Walk the Talk: Is Your Log-based Software Reliability Maintenance System Really Reliable?

Minghua He[†], Tong Jia[†‡*], Chiming Duan[†], Pei Xiao[†],
Lingzhe Zhang[†], Kangjin Wang[§], Yifan Wu[†], Ying Li[†*], and Gang Huang[†‡]
[†]Peking University, Beijing, China
[‡]National Key Laboratory of Data Space Technology and System, Beijing, China
[§]Alibaba Group, Hangzhou, China
{hemh2120, duanchiming, xiaopei, zhang.lingzhe}@stu.pku.edu.cn
{jia.tong, yifanwu, li.ying, hg}@pku.edu.cn, kangjin.wkj@alibaba-inc.com

***Abstract*—Log-based software reliability maintenance systems are crucial for sustaining stable customer experience. However, existing deep learning-based methods represent a black box for service providers, making it impossible for providers to understand how these methods detect anomalies, thereby hindering trust and deployment in real production environments. To address this issue, this paper defines a trustworthiness metric—diagnostic faithfulness—for models to gain service providers' trust, based on surveys of SREs at a major cloud provider. We design two evaluation tasks: attention-based root cause localization and event perturbation. Empirical studies demonstrate that existing methods perform poorly in diagnostic faithfulness. Consequently, we propose FaithLog, a faithful log-based anomaly detection system, which achieves faithfulness through a carefully designed causality-guided attention mechanism and adversarial consistency learning. Evaluation results on two public datasets and one industrial dataset demonstrate that the proposed method achieves state-of-the-art performance in diagnostic faithfulness.**

***Index Terms*—Software Reliability, Log Analysis, Anomaly Detection, Trustworthiness.**

I. Introduction

Software systems are growing increasingly large and complex, with their composition and operational logic becoming more intricate, while being subject to more failures [1]–[4]. In the real world where availability is critical to customer experience, service providers have designed software reliability maintenance systems that aim to rapidly and efficiently manage anomalous events to enhance reliability [5], [6].

Logs record system operational states and serve as the primary observable data utilized by software reliability maintenance systems. The use of logs for detecting system anomalies has been extensively studied [7]–[15]. Recent approaches focus on deep learning, particularly attention-based sequence models [16]–[20]. These methods employ sequential neural networks to extract system operational patterns from logs, then utilize attention mechanisms [21] to assign differential weights to distinct system events, and finally integrate diverse events to detect system anomalies. This strategy demonstrates the capability to learn historical operational patterns of systems, thereby enabling anomaly detection and has been proven effective in maintaining software system reliability [22]–[25].

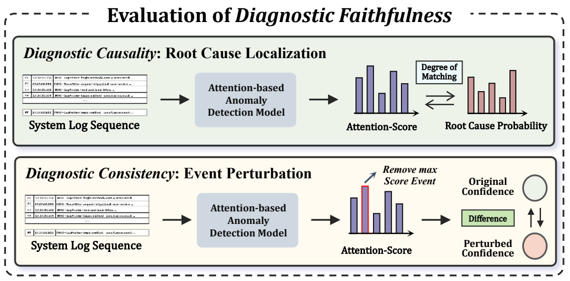
Evaluation of *Diagnostic Faithfulness*
Diagnostic Causality: Root Cause Localization
System Log Sequence
Attention-based Anomaly Detection Model
Attention-Score
Degree of Matching
Root Cause Probability
Diagnostic Consistency: Event Perturbation
Remove max Score Event
System Log Sequence
Attention-based Anomaly Detection Model
Attention-Score
Original Confidence
Difference
Perturbed Confidence

Fig. 1. The proposed attention-based diagnostic faithfulness assessment method for software reliability maintenance systems. These two tasks respectively evaluate the alignment between attention scores and anomaly root causes, and the consistency of between attention scores and detection results.

Although these methods have proven effective in detecting system anomalies, their black-box nature inherent to deep learning makes them difficult to deploy in real production environments. Specifically, these deep learning-based anomaly detection systems function as black boxes to service providers utilizing them [26]–[28], allowing providers only to obtain detection results without understanding how anomalies are detected, or whether the results are trustworthy. Providers cannot ascertain whether the anomaly detection system genuinely comprehends the operational patterns of the software system, or is merely overfitting to historical operational states. In other words, these methods intended to maintain software system reliability may themselves be unreliable. Consequently, to avoid unanticipated financial and operational losses, these deep learning-based approaches face significant barriers to deployment in actual production environments.

