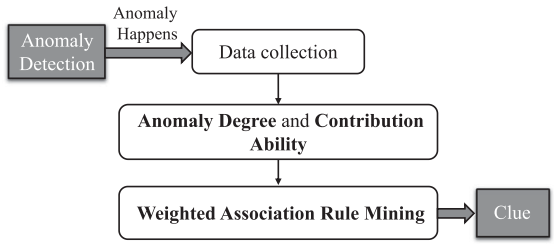


Fig. 3. The overview of RobustSpot.

4 Design

In this section, we first introduce Anomaly Degree in Section 4.1, followed by the description of Contribution Ability in Section 4.2. The WARM method is elaborated in Section 4.3.

4.1 Anomaly Degree

Anomaly detection for the total derived measure (e.g., SR) has been well studied for decades [10], [11], [12], [13], [14], [15]. In this work, we apply Bagel [13], an unsupervised metric anomaly detection method that has been demonstrated to be robust and efficient using real-world data, to automatically detect anomalies in the total SR. Note that applying Bagel for anomaly detection is not the main contribution of our work. After an anomaly of the total SR is detected, RobustSpot automatically collects the measurement data of the fundamental measures of leaf combinations (see Table 1 for example).

To determine whether a leaf combination is anomalous, and if it is, how severe the anomaly is, we first calculate its AD. Intuitively, for the derived measure of a leaf combination, we determine the anomaly severity of its measurement at time t from two aspects: how far it deviates from its historical measurement before t, and how prominent the anomaly is compared to the measurement of other leaf combinations at t.

Usually, for a derived measure, we measure how far its measurement deviates from its forecast measurement. The anomaly detection algorithm used in our work can report the moment t from which the anomaly starts. Therefore, the total SR is normal before t. Using the measurement within a period w before t to predict the normal value of t will introduce little bias. Considering the complex pattern (e.g., large fluctuation, very small magnitude) of measurement data, and the large number of leaf combinations, we apply a robust and efficient forecast algorithm, Moving Average (MA) [10], to obtain the forecast measurement for each leaf combination. Specifically, for a leaf combination ε, suppose that the total SR becomes anomalous at t, we apply the average of its collected measurement within a period w before t as its forecast measurement $f(\varepsilon)$. Typically, a measure's measurement

TABLE 4
Examples of Leaf Combinations and Their CA and AD

ε	f_s	m_s	f_o	m_o	f_d	m_d	AD	CA
(C=CDN1, B=500, D=PC)	5	5	100	100	0.05	0.05	0	0
(C=CDN2, B=500, D=iOS)	5	10	50	100	0.1	0.1	0	≈ 0
(C=CDN3, B=500, D=PC)	10	10	100	50	0.1	0.2	0.67	0.11
(C=CDN1, B=2000, D=iOS)	0	1	5	5	0	0.2	0.86	≈ 0
(C=CDN1, B=500, D=iOS)	0	0	100	100	0	0	0	0
(C=CDN3, B=500, D=iOS)	0	30	100	100	0	0.3	0.95	0.86
(C=CDN2, B=1200, D=PC)	15	4	55	55	0.27	0.07	0.67	-0.31
Total	35	60	510	510	0.07	0.12	-	-

constantly changes over time, and there are peaks and valleys within one day, and its patterns also change in different days. Consequently, it is very difficult to calculate $f(\varepsilon)$ using the pre-fixed values computed from historical measurements.

Furthermore, we determine the prominence of the anomaly of ε compared to the measurement of other leaf combinations. Specifically, given ε and the set of all leaf combinations $\mathbb{E}$, we can obtain the AD of ε as

$$AD(\varepsilon) = 1 - \frac{1}{\frac{n \times |m_d(\varepsilon) - f_d(\varepsilon)|}{|\sum_{i=1}^{n-1}(m_d(\varepsilon_i) - f_d(\varepsilon_i))|} + 1} \tag{1}$$

where $m_d(\cdot)$ and $f_d(\cdot)$ are the actual and forecast measurement of $(\cdot)$'s *derived* measure at t, respectively, $\varepsilon_i \in \mathbb{E}$ and ε_i / $= \varepsilon$, and n is the number of leaf combinations in $\mathbb{E}$. To get readers easily understand the concept of AD, we list several concrete examples of ε as well as their corresponding f_d, m_d, and AD in Table 4.

Obviously, the more prominently the actual measurement of ε's derived measure deviates from its forecast measurement, the larger $AD(\varepsilon)$ is. Let

$$\frac{n \times |m_d(\varepsilon) - f_d(\varepsilon)|}{|\sum_{i=1}^{n}(m_d(\varepsilon_i) - f_d(\varepsilon_i))|} = x \tag{2}$$

Fig. 4 shows how $AD(\varepsilon)$ changes as x increases. We can find that $AD(\varepsilon) \in [0, 1)$. Moreover, when the deviation of the actual measurement of the derived measure, $m_d(\varepsilon)$, from its forecast measurement, $f_d(\varepsilon)$, is not very significant, and thus x is not very large (say smaller than 2.0), $AD(\varepsilon)$ grows rapidly as the deviation becomes more significant. Therefore, $AD(\varepsilon)$ makes the deviation distinguishable even if it is not prominent. Overall, $AD(\varepsilon)$ is robust to determine the anomaly severity in most cases, even when $m_d(\varepsilon)$ does not deviate much from $f_d(\varepsilon)$ compared to other leaf combinations. The above is also the reason why we have not simply normalized AD.

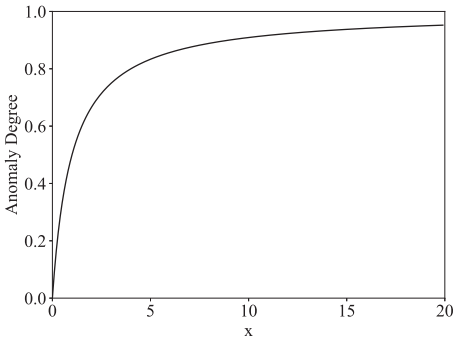


Fig. 4. How $AD(\varepsilon)$ changes as x increases.

Because we should only pay attention to the leaf combination that has large AD and thus suffers from a significant anomaly, we apply the Knee-point method [4], [16] to automatically determine the threshold of AD, i.e., τ_a. The Knee-point method has been demonstrated to be accurate and efficient for automatic threshold tuning in practice [4]. Note that applying this method for threshold tuning is not the main contribution of our work.

4.2 Contribution Ability

Now we have obtained the AD for each leaf combination to measure its anomaly severity. However, a leaf combination with large AD and thus having high anomaly severity does not necessarily contribute much to the anomaly of the total derived measure. For example, as listed in Table 4, when $\varepsilon = (C = CDN1, B = 2000, D = iOS)$ (the fifth row), $AD = 0.86$, which is very large (recall that $AD \in [0, 1)$), demonstrating that this leaf combination suffers from a severe anomaly. Nevertheless, it is obvious that the anomaly of the total derived measure ($f_d = 0.07 \rightarrow m_d = 0.12$) is mainly caused by the large fluctuation of the total SV ($f_s = 35 \rightarrow m_s = 60$), and the change of SV in ε ($f_s = 0 \rightarrow m_s = 1$) contributes little to this fluctuation.

