Eng Int Syst (2026) 5: 607–620
© 2026 CRL Publishing Ltd

Engineering
Intelligent Systems

Collaborative Fault Prediction and Intelligent Localization of Distributed Operation and Maintenance Platform Based on Spatiotemporal Graph Neural Network

Jianwei Ma[1,a*], **Tongfu Fu**[2,b], **Ming Lei**[2,c], **Zhongqiang Zhou**[1,d] **and Ling Liang**[1,e]

[1] *Power Dispatching and Control Center of Guizhou Power Grid Co., Ltd., Guiyang 550002, Guizhou, China*
[2] *Liupanshui Power Supply Bureau of Guizhou Power Grid Co., Ltd., Liupanshui 553001, Guizhou, China*

Traditional approaches to Distributed Systems Operations and Maintenance cannot adequately predict faults or locate root causes. To solve these problems, this paper presents a Spatio-Temporal Graph Neural Network (STGNN) as an intelligent framework. The STGNN framework leverages dynamic graphs for both the topological and the state evolution models of the distributed system. The prediction component utilizes deep learning algorithms. The localization component applies cross-node attention and graph reasoning to automatically reconstruct the chain of fault propagation and identify the node that caused the fault. Experimental results indicate that the proposed approach achieves a fault prediction accuracy of 90.2%, and reduces the average root cause localization time for write latency in databases by 35.6%. This paper provides a more effective and reliable means of enabling intelligent operations and maintenance in distributed systems.

Keywords: distributed operations management; spatiotemporal graph neural network; dynamic graph structure; cross - node attention; fault propagation chain

1. INTRODUCTION

The operation of systems that distribute work across multiple components requires approaches that maintain and provide services via digital methods. The components of these systems comprise relationships that change with time and differ between locations. Hence, the problems that occur in one area can affect other areas because of the connections between parts that call other parts and parts that use resources [1–2]. Therefore, there is a need to develop methods that can anticipate problems in addition to locating the source of a problem when it occurs, without requiring human intervention [3–4]. Approaches that examine data from single points, or use structures that do not change, are unable to address these two issues effectively because problems move between components, and this produces delays in providing

*Corresponding author. [a]Email: majianwei2005@163.com, [b]futongfu@163.com, [c]leim0930@163.com, [d]gzdwzzq@163.com, [e]lianglingaa@163.com

information about problems, and produces errors in locating the sources of problems [5–6]. The work presented in this study develops an approach that uses data that changes over time in order to find both the likelihood that problems will occur and the source of the problems. This approach provides improvements for maintaining function and for rectifying a problem without requiring individuals to make decisions.

The analysis of problems in systems operating in multiple locations shows that these conditions (There are temporal correlations between the components of the system, and these correlations vary across different geographical locations) reduce the capacity of approaches that provide prediction of problems and that identify origins of problems. The structure of the system changes over time, and as do the relationships between service operations and the conditions of resources, which produces data showing a strong relationship between system structure and time [7]. The paths that problems follow when spreading through the system are not certain; a problem that appears in one location may spread along different chains of relationships, making their origins unclear [8–9]. This lack of certainty means that prediction models must use reasoning across multiple paths, and models for finding origins must identify the actual source from as many different chains as possible. The relationship between changes in performance at specific locations, and the problems affecting the overall system, is often not clear; hence, models must obtain patterns showing problems in both the relationships between structural elements and in changes over time [10–11]. Therefore, the part of the approach that obtains features must combine information relating to structure and information relating to time in a comprehensive manner that shows differences and that supports the prediction of problems and identifies their origins.

Previous research has advanced in two directions: temporal prediction and graph structure learning, although neither has met the integrated requirements for collaborative fault prediction and the location of the root cause. Methods based on Long Short-Term Memory (LSTM), Temporal Convolutional Network (TCN), and Gated Recurrent Unit (GRU) can model the sequence of single-node indicators and capture temporal patterns, but they neglect cross-node structural relationships [12–13]. While graph neural network methods based on Graph Convolutional Network (GCN) and Graph Attention Network (GAT) can aggregate neighbor information, they often rely on static topology and struggle to adapt to dynamically changing dependencies [14–15]. Recent work has attempted to use temporal graph networks for joint modeling, but the feature fusion stage often employs serial or parallel structures, failing to achieve deep coupling between topological evolution and state evolution in a unified representation [16–17]. This shallow fusion limits the model's ability to resolve complex spatio-temporal dependencies, leading to errors in inferring dynamic propagation paths and making it difficult to accurately reconstruct multi-path fault chains.