A comprehensive solution to this problem necessitates opening deep learning's black box, rendering its decision logic fully transparent to service providers, yet substantial research [29]–[33] demonstrates this is exceptionally challenging. Therefore, this paper aims to establish a framework for anomaly detection systems to gain service providers' trust, ensuring their trustworthiness when deployed in real production environments. Specifically, through extensive surveys of SREs at a major cloud service provider, we define **Diagnostic Faithfulness**, which encompasses two critical aspects prioritized by service providers: 1) **Diagnostic Causality**: whether the anomaly

[*] Corresponding Authors

detection system genuinely comprehends the root cause events when anomalies are detected? 2) **Diagnostic Consistency**: whether the detection results align with the system events the model considers most significant?

To comprehensively assess the diagnostic faithfulness of anomaly detection systems, we designed an attention mechanism-based evaluation method as illustrated in Figure 1. Specifically, it evaluates: 1) the alignment degree between attention scores and root causes when system anomalies are detected; 2) the magnitude of detection result changes when perturbing system events with highest attention scores, to quantify the diagnostic faithfulness of models. Subsequently, we conducted an empirical study on the Thunderbird supercomputer system [34]. Results demonstrate that state-of-the-art existing methods exhibit suboptimal performance in diagnostic faithfulness, rendering their detection outcomes untrustworthy.

To address this issue and enhance the diagnostic faithfulness of anomaly detection systems, we propose **FaithLog**, a service-provider-faithful log-based failure prediction method. Its core principles are: 1) guiding attention score allocation toward root cause events of anomalies; 2) aligning detection results with system event consistency through adversarial training. Evaluations on two public datasets and one industrial dataset demonstrate that the proposed method remains faithful to service providers and yields trustworthy detection results.

In summary, our contributions are as follows:

- We present the first investigation into the trustworthiness of log-based anomaly detection systems.
- We define diagnostic faithfulness for log-based anomaly detection systems and establish its evaluation method.
- An empirical study revealing that existing log-based anomaly detection systems lack faithfulness.
- A service-provider-faithful log-based anomaly detection system, with evaluations confirming its faithfulness.

II. Empirical Study

This section evaluates diagnostic faithfulness of existing log-based anomaly detection methods using the Thunderbird system log dataset collected from a supercomputer system [34]. We investigate state-of-the-art attention-based log anomaly detection methods, specifically including LogRobust [16], PLELog [19], NeuralLog [17], and SwissLog [18].

A. Can existing methods comprehend root causes when detecting anomalies?

This study evaluates the diagnostic causality of existing methods, employing a root cause localization task for assessment. Specifically, we first train baseline methods on anomaly detection tasks then utilize their attention scores for root cause localization. Using Hit Rate (HR@k) as the metric with k=1, 3, 5 for evaluation, the results are presented in the table I. Overall, baseline methods achieve only 13.70% HR@1, 30.42% HR@3, and 41.13% HR@5. This indicates that current log-based anomaly methods fail to assign attention scores to root cause events despite detecting system anomalies, rendering model-generated anomaly alerts unrelated to actual root causes, thus making their detection results untrustworthy.

TABLE I
Diagnostic Causality Evaluation of Existing Methods

Model	**Metrics**		
	HR@1	**HR@3**	**HR@5**
LogRobust [16]	17.19	31.83	43.14
PLELog [19]	17.36	29.09	38.44
NeuralLog [17]	13.58	39.54	53.46
SwissLog [18]	6.67	21.22	29.48
Average	**13.70**	**30.42**	**41.13**

B. Whether the detection results align with the system events the model considers most significant?

This study evaluates diagnostic consistency of existing methods, for which we designed an event perturbation task. Specifically, after model decision-making, we remove the system event with highest attention score and observe confidence changes in new detection results. If confidence decreases, it indicates that the highest-scored event suppresses detection results; conversely, increased confidence suggests support. We employ Support Rate (SR) as the metric, representing the proportion of supportive cases; higher SR indicates better alignment between detection results and most critical system events. Table II presents our results, showing an average SR of merely 61.31%, demonstrating that existing methods do not rely on their identified most critical system events for decision-making, rendering detections untrustworthy.

