Journal of Computer Technology and Software

ISSN: 2998-2383

Vol. 3, No. 7, 2024

Causal Discriminative Modeling for Robust Cloud Service Fault Detection

Heyi Wang
Illinois Institute of Technology, Chicago, USA
helenwangheyi@gmail.com

Abstract: In this paper, a detection method based on a causal discrimination mechanism is proposed to improve the fault identification ability of the system in the complex dynamic environment to solve the modeling problem of service fault detection tasks in the cloud computing environment. This method explicitly models the causal dependency between multi-dimensional input features by constructing a structured representation module and causal path extraction mechanism and then generates intermediate representation with structural semantics for fault state discrimination. The overall architecture consists of input encoding, causal encoder, structure selector, and unified discriminator, which can effectively capture the key interaction paths between service components and enhance the perception ability of the fault propagation chain. In the process of structural modeling, the method combines causal graph information and structure selection function to extract stable and interpretable features for final prediction, which significantly enhances the robustness and transferability of the model under input perturbation and distribution change. In order to fully verify the effectiveness of the proposed method, this paper designs multiple experimental tasks, covering the sensitivity evaluation of multiple key variables such as the number of candidate paths, encoding dimension, noise perturbation, etc., and compares them with representative methods in recent years. Experimental results show that the proposed method performs well in multiple indicators such as accuracy, structural consistency, and stability, verifying the applicability and effectiveness of causal modeling mechanisms in cloud service fault detection.

Keywords: Causal discriminant modeling; structural consistency; input perturbations; cloud service stability

1. Introduction

With the rapid development of digital infrastructure, cloud computing has become a key platform for supporting various applications and services. Its elasticity, high availability, and on-demand scalability bring unprecedented flexibility to enterprises and institutions[1]. However, as cloud architectures become more complex, the underlying resource scheduling, multi-tenant concurrent access, and coordination among heterogeneous components exhibit increasing dynamics and uncertainty. This makes system failures more frequent and harder to detect. A single failure in cloud services may affect more than one component and even trigger cascading effects at the system level. In severe cases, it can cause large-scale service interruptions and data loss, resulting in significant economic and reputational damage. Therefore, accurately and promptly detecting potential failures in cloud services is critical to ensuring the stable operation of cloud computing systems[2].

Traditional failure detection approaches in cloud services often rely on statistical thresholds, rule matching, or machine learning models to identify anomalies based on monitoring indicators. These methods can be effective for isolated components or clear anomaly patterns. However, they lack generalization and robustness in complex distributed systems with highly dynamic environments. On one hand, traditional models often fail to capture the underlying causal relationships within systems and are prone to false positives triggered by non-fault factors. On the other hand, failures in cloud environments do not always manifest through a single observable feature. They are often caused by interactions among multiple latent factors, which are difficult to represent using static models. As a result, relying solely on surface-level observations is no longer sufficient to meet the growing demand for system stability in modern cloud infrastructures[3].

In recent years, with advances in causal inference theory, more research has focused on incorporating causal modeling into anomaly detection. This approach aims to better understand the driving mechanisms behind system behavior. Causal discrimination models build causal dependencies between variables. They enable a more accurate distinction between true faults and superficial anomalies, even under noise disturbances or representation shifts. This improves both reliability and interpretability[4]. In cloud service scenarios, where components involve complex invocation paths and resource contention, causal modeling helps identify the root causes of failures. It can reconstruct potential propagation chains from multi-dimensional data, revealing the internal logic behind failures. Introducing a causal perspective provides a new theoretical and technical foundation for designing more generalizable and stable fault detection algorithms.

Applying causal discrimination models to cloud failure detection carries significant research and practical value. First, these models break the dependency on labeled data and pattern memorization. They maintain high recognition performance even on unseen anomaly types. Second, they offer inherent interpretability, providing system operators with clearer diagnostic paths and actionable intervention suggestions. This

reduces the time needed for fault localization and recovery. More importantly, causal modeling can effectively handle data imbalance and observational bias, which are common in cloud environments. It helps build a more fair and robust detection system. By constructing causal structures among metrics, events, and services, the approach improves model generalization and supports downstream tasks such as root cause analysis, autonomous recovery, and preventive strategies[5].