This current work provides a framework for examining system issues, using a particular model that considers both structure and time. By means of the proposed approach, systems are created that change as a single structure showing relationships and development, combining features from organization and data that change over time. The framework uses a method for understanding the influence that components have on each other, thereby enabling the identification of issue sources and the path showing how problems develop from the combined representation. This provides a model that identifies and analyses problems in order to find their origins. The study develops a structure showing how a system's organization and condition change simultaneously. The proposed design combines multiple levels to integrate structural relationships with features that may change over time. The framework applies a mechanism using focus and reasoning about structure to reconstruct the path that problems follow, making it esier to find their origins. Results from tests indicate that this framework performs better than other current methods in terms of for both anticipating issues before they occur and finding the sources of problems.

2. A COLLABORATIVE FAULT ANALYSIS FRAMEWORK BASED ON SPATIOTEMPORAL GRAPH NEURAL NETWORKS

2.1 Framework Overview and Collaborative Workflow

In settings where operations occur across systems, different nodes providing service have relationship structures that change over time. These include multiple dimensions in the patterns of calls between components, in the way resources depend on other resources, and in the data from operation records. In this current study, a model is developed to show hoe these relationships change over time. This is done by using a structure that combines space and time in a single form. This approach allows features relating to structure, features relating to time, and effects that occur across nodes to function in a coordinated way within the same framework for analysis, thereby assisting with the prediction of faults and identifying their sources. In the process overall, the modules connect to other modules using operations on the structure, and this produces a flow of data that continues through the system, and the flow results in the organization of the system, as shown in Figure 1.

Figure 1 presents the complete mechanism from input modeling to graph inference and then to decision output, demonstrating the linkage between spatio-temporal feature coupling, cross-node influence measurement, and propagation chain reconstruction. The input and modeling layers construct a dynamic spatio-temporal graph using runtime data, enabling the original indicators to acquire an interpretable carrier within the structural space. The spatio-temporal representation learning layer generates a unified vector under the dual effects of spatial dependence and temporal recursion, giving the system behavior stable expressive power in continuous time [35–36]. The cross-node association modeling layer quantifies the influence strength between nodes in the fused representation, strengthens potential propagation paths through the attention matrix, and forms a sustainable influence mapping on the dynamic structure.

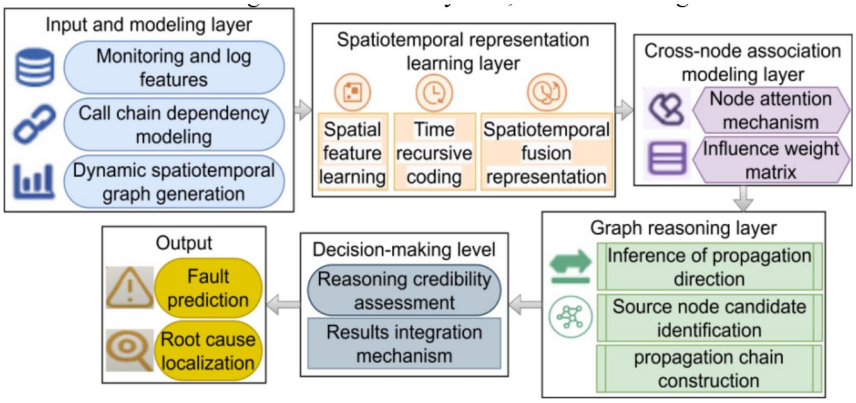


Figure 1 Schematic diagram of the overall framework for collaborative fault prediction and intelligent localization based on spatiotemporal graph neural networks.

The graph inference layer reconstructs the links based on the propagation direction and influence gradient, making the anomaly diffusion process exhibit structural consistency and temporal continuity in the graph domain. The layer that combines results uses a reliability measure that indicates the vadidity of the inference to support prediction work and the process finding root causes [37–38]. This approach allows these functions to produce outputs reflecting multiple sources of evidence combined in analysis. The overall structure used by the method shows complex processes of propagation through connections across multiple layers. This structure allows the identification of patterns appearing in system behavior in terms of space and over time.