TABLE II
Diagnostic Consistency Evaluation of Existing Methods

Model	**Metrics**
	SR
LogRobust [16]	71.37
PLELog [19]	70.69
NeuralLog [17]	40.42
SwissLog [18]	62.76
Average	**61.31**

Summary. Current log-based anomaly detection methods neither comprehend anomaly root causes during detection nor maintain consistency between results and their prioritized system events, resulting in unreliable outcomes.

III. Methodology

A. Overview

In this paper, we propose FaithLog, a service-provider-faithful log anomaly detection method. For input log event sequences, FaithLog first employs the advanced log parser Drain [35] to process unstructured raw logs, to obtain structured log event sequences. Subsequently, following prior work [14], we extract semantic embeddings for each log event, which are fed into a Transformer Encoder-based sequence

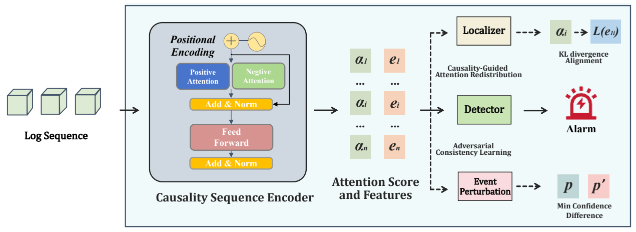

Fig. 2. The overview pipeline of FaithLog.

encoder [36]. Finally, the features encoded by the sequence encoder are delivered to a detector for anomaly identification. During the training phase, FaithLog incorporates a causality-guided attention mechanism and an adversarial consistency learning process, where the former directs attention score allocation toward root cause events of anomalies, while the latter aligns detection results with system event consistency through adversarial learning. Figure 2 illustrates the FaithLog's pipeline.

B. *Causality-Guided Attention Mechanism*

Existing log-based anomaly detection methods fail to assign attention scores to root cause events when detecting system anomalies, resulting in model alerts unrelated to actual causes and untrustworthy detection outcomes. To address this issue, FaithLog designs a Causality-Guided Attention Mechanism that aims to allocate maximal attention scores to root cause events, thereby enhancing diagnostic causality.

Specifically, FaithLog first incorporates a negative pathway into the attention mechanism, enabling the emergence of negative attention scores. When assigned negative scores, system events exhibit low probability of being anomaly root causes, yet their magnitude signifies the event's importance, a strategy that enhances the representational capacity of attention scores. Given input log event embeddings $[e_1, e_2, ..., e_n]$, for the i-th event embedding e_i, the FaithLog employs a Sinusoidal positional encoding function PE to encode its sequential position vector p_i.

$$
\begin{aligned}
PE_{(i,2j)} &= sin(\frac{i}{10000^{\frac{2j}{d_{model}}}}),\\
PE_{(i,2j+1)} &= cos(\frac{i}{10000^{\frac{2j}{d_{model}}}}),
\end{aligned} \tag{1}
$$

Where d_{model} signifies the dimension of e_i, with j indexing individual components within e_i. The resulting $E = e_i + p_i$, incorporating positional information, is then input to the encoder as the log vector. Inside the encoder, the causality-driven attention mechanism autonomously learns both positive and negative attention weight matrices, calculating attention scores for each log event as the arithmetic difference between positive and negative components.

$$
\begin{aligned}
Q_{pos} = EW^Q_{pos}, K_{pos} = EW^K_{pos}, V_{pos} = EW^V_{pos},\\
Q_{neg} = EW^Q_{neg}, K_{neg} = EW^K_{neg}, V_{neg} = EW^V_{neg},
\end{aligned} \tag{2}
$$

$$
A = Softmax(\frac{Q_{pos}K^T_{pos}}{\sqrt{d_k}})V_{pos} - Softmax(\frac{Q_{neg}K^T_{neg}}{\sqrt{d_k}})V_{neg}, \tag{3}
$$

The computed attention scores are utilized to integrate the final features, and optimize the detector by minimizing cross-entropy loss.

$$
\arg\min_{\theta}\ \mathcal{L}_{\text{CE}}(\theta) = \sum_{i=1}^{n}[-(y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i))], \tag{4}
$$