With the widespread deployment of large-scale distributed systems and multi-tenant hybrid architectures, traditional statistical feature-based detection mechanisms can no longer meet the dual requirements of timeliness and accuracy in cloud failure response. Developing intelligent detection methods with causal discrimination capabilities has become a key technical direction to enhance cloud service availability and user experience. This direction supports a paradigm shift from observing anomalies to understanding failures. It also facilitates the transformation of cloud operation from experience-driven to mechanism-driven, and from data centralization to causal logic. As causal learning techniques continue to evolve, their role in cloud failure management will become increasingly important. They are expected to serve as a foundational component in building intelligent and autonomous cloud infrastructure.

2. Background & Motivation

2.1 Background

In modern cloud computing systems, service components are highly distributed and strongly interdependent. Frequent interactions and resource sharing among services make failures more complex and harder to localize. Different types of abnormal behaviors may originate from the underlying hardware, virtualization platforms, middleware, or business logic. These sources interact with each other, creating coupled effects that obscure the root cause of failures. Monitoring data from a single dimension is often insufficient to capture the true cause. In addition, frequent updates to service deployments and continuous changes in system state introduce significant temporal variations in runtime data. As a result, traditional detection mechanisms based on fixed patterns are easily disturbed and lack stability in real-world scenarios[6].

In practice, most mainstream anomaly detection methods rely on unsupervised learning or statistical feature extraction. Although these methods offer a degree of automation, they lack a deep understanding of the internal relationships among multidimensional data. Judgments based on representation bias tend to misclassify transient resource fluctuations or non-critical anomalies as serious faults. This leads to high false alarms and miss rates. Moreover, current methods struggle to model the dynamic evolution of cross-node and cross-service anomaly chains. The resulting diagnostic view is often fragmented, and observations of the system lack coherence.

Furthermore, in multi-tenant shared environments, implicit conflicts frequently arise between system load fluctuations and resource scheduling strategies. These conflicts often do not produce obvious metric anomalies. Instead, they affect service stability through indirect effects. Such hidden causal relationships are often overlooked by existing monitoring systems, leading to root cause analysis that deviates from the real issue. In addition, data incompleteness, limited labels, and annotation delays are common in cloud systems. These challenges reduce the effectiveness of traditional data-driven methods and significantly limit the applicability and scalability of detection models.

2.2 Motivation

In highly dynamic and structurally complex cloud service environments, existing fault detection techniques face critical limitations in generalization and interpretability. Most approaches work well only under specific conditions. When the environment or business logic changes, their performance often drops sharply. This lack of adaptability limits their effectiveness in real-world operations. Researchers have started to introduce structural modeling and semantic-level understanding to maintain stable identification of abnormal patterns, even when service behavior changes unexpectedly. Therefore, it is urgent to explore a detection path that can understand internal system mechanisms and support transferability, to handle diverse failure characteristics under complex conditions[7].

From a practical perspective, the maintenance burden of cloud platforms continues to grow. Manual rule updates and model retraining are costly and cannot meet the demands of large-scale systems for low-latency and highly interpretable automated detection. In this context, building detection frameworks with causal discrimination ability has become an important direction. This approach makes judgments based on the influence between events rather than relying on isolated anomalies. It helps reduce false alarms and improves fault localization accuracy. The core idea is to transform the question of why a failure occurs into a systematic reasoning process. This enhances both the adaptability and depth of inference, aligning with the engineering requirements of large-scale deployment.

At the same time, in real deployments, traditional models often rely on large amounts of labeled data. However, in practical environments, high-quality labels are scarce. This is especially true for rare or novel faults, which are difficult to detect. As a result, there is an urgent need to develop a more general model architecture that can perform anomaly detection under weak supervision or even in unsupervised settings. By building task-independent causal identification mechanisms, it is possible to break the dependency on labeled data and maintain strong robustness in complex conditions. This not only improves the overall fault tolerance of the system but also provides a theoretical foundation and practical path toward more intelligent fault management.

3. Method

3.1 Overall Framework

This study constructs a causal discrimination modeling framework for cloud service fault detection tasks, aiming to improve the ability to identify fault behaviors in complex environments by modeling the causal dependencies between key variables within the system. The overall approach takes

multidimensional monitoring indicator sequences and system event logs as inputs, first constructs a structured variable representation, and extracts candidate causal paths based on structural assumptions. Subsequently, the relationship between variables is modeled by introducing a causal encoder to generate an intermediate representation with causal semantics to support subsequent fault discrimination and relational reasoning. This framework not only supports the location of abnormal behaviors in time series, but also reveals the potential mechanism of action between variables, improving the interpretability and generalization of the detection model from a structural perspective. The overall model architecture is shown in Figure 1.

