PRiskLoc: An Enhanced Multi-dimensional Root Cause Localization Algorithm Aided by a Fine-Grained Evaluation System

Tian Ding[1,2(✉)] and Yanli Wang[1,2]

[1] Intelligence Service Lab, Samsung Electronics (China) R&D Center, Yuhuatai District, Nanjing, Jiangsu, China
tian123.ding@samsung.com
[2] R&D Center, Yuhuatai District, Nanjing, Jiangsu, China

Abstract. Root cause localization within multi-dimensions is a challenging task due to its large search space within a limited time. There are a series of algorithms to handle this task, but to our knowledge, there is no evaluation system to help users analyse or optimize them according to their specific data and needs. In this paper, there are two main contributions: first, we provide a multi-dimensional evaluation system to evaluate the performance of algorithms in full aspects, which can help us comprehensively and finely analyse, compare, and choose the applicable scenario of algorithms or optimize targeted algorithms; second, we analyse and find the weakness of the SoTA algorithm RiskLoc based on our contributed evaluation system, aiming at its weakness. To tackle the issue of RiskLoc found by our evaluation system, we present PRiskLoc, an efficient and effective multi-dimensional root cause localization algorithm. We demonstrate that PRiskLoc consistently outperforms state-of-the-art baselines, especially in more challenging root cause scenarios, with the F1 improved from 0.635049 to 0.724687.

Keywords: Multi-dimensional Root Cause · Potential Score · Anomaly Detection

1 Introduction

In the process of system operation and maintenance, abnormal changes in key indicators often mean service abnormalities, system failures and so on. Therefore, we need to check key indicators frequently [1–3]. In practice, anomaly detection models are applied to these collected time series and quickly positioning abnormalities.

However, various measures with many attributes are accompanied by a huge search space, which means that this is a very challenging task. Table 1 shows an example with two attributes where the aggregated measure (total) is abnormal with the root cause being $\{(\text{BE}, *)\}$, where $*$ indicates an aggregated attribute. Due to the nature of the problem, the root cause is a set of elements with different levels of aggregation. The main challenge is the huge search space since we need to consider all possible combinations of any number of attribute values. For a measure with dimensions each with n values, the number of

M. Zhang et al. (Eds.): CCF NCCA 2023, CCIS 1959, pp. 3–18, 2024.
https://doi.org/10.1007/978-981-99-8764-1_1

valid elements is $\sum_{i=1}^{d} \binom{d}{i} n^i = (n+1)^d - 1$, which gives $2^{(n+1)^d - 1} - 1$ number of possible combinations.

Recent works have made great efforts to rapidly search for multi-dimensional root causes [4–7]. However, most of them only work for specific application scenarios. Adtributor [4] is limited to identifying root causes in one dimension. Squeeze [5] and AutoRoot [6] are only applicable in a scenario with a relatively large difference in abnormal amplitude and are sensitive to the clustering outcome. RiskLoc [7] has almost no restrictions on the applicable scenario and is significantly better than the previous algorithm in terms of effectivity and efficiency.

The evaluation of these algorithms is based on individual datasets, so the results presented are not sufficiently convincing, and there is a lack of uniform standards for the evaluation of algorithms. To address this problem, we proposed an algorithm evaluation system that can assist us with algorithm evaluation, selection and optimization. With the help of our evaluation system, we analysed how the algorithms perform in various scenarios and found that RiskLoc performs best in complex scenarios but not well when the magnitude of the anomaly is not obvious. To address this issue, we propose PRiskLoc, which can significantly improve the performance of RiskLoc when the abnormality is not significant, which is a universally difficult topic in current SOTA algorithms. For the improved PRickLoc, we also use the evaluation system to demonstrate its effectiveness.

Table 1. Example of a multi-d measure with two attributes.

Country	Device Series	Actual	Forecast
BE	G71	1500	50
BE	G72	900	100
BR	G70	2000	1990
BR	G72	1005	1000
BR	G71	2005	2000
Total		7410	5140

Our main contributions are as follows:

- We propose an evaluation system that allows for a comprehensive and efficient comparison of algorithms.
- Based on the evaluation system we constructed, we propose PRiskLoc, which can compensate for the lack of RiskLoc. We propose an augmented density-based partitioning approach that compensates for the inadequacy of RiskLoc in cases where the magnitude of the anomaly is not obvious.
- We design the experiment and complete the effectiveness validation of PRiskLoc based on the evaluation system we constructed.

2 Background

In this section, we begin by defining the problem, the related notation, and terminology. Then, we briefly introduce the ripple effect [9] and grounded theory of root cause algorithms.

2.1 Definition

Many time-series indicators can be broken down into a number of dimensions, each with a different value domain, and when anomalies occur, each dimension can be a potential root cause.