2.2 Dynamic Operation and Maintenance Spatiotemporal Diagram Modeling

In distributed environments, operational systems develop relationships that vary depending on service type, resource structure, and hierarchy, revealing the interrelationships between components [18]. This study employs a representation method to depict the overall operational environment of these systems at time t. This representation method uses $G_t = (V, E_t)$ to represent the system structure. The element V represents the set of independent points in the structure. Each point corresponds to a specific service or resource type used by a running system. The element E_t represents the set of connections between points, which change over time. These connections indicate how services call other services and how resources are interconnected, and these patterns also change over time. This study uses $X_t \in R^{|V|\times d}$ to describe the characteristics of points, with dimensions representing the total number of points and d. The d value is a comprehensive dimension that includes monitoring metrics, log data representation, and inter-service call status. This study uses $Z_t \in R^{|E_t|\times q}$ to describe the characteristics of connections, with dimensions representing the total number of connections and q. The q value represents performance status, call latency, available bandwidth, and system load. The structure representing these relationships also changes over time. This study uses the form $\{G_t\}_{t=1}^{T}$ to present these changes as time-varying patterns, where t ranges from 1 to T. This method can be used to examine the evolution of the system structure during operation.

The directed matrix with elements $A_t \in R^{|V|\times|V|}$ showing relationships between nodes is generated using E_t at time t, and values $A_t(i, j)$ show the strength of connection between node i and node j. This strength measure is determined using factors that include how much of the resources are used, how often calls occur between nodes, and how long signals take to move between nodes. The measure combines these factors to provide a single value showing the relationship that results from both the structure of connections and the current state of operations. To maintain the meaning of the structure in the space dimension as conditions change over time, the representation $H_t \in R^{|V|\times h}$ for nodes is updated using operations that apply across the structure. The relationships between points in the structure are obtained with equation 1:

$$H_t = \sigma(\widetilde{A}_t X_t W_s) \tag{1}$$

where, $\widetilde{A}_t$ shows the adjacency matrix that uses a form with changes applied to the values. $W_s \in R^{d\times h}$ indicates the parameter that provides mapping in the structure of the data. The function $\sigma(\cdot)$ represents the operation that applies a change that is not following a direct pattern. The structure itself shows features that differ across time. Ultimately, the operation shows the structure at a particular point in time [19–20].

The operational patterns in the time dimension form state trajectories through node time-encoded sequences $\{X_t\}_{t=1}^{T}$. To characterize the evolutionary patterns across multiple time periods, the temporal recursive unit establishes a recursive relationship between historical and current structural representations as follows.

$$H_t^{temp} = f(H_{t-1}^{temp}, H_t) \tag{2}$$

where, $f(\cdot)$ is a parameterized time transformation function that captures the continuous evolution of performance fluctuations and potential fault signs over time in a state transfer manner.

The graph structure itself also changes over time. To ensure the learnability of the dynamic topology, the evolution of the directed weighted adjacency matrix is represented as follows:

$$A_t = g(A_{t-1}, X_t, Z_t) \tag{3}$$

where, the process updating structure allows the calculation of weights that differ from the previous form. It uses the structure from the prior stage as well as the state of individual points and the state of connections between these points. This process allows ongoing changes in relationships between parts of the system. The relationships between points show changes that follow the particular conditions occurring in the system. The form that results from this approach shows the state of points and also shows dependencies between points in addition to the process that develops over time. This provides a form that combines these different aspects in an approach that is consistent. The representation presents the features of systems that include multiple dimensions in a form that shows structure comprehensively [21–22]. This form provides a basis for the work that follows, including an analysis of the features that occur across space and time. It also allows measurement of effects that occur between different points. that the analysis also shows how problems develop from one point to other points in the system.