Intuitively, to measure how the fluctuation of a leaf combination ε contribute to that of the total derived measure, we assume that only ε suffers from an anomaly (i.e., using its actual measurement) and other leaf combinations are normal (i.e., using their forecast measurement), and determine how the measurement of the total derived measure in this case deviate from its forecast measurement. Specifically, we calculate $CA(\varepsilon)$ as

$$CA(\varepsilon) = \frac{\theta(\varepsilon) - f_d(T)}{f_d(T)} \tag{3}$$

where $f_d(T)$ $(= f_s(T)/f_o(T))$ is the forecast measurement of the total derived measure, i.e., SR. $f_s(T)$ and $f_o(T)$ are the forecast measurement of the total fundamental measure, i.e., SV and OV, respectively. And $\theta(\varepsilon)$ is

$$\theta(\varepsilon) = \frac{f_s(T) + (m_s(\varepsilon) - f_s(\varepsilon))}{f_o(T) + (m_o(\varepsilon) - f_o(\varepsilon))} \tag{4}$$

where $m_s(\varepsilon)$ and $m_o(\varepsilon)$ are the actual measurements of ε's SV and OV, respectively, $f_s(\varepsilon)$ and $f_o(\varepsilon)$ are the forecast measurement of ε's SV and OV, respectively. The maximum and minimum values of CA are different in different cases. Theoretically, the possible maximum and minimum of CA are positive infinity and -1.0, respectively. When $f_d(T \rightarrow 0)$

TABLE 5
Examples of AD and CA to Show That CA Alone is Not Sufficient to Determine the Prominently Anomalous Leaf Attributions

ε	f_s	m_s	f_o	m_o	f_d	m_d	AD	CA
(C=CDN4, B=500, D=iOS)	50	200	1000	500	0.05	0.4	0.97	1.77
(C=CDN5, B=1200, D=iOS)	50	100	500	1000	0.01	0.01	0	0.39
(C=CDN5, B=500, D=PC)	0	100	4000	4000	0	0.025	0.125	1
Total	100	400	5500	5500	0.018	0.073	-	-

and $\theta > 0$, $CA(\varepsilon) \to \infty$. Additionally, when $\theta \to 0$, $CA(\varepsilon) \to -1$. Moreover, the larger the actual measurement of ε's SV compared to its forecast measurement (i.e., $m_s(\varepsilon) - f_s(\varepsilon)$), and the smaller the actual measurement of ε's OV compared to its forecast measurement (i.e., $m_o(\varepsilon) - f_o(\varepsilon)$), the more ε contributes to the anomaly of the total SR, and the larger $CA(\varepsilon)$ becomes At this time, from Eq. (3) and Therefore, we can infer that the maximum $CA(\varepsilon)$ indeed denotes the maximum contribution.

In this way, as listed in Table 4, when $\varepsilon = (C = CDN1, B = 2000, D = iOS)$ (the fifth row), $CA(\varepsilon) \approx 0$, demonstrating that ε has a small contribution to the anomaly of the total SR. On the contrary, when $\varepsilon = (C = CDN3, B = 500, D = iOS)$ (the seventh row), $CA(\varepsilon) = 0.86$, $AD(\varepsilon) = 0.95$, showing that ε suffers from a significant anomaly and it makes a great contribution to the anomalous total SR. This is mainly because its change of SV ($f_s = 0 \to m_s = 30$) has a large contribution to the change of the total SV ($f_s = 35 \to m_s = 60$). From Table 4, we can also see that some fluctuation in the leaf combination may have a negative contribution to the anomaly of the total SR. For instance, when $\varepsilon = (C = CDN2, B = 1200, D = PC)$ (the eighth row), $CA(\varepsilon) = -0.31$, demonstrating that ε contributes negatively to the anomaly of the total SR. It is because its SV decreases ($f_s = 15 \to m_s = 4$) while the total SV increases ($f_s = 35 \to m_s = 60$).

It can be easily obtained from Eq. (3) that, when $CA(\varepsilon)$ approximates to 0, or is negative, ε has a little or negative contribution to the anomaly of the total SR. Only when $CA(\varepsilon) > 0$, ε has a positive contribution to the anomaly of the total SR. Therefore, we set the threshold of CA, i.e., $\tau_c = 0$, and automatically prune the leaf combinations whose $CA \leq \tau_c$. A positive or negative CA has a different physical meaning. When $CA(\varepsilon)$ is positive (negative), ε has a positive (negative) contribution to the anomalous total SR. Therefore, we do not normalize CA. In this way, only $\varepsilon = (C = CDN3, B = 500, D = PC)$ (the fourth row) and $\varepsilon = (C = CDN3, B = 500, D = iOS)$ (the seventh row) in Table 4 are preserved.

Although CA can measure the contribution of a leaf combination to the anomalous total SR, using CA alone is not sufficient in the anomaly localization process. For instance, as listed in Table 5, when $\varepsilon = (C = CDN5, B = 1200, D = iOS)$ (the second row), $CA = 0.39 > 0$, demonstrating that ε has a positive contribution to the anomaly of the total SR. Nevertheless, ε's actual SR is as expected ($f_d = 0.01 \to m_d = 0.01$), and thus it is not anomalous ($AD = 0$). Therefore, ε should not be preserved for further anomaly localization process. Similarly, we do not retain $\varepsilon = (C = CDN5, B = 500, D = PC)$ (the third row) because its AD is very small.

4.3 Weighted Association Rule Mining

Now we have obtained the leaf combinations that suffer from prominent anomalies and contribute significantly to the anomalous total derived measure, and they constitute the set of preserved leaf combinations, i.e., $\mathbb{P}$. However, we cannot directly use these leaf combinations as clues because: (1) Usually there are a large number of leaf combinations satisfying the above two conditions when the total derived measure is anomalous. Operators have to spend a lot of time checking every combination if using them as clues. (2) These leaf combinations are too fine-grained to capture the pattern of the clue, and operators need those more coarse-grained combinations that better represent the pattern of clue. Consequently, we propose a simple yet effective method, WARM, to narrow down the anomaly localization results to a small number of combinations capturing the pattern of the clue.

After investigating extensive real-world anomalous SR cases, we have the following two observations:

1) The combinations in the clue set are *significantly* anomalous (with a high AD) and contribute *the most* to the anomaly of the total SR (with a high CA), and thus cover most of the above preserved leaf combinations.
2) The measurement of OV (i.e., m_o) with different attribute values can be highly imbalanced. For example, as shown in Table 4, we have

$$\frac{m_o(C = *, B = 500, D = *)}{m_o(Total)} = \frac{450}{510} = 88\% \tag{5}$$

It denotes that 88% online viewers set video Bitrate to 500, and only 12% online viewers use other types of Bitrate. It can easily cause an anomaly localization method ignoring the imbalanced distribution to mistakenly consider that $(B = 500)$ is always the clue.

We thus design WARM motivated by association rule mining [17], which is a rule-based machine learning method for discovering interesting relations between variables. In our scenario, the scale of the data is significant, and thus the designed rule mining algorithm should be computationally efficient. Additionally, operators want to implement a simple rule mining algorithm that is easy to understand. WARM is efficient and straightforward. Therefore it is suitable in our scenario. It consists of two core ideas: (1) creating a weighted database according to AD and CA to capture the hidden relationship across different attributes, and (2) balancing each combination's proportion (i.e., contribution) in the weighted database based on its fundamental measure. Specifically, to apply the rule-based machine learning method for efficiently capturing the pattern of clue combinations, we first create two *databases*, i.e., $\mathbb{D}_p$ and $\mathbb{D}_o$, respectively.