More formally, for a measure, we assume that it has n dimensions of attributes $(A_1, A_2, \ldots, A_n)$, and each dimension has values $(V_1, V_2, \ldots, V_n)$. We can obtain different subsequences depending on the degree of decomposition, which form sets of elements called cuboids. For the most fine-grained scenarios $\left(A_{1,2,\ldots,n}\right)$, we can obtain $V_1 * V_2 * \cdots * V_n$ measures, which we call leaf elements, and for the coarsest-grained decomposition $(A_1, *) \cup (A_2, *) \cup \cdots \cup (A_n, *)$, we can obtain $V_1 + V_2 + \cdots + V_n$ sub measures. For example, the scene of selling cell phones has the following dimensions: country, device and quarter. Each dimension has its own attributes, for example, {Country1, Country2, Country3}, {Device1, Device2, Device3}, {Quarter1, Quarter2, Quarter3, Quarter4}. We can analyse sales from country, device, quarter, respectively, or combine their attributes, such as (Country, Device), or more fine-grained (Country, Device, Quarter) (Fig. 1).

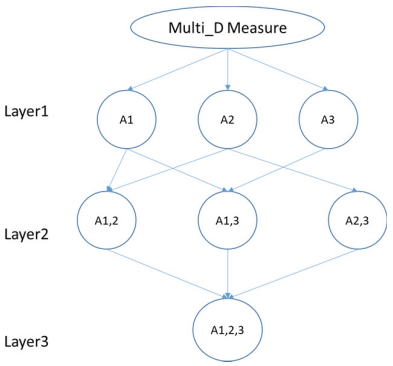

Fig. 1. Cuboid relation graph with n = 3

2.2 Deviation Score

The deviation score is presented by Li [5], and is derived from the ripple effect [9]. The deviation score of element e is defined as:

$$ds(e) = 2 \cdot \frac{f(e) - v(e)}{f(e) + v(e)} \tag{1}$$

$f(e)$ is the predicted value, and $v(e)$ is the true value. It shows the quantitative relationship between elements caused by the same cause. If A_1 is a root cause, leaf elements that are inherited from A_1 necessarily contribute together to the anomalous change, and the proportion of the change in the root cause at this time will be assigned to all its leaf elements in proportion to the predicted value, which means that leaf elements in the same root cause have similar deviation scores. Generally, if the prediction algorithm is accurate enough, their forecast residuals are small, which means that leaf elements with small deviation scores can be considered normal. In the case of abnormalities, the difference between the predicted and true values is relatively large, and the deviation score will be far from zero. Then, at this point, the problem is how to find combinations inside the huge search space to satisfy leaf elements with similar ds values, and these combinations are the root cause of the anomaly. It is noteworthy that the targeted combinations should satisfy two requirements: 1) the leaf elements of the targeted combination satisfy the ripple effect, and 2) most of the elements in the targeted combination are anomalous.

2.3 Related Work

In the following, we will present some algorithms based on the ripple effect.

- HotSpot

HotSpot [9] proposes a metric called PS (Potential Score) to measure how well all leaf elements of a given combination follow the ripple effect. PS is actually a combined attribute, which means that it can be compared between all element combinations. Facing such a huge search space, HotSpot adopts MCTS to optimize the search. From layer 1 to layer L (L is the number of layers), the root cause set RSet (ps(RSet) $>$ PT) is obtained, where PT (ps Threshold) is a threshold that we think is large enough to be regarded, and finally, we obtain the combination with the highest PS among all cuboids.

HotSpot assumes that all root causes are in a cuboid and HotSpot is not available for the derived measure. At the same time, the accuracy of the results is greatly influenced by the number of MCTS iterations; the more iterations there are, the more accurate the results, but the corresponding time cost will be longer.

- Squeeze

Unlike the MCTS used by HotSpot, the first step of Squeeze [5] is filtering to reduce the search space. Squeeze includes two major parts: 1) from bottom to top, filtering most of the normal data, and the abnormal data clustered based on ds values; 2) cluster internal location from top to bottom. Similar to HotSpot's PS, Squeeze proposes GPS as a quantitative criterion for root cause.

Squeeze extends the theory based on HotSpot to increase the generality and robustness of the algorithm. However, actual business scenarios may have some impact on the accuracy of the algorithm: 1) for different data distributions, the filtering algorithm is not always effective, which could lead to errors in the later analysis; 2) it is highly dependent on the accuracy of the cluster; and 3) it cannot be used when there are multiple anomalies with similar ds.

- AutoRoot

Similar to Squeeze, AutoRoot [6] can be divided into filtering, aggregation, and root cause search within the cluster, and the different points from Squeeze are mainly as follows: 1) empirical values are used when filtering normal data; 2) the measurement of root cause search within the cluster is no longer GPS but RS instead, which is more comprehensive.

AutoRoot has a large improvement over Squeeze in terms of performance and efficiency, but it is still unable to avoid the drawbacks of clustering: 1) reliance on accurate clustering and 2) inability to be used when there are multiple anomalies with similar ds.