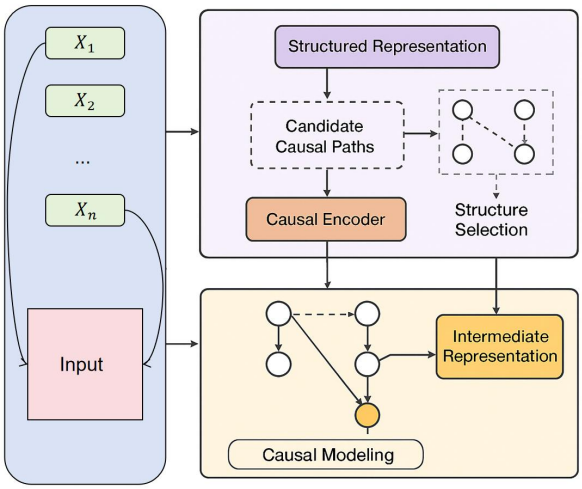

Figure 1. The overall model architecture diagram of this algorithm

In the causal modeling layer, let the variable set be $V = \{X_1, X_2, ..., Xn\}$, and the causal relationship $G = (V, \varepsilon)$ between the variables is represented by a directed graph structure, which ε represents the causal edge set. The framework maps the original input into a causal representation space by constructing a function family $f : X \to Z$, which is then used for predicting the fault state:

$$Z = f(X_1, X_2, ..., X_n)$$

At the same time, the structural selection function $g(\cdot)$ is introduced in the reasoning stage to screen and integrate the candidate causal paths to form the final representation vector for discrimination:

$$H = g(Z, G)$$

This mechanism enables the model to dynamically perceive the key causal factors in the structure, thereby achieving stable fault identification capabilities under complex system states.

3.2 Optimization Objective

In the causal discrimination framework constructed in this study, the input consists of a multidimensional monitoring indicator sequence $\{X_1, X_2, ..., X_n\}$ and a system event log. First, the original input is uniformly encoded to form a basic feature vector set $X = [x_1, x_2, ..., x_n]^T \in R^{n \times d}$. These input features are processed by the structured mapping module and projected into the potential causal representation space, thereby establishing an initial structure diagram for subsequent modeling. To capture the action path between variables, the framework applies the structure function $f_{struct}(\cdot)$ to X to obtain a preliminary structure candidate set $S \in \{0,1\}^{n \times n}$, where each element $s_{i,j}$ represents the possibility of causal influence of x_i on x_j .

Next, the candidate structure S is mapped into a high-order causal representation vector $Z \in R^{n \times h}$ through the causal encoding module $f_{causal}(\cdot)$, that is:

$$Z = f_{causal}(X, S)$$

This representation not only retains the contextual information of the original input but also incorporates structural dependency information, which facilitates accurate recognition of downstream tasks. Subsequently, the framework maps Z to a unified representation $H \in R^{1 \times h}$ through the relational abstraction function $g_{proj}(\cdot)$ to express potential failure modes or system behavior states in a global dimension:

$$H = g_{proj}(Z)$$

In the causal reasoning module, based on the aggregation operation of the reasoning path in the structure graph, the causal decision representation $C \in R^{1 \times h}$ is further generated, which is calculated as follows:

$$C = \sum_{i=1}^{n} a_i \cdot z_i$$

Where a_i represents the path weight calculated by the structure selection mechanism. This aggregation process strengthens the role of key causal paths in the final reasoning, enabling the model to understand and abstract potential anomalies or failures around the causal backbone structure.

Finally, the discriminant layer inputs the causal representation C into the nonlinear mapping module $f_{pred}(\cdot)$ and outputs the fault state prediction result $y \in R^{1 \times k}$, namely:

$$\hat{y} = f_{pred}(C)$$

The prediction vector can be used for multi-category classification or status scoring, providing the system with real-time risk indication and fault judgment basis. The overall process maintains structural explicitness and representation

traceability at each stage, ensuring that the model has clear data flow and structural perception capabilities.

4. Experimental setup & Dataset

4.1 Experimental setup

All experiments in this study were conducted in a high-performance computing environment. The system was configured with 128 GB of memory, a 32-core CPU, and an NVIDIA A100 GPU. The operating system was Ubuntu 22.04 based on Linux. The model was implemented using PyTorch. All experiments were run with the same random seed to ensure reproducibility.