TABLE 6
$\mathbb{D}_p$, the Weighted Database Generated by the Preserved Leaf Combinations in Table 4

$\mathbb{N}_p(\varepsilon)(\lambda = 100)$	C	B	D
$\varepsilon = (C = CDN3, B = 500, D = PC)$	CDN3	500	PC
$\lfloor 0.67 \times 0.11 \times 100 \rfloor = 7(Trans)$	...	...	...
	CDN3	500	PC
$\varepsilon = (C = CDN3, B = 500, D = iOS)$	CDN3	500	iOS
$\lfloor 0.95 \times 0.86 \times 100 \rfloor = 81(Trans)$	...	...	...
	CDN3	500	iOS

TABLE 7
$\mathbb{D}_o$, the Weighted Database Generated by All Leaf Combinations in Table 4

$\mathbb{N}_o(\varepsilon)$	C	B	D
$\varepsilon = (C = CDN1, B = 500, D = PC)$	CDN1	500	PC
$m_o = 100(Trans)$	...	...	...
	CDN1	500	PC
⋮	⋮	⋮	⋮
$\varepsilon = (C = CDN2, B = 1200, D = PC)$	CDN2	1200	PC
$m_o = 55(Trans)$	...	...	...
	CDN2	1200	PC

- $\mathbb{D}_p$. We create $\mathbb{D}_p$ by weighting every *preserved* leaf combinations based on their AD and CA. In this database, ε's number of *transactions*, $\mathbb{N}_p(\varepsilon)$, is

$$\mathbb{N}_p(\varepsilon) = \lfloor AD(\varepsilon) \times CA(\varepsilon) \times \lambda \rfloor \tag{6}$$

  where λ is an amplification factor. The objective of applying λ is to construct a large database to facilitate the application of data mining algorithms. In the same scenario, we set the same value for all preserved leaf combinations. In different scenarios, if the value of $AD(\varepsilon) \times CA(\varepsilon)$ is small, a large λ, say 1000, is preferred. Otherwise, we can set λ to a small value, e.g., 100 in our scenarios (see Section 5.1.5 for more details). In this way, we incorporate every preserved leaf combination as well as their AD and CA in $\mathbb{D}_p$. To get readers better understand the database, in Table 6 we list the $\mathbb{D}_p$ generated from the preserved leaf combinations of Table 4.
- $\mathbb{D}_o$. We create $\mathbb{D}_o$ by weighting all the leaf combinations based on their actual measurement of OV (i.e., m_o). More specifically, ε's number of *transactions* in $\mathbb{D}_o$ is: $\mathbb{N}_o(\varepsilon) = m_o$. For example, in Table 7 we list $\mathbb{D}_o$ generated from the raw dataset listed in Table 4.

Now we define $Support_p$, $Support_o$ and $Support_b$ in our scenario to mine the pattern of clue combinations and solve the challenge induced by the imbalanced distribution of OV.

- $Support_p$ and $Support_o$. Based on Eq. 6, we can conclude that for a leaf combination ε, the larger $AD(\varepsilon)$ and $CA(\varepsilon)$ are, the more transactions ε occupies in $\mathbb{D}_p$. Based on Observation 1, we can conclude that the combinations of attribute values in the clue set frequently occur in $\mathbb{D}_p$. Therefore, we calculate the occurrence frequency of every combination using Support. The Support of a combination is the proportion of transactions in which it appears. More specifically

$$Support(\eta_1, \ldots, \eta_m) = P(\eta_1 \cup ... \cup \eta_m) \tag{7}$$

  where η_i is a combination, and $\eta_1, \ldots, \eta_m$ is a set containing m combinations. For a combination, its $Support_p$ and $Support_o$ are its Support in $\mathbb{D}_p$ and $\mathbb{D}_o$, respectively. For example, as listed in Tables 6 and 7, $Support_p(C = CDN3, B = 500, D = *) = P(C = CDN3, B = 500, D = *) = 1.0$ and $Support_o(C = *, B = 500, D = *) = P(C = *, B = 500, D = *) = 0.88$, respectively. We can find that $Support_o$ represents the distribution of OV from with respect to attribute values.
- $Support_b$. As discussed in Observation 2, for an attribute, the m_o (i.e., actual measurement of OV) of its different attribute values can be highly imbalanced. To address this challenge, we should consider OV's distribution of different attribute values. Since combinations' $Support_o$ reveals OV's distribution concerning different attribute values, we propose a simple yet effective indicator, support balancer ($Support_b$), integrating their Support in both $\mathbb{D}_p$ and $\mathbb{D}_o$ as follows:

$$\begin{aligned} Support_b(\eta_1, \ldots, \eta_m) = {} & Support_p(\eta_1, \ldots, \eta_m) \\ & - Support_o(\eta_1, \ldots, \eta_m) \end{aligned} \tag{8}$$

  Specifically, to alleviate the impact of the imbalanced data distribution of OV, $Support_b$ applies the difference between combinations' $Support_p$ and $Support_b$ to measure how much these combinations contribute to the anomalous total derived measure.

Assuming that all η that appears in $\mathbb{D}_p$ form a set $\mathbb{C}$, the anomalous total derived measure can be caused by a single combination $\eta_i \in \mathbb{C}$ (single-clue), or a subset of two or more combinations $\{\eta_i, \ldots, \eta_j\} \subset \mathbb{C}$ (multi-clue), where $\eta_i, \ldots, \eta_j$ become prominently anomalous simultaneously and together contribute significantly to the anomaly of the total derived measure. Usually, $\eta_i, \ldots, \eta_j$ do not share any non-wildcard common attribute value [4], [5], [6]. For example, $\{(B = 500, D = PC), (D = PC)\}$ cannot be a multi-clue.

As aforementioned, $Support_b$ integrates $Support_p$ which measures how prominently a set of combinations become anomalous and how significantly they contribute to the total anomalous derived measure, with $Support_o$ describing OV's distribution concerning different attribute values. Therefore, we first select the subsets of $\mathbb{C}$ where there are no two combinations that share common non-wildcard attribute values, and these subsets constitute $\mathbb{S}$. Usually, the number of subsets in $\mathbb{S}$ is not very large, e.g., < 1000 in our scenario. Then we sort the subsets in $\mathbb{S}$ based on their $Support_b$ and choose the Top k subsets to constitute the candidate clue set $\mathbb{R}$.

To get readers better understand the design of RobustSpot, we illustrate its procedure from input to result in Algorithm 1.

5 Evaluation

To evaluate the effectiveness and efficiency of RobustSpot, we perform extensive experiments to answer the following research questions.

- *RQ1 (Localization accuracy)*: How accurate is RobustSpot in localizing the anomaly of multi-dimensional derived measure for real-world online video services?

- *RQ2 (Main components)*: In this work, we propose two simple yet effective indicators, i.e., AD and CA, and a new WARM method. They serve as the main technical contributions of our work. How prominently do they impact the accuracy of RobustSpot?
- *RQ3 (Computational efficiency)*: Is the computational efficiency of RobustSpot sufficient for anomaly localization of multi-dimensional derived measure in real-world?

