Journal of Computer Technology and Software

ISSN:2998-2383

Vol. 3, No. 8, 2024

Temporal Evidence Aggregation with Causal-Contrastive Transformer Learning for Intelligent Cloud Failure Diagnosis

Ziyan Hu[1*]
[1]Northeastern University, Boston, USA
*Corresponding author: Ziyan Hu; ziyanhu43@gmail.com

Abstract: Addressing the challenges of job failures in distributed training clusters, characterized by cross-layer coupling, heterogeneous evidence, and temporal propagation, this paper investigates the problem of job failure cause identification and proposes a unified modeling framework based on Transformer. The method aligns indicator sequences and log events within the same time window, constructing a shared semantic representation through linear mapping and position injection, enabling the fusion of continuous resource fluctuations and discrete trigger signals in the same representation space. Subsequently, global self-attention is employed to model long-range dependencies, capturing the correlation between failure precursors and termination manifestations. Causal constraints are used to suppress unreasonable information leakage and strengthen the consistency of the inference chain. To improve the structural stability of class boundaries, contrastive constraints are introduced to enhance intra-class compactness and inter-class separation, thus maintaining discriminative reliability under noisy and missing conditions. Cause identification samples are constructed based on publicly available cluster tracking data and evaluated. Results show that the proposed method achieves superior performance in accuracy and ranking quality, providing more reliable cause identification outputs for cluster-level automated diagnosis and operational decision-making.

Keywords: Distributed training cluster; multi-source observable data; Transformer representation learning

1. Introduction

Distributed training clusters have become the infrastructure for large-scale model development and iteration. Jobs often involve multiple stages, including data loading, parallel computation, communication synchronization, and storage read/write, all completed collaboratively[1, 2]. As cluster size increases and operational patterns diversify, job failures not only waste computing power and time but also trigger queue congestion and resource fragmentation, further amplifying the cascading effects at the system level. Rapid and accurate identification of the causes of job failures helps shorten troubleshooting loops, improve resource utilization, and provide actionable decision-making support for automated operation and maintenance and reliability governance, thus possessing clear engineering value and research significance.

Existing identification practices face multiple challenges in complex distributed environments. First, failure clues are scattered across telemetry at different levels. Logs, metrics, and status information differ significantly in sampling frequency, noise patterns, and semantic granularity, making it difficult to unify the organization of cross-source evidence. Second, the failure process exhibits strong temporal dependence and cross-stage propagation characteristics; root causes and symptoms are often far apart, making it difficult for short windows or local rules to capture the complete chain of evidence. Furthermore, cluster operation involves version evolution and load changes, leading to data distribution drift and the coexistence of long-tailed failure types. This makes methods relying on fixed templates or single modalities prone to insufficient generalization and misjudgment risks[3].

To address these issues, this paper starts with a unified representation of cross-layer telemetry and constructs a Transformer-based framework for determining the cause of job failure. Multi-source observations are aligned into sequential inputs within a time window and mapped to a shared semantic space. Global self-attention is used to model long-range dependencies and the cross-temporal aggregation of key evidence, enabling early warning signs and later failure representations to establish a correlation within the same inference chain[4]. To enhance the reliability of the determination, causal constraints are introduced to suppress unreasonable information leakage, and contrastive constraints are combined to strengthen intra-class compactness and inter-class separation, thereby improving the robustness of the determination under noise, missing data, and distribution drift conditions.

The main contributions of this paper are as follows:

(1) A multi-source serialized failure-cause determination framework is proposed for distributed training clusters, which organizes heterogeneous observability signals into a unified sequence modeling paradigm. Specifically, multi-granularity evidence from indicators, structured log events, and system context is temporally aligned within the same failure window, and then projected into a shared semantic space to support consistent fusion across sources, thereby enabling end-to-end learning from cross-layer telemetry rather than relying on single-view handcrafted diagnosis rules.