- RiskLoc

Abandoning the idea of cluster, RiskLoc [7] presents a new idea that employs three main components in its search for the root cause set: 1) leaf element partitioning and weighting; 2) risk score; and 3) element search and iteration. Specifically, RiskLoc separates leaf elements into a normal and an abnormal set with a simple 2-way partitioning scheme at the first step, and each leaf element is given a weight corresponding to the distance from the partitioning point. In this way, RiskLoc believes that it can mitigate the effects of incorrect partitioning. Then, RiskLoc proposes a risk score to identify potential root cause elements in each cuboid. Finally, RiskLoc searches from the lower layer to the higher layer, and when a combination satisfies the condition, its leaf elements are deleted, and the iteration is repeated until none of the remaining elements can satisfy the condition. Since iterations rather than ds-based clustering are used, RiskLoc can handle multiple anomalies with ds-like conditions very well.

RiskLoc [7] has almost no restrictions on the applicable scenario and is significantly better than the previous algorithm in terms of effectivity and efficiency. However, the accuracy of the partitioning in the first step has a large impact on the results, which leads to limitations in use.

3 Evaluation

3.1 Evaluation System

Multi-dimensional root cause localization is a complex problem, and there are various algorithms. However, algorithms are often validated on specific datasets, and these datasets lack specific descriptions, which presents a huge challenge to choose according to your needs or targeted optimization. Therefore, a comprehensive evaluation system is necessary.

As stated earlier, the deviation score (ds) is the cornerstone of multi-dimensional root cause localization algorithms. By analysing the false cases of various algorithms, we propose an evaluation system with four main factors that affect ds:

- The abnormal amplitude (the smaller the amplitude, the harder to locate);
- The layer of anomalies (the higher the layer, the harder to locate);
- The amplitude between different anomalies (the more similar, the harder to locate);
- The anomaly number (the more, the harder to locate).

We construct fine-grained datasets considering these factors. Meanwhile, we verify that the evaluation system is effective.

3.2 Dataset Generation

The existing public datasets have insufficient information about anomaly injection, and a single anomaly case cannot provide a comprehensive algorithm performance comparison. To generate datasets for different scenarios, we employ an approach to generate datasets [7]. Each element has only a single actual value $v(e)$ and a single forecast value f (e). Actual values $v(e)$ are sampled from a one-parameter Weibull distribution with $\alpha \sim U$ [0.5, 1.0], where U is a uniform distribution. Actually, the accuracy of the prediction algorithm has a relatively large impact on the results. To reduce the interference term, we assume that the prediction is accurate and simulate it by adding prediction residuals to the true value:

$$f(e) = v(e) \times N(1, \sigma) \tag{2}$$

When anomaly injection occurs, there are several steps: 1) select the anomaly layer; 2) select the anomaly combinations; 3) set the anomaly magnitude; and 4) change the actual value of the target combination according to the set anomaly magnitude. To better simulate reality, we make the magnitude change obey $N(s, d)$, where s means anomaly severity and d means anomaly deviation. All elements in a single anomaly are scaled the same (i.e., following the ripple effect [9]) with:

$$x = max(x * (1 - N(s, d)), 0) \tag{3}$$

where $x = v(e)$, if $\sum v(e) > \sum f(e)$; otherwise, $x = f(e)$, which ensures the balance of the anomaly direction. We can generate datasets with different magnitudes by changing the value of s, and a smaller s means that the anomaly magnitude is less obvious and hard to locate.

Based on our evaluation system, combined with the data synthesis approach described above, we construct multi-dimensional datasets. Dataset S is provided by RiskLoc, while S1 and S2 are the comprehensive datasets constructed by us. Dataset L is a combination of different anomaly layers. The difference between datasets F and L is that F has only one anomaly, while L has 1–3 anomalies. Dataset D is a set of datasets with different anomaly magnitudes. The details of each generated dataset can be found in Appendix A.

3.3 Evaluation Metrics

We assess the effectiveness of the methods using the F1-score. TP means true positive, FP means false positive and FN means false negative.

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{4}$$

3.4 Results and Discussion

Based on our evaluation system and the corresponding fine-grained datasets, we performed the following experiments. Due to the long execution time of HotSpot, we only compare Squeeze, AutoRoot and RiskLoc in this paper. The comprehensive comparison of algorithms is displayed in Fig. 2.RiskLoc and AutoRoot perform far better than Squeeze, and RiskLoc is undoubtedly the best. In the following, we will compare each algorithm in terms of each dimension of the evaluation system.