The rest of this section is organized as follows. We first depict the details of the experimental setup, including datasets, compared methods, metrics, and implementation in Section 5.1, and evaluate the accuracy of RobustSpot in Section 5.2. After that, we study how significantly AD, CA, and WARM contribute to the accuracy of RobustSpot in Section 5.3, followed by the introduction of RobustSpot's computational efficiency in 5.4.

Algorithm 1. The Design of RobustSpot

Input: $m_s(\varepsilon)$, $m_o(\varepsilon)$, $m_d(\varepsilon)$, where $\varepsilon \in \mathbb{E}$
Result: Candidate clue set $\mathbb{R}$ (Top k)

$\mathbb{P} = \varnothing, \mathbb{C} = \varnothing, \mathbb{S} = \varnothing$;
for $\varepsilon \in \mathbb{E}$ **do**
  $f_s(\varepsilon)$, $f_o(\varepsilon)$, $f_d(\varepsilon) \leftarrow$ MA;
  Calculate $AD(\varepsilon)$ using $m_d(\varepsilon), f_d(\varepsilon)$ (Eq. 1);
  Calculate $CA(\varepsilon)$ using $m_s(\varepsilon), m_o(\varepsilon), f_s(\varepsilon), f_o(\varepsilon)$ (Eq. 3);
  $\tau_a \leftarrow$ Knee-point($AD(\varepsilon)$), $\tau_c = 0.0$;
  if $AD(\varepsilon) > \tau_a \text{and} CA(\varepsilon) > \tau_c$ **then**
    $\mathbb{P} \leftarrow \varepsilon$;
  end
end
$\mathbb{D}_p \leftarrow$ weight $\mathbb{P}$ based on AD and CA (Eq. 6);
$\mathbb{D}_o \leftarrow$ weight $\mathbb{E}$ based on m_o;
Calculate $Support_p$ using $\mathbb{D}_p$ (Eq. 7);
Calculate $Support_o$ using $\mathbb{D}_o$ (Eq. 7);
$Support_b = Support_p - Support_o$ (Eq. 8);
for η *appears in* $\mathbb{D}_p$ **do**
  $\mathbb{C} \leftarrow \eta$;
end
for $\{\eta_i, \ldots, \eta_j\} \subset \mathbb{C}$ **do**
  if $\forall \eta_x, \eta_y \in \{\eta_i, \ldots, \eta_j\}$, η_x *and* η_y *do not share any common attribute value* **then**
    $\mathbb{S} \leftarrow \{\eta_i, \ldots, \eta_j\}$;
  end
end
$\mathbb{R} \leftarrow$ sort $\mathbb{S}$ by $Support_b$ and select Top k elements;
return $\mathbb{R}$;

5.1 Experimental Setup

5.1.1 Datasets

The real-world dataset used in our experiments is collected from a top-tier global online video service provider H, which provides services for 170+ million monthly active viewers. It consists of 135 anomalous SR cases that have been manually investigated by experienced operators. These cases are *randomly* selected within 10 months period. For each anomalous case, we collect its records of fundamental measures including SV and OV, and calculate their derived measure, i.e., SR, within the range between 25 minutes before and after the anomaly. Both of the two fundamental measures have five attributes (dimensions), i.e., CDN (C), Bitrate (B), Device Type (D), P2P (P), ISP (I), and the number of their different attribute values are 13, 24, 9, 3, 94, respectively.

TABLE 8
The Distribution of the Studied 135 Cases

Notation	#Combination	#Attribute	#Case	#Candidate clues
Single-1	1	1	71	About 6.05×10^9 per case
Single-2	1	2	45	
Single-3	1	3	11	
Multi	2	–	8	

As aforementioned, the anomalous derived measures, according to their clues, can be classified into two categories: single-clue and multi-clue. The distribution of the above 135 cases is listed in Table 8. Single-1/2/3 denotes that the clue is one single combination, and the numbers of non-wildcard attributes (dimensions) in the combination are 1, 2, 3, respectively. As for "Multi", two or more combinations simultaneously contribute to the total anomalous SR and together serve as the clue. Note that this distribution is consistent with that of all anomalies in the studied company.

5.1.2 Compared Methods

We compare RobustSpot with the state-of-the-art anomaly localization methods for multi-dimensional measures, including HotSpot [5], Squeeze [4], ImpAPTr [7] and iDice [6]. Additionally, Adtributor [3] can address the anomaly localization problem for single-dimensional scenarios, and we thus also compare RobustSpot with it. Note that the parameters of these baseline methods are set best for accuracy.

5.1.3 Evaluation Metrics

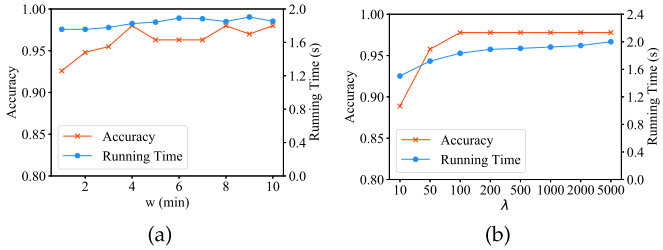
To evaluate the accuracy of RobustSpot and baseline methods, we need an appropriate evaluation metric. Because the anomaly localization methods usually generate a rank of potential clues for each case, we can evaluate the performance of each method based on the position of the actual clue in the rank [18]. Specifically, suppose γ is the actual clue of an anomaly case, $\Re$ is the set of the clues of the 135 cases, and thus we have $\gamma \in \Re$. Let ρ_γ represent the position of γ output by an anomaly localization method. Given k, γ_k denotes that whether the first k entries output by a method include γ

$$\gamma_k = \begin{cases} 1, & \rho_\gamma \le k \\ 0, & \rho_\gamma > k \end{cases} \tag{9}$$

The accuracy at top k, i.e., $ACC@Top\ k$, is the probability that the top k entries output by a method contain the actual clue

$$ACC@Top\ k = \frac{\sum_{\gamma \in \Re} \gamma_k}{\|\Re\|} \tag{10}$$

Obviously, the objective of an anomaly localization method is to obtain a high accuracy when k is relatively small. We set k from 1 to 5 in our evaluation.



(a) (b)

Fig. 5. $ACC@Top\ 5$ and average running time under different w and λ.

In addition, we also use F-score to evaluate the accuracy of RobustSpot, i.e., $F-score = (2 \times Precision \times Recall)/(Precision + Recall)$, where $Precision = TP/(TP + FP)$ and $Recall = TP/(TP + FN)$. TP (true positive) is the number of cases where the clue is correctly reported, and FP (false positive) is the number of cases where the clue is wrongly reported. FN (false negative) is the number of cases where the clue is not reported. Here we only calculate the F-score of the best clue ($Top\ 1$). At the same time, we also use F-score to express the performance of Bagel, the anomaly detection algorithm used in this paper.

5.1.4 Implementation

We implement RobustSpot and baseline methods with Python, and run them on a server with 12 × Intel(R) Xeon (R) CPU E5-2650 v4 @ 2.20GHz and 128G RAM. To get readers more easily understand our work, we have made the data and code publicly available [8].