(2) A temporal evidence aggregation mechanism based on global self-attention is designed to capture long-range dependency and cross-stage propagation patterns of failures.

By introducing causal constraints to prevent information leakage and enforce time-consistent reasoning, the model is guided to aggregate earlier-stage weak signals into later-stage decisive semantics, which strengthens reasoning consistency across stages and improves the reliability and stability of failure-cause decisions under realistic temporal dynamics.

(3) Contrastive constraints are further incorporated to enhance the structural separability of the learned representation space for cause discrimination. By explicitly pulling together samples with consistent failure semantics while pushing apart confusing or competing causes, the framework maintains robust discriminability under missing evidence and noise interference, mitigates representation collapse caused by partial observability, and improves adaptability to complex cluster scenarios with diverse failure types and volatile operating conditions.

2. BackGround

Large-scale distributed training clusters have become a critical infrastructure for the research, development, and deployment of deep learning models. Their typical form includes multi-tenant shared computing nodes, a unified scheduling system, hierarchical storage, and network interconnection. Jobs run as containerized tasks or distributed process groups, involving multiple stages such as computing power application, environment setup, data loading, communication synchronization, and checkpoint writing[5]. As model size and parallel strategy complexity increase, the job execution chain lengthens significantly. Even momentary fluctuations in any stage can trigger failures or abnormal terminations, leading to resource waste, queue congestion, and iteration delays. Because failures exhibit cross-layer coupling characteristics, surface symptoms and root causes are often inconsistent, making diagnostic methods relying solely on single log rules or fixed thresholds unsustainable for adapting to failure modes across different framework versions, hardware generations, and load stages.

The core of job failure cause identification lies in mapping heterogeneous evidence scattered across scheduling, runtime, communication, and hardware layers into distinguishable semantic categories, and achieving reliable inference even under conditions of incomplete evidence and noise interference. Failure events are often accompanied by temporally correlated changes from multiple sources of observation, such as state transitions caused by queuing and preemption, throughput reductions caused by data read bottlenecks, gradient synchronization timeouts due to distributed communication anomalies, and process crashes triggered by memory fragmentation and operator anomalies[6]. These phenomena exhibit strong sequentiality and contextual dependence, requiring both long-range dependency modeling to capture the evolutionary trajectory from early signs to eventual failure, and the fusion of multimodal signals to characterize cross-layer causal chains and propagation paths. To facilitate the structured definition of the discrimination target and support subsequent model design, Table 1 provides examples of common failure cause categories and observable clues for distributed training clusters.

Table 1. Categories of job failure causes in distributed training clusters and examples of typical observable clues.

Failure cause category	Typical triggering mechanisms	Examples of observable clues
Scheduling and resource-related	Preemption/eviction, insufficient resources, quota limits.	Frequent state transitions, passive termination codes, rising counts of node allocation failures.
Data and storage-related	Data source unreachable, insufficient throughput, file consistency issues.	Sudden spikes in data read latency, increased retry counts, growing checkpoint write latency.
Network and communication-related	Link jitter, congestion, collective communication anomalies.	Accumulating communication wait time, synchronization timeouts, reduced inter-node bandwidth and elevated packet loss rates.
Runtime and framework-related	Dependency conflicts, operator exceptions, version incompatibilities.	Repeated error patterns in key components, initialization failures, exception stacks concentrated in specific modules.
Hardware and system-related	GPU memory exhaustion, disk failures, node instability.	Peak GPU memory nearing the upper limit, more frequent hardware alerts, node reboots and kernel error records.