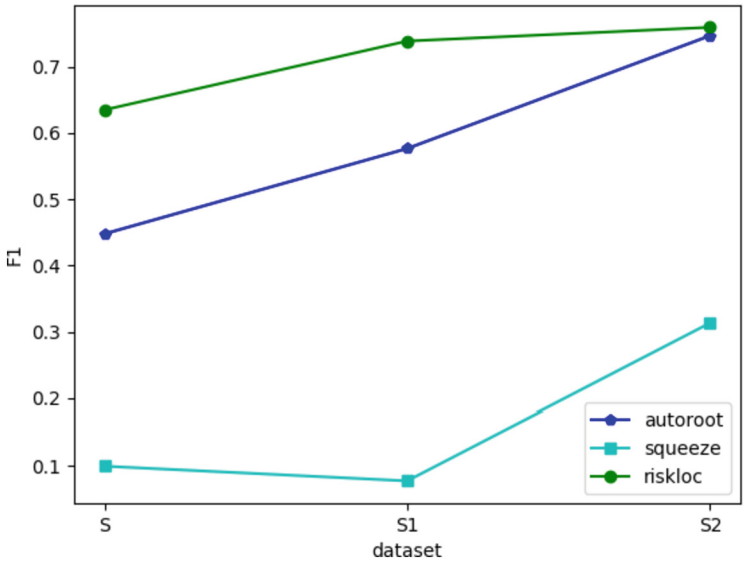

Fig. 2. F1 of Comprehensive datasets

Anomaly Amplitude. Dataset D is employed to complete this task, and D1 to D9 indicate that the anomaly severity in Eq. (3) changes from 0.1 to 0.9, which indicates that the anomaly is becoming increasingly obvious. When the abnormal amplitude is small, RiskLoc does not perform as well as AutoRoot (Fig. 3).

Anomaly Layer. Datasets F and L are both for layer level comparison, F1 or L1 means the anomaly is in the first layer, while F2 or L2 means the anomaly is in the second layer. From Fig. 4 and Fig. 5, we found that the deeper layer the anomaly in, the worse the performance.

Amplitude Between Different Anomalies. In this dimension, we can see the different performances of S1 and S2 (Fig. 2), while the only difference between S1 and S2 is that S1 has similar anomaly magnitudes and S2 has different anomaly magnitudes. The results show that RiskLoc performs better when the magnitudes of the anomalies are similar.

Anomaly Number. The difference between datasets L and F is that there are multiple anomalies in L, while there is only one anomaly in F. A comparison of Fig. 4 and Fig. 5

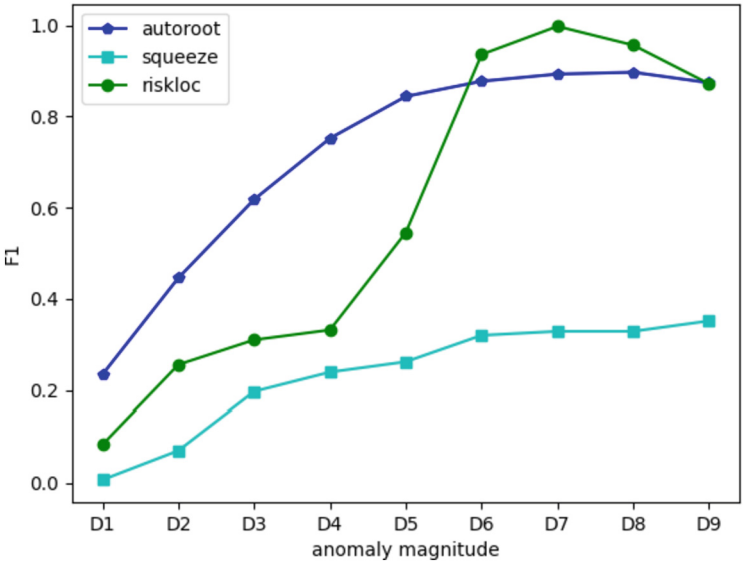

Fig. 3. F1 of D

shows the performance of different algorithms in the dimension of anomaly number, and RiskLoc has advantages when the anomalies are more complex.