The input data were uniformly preprocessed and then divided into training, validation, and test sets with a ratio of 7:1:2. During training, early stopping was applied to prevent overfitting. Each training iteration used mini-batch inputs, with a batch size of 64. The optimizer was Adam, and the initial learning rate was set to 1e-4.

To comprehensively evaluate the applicability and stability of the proposed method, experiments were conducted under various parameter settings. These included different structure selection depths, causal encoding dimensions, and representation aggregation strategies. All models were trained with the same number of epochs and under identical initialization conditions. Performance comparisons were made using unified evaluation metrics.

The core hyperparameter settings used in the experiments are summarized in the table. These details are provided to ensure the method can be accurately reproduced by others. Detailed experimental settings are presented in Table 1.

Table 1: Experimental detailed parameter settings

Configuration items	Value
Batch size	64
Learning rate	1e-4
Encoder hidden size	256
Structure candidates	20
Aggregation method	Weighted sum
Training/Validation/Test	70% / 10% / 20%
Early stopping patience	10

4.2 Dataset

This study uses the publicly available Alibaba Cluster Trace 2018 dataset as the primary experimental data source. The dataset was collected from real production environments of Alibaba Cloud. It contains multi-dimensional operational information on resource scheduling, container lifecycles, and service state changes in large-scale distributed systems. It is widely used in research on fault prediction and resource optimization in cloud computing scenarios. The dataset exceeds 400 GB in total size and provides rich traces of system state changes and diverse patterns of abnormal behavior. It offers a strong basis for validating causal modeling algorithms under complex conditions.

The dataset includes millions of structured records related to containers, machines, and services. It spans eight days and uses a sampling interval of five seconds. Key indicators include CPU usage, memory usage, disk I/O, service start and stop events, container migrations, and machine reboots. Each record contains detailed timestamps, resource usage values, and service instance identifiers. These fields provide a complete foundation for constructing temporal causal structures. Based on this dataset, it is possible to model system operation and extract potential abnormal patterns for training and evaluating detection models.

To facilitate processing and modeling, the raw data were standardized through several steps. These included outlier removal, time series alignment, missing value imputation, and variable normalization. In addition, annotated samples were created using labels from log records that indicate container restarts, task failures, and resource overuse events. The entire processing workflow preserved the original structural and temporal characteristics of the data. This provides a reliable basis for subsequent causal modeling and discriminative reasoning.

5. Experimental Results

In the experimental results section, the relevant results of the comparative test are first given, and the experimental results are shown in Table 2.

Table 2: Comparative experimental results

Method	Accuracy	FPR	Structural Consistency
Deeplog[8]	87.3	12.6	62.1
LogAnomaly[9]	89.1	10.2	65.3
NeuralLog[10]	90.7	9.8	70.4
CausalDetect[11]	92.8	7.4	78.9
Ours	94.5	5.9	85.6

In terms of overall accuracy, the proposed causal discrimination model achieves the highest detection accuracy of 94.5% in the fault detection task. This significantly outperforms existing representative methods. The result shows that introducing causal structure modeling can effectively improve the model's ability to distinguish fault behaviors. Compared with traditional approaches based on log pattern matching or neural encoding, causal modeling captures the underlying mechanisms between variables more deeply. It enhances the ability to identify complex anomaly patterns, especially in environments with highly nonlinear system states and multivariable interference.

Regarding false positive rate (FPR), the proposed model also demonstrates strong robustness, reaching only 5.9%, which is the lowest among all compared methods. Traditional models often misclassify anomalies under multi-tenant resource fluctuations or transient abnormal states due to the lack of structural awareness. In contrast, the causal discrimination model identifies abnormal paths and avoids noise that is irrelevant to actual faults. The model not only detects fault states but also avoids classifying sporadic fluctuations as anomalies. This is essential for maintaining the operational stability of cloud platforms.

For the structural consistency metric, the proposed method reaches 85.6%, which is significantly higher than CausalDetect

and other non-causal models. This indicates that the model not only focuses on the final classification result but also emphasizes maintaining the true structural logic among input variables. Structural consistency reflects the model's ability to preserve the system's operational mechanisms. A high score shows that the method is more suitable for operation scenarios where interpretability of internal system behavior is required. It adds practical value in terms of model traceability and debugging convenience.