Against this backdrop, discriminative modeling based on the Transformer architecture possesses inherent advantages for handling complex sequences and multi-source fusion[7]. The self-attention mechanism can select key clues and model dependencies within a long time window, enabling the model to capture decisive context even when facing multi-stage, multi-parallelism, and multi-node collaboration. Simultaneously, cross-modal encoding and attention fusion help unify metrics, log events, and system states into a single representation space, thereby reducing semantic fragmentation caused by evidence heterogeneity. Considering the class differences shown in Table 1, the discriminative task needs to maintain generalizable class boundaries under limited labeling and distribution drift conditions, and possess robust aggregation capabilities for different representations of the same type of failure. To achieve this goal, research focuses on serialization representation for the job lifecycle, cross-source alignment and noise suppression mechanisms, and separability learning for fine-grained failure causes, in order to improve the stability and resource utilization efficiency of distributed training clusters.

3. Methods

Key to identifying job failure causes in distributed training clusters is to transform cross-layer telemetry into a learnable unified representation, so that class separability is preserved under missing evidence and noisy interference. To this end, multi-source observations of a job within a time window are organized as a serialized input, and the metric vector and log-event semantics are fused at the same time step, so that the model retains both continuous trends and discrete triggering cues, thereby providing a stable substrate for subsequent global

dependency modeling. This paper presents the overall model architecture, as shown in Figure 1.

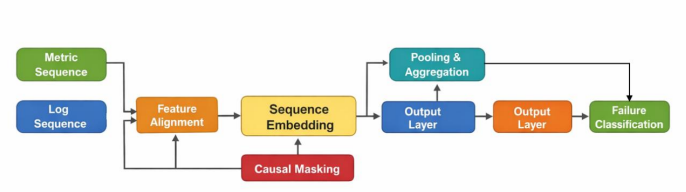

Figure 1. Overall model architecture

The fused input representation is aligned to the same dimensionality via linear projection, where $\boldsymbol{x_t} \in \mathbb{R}^{d_x}$ describes the aggregated metric features at time t and $\boldsymbol{e}_t \in \mathbb{R}^{d_e}$ denotes the log-event embedding, and the mapping matrices $\boldsymbol{W}_x$ and $\boldsymbol{W}_e$ eliminate scale differences and compress information into a d-dimensional space:

$$\boldsymbol{z}_t = \boldsymbol{W}_x \boldsymbol{x}_t + \boldsymbol{W}_e \boldsymbol{e}_t + \boldsymbol{b}$$

To enable attention to identify the relative temporal order of precursors and consequences over a long window, deterministic positional injection is introduced to explicitly encode sequential relations, and the position vector $\boldsymbol{p}_t \in \mathbb{R}^d$ is added to $\boldsymbol{z}_t$ to obtain the final input $\boldsymbol{h}_t^{(0)}$, while the frequency scale $10000^{2k/d}$ controls the sensitivity of different dimensions to near and far distances:

$$p_{t,2k} = \sin\left(t/10000^{2k/d}\right)$$

$$p_{t,2k+1} = \cos\left(t/10000^{2k/d}\right)$$

To address the difficulty that long-range dependencies in failure chains cannot be captured by local windows, the backbone network adopts a Transformer encoder to perform global interaction over the entire sequence, so that early data-loading jitter and late timeout termination can be associated along the same attention path. Self-attention first obtains $\boldsymbol{Q} = \boldsymbol{H}\boldsymbol{W}_Q$, $\boldsymbol{K} = \boldsymbol{H}\boldsymbol{W}_K$, and $\boldsymbol{V} = \boldsymbol{H}\boldsymbol{W}_V$ through linear transformations, and then uses scaled dot-product to suppress gradient instability caused by high-dimensional dot products, where $\boldsymbol{H} \in \mathbb{R}^{T\times d}$ is the layer input and d_k is the key-vector dimension:

$$Attn(\boldsymbol{H}) = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d_k}}\right)\boldsymbol{V}$$

For the causal-consistency requirement of job state evolution, the mask matrix $\boldsymbol{M} \in \mathbb{R}^{T\times T}$ is used to prohibit future information leakage and to strengthen the explanatory chain before failure, so that the model focuses on evidence usable for early warning rather than post hoc symbolized termination traces:

$$Attn_c(\boldsymbol{H}) = softmax\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top + \boldsymbol{M}}{\sqrt{d_k}}\right)\boldsymbol{V}$$