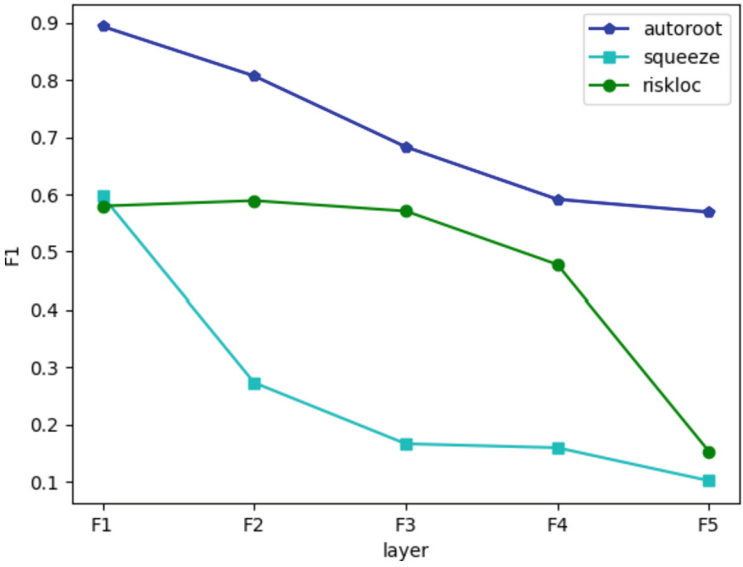

Fig. 4. F1 of D

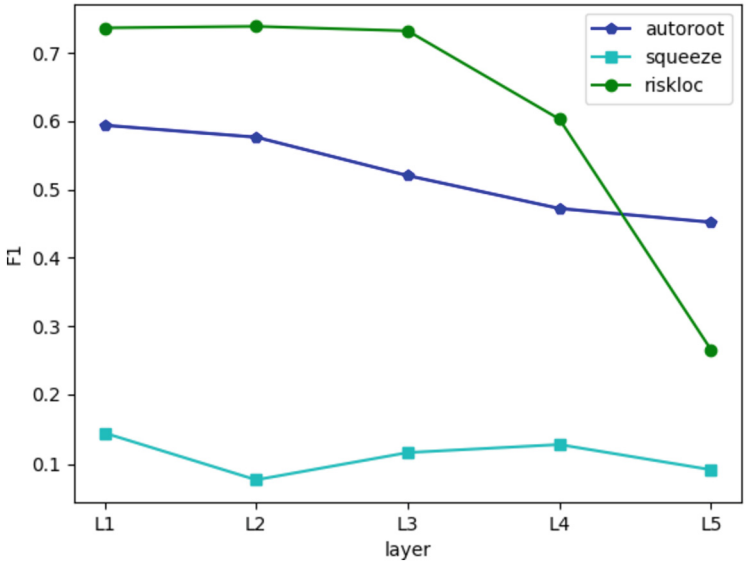

Fig. 5. F1 of D

From the results, the performance of algorithms is influenced by abnormal amplitude and abnormal layer, and there is no doubt that the less obvious the abnormal magnitude is, the more difficult it is to locate, and the higher the abnormal layer is, the harder it is to locate. RiskLoc and AutoRoot are actually complementary, and RiskLoc is good at handling scenarios with multiple anomalies of similar magnitude, while AutoRoot is skilled in handling cases where the anomaly magnitude is not obvious or there is only one anomaly. However, on the comprehensive datasets, RiskLoc consistently outperforms other algorithms, which also shows that further optimization of RiskLoc is necessary. It is worth noting that Squeeze performs poorly and is very sensitive to changes in the number of anomalies.

4 PRiskloc

RiskLoc is significantly better than the other algorithms but not well when the magnitude of the anomaly is not obvious. In this section, we will analyse the inefficiency of RiskLoc and propose an optimization plan to obtain PRiskLoc, and our evaluation system verifies the effectiveness of PRiskLoc.

4.1 Inefficiency of 2-Way Partitioning Scheme

In the following, we have performed an experiment on the datasets S and L presented by RiskLoc, where we partitioned the data according to the actual situation before the root cause search, i.e., the partitioning accuracy was 100%, and then the results were

Table 2. F1 of Datasets S and L with RiskLoc

Dataset	S	L
No partitioned	0.676724	0.6767
Partitioned	0.808694	0.757839

compared with no advanced partitioning. Table 2 shows that the accuracy of the first step of partitioning has a significant impact on the overall result.

With the help of self-constructed fine-grained datasets, we found that the abnormal amplitude is an important factor. The basis of the 2-way partitioning approach is that the normal element ds is smaller and the abnormal element ds is larger. When some abnormal elements have $|ds| <$ cutpoint (meaning the abnormal is not significant), these points will be mistakenly divided into the normal set, as shown in Fig. 6. Although RiskLoc uses weight to dilute the impact of partition errors, the effect is not obvious, which will also influence subsequent abnormal localization.