This paper also provides a detailed analysis of how varying the dimensionality of causal encoding influences the overall performance of the model. The discussion focuses on the relationship between encoding capacity and the model's ability to capture underlying structural dependencies among input variables. By examining different dimensional settings, the paper explores how the richness of causal representation affects the model's reasoning ability and generalization in complex environments. The corresponding experimental results are illustrated in Figure 2.

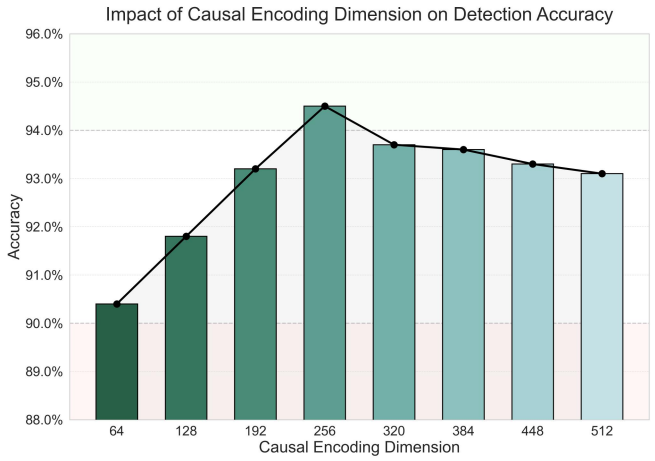

Figure 2. Analysis of the impact of different causal encoding dimensions on model performance

As shown in the figure, the detection accuracy of the model increases significantly when the causal encoding dimension rises from 64 to 256. This trend indicates that with a small encoding dimension, the model has a limited ability to represent causal structures. It fails to fully capture the dependencies and underlying mechanisms among input variables. However, as the dimension increases, the model gains richer representation capacity. This leads to stronger generalization and deeper reasoning in both structural modeling and fault discrimination tasks.

When the encoding dimension reaches 256, the model achieves its highest accuracy. This suggests that the causal representation is most complete at this dimension, and the structural relationships are effectively captured. The intermediate representation space at this point balances information density and discriminative power. It represents an optimal structural point in the causal modeling process. This result further confirms the positive impact of introducing causal modeling mechanisms on fault detection in cloud services. The improvement is especially notable in scenarios with complex variable interactions.

However, when the encoding dimension continues to increase to 384 and beyond, the model performance shows a slight decline. This suggests that overly high-dimensional representations may introduce redundant features or noise. Such interference can affect causal path aggregation and reduce the stability of final predictions. This phenomenon shows that causal structure modeling is sensitive to dimensional settings. It is necessary to find a balance between complexity and expressiveness to ensure effective use of information.

In addition, the paper conducts a sensitivity evaluation on how the number of candidate causal paths influences detection accuracy. This analysis investigates the trade-off between structural coverage and redundancy, aiming to understand how varying the number of causal paths affects the model's ability to capture meaningful interactions within the system. By assessing different path quantities, the paper highlights the importance of structural selection in maintaining effective and focused causal reasoning. The related experimental results are presented in Figure 3.

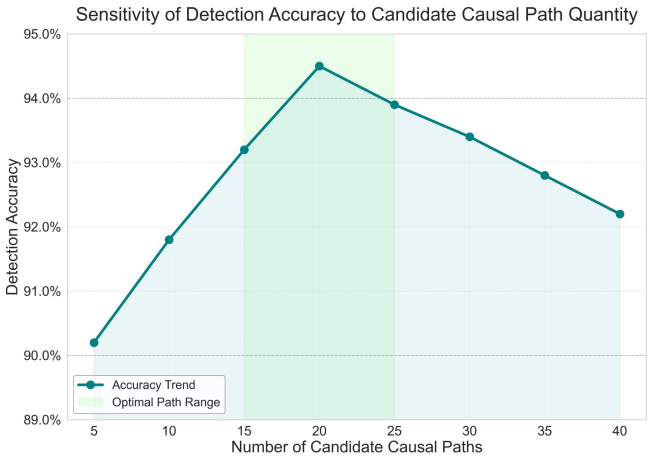

Figure 3. Sensitivity evaluation of the number of candidate causal paths on detection accuracy

As shown in the figure, the number of candidate causal paths has a significant impact on the model's detection accuracy. When the number of candidate paths is small, such as between 5 and 10, the model shows lower accuracy. This indicates that the causal modeling information is insufficient at this stage. The structural space is limited and fails to cover important potential variable relationships in the system. As a result, the overall discrimination performance is affected. This reflects that overly sparse structural awareness reduces the ability to identify fault propagation chains in cloud service environments.