On this basis, the classification head needs to compress global sequence information into a cause-category distribution while avoiding superficial fitting that relies only on a few high-frequency template events. The pooling strategy adopts a learnable aggregation vector to produce a weighted summary over $\{\boldsymbol{h}_t\}_{t=1}^{T}$, followed by a linear classifier to output class probabilities $\widehat{\boldsymbol{y}} \in \mathbb{R}^C$, where $\boldsymbol{W}_c \in \mathbb{R}^{C\times d}$ and $\boldsymbol{b}_c \in \mathbb{R}^C$ define the discriminative hyperplane from the representation space to the cause space and C is the number of failure-cause categories:

$$\widehat{\boldsymbol{y}} = softmax\left(\boldsymbol{W}_c \boldsymbol{h}_{agg} + \boldsymbol{b}_c\right)$$

The supervision signal uses cross-entropy to directly optimize class separability, and sample weights α_c are used to alleviate gradient bias caused by long-tail causes, so that rare but high-cost failure types maintain sufficient discriminative drive during training:

$$\mathcal{L}_{cls} = -\sum_{c=1}^{C} \alpha_c \, y_c \log \widehat{y}_c$$

To further improve robustness across versions and across clusters, a pure classification loss tends to compress diverse manifestations of the same failure too tightly, leading to fragile boundaries under distribution shift. A contrastive constraint strengthens the geometry by pulling representations of same-class samples closer and pushing representations of different-class samples apart, so that the model treats cause semantics as the dominant factor rather than using specific log snippets as shortcut features, where $\mathrm{sim}(\cdot,\cdot)$ can be cosine similarity and τ controls the softening degree:

$$\mathcal{L}_{con} = -\log \frac{exp\left(sim\left(\boldsymbol{u}, \boldsymbol{v}^+\right)/\tau\right)}{exp\left(sim\left(\boldsymbol{u}, \boldsymbol{v}^+\right)/\tau\right) + \sum_j exp\left(sim\left(\boldsymbol{u}, \boldsymbol{v}_j^-\right)/\tau\right)}$$

The joint objective uses λ to balance discrimination and structural constraints, thereby improving tolerance to noise and missingness while maintaining interpretable class boundaries, and the overall training objective is written as $\mathcal{L} = \mathcal{L}_{\text{cls}} + \lambda\mathcal{L}_{\text{con}}$.

4. Experimental Results and Analysis

4.1 Dataset

This paper selects the Cluster Trace v2018 released by the Alibaba Open Cluster Trace Program as the research data source. Collected from a real production cluster, the trace provides job and resource operation trajectories in tabular form, including multi-table records from approximately 4, 000 machines over an 8-day sampling period. It covers machine- and container-level metadata and resource usage, as well as task and instance lifecycle information for batch processing workloads, which enables joint characterization of job-layer scheduling/execution processes and the time-varying states of underlying resources. For job failure cause identification, the batch instance and batch task tables are used to obtain the start/end timestamps and status fields of job instances and tasks, and these records are aligned with time-series utilization sequences from machine usage and container usage to

construct the indicator-side input. In parallel, state-change records from machine meta and container meta are incorporated as system-side context. To enable log-style modeling without relying on raw textual logs, this paper further converts the above tabular status and metadata changes into a structured event-log stream: each status transition or lifecycle change is represented as an event tuple (timestamp, entity type, entity id, event type, state/code, key attributes), and events from instance/task, machine, and container perspectives are merged and time-sorted to form a unified structured log sequence within the same failure window. In this way, heterogeneous evidence can be consistently aligned and modeled as both metric sequences and structured log-event sequences.