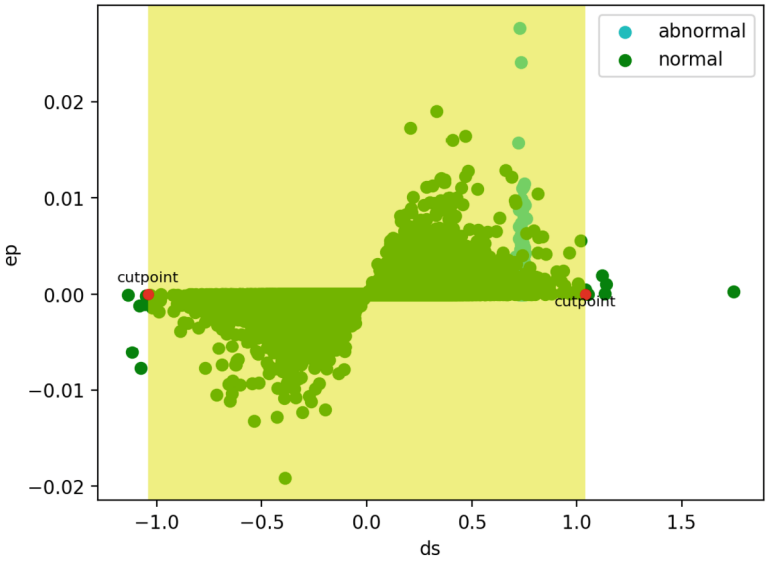

Fig. 6. Partition by Cutpoint

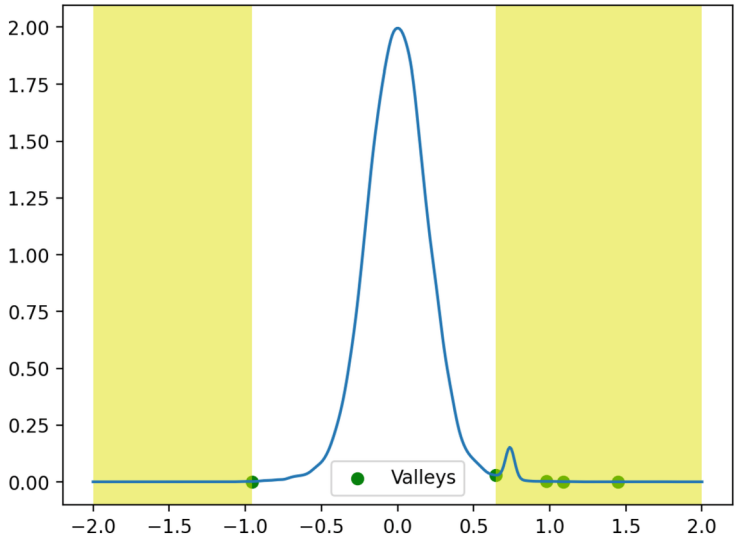

Fig. 7. Gaussian KDE of ds

4.2 Partition by KDE

We apply simple but effective kernel density estimation (KDE) [10] with a Gaussian kernel to obtain the distribution density of the deviation score of the normal part partitioned by RiskLoc, and Fig. 7 shows the KDE plot. After clustering, the relative maximum values are the centers of each cluster, and the nearby relative minimum values are the clusters' boundaries. Thus, we can obtain different clusters. Based on the heuristics that anomalies always hold a small part, except for the data in the largest cluster, we put the other data to the abnormal part. The overall framework refers to Fig. 8.

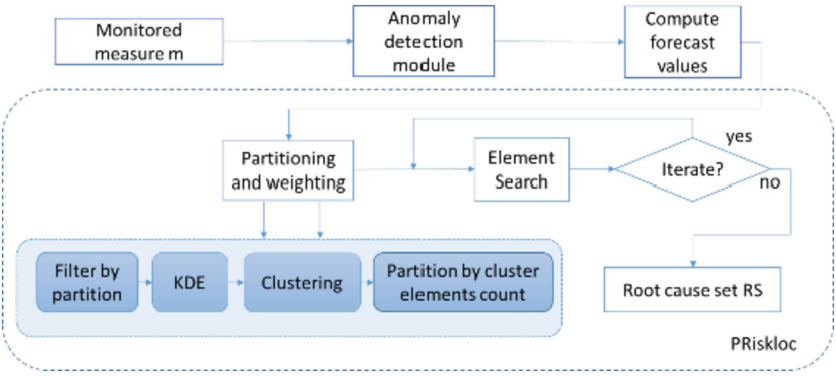

Fig. 8. System framework of PRiskLoc

4.3 Evaluation of PRiskLoc

We also use the evaluation system in part 3 to evaluate PRiskLoc. As shown in Fig. 10, the improvement of PRiskLoc is very obvious when the abnormal severity >0.2, and from the perspective of layer, the lower layer, the more obvious the effect (see Fig. 11 and Fig. 12). Figure 9 shows that PRiskLoc consistently outperforms RiskLoc on comprehensive datasets, which proves that the insufficiency of RiskLoc has been greatly improved by PRiskLoc (the detailed results can be found in Appendix B). It is worth noting that PRiskLoc did not achieve the expected effect for anomaly severity < 0.2 and layer > 3 because when layer > 3, the leaf elements of abnormal are less, and the abnormal element cannot be found during the KDE analysis, and when the anomaly severity < 0.2, the density peaks formed by the anomalous data are superimposed on the normal data, resulting in the inability to form multiple peaks.