5.1.5 Parameter Tuning

As aforementioned, we tune τ_a automatically using the Knee-point method, and set $\tau_c = 0$ empirically. Additionally, we tune w and λ (see Section 4.1 and Section 4.3 for more details, respectively) as follows. As shown in Fig. 5, we calculate the accuracy (in terms of $ACC@Top\ 5$) and computational complexity (in terms of the average running time per case) of RobustSpot. As shown in Fig. 5a, we can see that the computational complexity is insensitive to the value of w. Moreover, RobustSpot's accuracy is also not very sensitive to w, and becomes stable when $w \geq 4$. Therefore, we set $w = 4$. Additionally, from Fig. 5b) we can find that when $\lambda = 100$, RobustSpot achieves the best accuracy, and becomes stable when $\lambda \geq 100$. In addition, the computational complexity increases as λ increases. Therefore, we set $\lambda = 100$.

5.2 Overall Accuracy (RQ1)

Due to the application of Bagel [13] for anomaly detection is not our main contribution, here we just present its performance simply. The precision and recall of Bagel are 0.97 and 0.95, respectively, with an F-score of 0.96, proving its superior effectiveness.

Table 9 lists the accuracy (in terms of $ACC@\ k$) of RobustSpot and five baseline methods on the 135 real-world cases. We can see that RobustSpot outperforms the five baseline methods in terms of $ACC@Top1$, $ACC@Top2$, $ACC@Top3$, $ACC@Top4$, and $ACC@Top5$. More specifically, its $ACC@Top5$ is 98%, which is 26% higher than the best baseline method. Similarly, its F-score is 0.8, which is 0.25 higher than the best baseline method.

TABLE 9
The $ACC@Top\ k$ of RobustSpot and Baseline Methods

Method	$Top\ 1$	$Top\ 2$	$Top\ 3$	$Top\ 4$	$Top\ 5$	F-score
RobustSpot	0.67	0.84	0.87	0.95	0.98	0.8
Adtributor	0.38	0.41	0.45	0.45	0.45	0.55
iDice	0.44	0.59	0.61	0.63	0.63	0.62
HotSpot	0.07	0.1	0.12	0.13	0.14	0.13
Squeeze	0.04	0.05	0.05	0.05	0.05	0.09
ImpAPTr	0.1	0.21	0.31	0.43	0.5	0.19

To better understand the results, we show the accuracy of different methods on different types of datasets in Fig. 6. We can see that RobustSpot achieves close-to-one $ACC@Top5$ when the clue contains only one combination. When there are two combinations in the actual clue, the $ACC@Top5$ is 88%, which is much higher than the best performed method, i.e., Adtributor, by 75%. Moreover, the scenario where there are two or more combinations in the clue does not occur often (5.93% in our scenario). Therefore, operators are satisfied with the accuracy of RobustSpot for the multi-clue scenario.

Both Adtributor and iDice achieve good performance when the clue has only one non-wildcard attribute value. However, Adtributor can only be applied for the scenario where the clue has a single non-wildcard attribute value (dimension), and cannot be used for multi-dimensional anomaly localization (Single-2/3). Moreover, neither iDice nor ImpAPTr can be applied to the scenario where there are two or more combinations serving as the clue (multi-clue), and neither of them performs as well as RobustSpot when there is only one combination in the clue. This is because to improve localization efficiency, iDice heavily relied on pruning strategies, which caused iDice to mistakenly consider too coarse-grained combinations as the clues. Similarly, ImpAPTr also depended on pruning strategies, which only considered how large the measurement value of a combination was, rather than how significantly it became anomalous. This causes ImpAPTr to generate much more false alarms than RobustSpot. Therefore, none of Adtributor, iDice, or ImpAPTr is applicable in our scenario. In addition, as aforementioned, both HotSpot and Squeeze work only when the data complies with the Ripple Effect rule, but the real-world data in our scenario does not comply with this rule. Consequently, neither HotSpot nor Squeeze performs well in our scenario, no matter how many non-wildcard attribute values in the single-clue, or how many combinations in the multi-clue.

5.3 Effect of AD, CA, and WARM (RQ2)

As aforementioned, we propose AD and CA to capture the complex pattern of derived measures. Moreover, we present a new method, WARM, to capture the hidden relationship across different attribute values and thus learn the propagation pattern from clue attribute values to other attribute values. AD, CA and WARM serve as the main technical contributions of our work. To evaluate the effect of the three main components, we evaluate the accuracy of RobustSpot

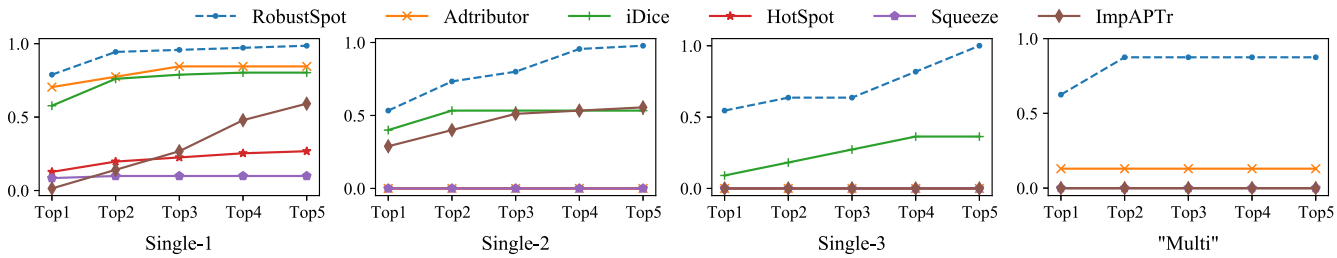


Fig. 6. The accuracy of RobustSpot and baseline methods on different types of datasets.

when each of them is removed from it. Specifically, we have three different versions of RobustSpot as follows.

- "*RobustSpot without AD*": We remove AD from RobustSpot, and only apply CA instead of the combination of AD and CA to obtain the preserved ε.
- " *RobustSpot without CA*": Similarly, we remove CA from RobustSpot, and only apply AD instead of the combination of AD and CA to obtain the preserved ε.
- "*RobustSpot without WARM*": As aforementioned, creating the weighted database $\mathbb{D}_p$ according to AD and CA as well as calculating the balanced support $Support_b$ are the two core ideas of WARM. Therefore, we evaluate RobustSpot's accuracy when the above two ideas are removed from it. More specifically, after calculating AD and CA, we can obtain the set of preserved leaf combinations (i.e., $\mathbb{P}$). Then we create a database constituted by these combinations, instead of weighting $\varepsilon \in \mathbb{P}$ based on their AD and CA. After that, we calculate the $Support_p$ of every combination in the database, and deduce the clue combinations based on their $Support_p$ instead of $Support_b$.

Fig. 7 shows the accuracy of the above three versions of RobustSpot as well as that of the whole RobustSpot. As aforementioned, if we ignore how prominently ε becomes anomalous, we may mistakenly retain normal εs (see Table 5 for more details). Therefore, *RobustSpot without* **AD** suffers from low accuracy. Similarly, the εs contributing little to the anomalous total SR may be preserved when we ignore CA (see Table 4 for more details), and thus *RobustSpot without* **CA** performs not well compared to the whole RobustSpot. In our scenario, for most cases, many leaf combinations suffer from anomalies when the total SR becomes anomalous. However, only a small number of these anomalous leaf combinations significantly contribute to the total anomalous SR (more details can be seen in Challenge 1 of Section 2.2). Therefore, when we remove CA from RobustSpot, a large number of false alarms will be generated, and thus the accuracy is prominently degraded.