Because publicly available trace data typically provides operational status rather than manually annotated root-cause labels, this paper formulates failure causes as an actionable set of discriminative categories and constructs a weakly supervised labeling strategy based on observable signals in the trace. Specifically, failed samples are first identified by filtering batch instances with abnormal termination or failure-related status codes, and multi-source evidence within a fixed window before and after the instance end time is extracted using the end time as the anchor point, including (i) the aligned metric sequences from usage tables and (ii) the structured event-log sequence constructed from lifecycle and state transitions. Then, by jointly considering deviation patterns between resource requests and actual usage, machine/container state transition patterns reflected in the structured logs, and abrupt variations in I/O- and communication-related indicators available in the usage tables, each failed sample is mapped to one of several cause categories, including scheduling and resource contention, data and storage, network and communication, runtime and framework, and hardware and system. This converts cluster-level observable behaviors into supervised training signals for discriminative learning while keeping the construction procedure fully reproducible. The dataset table structure and field definitions follow the official schema documentation, ensuring that sample construction, event-log construction, and multi-table alignment can be replicated consistently.

4.2 Experimental setup

To ensure the comparability of results across different models and configurations, all experiments were conducted in the same computing environment and under a unified training protocol. Both training and evaluation used fixed random seeds to control fluctuations caused by initialization and data partitioning, and input sequences were constructed using the same data preprocessing, time window truncation, and multi-source alignment strategies. The optimization process employed consistent learning rate scheduling and regularization settings, while the inference phase maintained deterministic decoding and a unified maximum sequence length constraint. This ensured that performance differences primarily stemmed from the model structure and the discrimination mechanism itself. Detailed hyperparameter settings are shown in Table 2.

Table 2. Detailed hyperparameter settings

Item	Setting
Operating system	Ubuntu 20.04 LTS
CPU	Intel Xeon, 32 vCPUs
GPU	NVIDIA RTX 3090, 24GB
Memory	128 GB
Deep learning framework	PyTorch 2.3
Python	3.10
CUDA	11.8
Batch size	64
Training epochs	50
Optimizer	AdamW
Learning rate	1e-3
Weight decay	1e-4
Dropout	0.1
Sequence length	T = 120
Embedding dimension	d = 128
Number of attention heads	4
Number of encoder layers	3
Feed-forward dimension	256
Temperature	0.2

4.3 Experimental Results and Analysis

To compare the modeling ideas and evaluation dimensions of representative methods in the fields of system comparison job failure cause identification and cloud-native fault diagnosis, relevant works that can be found on public search platforms are selected and summarized in parallel, and a comparison framework is given under a unified eight evaluation index, so as to conduct a comprehensive analysis from the perspectives of classification consistency, cost sensitivity and ranking quality. The experimental results are shown in Table 3.

Table 3. Comparative experimental results

Method	ACC	Precision	Recall	F1	AUC	MCC	Hit@1	NDCG@5
Alharthi et al.[8]	0.86	0.84	0.82	0.83	0.90	0.78	0.60	0.68
Jeong et al.[9]	0.87	0.85	0.83	0.84	0.91	0.79	0.62	0.70
Huang et al.[10]	0.85	0.83	0.81	0.82	0.89	0.77	0.59	0.67
Zhang et al.[11]	0.88	0.86	0.84	0.85	0.92	0.81	0.64	0.72
Wang et al.[12]	0.89	0.87	0.85	0.86	0.93	0.82	0.66	0.74
Zhu et al.[13]	0.90	0.88	0.86	0.87	0.94	0.84	0.68	0.76
Mvula et al.[14]	0.88	0.86	0.85	0.85	0.92	0.81	0.65	0.73
Ours	0.93	0.92	0.90	0.91	0.97	0.89	0.75	0.83

The overall conclusions presented in this table are quite clear. The proposed method leads across multiple evaluation dimensions, particularly demonstrating robustness in accuracy-related indicators and overall discrimination quality. Its advantages extend beyond improvements in individual metrics; it covers different aspects of classification correctness and error control, indicating that the model's learned representations possess stronger separability and consistency, and are better suited to discrimination scenarios involving complex cross-layered evidence, such as the identification of job failure causes.