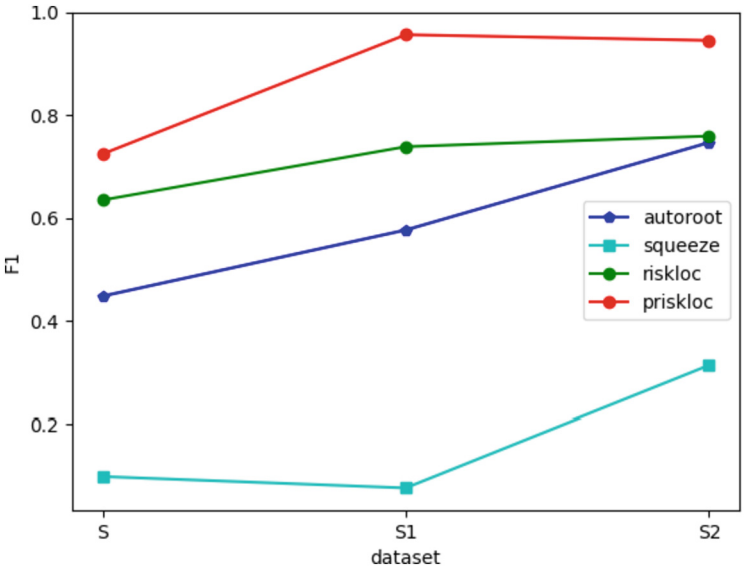

Fig. 9. F1 of Comprehensive dataset

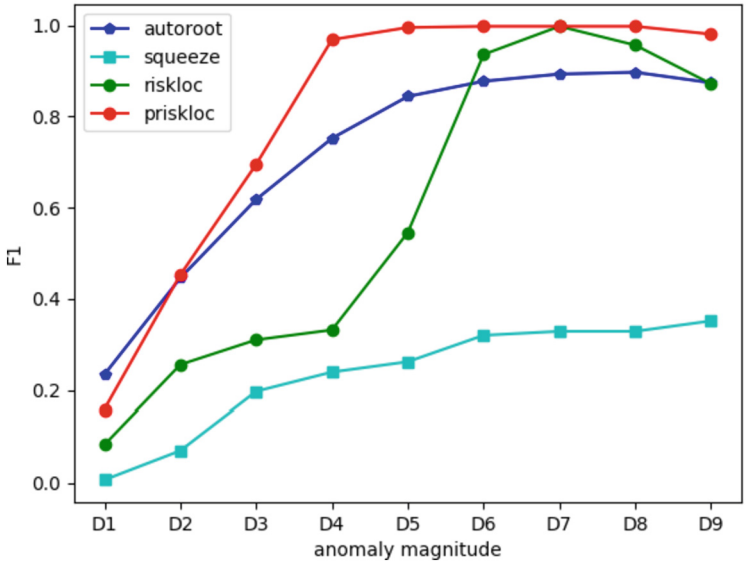

Fig. 10. F1 of D

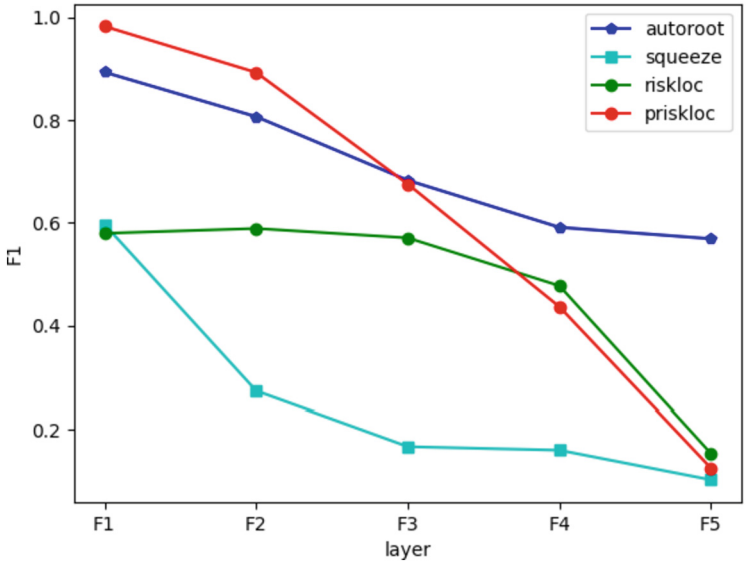

Fig. 11. F1 of F

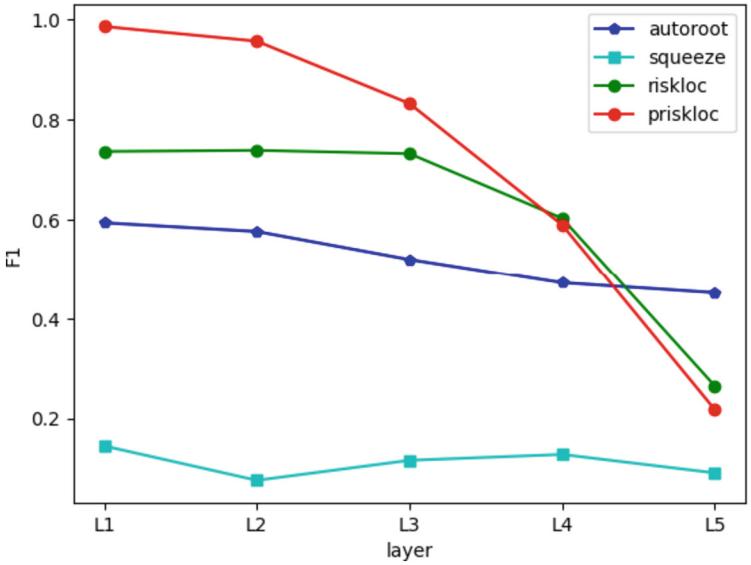

Fig. 12. F1 of L

5 Conclusion

In this paper, we provide a multi-dimensional evaluation system to evaluate the performance of algorithms in full aspects, which can help us comprehensively and finely analyse, compare, and choose the applicable scenario of algorithms or optimize targeted algorithms. With the help of our evaluation system, we found the weakness of the SoTA algorithm RiskLoc. Then, we present PRiskLoc, which consistently outperforms state-of-the-art baselines. In particular, in more challenging root cause scenarios, the F1 improved from 0.635049 to 0.724687.