When the number of candidate paths increases to 20, the model reaches its highest accuracy. This suggests that, in this range, the number of causal paths achieves a balance between modeling capacity and redundancy control. At this point, the structural graph covers most of the key interaction dependencies. The model can more comprehensively extract latent causal features for reasoning. This trend confirms the

importance of structural richness in enhancing model discrimination. It is also one of the core mechanisms behind the accuracy improvement observed in this study.

However, when the number of candidate paths continues to increase to 30 and above, the accuracy begins to decline. This indicates that too many paths introduce structural redundancy. Non-critical paths may interfere with the reasoning process and reduce the focus of causal feature extraction. In high-dimensional causal graphs, redundant paths not only degrade representation quality but may also cause the model to over-attend to irrelevant variables. This reduces the precision of reasoning.

Moreover, the paper introduces a disturbance experiment designed to assess the model's ability to recognize input structures under varying levels of noise injection. This evaluation aims to examine how external perturbations affect the stability and reliability of the structural perception module. By simulating noisy conditions, the study explores the sensitivity of causal modeling to input quality and highlights the role of structural integrity in maintaining accurate fault reasoning. The corresponding experimental results are illustrated in Figure 4.

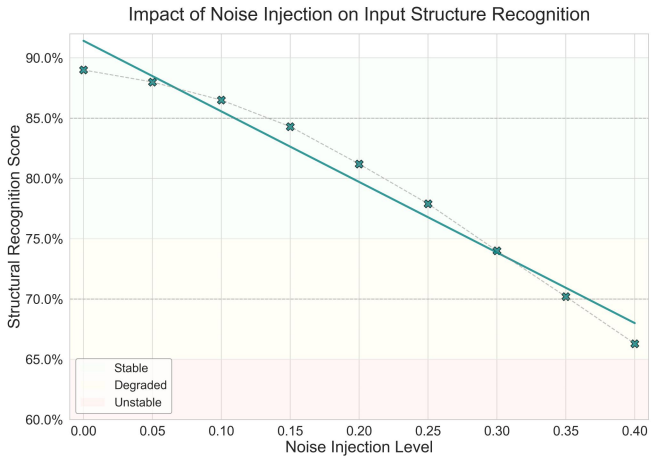

Figure 4. Experiment on the disturbance of input structure recognition ability by noise injection

As shown in the figure, the model's ability to recognize input structure decreases steadily as the level of noise injection increases. This trend indicates that the model's perception of the causal structure among original variables is significantly disturbed when handling noisy data. The decline becomes more pronounced when the noise intensity exceeds 0.2. This result reveals the sensitivity of the structural perception module to input quality. It is one of the key factors affecting the effectiveness of causal modeling.

When the noise level is low, in the range of 0.0 to 0.1, the model still maintains a high structural recognition score. This suggests that the proposed causal representation mechanism shows robustness under mild perturbations. In this stage, the model relies on stable causal features for reasoning. Local noise in the input structure does not break the overall variable dependencies. The structure preservation remains in a "stable" region, which aligns with expectations in cloud environments with minor resource fluctuations.

As noise increases to a moderate level, between 0.15 and 0.25, the structural score begins to decline rapidly. The model output gradually enters a "degraded" state. This shows that with accumulating noise, the influence of key paths or variables becomes weakened or distorted. The model struggles to accurately model the system behavior. This type of interference is common in complex distributed service environments. It reflects the model's strong dependence on structural completeness in real applications.

When the noise level reaches a high intensity of 0.3 or above, the structural recognition drops into an "unstable" region. The model can no longer maintain its ability to model the causal structure of input variables. The recognition of fault propagation paths becomes confused or disrupted. This result further confirms the central role of structural constraints in ensuring model robustness in causal modeling. It highlights the need for structure-enhancing mechanisms and noise-suppression strategies during real deployment. These are essential for preserving reliable structural recognition under unstable operating conditions.