Subsequently, FaithLog learns root cause scores and guides attention score redistribution. For the input sequence $X = [e_1, e_2, ..., e_n]$, FaithLog establishes a locator L to encode root cause scores $L(e_i)$. Root cause score learning is based on ranking loss [23], [37].

$$
\arg\min_{\theta}\ \mathcal{L}_{\text{Rank}}(\theta) = \max(0, 1 + \max_{e_i \in \mathbf{X}_n} L(e_i) - \max_{e_i \in \mathbf{X}_a} L(e_i)), \tag{5}
$$

Where X_n and X_a represent normal and anomalous runtime log sequences, respectively. Finally, FaithLog aligns attention scores to root cause scores based on KL divergence, to enhance diagnostic causality.

$$
\arg\min_{\theta}\ \mathcal{L}_{\text{KL}}(\theta) = \ KL(L(X)\|A) = \sum_{i} L(e_i)\log\frac{L(e_i)}{A(e_i)}, \tag{6}
$$

C. *Adversarial Consistency Learning*

Existing log-based anomaly detection methods exhibit inconsistency between detection outcomes and highest-attention system events during decision-making, resulting in untrustworthy results. To address this issue, FaithLog designs Adversarial Consistency Learning, aiming to align detection results with the most critical system events.

Specifically, during training, FaithLog first employs detector D for initial detection, and identifies the system event with highest attention score $e_{max} = \arg\max_{i \in 1,2,...,n} a_i$, with initial detection confidence $p = D(X)$. Then, FaithLog removes the highest-attention system event e_{max}, and performs a secondary detection, yielding new confidence $p' = D(X \setminus e_{max})$. FaithLog establishes the following constraint during training, which encourages reduced detection confidence after removing the most critical system event, thereby enhancing diagnostic consistency.

$$
\arg\min_{\theta}\ \mathcal{L}_{\text{Consistency}}(\theta) = \max(1 + p' - p), \tag{7}
$$

D. *Training*

Following this approach, FaithLog can be deployed on real industrial software systems, to deliver reliability maintenance strategies faithful to service providers. FaithLog undergoes end-to-end training by integrating the aforementioned loss functions:

$$
\arg\min_{\theta}\ \mathcal{L}(\theta) = \lambda_1\mathcal{L}_{\text{CE}}(\theta) + \lambda_2\mathcal{L}_{\text{Rank}} + \lambda_3\mathcal{L}_{\text{KL}} + \lambda_4\mathcal{L}_{\text{Consistency}}, \tag{8}
$$

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are hyperparameters for balancing the loss functions.

TABLE III
DIAGNOSTIC CAUSALITY EVALUATION OF FAITHLOG

Dataset	Model	Metrics							
		HR@1	**HR@3**	**HR@5**	**PR@3**	**PR@5**	**MAP@3**	**MAP@5**	**MRR**
BGL	LogRobust [16]	37.98	65.13	67.89	39.90	41.15	39.26	39.94	52.78
	PLELog [19]	44.32	65.30	73.52	47.59	51.27	46.07	51.27	58.22
	NeuralLog [17]	37.04	49.83	52.28	40.87	42.69	39.09	40.38	43.49
	SwissLog [18]	31.89	37.04	39.24	31.55	31.90	31.98	32.28	36.84
	FaithLog	**70.03**	**77.29**	**79.42**	**72.56**	**73.70**	**71.31**	**72.18**	**74.29**
Thunderbird	LogRobust [16]	17.19	31.83	43.14	24.11	32.24	20.53	24.40	20.76
	PLELog [19]	17.36	29.09	38.44	23.22	30.07	20.29	23.47	29.66
	NeuralLog [17]	13.58	39.54	53.46	31.14	43.11	22.77	29.86	30.33
	SwissLog [18]	6.67	21.22	29.48	14.36	19.89	10.78	13.90	17.93
	FaithLog	**34.13**	**48.93**	**55.86**	**40.82**	**47.84**	**37.54**	**40.97**	**45.86**
System A	LogRobust [16]	44.15	54.24	60.58	45.28	47.50	44.70	45.56	53.40
	PLELog [19]	52.41	63.13	67.93	53.29	54.84	52.80	53.46	59.14
	NeuralLog [17]	52.45	67.19	74.72	53.71	56.32	52.93	54.05	61.64
	SwissLog [18]	43.57	54.14	56.90	45.20	47.17	44.34	45.28	49.41
	FaithLog	**58.41**	**76.73**	**81.63**	**62.13**	**65.06**	**60.30**	**61.95**	**68.36**