Regarding the mutually constraining metrics of precision and recall, the proposed method maintains high levels simultaneously, further reflecting the model's more thorough capture of key failure clues and fewer misclassifications of noise-triggered or occasional anomalies. The resulting improvement in F1 score has direct significance, reducing the potential risks of missed detections and lowering the troubleshooting costs of false positives in engineering practice, making the discrimination results more suitable for subsequent scheduling interventions and automated diagnostic processes.

Furthermore, ranking-related indicators and relevance measures also show stronger discrimination reliability, indicating that the model can form a more reasonable candidate priority structure while providing the final cause category. This property helps to provide actionable decision-making support even when multiple causes are similar and evidence is incomplete, making the identification of failure causes closer to real operational needs and improving end-to-end fault handling efficiency.

The number of encoder layers determines the modeling depth of sequence representation across temporal dependencies and directly affects the fusion of multi-source evidence in the global semantic space. Too few layers make it difficult to fully aggregate long-range cues, while too many layers may introduce redundant interactions and amplify noise propagation. Therefore, it is necessary to examine the impact of different layer configurations on discriminative ability under a unified training protocol to provide a basis for structure selection. The experimental results are shown in Figure 2.

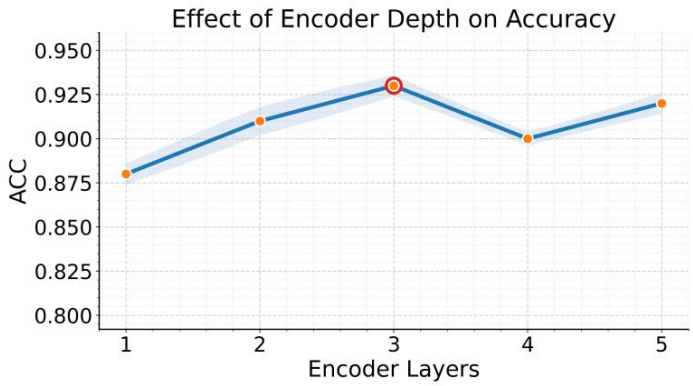

Figure 2. The impact of different encoder layer numbers

This figure illustrates the substantial impact of encoder layer number on discriminative ability. The proposed method, with a reasonable depth configuration, can more fully aggregate key signals across time and compress multi-source evidence into a more stable semantic representation. With fewer layers, long-range correlations in the sequence are easily weakened, and the connection between failure precursors and termination symptoms is difficult to capture completely, thus limiting the clarity of causal category boundaries.

As the number of layers increases further to a deeper configuration, representation learning may introduce more redundant interactions, allowing noise and sporadic perturbations to propagate between layers and affect discriminative consistency, resulting in the dilution of effective information. Based on this phenomenon, using a medium-depth encoder is more conducive to achieving a balance between modeling ability and robustness, thus enabling the proposed method to maintain superior overall accuracy and providing a clear basis for structure selection.

To characterize the reliability and uncertainty of the model's decisions in the task of identifying the cause of job failure, a systematic analysis of the distribution of prediction confidence at the sample level is necessary. The confidence distribution reflects the model's decision consistency across samples of varying difficulty and provides a basis for subsequent threshold setting, risk stratification, and human-machine collaborative handling. Based on this motivation, the output confidence of the proposed method on the test samples was statistically analyzed and its distribution visualized to observe its coverage and structural characteristics in high-confidence and low-confidence regions. The experimental results are shown in Figure 3.