6. Conclusion

This study addresses the core challenge of service failure detection in cloud computing environments by proposing a detection algorithm framework based on causal discrimination modeling. The method explicitly models the causal relationships among multi-dimensional input variables. It enables accurate recognition of fault states and structural awareness. This approach overcomes the limitations of traditional methods, which often rely heavily on predefined anomaly patterns and lack generalization. The model focuses not only on semantic-level representation learning but also emphasizes structural consistency and interpretability of system behavior. This significantly enhances its adaptability and stability in complex environments.

The proposed method integrates structural representation generation, causal path construction, and discriminative encoding into a unified modeling process. This forms a complete structure-aware pathway that allows the model to maintain effective fault chain modeling under information perturbation and data drift. Experimental results show significant improvements in accuracy, structural consistency, and robustness. These findings demonstrate the potential of the causal modeling paradigm for intelligent operations in cloud services. In addition, sensitivity experiments reveal the model's dependence on key structural hyperparameters. This provides theoretical and empirical support for the stable deployment of causal discrimination mechanisms in engineering systems.

Methodologically, this study achieves a paradigm shift from feature awareness to structural awareness. The model focuses on learning mechanism-level logic rather than fitting surface-level patterns. This aligns with the operational characteristics of cloud computing systems, which are highly dynamic, strongly coupled, and heterogeneous. Therefore, the work contributes directly to improving fault detection performance. It also provides practical value for integrating

causal reasoning into intelligent operations. The method offers a new path for building detection models that are more interpretable, more transferable, and more stable. It also presents a feasible reference for introducing causal modeling into related fields.

For future work, several directions are worth exploring. One direction is to further enhance the model's ability to integrate multi-source heterogeneous data. This would improve the adaptability of causal structures in complex data environments. Another direction is to couple causal discrimination mechanisms with modules for root cause localization and anomaly prediction. This would support a more comprehensive intelligent fault management system for cloud computing. As automatic causal structure learning continues to advance, how to construct graphs directly from raw observational data and use them to drive efficient detection will also become an important research topic. Overall, this study provides a systematic framework, methodology, and empirical foundation for applying causal reasoning in intelligent cloud operations. It holds strong theoretical value and engineering potential.

References

[1] Kumari P, Kaur P. A survey of fault tolerance in cloud computing[J]. Journal of King Saud University-Computer and Information Sciences, 2021, 33(10): 1159-1176.

[2] Soldani J, Brogi A. Anomaly detection and failure root cause analysis in (micro) service-based cloud applications: A survey[J]. ACM Computing Surveys (CSUR), 2022, 55(3): 1-39.

[3] Kalaskar, C., & Thangam, S. (2023). Fault tolerance of cloud infrastructure with machine learning. Cybernetics and Information Technologies, 23(4), 26-50.

[4] Tang W, Yang Q, Hu X, et al. Deep learning-based linear defects detection system for large-scale photovoltaic plants based on an edge-cloud computing infrastructure[J]. Solar Energy, 2022, 231: 527-535.

[5] Li H, Hu G, Li J, et al. Intelligent fault diagnosis for large-scale rotating machines using binarized deep neural networks and random forests[J]. IEEE Transactions on Automation Science and Engineering, 2021, 19(2): 1109-1119.

[6] Guntupalli, R. (2023). Optimizing Cloud Infrastructure Performance Using AI: Intelligent Resource Allocation and Predictive Maintenance. Available at SSRN 5329154.

[7] Soni D, Kumar N. Machine learning techniques in emerging cloud computing integrated paradigms: A survey and taxonomy[J]. Journal of Network and Computer Applications, 2022, 205: 103419.

[8] Du M, Li F, Zheng G, et al. Deeplog: Anomaly detection and diagnosis from system logs through deep learning[C]//Proceedings of the 2017 ACM SIGSAC conference on computer and communications security. 2017: 1285-1298.

[9] Meng W, Liu Y, Zhu Y, et al. Loganomaly: Unsupervised detection of sequential and quantitative anomalies in unstructured logs[C]//IJCAI. 2019, 19(7): 4739-4745.

[10] Chen Z, Gao Q, Moss L S. NeuralLog: Natural language inference with joint neural and logical reasoning[J]. arXiv preprint arXiv:2105.14167, 2021.

[11] Zhang, Y., Gong, M., Liu, T., Niu, G., Tian, X., Han, B., ... & Zhang, K. (2021). Causaladv: Adversarial robustness through the lens of causality. arXiv preprint arXiv:2106.06196.