TABLE IV
DIAGNOSTIC CONSISTENCY EVALUATION OF FAITHLOG

Dataset	Model	Metrics
		SR
BGL	LogRobust [16]	77.86
	PLELog [19]	81.76
	NeuralLog [17]	66.44
	SwissLog [18]	54.58
	FaithLog	**88.58**
Thunderbird	LogRobust [16]	71.37
	PLELog [19]	70.69
	NeuralLog [17]	40.42
	SwissLog [18]	62.76
	FaithLog	**82.74**
System A	LogRobust [16]	75.73
	PLELog [19]	52.73
	NeuralLog [17]	48.07
	SwissLog [18]	66.34
	FaithLog	**83.28**

IV. PRELIMINARY EVALUATION

A. Experimental Setup

We conducted experiments on two public datasets (BGL and Thunderbird) [34] and an industrial log dataset System A collected from real-world systems, to assess FaithLog's diagnostic faithfulness. Detailed experimental configurations are provided in Section II. For a more comprehensive assessment, in the root cause localization task, we employ HR@k, PR@k, MAP@k, and MRR for evaluation [38].

B. Diagnostic Causality of FaithLog

This section evaluates whether FaithLog comprehends root causes when detecting anomalies, with Table III presenting comparative results against existing methods. Overall, FaithLog achieves superior performance across all three evaluation datasets. Existing methods employ naive attention mechanisms, where attention scores only indicate importance without representing causality, potentially assigning high scores even to normal system events. Conversely, FaithLog incorporates a carefully designed causality-guided attention mechanism, assigning maximally possible attention scores to root cause events, effectively enhancing diagnostic causality. These findings demonstrate FaithLog's substantial potential for direct application in root cause localization.

C. Diagnostic Consistency of FaithLog

This section evaluates whether FaithLog's detection results maintain consistency with the system events it deems most critical, with Table IV presenting results. Notably, SR of existing methods are suboptimal across all evaluated datasets. This occurs because existing methods merely utilize attention to combine features of system events, where event weights fail to represent their capacity to influence detection, resulting in attention scores being incapable of explaining the model's decision logic. Conversely, FaithLog benefits from meticulously designed adversarial consistency learning, maintaining high consistency between detection results and its prioritized system events, thereby enabling attention scores to reflect the model's decision logic, consequently empowering it to deliver trustworthy anomaly detection results for service providers.

V. CONCLUSION AND FUTURE WORK

In this paper, we present the first study on the trustworthiness of software reliability maintenance systems. Through surveys of SREs at a major cloud service provider, we designed a metric—diagnostic faithfulness—for evaluating the trustworthiness of software reliability maintenance systems, and developed two evaluation tasks: attention-based root cause localization and event perturbation. Empirical studies demonstrate that existing methods fail to comprehend root causes when detecting system anomalies, and exhibit inconsistency between detection outcomes and their prioritized system events, rendering their detection results untrustworthy. Therefore, we propose FaithLog, a faithful log-based anomaly detection system. This work provides a pathway for assessing the trustworthiness of maintenance systems, and we plan to deliver transparent, trustworthy, and explainable reliability maintenance systems in future work.

ACKNOWLEDGMENT

This work was supported by National Key Laboratory of Data Space Technology and System.

References

[1] S. Niu, J. Jin, X. Huang, Y. Wang, W. Xu, and Y. Kong, "Locating faulty applications via semantic and topology estimation," in *Companion Proceedings of the ACM Web Conference 2023*, 2023, pp. 528–532.

[2] Q. Lin, T. Li, P. Zhao, Y. Liu, M. Ma, L. Zheng, M. Chintalapati, B. Liu, P. Wang, H. Zhang *et al.*, "Edits: An easy-to-difficult training strategy for cloud failure prediction," in *Companion Proceedings of the ACM Web Conference 2023*, 2023, pp. 371–375.