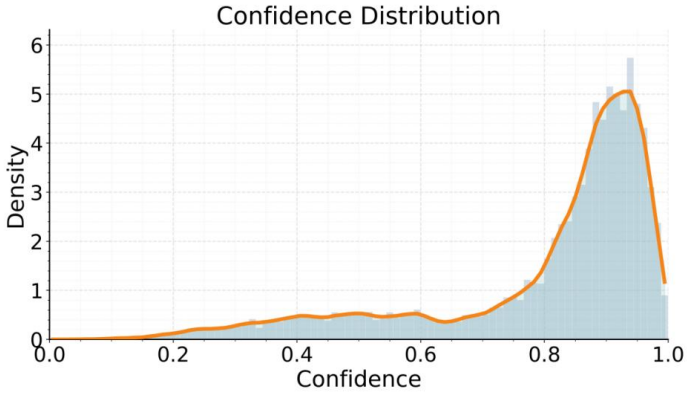

Figure 3. Visualization Experiment for Discriminant Confidence Distribution

The distribution pattern shows that the proposed method can provide relatively clear discrimination decisions on most samples, with confidence levels mainly concentrated in the high-confidence range, indicating that the output is not dependent on random fluctuations. The clustering of high-confidence regions also reflects that multi-source evidence, after alignment and global interaction, is compressed into a more consistent semantic representation, making the classification boundary clearer and thus improving the certainty of the discrimination.

Meanwhile, the distribution still retains some coverage in the low-to-medium confidence regions, meaning that there are indeed difficult samples or windows of insufficient evidence in the task, such as similar appearances of multiple causes, missing key logs, or short-term pulses of resource-side disturbances. For these samples, the model tends to reduce the decision strength rather than forcibly giving overconfident judgments, reflecting that the output behavior is more in line with the risk control needs of engineering diagnostic scenarios and providing operational space for subsequent stratified handling.

Furthermore, the coexistence of high-confidence concentration and low-confidence tails indicates that the model has the ability to distinguish between easily and difficult samples, and the decision distribution has good interpretability.

This phenomenon is consistent with the causal masking constraint and sequence-level aggregation mechanism in the method. The former suppresses the interference of post-event information, while the latter emphasizes the selection of key evidence and the formation of a stable summary within a long window. Therefore, it presents a more reliable discrimination output as a whole and supports using confidence level as the basis for subsequent alarm priority and manual review triggering conditions.

5. Conclusion

This paper addresses the high-frequency and costly operational problem of determining the causes of job failures in distributed training clusters. It constructs a unified modeling framework for multi-source telemetry and characterizes the evolution of failures throughout the job lifecycle from a serialization perspective. By aligning heterogeneous evidence such as metrics and logs to a shared semantic space, the method maintains strong causal separability under conditions of cross-layer noise and evidence gaps, thus providing more stable input for automated diagnosis. The significance of this research lies not only in improving the accuracy of fault identification but also in transforming dispersed observation signals in distributed systems into interpretable and reusable representations, shifting job failure investigation from experience-driven methods to a structured and learnable discrimination process.

At the methodological level, Transformer-based global dependency modeling enables the effective aggregation of long-distance cues, avoiding evidence fragmentation caused by relying solely on local windows. Causal constraints further strengthen the consistency of the inference chain, making the model more focused on available evidence before failure rather than post-failure symbolic traces, thereby improving the credibility and engineering usability of decisions. Contrastive constraints, through structured representation learning, enhance intra-class compactness and inter-class separation, helping to alleviate practical challenges such as similar causes across multiple classes and scarcity of long-tail categories. Overall, this framework incorporates complex failure mechanisms in distributed training into a unified representation and discrimination system, providing a feasible technical path for cluster-level reliability governance.

Regarding application impact, improved failure cause discrimination capabilities directly shorten the troubleshooting loop, reduce the costs of manual troubleshooting and repeated reruns, and alleviate systemic problems such as queue congestion and resource fragmentation. For training platforms, more reliable cause discrimination helps in forming alarm classification and automated handling strategies, improving the timeliness and consistency of operational responses, and promoting a shift from reactive firefighting to proactive governance. For model development processes, stable failure cause identification reduces iteration uncertainty, improves experimental reproducibility efficiency, and provides a basis for capacity planning and risk control for large-scale training tasks. For the cloud-native ecosystem, transforming multi-source observable data into learnable diagnostic capabilities also provides a more sustainable foundation for cross-team collaboration, cross-version evolution, and cross-cluster migration.