Additionally, when we remove WARM from RobustSpot, its accuracy degrades dramatically. That is because WARM can not only capture the hidden relationship across different attributes (dimensions), but also take the distribution of different attribute values of each attribute into account. Without WARM, RobustSpot can mistakenly consider too fine-grained combinations as clues.

5.4 Computational Efficiency (RQ3)

Fig. 8 shows the CDF of RobustSpot's running time on the 135 anomalous total SR cases. We can see that more than 75% of these cases are localized within 2s. The maximum anomaly localization time is less than 12s. Specifically, the average anomaly localization time of RobustSpot is 1.83s. It is much faster than the manual way, which usually costs operators $1h+$ to localize the clue of an anomalous total SR case. Moreover, since these measurement records are collected every minute (i.e., the shortest time interval between two anomalies is one minute), the average localization time of 1.83s is quite satisfying in our scenario. Additionally, clue localization is triggered only when the total SR becomes anomalous, which occurs not often (less than ten times per day). Therefore, a server is enough, with RobustSpot, to localize the clues of anomalous total SR. Additionally, as listed in Table 10, the average anomaly localization time of HotSpot [5] and Squeeze [4] is slightly higher than that of RobustSpot. Although ImpAPTr [7], iDice [6], and Adtributor [3] achieve higher efficiency, they are not effective in our scenario as shown in Fig. 6.

6 Case Sutdy

RobustSpot has been implemented and deployed in a top-tier global online video service provider H, which provides services for 170+ million monthly active viewers, for more

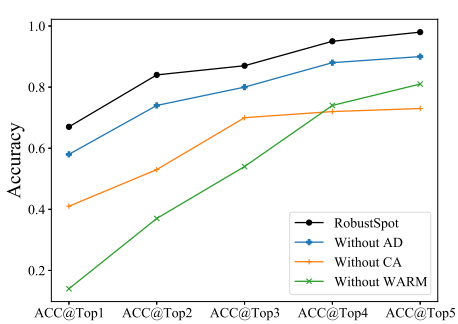


Fig. 7. The accuracy (measured by $ACC@K$ of different versions of RobustSpot).

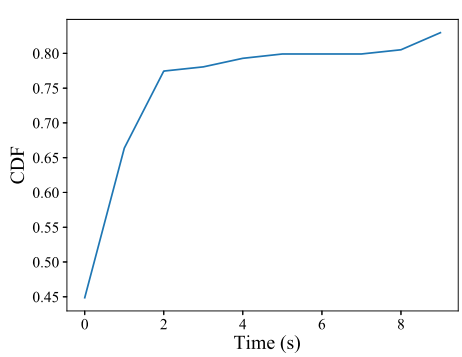


Fig. 8. The CDF of the running time of all 135 cases.

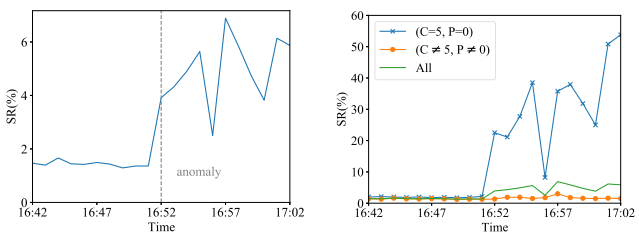
TABLE 10
The Average Anomaly Clue Localization Time Per Case for RobustSpot and Baseline Methods

Method	RobustSpot	Adtributor	iDice	HotSpot	Squeeze	ImpAPTr
Time (s)	1.83	0.09	0.08	2.30	2.11	0.07

than ten months. When the anomaly detection system reports an anomaly, RobustSpot will be triggered. The timestamp of the anomaly and the monitoring data near before the anomaly will be input into RobustSpot. The data has five dimensions: CDN(C), Bitrate (B), Device Type (D), P2P (P), ISP (I). For each anomaly case, RobustSpot reports the Top 5 candidate clues to operators. For the present, it has been used to handle more than 2700 anomalous total SR cases. Before the application of RobustSpot, operators have to use the graph visualization tools to manually identify possible clues. For each case, the typical time used for manual clue localization and confirmation is over 1 hour. RobustSpot, on average, generates clue recommendations in 1.83 seconds. According to the randomly selected 135 of the above 2700+ cases, we demonstrate that RobustSpot achieves 65%+ *ACC@Top* 1 and 90%+ *ACC@Top* 5 (see Section 5 for more details). Specifically, we present two successful and representative stories as follows.

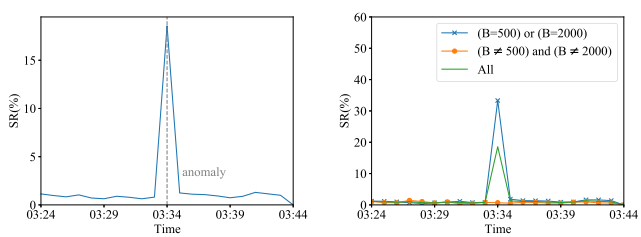
One Combination of Two Non-Wildcard Attribute Values is the Clue. As shown in Fig. 9a, the total SR suffered from a level shift (i.e., a sudden change in the monitored time series data) and the anomaly detection system reported an alarm at 16:52, 21 January, 2021, and this anomaly lasted 30+ minutes. This anomaly was then input into RobustSpot, which reported that the combination of $(C = 5, P = 0)$ was the clue in less than 1 s, i.e., the service quality of viewers with CDN=5 and P2P=0 was severely degraded, which led to the anomalous total SR. The result was confirmed by operators. This incident was induced by a misconfiguration of the P2P traffic, which was triggered by the service provider of CDN5. The service provider mitigated these misconfigurations half an hour later. Fig. 9b shows the total SR, the SR with $(C = 5, P = 0)$, and the SR with $(C \neq 5, P \neq 0)$, respectively. We can see that the fluctuation of the total SR is highly consistent with that of the SR with $(C = 5, P = 0)$, and the latter one is much more significant than the former one. In addition, in order to verify the effectiveness of CA and AD, we analyzed their values during the localization process of this case. Since it is impossible to display the intermediate experimental results of all εs, we calculated and displayed the average AD and CA values of the clue εs and the non-clue εs. The result shows that the average AD and CA of εs with $(C = 5, P = 0)$ are 0.86, 0.33, respectively. However, the average AD and CA of εs without $(C = 5, P = 0)$ are 0.13 and -0.005, significantly smaller than those of εs containing $(C = 5, P = 0)$, respectively. It can be seen that AD and CA well capture the anomalous εs and εs that contribute the most to the total anomaly.