[3] Y. Li, H. Yuan, Z. Fu, X. Ma, M. Xu, and S. Wang, "Elastic: edge workload forecasting based on collaborative cloud-edge deep learning," in *Proceedings of the ACM Web Conference 2023*, 2023, pp. 3056–3066.

[4] C. Luo, P. Zhao, B. Qiao, Y. Wu, H. Zhang, W. Wu, W. Lu, Y. Dang, S. Rajmohan, Q. Lin *et al.*, "Ntam: neighborhood-temporal attention model for disk failure prediction in cloud platforms," in *Proceedings of the Web Conference 2021*, 2021, pp. 1181–1191.

[5] M. Jiang, M. A. Munawar, T. Reidemeister, and P. A. Ward, "Efficient fault detection and diagnosis in complex software systems with information-theoretic monitoring," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 4, pp. 510–522, 2011.

[6] S. Zhang, S. Xia, W. Fan, B. Shi, X. Xiong, Z. Zhong, M. Ma, Y. Sun, and D. Pei, "Failure diagnosis in microservice systems: A comprehensive survey and analysis," *arXiv preprint arXiv:2407.01710*, 2024.

[7] H. Guo, J. Yang, J. Liu, J. Bai, B. Wang, Z. Li, T. Zheng, B. Zhang, J. Peng, and Q. Tian, "Logformer: A pre-train and tuning pipeline for log anomaly detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 1, 2024, pp. 135–143.

[8] J. Sun, T. Jia, M. He, Y. Wu, Y. Li, and G. Huang, "Exploring variable potential for llm-based log parsing efficiency and reduced costs," in *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, 2025, pp. 596–600.

[9] X. Zhao, T. Jia, M. He, Y. Wu, Y. Li, and G. Huang, "From few-label to zero-label: An approach for cross-system log-based anomaly detection with meta-learning," in *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering*, 2025, pp. 661–665.

[10] J. Tong, L. Ying, T. Hongyan, and W. Zhonghai, "An approach to pinpointing bug-induced failure in logs of open cloud platforms," in *2016 IEEE 9th International Conference on Cloud Computing (CLOUD)*. IEEE, 2016, pp. 294–302.

[11] C. Ebert, G. Gallardo, J. Hernantes, and N. Serrano, "Devops," *Ieee Software*, vol. 33, no. 3, pp. 94–100, 2016.

[12] K. Yin, M. Yan, L. Xu, Z. Xu, Z. Li, D. Yang, and X. Zhang, "Improving log-based anomaly detection with component-aware analysis," in *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2020, pp. 667–671.

[13] J. Kim, V. Savchenko, K. Shin, K. Sorokin, H. Jeon, G. Pankratenko, S. Markov, and C.-J. Kim, "Automatic abnormal log detection by analyzing log history for providing debugging insight," in *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Software Engineering in Practice*, 2020, pp. 71–80.

[14] C. Zhang, T. Jia, G. Shen, P. Zhu, and Y. Li, "Metalog: Generalizable cross-system anomaly detection from logs with meta-learning," in *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, 2024, pp. 1–12.

[15] T. Jia, Y. Li, Y. Yang, and G. Huang, "Hilogx: noise-aware log-based anomaly detection with human feedback," *The VLDB Journal*, vol. 33, no. 3, pp. 883–900, 2024.

[16] X. Zhang, Y. Xu, Q. Lin, B. Qiao, H. Zhang, Y. Dang, C. Xie, X. Yang, Q. Cheng, Z. Li *et al.*, "Robust log-based anomaly detection on unstable log data," in *Proceedings of the 2019 27th ACM joint meeting on European software engineering conference and symposium on the foundations of software engineering*, 2019, pp. 807–817.

[17] V.-H. Le and H. Zhang, "Log-based anomaly detection without log parsing," in *2021 36th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE, 2021, pp. 492–504.

[18] X. Li, P. Chen, L. Jing, Z. He, and G. Yu, "Swisslog: Robust anomaly detection and localization for interleaved unstructured logs," *IEEE Transactions on Dependable and Secure Computing*, 2022.

[19] L. Yang, J. Chen, Z. Wang, W. Wang, J. Jiang, X. Dong, and W. Zhang, "Semi-supervised log-based anomaly detection via probabilistic label estimation," in *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)*. IEEE, 2021, pp. 1448–1460.