A Dataset Details

See Table 3.

Table 3. Dataset Details

Dataset	N	D	Elements	Anomaly num	Anomaly layer	Anomaly severity	Amplitude similar between different abnormalities
S	1000	5	480000	[1, 9]	[1, 5]	[0.2,1.0]	Both
S1	1000	5	480000	[1, 3]	2	[0.2,0.9]	Yes
S2							No
L	1000	4	36000	[1, 5]	4	[0.5,1.0]	No

(*continued*)

Table 3. (*continued*)

Dataset	N	D	Elements	Anomaly num	Anomaly layer	Anomaly severity	Amplitude similar between different abnormalities
L1	1000	5	480000	[1, 3]	1	[0.2,0.9]	Yes
L2					2		
L3					3		
L4					4		
L5					5		
F1	1000	5	480000	1	1	[0.2,0.9]	-
F2					2		
F3					3		
F4					4		
F5					5		
D1	200	5	480000	[1, 3]	2	0.1	Yes
D2						0.2	
D3						0.3	
D4						0.4	
D5						0.5	
D6						0.6	
D7						0.7	
D8						0.8	
D9						0.9	

B Result Details

See Tables 4, 5 and 6.

Table 4. F1-score of L and F

Algorithm	L1	L2	L3	L4	L5	F1	F2	F3	F4	F5
Squeeze	0.145608	0.076923	0.116894	0.128627	0.09173	0.597074	0.273303	0.166527	0.159615	0.101911
AutoRoot	0.594142	0.576687	0.52037	0.472305	**0.452158**	0.89377	0.807187	**0.68316**	**0.591513**	**0.569174**
RiskLoc	0.736708	0.738944	0.732294	**0.603129**	0.265487	0.579802	0.589247	0.570931	0.477546	0.15288
PRiskLoc	**0.986328**	**0.957026**	**0.832447**	0.589744	0.218144	**0.983202**	**0.893358**	0.675841	0.436701	0.124502

Table 5. F1-score of D

Algorithm	D1	D2	D3	D4	D5	D6	D7	D8	D9
Squeeze	0.00641	0.07028	0.197647	0.239829	0.262366	0.320346	0.32906	0.32906	0.351893
AutoRoot	**0.234667**	0.446009	0.618026	0.752066	0.843882	0.877193	0.892857	0.896861	0.873874
RiskLoc	0.083333	0.256158	0.310395	0.332121	0.545455	0.935867	**0.997506**	0.956311	0.87156
PRiskLoc	0.159259	**0.45283**	**0.694949**	**0.96837**	**0.995025**	**0.997506**	**0.997506**	**0.997506**	**0.980392**

Table 6. F1-score of S

Algorithm	S	S1	S2
Squeeze	0.099115	0.076923	0.313357
AutoRoot	0.447842	0.576687	0.747053
RiskLoc	0.635049	0.738944	0.759644
PRiskLoc	**0.74687**	**0.957026**	**0.945873**

References

1. Meng, W., et al.: LogAnomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs. In: IJCAI, vol. 19, pp. 4739–4745 (2019)
2. Zhang, S., et al.: Rapid and robust impact assessment of software changes in large internet-based services. In: Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies, pp. 1–13 (2015)
3. Zhang, S., et al.: Funnel: assessing software changes in web-based services. IEEE Trans. Serv. Comput. **11**(1), 34–48 (2016)
4. Bhagwan, R., et al.: Adtributor: revenue debugging in advertising systems. In: Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation, NSDI 2014, Seattle, WA, pp. 43–55. USENIX Association, USA (2014)
5. Li, Z., et al.: Generic and robust localization of multi-dimensional root causes. In: 2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE), pp. 47–57. IEEE (2019)
6. Jing, P., Han, Y., Sun, J., Lin, T., Hu, Y.: AutoRoot: a novel fault localization schema of multi-dimensional root causes. In: 2021 IEEE Wireless Communications and Networking Conference (WCNC), pp. 1–7. IEEE (2021)
7. Kalander, M.: RiskLoc: localization of multi-dimensional root causes by weighted risk. arXiv preprint arXiv:2205.10004 (2022)
8. Silverman, B.W.: Density Estimation for Statistics and Data Analysis, vol. 26. CRC Press (1986)
9. Sun, Y., et al.: HotSpot: anomaly localization for additive KPIs with multi-dimensional attributes. IEEE Access **6**, 10909–10923 (2018)
10. Davis, R.A., Lii, KS., Politis, D.N.: Remarks on some nonparametric estimates of a density function. In: Davis, R., Lii, KS., Politis, D. (eds.) Selected Works of Murray Rosenblatt. Selected Works in Probability and Statistics. Springer, New York (2011). https://doi.org/10.1007/978-1-4419-8339-8_13