Two Attribute Combinations are the Clue. Fig. 10a shows the second case. The total SR suffered from an abnormal spike at 3:34, 5 February, 2021, and returned to normal status 1 min later. Immediately after this anomaly was detected, RobustSpot reported that the clue was two combinations, i.e., $(B = 500)$ and $(B = 2000)$, respectively. This means the service quality of viewers with Bitrate=500 and those with Bitrate=2000 was severely degraded, which leads to the anomalous total SR. That is, as shown in Figs. 10c and 10d, both of the two combinations became prominently anomalous and contributed significantly to the anomaly of the total SR. The anomaly localization result was validated by operators, because the SR with $B \neq 500$ and $B \neq 2000$ did not experience any anomaly as shown in Fig. 10b. More specifically, a software update was deployed at 3:33, which triggered the reboot of a collection of servers providing services for the online videos with Bitrate=500 and those with Bitrate=2000. After these servers were rebooted and the software update was successfully deployed, the above online video services returned to normal. Because this software update was rolled

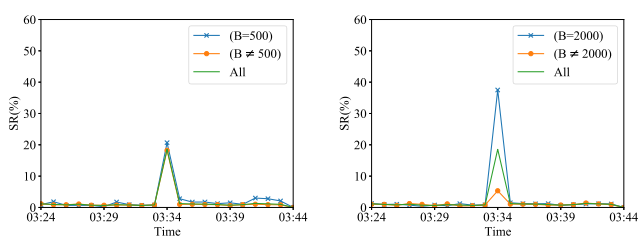


(a) The total SR became anomalous (b) The SR with (C= 5, P= 0) became anomalous, while the SR with $(C \neq 5, P \neq 0)$ were normal

Fig. 9. An anomalous total SR case where a combination of two non-wildcard attribute values was the clue.



(a) The total SR became anomalous (b) The SR with $B = 500$ or $B = 2000$ became anomalous, while the SR with $B \neq 500$ and $B \neq 2000$ were normal



(c) Both the SR with $B = 500$ and that with $B \neq 500$ became anomalous (d) Both the SR with $B = 2000$ and that with $B \neq 2000$ became anomalous

Fig. 10. An anomalous total SR case where two attribute combinations were the clue.

TABLE 11
Qualitative Comparison of Related Works

Algorithms	Multi-dimensional	Fundamental measure	Derived measure	Data distribution requirement
Adtributor	No	Yes	Yes	No
iDice	Yes	Yes	No	No
HotSpot	Yes	Yes	No	Ripple Effect
Squeeze	Yes	Yes	Yes	Ripple Effect
ImpAPTr	Yes	Yes	Yes	No
RobustSpot	**Yes**	**Yes**	**Yes**	**No**

out at midnight, the experience of only a small number of viewers was affected. The average AD and CA of εs with $(B = 500)$ or $(B = 2000)$ were 0.61, 1.75, respectively. Moreover, the average AD and CA of εs with $(B \neq 500 \& B \neq 2000)$ were 0.14 and -0.003, respectively, significantly smaller than those of εs with $(B = 500)$ or $(B = 2000)$.

7 Related Work

Recently, a large number of failure diagnosis works have been conducted by studying how anomalies propagate among *different* metrics, traces, or logs [18], [19], [20], [21], [22], [23], [24], [25], [26], [27]. In this work, however, we study how anomalies propagate among different attributes (dimensions) of the *same* multi-dimensional derived metric. Several similar works have been proposed in recent years, including Adtributor [3], iDice [6], HotSpot [5], Squeeze [4] and ImpAPTr [7]. Table 11 summarizes the characteristics of these works, including whether they can be applied for multi-dimensional measures, fundamental measures, and derived measures, and their requirement of data distribution. We then describe these works as follows.

Adtributor [3] assumed that all clues were one-dimensional, and used "Surprise" and "Explanatory power" to identify clues. However, the clues of many anomalous derived or fundamental measures were multi-dimensional, and thus Adtributor could not be applied in these scenarios. iDice [6] proposed "Isolation Power" to capture the pattern of clues based on information entropy. It applied a variety of pruning strategies to reduce complexity, which caused it to mistakenly consider too coarse-grained combinations as clues.

HotSpot [5] applied a reinforcement leaning method to infer the clues of anomalous *additive fundamental* measures. After that, Squeeze [4] was proposed to improve HotSpot to be fit for derived measures. Both HotSpot and Squeeze believed that all the combinations complied with the Ripple Effect. However, as aforementioned, some scenarios, including ours, do not satisfy the Ripple Effect. Therefore, they could not be applied to address the anomaly localization problem for online video services.

ImpAPTr [7] was another anomaly localization method for multi-dimensional derived measures. It was designed for the cases where there was only one combination in the clue, and could not be applied for those cases where two or more combinations simultaneously contribute to the anomalous total derived measures. Additionally, the pruning strategies used in ImpAPTr made it likely to generate too coarse-grained combinations as clues.

8 Conclusion

Anomaly localization of multi-dimensional derived measure is of vital importance for online video services. In this work, we propose RobustSpot, a robust anomaly localization framework for multi-dimensional derived measure. It integrates two new indicators, AD and CA, to accurately capture the complex patterns of derived measures, with a simple yet effective method, WARM, which robustly captures the hidden relationship and learns the propagation patterns across different dimensions, and balances the biased distribution of fundamental measures. Extensive experiments using 135 real-world cases demonstrate that RobustSpot achieves 98% accuracy (in terms of $ACC@Top5$), significantly outperforming baseline methods. RobustSpot has deployed in a top-tier global online video service provider, which provides services for 170+ million monthly active users.

Theoretically, RobustSpot can be applied to any derived measure with multi-dimensional attributes. Therefore, as long as a Web service pays attention to derived measures with multi-dimensional attributes, RobustSpot can be applied for its anomaly clue localization. Additionally, the baseline methods, e.g., iDice, HotSpot, Squeeze, ImpAPTr, and Adtributor, have demonstrated the importance and feasibility of robust anomaly clue localization for other Web services, including online office, advertising, search engine, online shopping, online banking, etc. We believe that RobustSpot can also be applied to these Web services.

References

[1] M. Seufert, P. Casas, N. Wehner, L. Gang and K. Li, "Features that matter: Feature selection for on-line stalling prediction in encrypted video streaming," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2019, pp. 688–695.

[2] S. Wassermann, N. Wehner, and P. Casas, "Machine learning models for youtube qoe and user engagement prediction in smartphones," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 46, no. 3, pp. 155–158, 2019.

[3] R. Bhagwan *et al.*, "Adtributor: Revenue debugging in advertising systems," in *Proc. 11th USENIX Symp. Netw. Sys. Des. Implementation*, 2014, pp. 43–55.

[4] Z. Li *et al.*, "Generic and robust localization of multi-dimensional root causes," in *Proc. IEEE 30th Int. Symp. Softw. Rel. Eng.*, 2019, pp. 47–57.

[5] Y. Sun *et al.*, "HotSpot: Anomaly localization for additive KPIs with multi-dimensional attributes," *IEEE Access*, vol. 6, pp. 10 909–10 923, 2018.

[6] Q. Lin *et al.*, "iDice: Problem identification for emerging issues," in *Proc. 38th Int. Conf. Softw. Eng.*, 2016, pp. 214–224.

[7] G. Rong *et al.*, "Locating the clues of declining success rate of service calls," in *Proc. IEEE 31st Int. Symp. Softw. Rel. Eng.*, 2020, pp. 335–345.

[8] Code and Data, 2022. [Online]. Available: https://github.com/robustspotproject/RobustSpot

[9] L. Liao *et al.*, "Locating performance regression root causes in the field operations of web-based systems: An experience report," *IEEE Trans. Softw. Eng.*, to be published, doi: 10.1109/TSE.2021.3131529.