[20] M. He, T. Jia, C. Duan, H. Cai, Y. Li, and G. Huang, "Llmelog: An approach for anomaly detection based on llm-enriched log events," in *2024 IEEE 35th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2024, pp. 132–143.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[22] C. Lee, T. Yang, Z. Chen, Y. Su, and M. R. Lyu, "Eadro: An end-to-end troubleshooting framework for microservices on multi-source data," in *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*. IEEE, 2023, pp. 1750–1762.

[23] M. He, T. Jia, C. Duan, H. Cai, Y. Li, and G. Huang, "Weakly-supervised log-based anomaly detection with inexact labels via multi-instance learning," in *2025 IEEE/ACM 47th International Conference on Software Engineering (ICSE)*. IEEE Computer Society, 2025, pp. 726–726.

[24] V.-H. Le and H. Zhang, "Prelog: A pre-trained model for log analytics," *Proceedings of the ACM on Management of Data*, vol. 2, no. 3, pp. 1–28, 2024.

[25] C. Duan, T. Jia, H. Cai, Y. Li, and G. Huang, "Afalog: A general augmentation framework for log-based anomaly detection with active learning," in *2023 IEEE 34th International Symposium on Software Reliability Engineering (ISSRE)*. IEEE, 2023, pp. 46–56.

[26] F. Poursabzi-Sangdeh, D. G. Goldstein, J. M. Hofman, J. W. Wortman Vaughan, and H. Wallach, "Manipulating and measuring model interpretability," in *Proceedings of the 2021 CHI conference on human factors in computing systems*, 2021, pp. 1–52.

[27] Y. Zhang, P. Tiňo, A. Leonardis, and K. Tang, "A survey on neural network interpretability," *IEEE transactions on emerging topics in computational intelligence*, vol. 5, no. 5, pp. 726–742, 2021.

[28] B. Li, P. Qi, B. Liu, S. Di, J. Liu, J. Pei, J. Yi, and B. Zhou, "Trustworthy ai: From principles to practices," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–46, 2023.

[29] D. Kaur, S. Uslu, K. J. Rittichier, and A. Durresi, "Trustworthy artificial intelligence: a review," *ACM computing surveys (CSUR)*, vol. 55, no. 2, pp. 1–38, 2022.

[30] Y. Liu, H. Li, Y. Guo, C. Kong, J. Li, and S. Wang, "Rethinking attention-model explainability through faithfulness violation test," in *International conference on machine learning*. PMLR, 2022, pp. 13 807–13 824.

[31] A. Jacovi and Y. Goldberg, "Towards faithfully interpretable nlp systems: How should we define and evaluate faithfulness?" in *58th Annual Meeting of the Association for Computational Linguistics, ACL 2020*. Association for Computational Linguistics (ACL), 2020, pp. 4198–4205.

[32] H. J. Rashkin, D. Reitter, G. S. Tomar, and D. Das, "Increasing faithfulness in knowledge-grounded dialogue with controllable features," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 704–718.

[33] C. S. Chan, H. Kong, and L. Guanqing, "A comparative study of faithfulness metrics for model interpretability methods," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 5029–5038.

[34] A. Oliner and J. Stearley, "What supercomputers say: A study of five system logs," in *37th annual IEEE/IFIP international conference on dependable systems and networks (DSN'07)*. IEEE, 2007, pp. 575–584.

[35] P. He, J. Zhu, Z. Zheng, and M. R. Lyu, "Drain: An online log parsing approach with fixed depth tree," in *2017 IEEE international conference on web services (ICWS)*. IEEE, 2017, pp. 33–40.

[36] L. Ma, W. Yang, B. Xu, S. Jiang, B. Fei, J. Liang, M. Zhou, and Y. Xiao, "Knowlog: Knowledge enhanced pre-trained language model for log understanding," in *Proceedings of the 46th ieee/acm international conference on software engineering*, 2024, pp. 1–13.

[37] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6479–6488.

[38] L. Zheng, Z. Chen, D. Wang, C. Deng, R. Matsuoka, and H. Chen, "Lemma-rca: A large multi-modal multi-domain dataset for root cause analysis," *arXiv preprint arXiv:2406.05375*, 2024.