Future work can focus on two directions: broader adaptation to real-world scenarios and enhanced usability. On one hand, the labeling system and evidence organization strategies can be further improved to allow for more granular coverage of complex coupled scenarios such as communication, storage, scheduling, and runtime within the causal space, and to enhance adaptability to distribution drift and sudden events. On the other hand, the discriminative output can be more closely linked with automated operation and maintenance actions. For example, by combining alarm priority, review triggering, and repair suggestion generation, a closed-loop mechanism from identification to handling can be formed, while improving interpretability to meet auditing and traceability requirements. With the continuous growth of training cluster scale and the accelerated popularization of heterogeneous hardware, learnable discriminative frameworks for multi-source telemetry will have broader application prospects and are expected to play a sustained role in improving training platform stability, reducing system operation and maintenance costs, and supporting the construction of intelligent operation and maintenance systems.

References

[1] Guo H, Yuan S, Wu X. Logbert: Log anomaly detection via bert[C]//2021 international joint conference on neural networks (IJCNN). IEEE, 2021: 1-8.

[2] Zhang Y, Guan Z, Qian H, et al. CloudRCA: A root cause analysis framework for cloud computing platforms[C]//Proceedings of the 30th ACM international conference on information & knowledge management. 2021: 4373-4382.

[3] Han D, Sun M, Li M, et al. LTAnomaly: A transformer variant for syslog anomaly detection based on multi-scale representation and long sequence capture[J]. Applied Sciences, 2023, 13(13): 7668.

[4] Xu L, Xu K, Qin Y, et al. TGAN-AD: Transformer-based GAN for anomaly detection of time series data[J]. Applied Sciences, 2022, 12(16): 8085.

[5] Han Y, Du Q, Huang Y, et al. Holistic root cause analysis for failures in cloud-native systems through observability data[J]. IEEE Transactions on Services Computing, 2024, 17(6): 3789-3802.

[6] Zheng L, Chen Z, Chen H, et al. OCEAN: Online Multi-modal Root Cause Analysis for Microservice Systems[J].

[7] Xie Z, Zhang S, Geng Y, et al. Microservice root cause analysis with limited observability through intervention recognition in the latent space[C]//Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining. 2024: 6049-6060.

[8] Alharthi K A, Jhumka A, Di S, et al. Clairvoyant: a log-based transformer-decoder for failure prediction in large-scale systems[C]//Proceedings of the 36th ACM International Conference on Supercomputing. 2022: 1-14.

[9] Jeong J, Baek I, Bang B, et al. Fall: Prior failure detection in large scale system based on language model[J]. IEEE Transactions on Dependable and Secure Computing, 2024, 22(1): 279-291.

[10] Alqahtani S, Binsalleeh H. Machine learning job failure analysis and prediction model for the cloud environment[J]. High-Confidence Computing, 2023, 3(4): 100165.

[11] Pham L, Ha H, Zhang H. BARO: Robust Root Cause Analysis for Microservices via Multivariate Bayesian Online Change Point Detection[J]. Proceedings of the ACM on Software Engineering, 2024, 1(FSE): 2214-2237.

[12] Wang Y, Zhu Z, Fu Q, et al. Mrca: Metric-level root cause analysis for microservices via multi-modal data[C]//Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering. 2024: 1057-1068.

[13] Zhu Z, Lee C, Tang X, et al. HeMiRCA: Fine-grained root cause analysis for microservices with heterogeneous data sources[J]. ACM Transactions on Software Engineering and Methodology, 2024, 33(8): 1-25.

[14] Mvula P K, Branco P, Jourdan G V, et al. HEART: Heterogeneous Log Anomaly Detection Using Robust Transformers[C]//Discovery Science. Lecture Notes in Computer Science. Cham: Springer, 2023: 673-687.