[10] D. Liu *et al.*, "Opprentice: Towards practical and automatic anomaly detection through machine learning," in *Proc. Internet Meas. Conf.*, 2015, pp. 211–224.

[11] H. Xu *et al.*, "Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications," in *Proc. World Wide Web Conf. World Wide Web*, 2018, pp. 187–196.

[12] J. Bu *et al.*, "Rapid deployment of anomaly detection models for large number of emerging KPI streams," in *Proc. IEEE 37th Int. Perform. Comput. Commun. Conf.*, 2018, pp. 1–8.

[13] Z. Li, W. Chen, and D. Pei, "Robust and unsupervised KPI anomaly detection based on conditional variational autoencoder," in *Proc. IEEE 37th Int. Perform. Comput. Commun. Conf.*, 2018, pp. 1–9.

[14] W. Chen *et al.*, "Unsupervised anomaly detection for intricate KPIs via adversarial training of VAE," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1891–1899.
[15] X. Zhang *et al.*, "Cross-dataset time series anomaly detection for cloud systems," in *Proc. USENIX Annu. Tech. Conf.*, 2019, pp. 1063–1076.
[16] V. Satopaa, J. Albrecht, D. Irwin and B. Raghavan, "Finding a kneedle in a haystack: Detecting knee points in system behavior," in *Proc. Int. Conf. Distrib. Comput. Syst. Workshops*, 2011, pp. 166–171.
[17] J. Han, J. Pei, and M. Kamber, *Data Mining: Concepts and Techniques*, Amsterdam, The Netherlands: Elsevier, 2011.
[18] J. Weng, J. H. Wang, J. Yang, and Y. Yang, "Root cause analysis of anomalies of multitier services in public clouds," *IEEE/ACM Trans. Netw.*, vol. 26, no. 4, pp. 1646–1659, Aug. 2018.
[19] Y. Gan *et al.*, "Seer: Leveraging big data to navigate the complexity of performance debugging in cloud microservices," in *Proc. 24th Int. Conf. Architect. Support Program. Lang. Oper. Syst.*, 2019, pp. 19–33.
[20] X. Zhou *et al.*, "Latent error prediction and fault localization for microservice applications by learning from system trace logs," in *Proc. 27th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, 2019, pp. 683–694.
[21] X. Guo *et al.*, "Graph-based trace analysis for microservice architecture understanding and problem diagnosis," in *Proc. 28th ACM Joint Meeting Eur. Softw. Eng. Conf. Symp. Found. Softw. Eng.*, 2020, pp. 1387–1397.
[22] J. Lin, P. Chen, and Z. Zheng, "Microscope: Pinpoint performance issues with causal graphs in micro-service environments," in *Proc. Int. Conf. Serv.-Oriented Comput.*, 2018, pp. 3–20.
[23] M. Ma *et al.*, "AutoMAP: Diagnose your microservice-based web applications automatically," in *Proc. Web Conf.*, 2020, pp. 246–258.
[24] X. Zhou *et al.*, "Fault analysis and debugging of microservice systems: Industrial survey, benchmark system, and empirical study," *IEEE Trans. Softw. Eng.*, vol. 47, no. 2, pp. 243–260, Feb. 2021.
[25] L. Wang, N. Zhao, J. Chen, P. Li, W. Zhang and K. Sui, "Root-cause metric location for microservice systems via log anomaly detection," in *Proc. IEEE Int. Conf. Web Serv.*, 2020, pp. 142–150.
[26] A. R. Chen, "An empirical study on leveraging logs for debugging production failures," in *Proc. IEEE/ACM 41st Int. Conf. Softw. Eng., Companion Proc.*, 2019, pp. 126–128.
[27] T. Jia, P. Chen, L. Yang, Y. Li, F. Meng and J. Xu, "An approach for anomaly diagnosis based on hybrid graph model with logs for distributed services," in *Proc. IEEE Int. Conf. Web Serv.*, 2017, pp. 25–32.

Yongqian Sun (Member, IEEE) received the BS degree in statistical specialty from Northwestern Polytechnical University, Xi'an, China, in 2012, and PhD degree in computer science from Tsinghua University, Beijing, China, in 2018. He is currently an assistant professor with the College of Software, Nankai University, Tianjin, China. His research interests include anomaly detection and root cause localization in service management.

Daguo Cheng received a BS degree in information security from the School of Computer Science and Cyberspace Security, Hainan University, Haikou, China, in 2019. He is currently working toward the MS degree with the College of Software, Nankai University, Tianjin, China. His research interests include anomaly detection and root cause localization.

Pengxiang Jin received the bachelor's degree in software engineering from Nankai University, Tianjin, China, in 2020. He is currently working toward the master's degree with the College of Software, Nankai University. His research interests include anomaly detection and anomaly localization.

Quan Ding received the BS degree in software engineering from the School of Software, Nankai University, Tianjin, China, in 2020. He is currently working toward the master's degree with the Institute of Computing Technology, University of Chinese Academy of Sciences. His research interests include anomaly detection and summarization on time series data.

Shenglin Zhang (Member, IEEE) received the BS degree in network engineering from the School of Computer Science and Technology, Xidian University, Xi'an, China, in 2012 and the PhD degree in computer science from Tsinghua University, Beijing, China, in 2017. He is currently an associate professor with the College of Software, Nankai University, Tianjin, China. His current research interests include failure detection, diagnosis and prediction for service management.

Xu Chen received the bachelor's degree in software engineering from Nankai University, Tianjin, China, in 2020. His research interests lie in data science and computer graphics with particular on physically-based simulation.

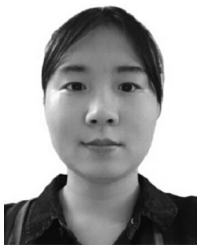
Yuzhi Zhang received the BS and MS degrees in computer science from the Department of Computer Science and Technology, Tsinghua University in 1985 and 1987, respectively, and the PhD degree in computer science from the Institute of Computing Technology, Chinese Academy of Sciences in 1991. He is currently dean with the College of Software, Nankai University, and is also a distinguished professor. His research interests include deep learning and other aspects in artificial intelligence.

Minghan Liang is currently a senior undergraduate with the College of Software at Nankai University, Tianjin, China. His research interests include data science, anomaly detection, and root cause localization.

Dan Pei (Senior Member, IEEE) received the BE and MS degrees in computer science from the Department of Computer Science and Technology, Tsinghua University in 1997 and 2000, respectively, and the PhD degree in computer science from the Computer Science Department, University of California, Los Angeles (UCLA) in 2005. He is currently an associate professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include network and service management in general. He is an ACM senior member.

Jianyan Zheng received the BS degree in electronic information science and technology from Shandong Agricultural University in 2014 and the MS degree in signal and information processing from the School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China, in 2017. He is currently an engineer with HUYA, Inc.

Sen Luo received the BA degree in financial management from the School of management, Guangzhou College of South China University of Technology, Guangzhou, China, in 2016. He is currently an engineer with HUYA, Inc.



Xinyu Tang received the BS degree in mathematics from Nanchang University, Nanchang, China, in 2016, and the MS degree in mathematics from Xi'an Jiaotong University, Xi'an, China, in 2019. Her research interests mainly include data mining and anomaly detection. She is currently an engineer with HUYA, Inc.