2.3 Spatiotemporal Feature Fusion and Collaborative Prediction Module

The main component of this module is a network that combines features across space and time to provide representations of high-quality nodes that support prediction. This network develops a representation $S_t^{(l)}$ that is commonly used and also includes dependencies relating to structure and patterns that occur over time by encoding together the conditions of nodes and changes in connections between nodes. This representation supports the process that determines the probability values in the prediction module and the process that produces values for attention in the module for finding locations. This establishes a connection between the two tasks that operates at a level that is substantial in terms of features. The process uses H_t to represent individual elements and H_t^{temp} to represent changes over time. These representations provide the basis for combining features that relate to structure and to changes over time. The approach that combines these features develops a single system. This system links patterns in connections between elements with patterns in sequences of operations. The structure channel examines relationships between elements. These relationships use the current connection pattern A_t and the current states of elements. The time channel examines patterns that develop across sequences. These patterns show changes that occur in the data over time. The process combines these channels into one structure. This structure allows relationships between elements and changes over time to interact. The interaction occurs in the same representation form that provides a shared basis for both types of patterns. This shared basis allows the system to show effects that involve both structure and time.

The fused comprehensive representation is denoted as $U_t \in R^{|V|\times h}$. To enable spatial dependence and temporal trajectory to form an additive shared semantic, they are compressed into the same mapping domain using a parameter matrix $W_f \in R^{h\times h}$.

$$U_t = \phi(H_t W_f + H_t^{temp}) \tag{4}$$

where, the nonlinear transformation $\phi(\cdot)$ is a continuously differentiable function that provides a unified way of representing the spatial and temporal behaviours. This representation eliminates the potential for feature fragmentation due to independent processing, while also allowing the spatial location and temporally linked historical information to reside together at the same node (i.e. use the same parameterization space), thereby maintaining the compactness of features over time and for all cases [23].

The operating environment shows changes that occur over time in a manner that differs from changes occurring at single points. The representation combining multiple sources of data requires a method to show continuity in the pattern that develops from one time period to the next time period. The approach uses the comprehensive state from the previous moment, which provides the basis for examining patterns across time. This state appears as U_{t-1} in the analysis. A term that indicates flow across time periods provides the means with which to examine continuity. The parameter matrix $W_c \in R^{h\times h}$ applies this term to show patterns developing over time. This produces equation (5) in the analysis. This formula shows the cumulative effect of actions taken over a period of time. It will show the cumulative fault characteristics of failures occurring in the system within a specific period, the cumulative load disturbance characteristics of disturbances occurring in the system, and the cumulative structural fluctuation characteristics of the call chain.

$$S_t = U_t + U_{t-1} W_c \tag{5}$$

This structure perpetuates the spatiotemporal coupling characteristics while maintaining inter-node correlations, enabling the system to form a continuous and traceable representation sequence S_t [24–25] under the combined effects of topology changes and performance fluctuations.

The deep semantics of spatiotemporal feature fusion are enhanced by a stacked structure, which yields the following expression through multi-layer mapping.

$$S_t^{(l)} = \phi(S_t^{(l-1)} W_l) \tag{6}$$

where, $S_t^{(0)} = S_t$ and $W_l \in R^{h\times h}$ represents the parameters of the l-th layer. Multi-layer mapping enables cross-scale representation from local dependencies to system-level propagation paths, allowing the network to maintain feature stability over long periods and respond to abrupt changes in state over short periods [26–27]. Each layer of the fusion structure operates on a unified graph domain, enabling the entire network to form a coherent representation of the joint laws governing instantaneous topology, historical sequences, and structural evolution, providing a high-resolution spatiotemporal semantic foundation for subsequent measurement of cross-node influence and propagation chain inference.

where, $S_t^{(0)} = S_t$ and $W_l \in R^{h\times h}$ represent parameters for layer l-th. The approach using multiple layers allows representation across different scales, showing local relationships and also system-level paths for the way effects occur. This allows the structure to show stable features across long periods and also allows response to changes that occur in short

periods. Each layer in this structure that combines different sources operates on the same form for the overall structure. This allows the structure to develop representation that shows the general patterns relating to immediate structure, patterns following over time, and changes in the overall form. The representation provides a basis with high detail in both space and time for the work that follows. This study examines how individual elements affect other elements and also examines the patterns showing how effects occur across the structure.

The prediction task anticipates the future state of nodes by examining spatio-temporal features. The main focus is on finding patterns that indicate potential problems before they appear. This indicates conditions and signals for later work that examines how problems spread and where they begin. This approach uses representation $S_t^{(l)}(v)$ that results from combining features of space and time. This is a simple-method approach that shows strong capacity for revealing complex relationships and providing estimates of probability for future time points. The prediction function follows this form:

$$P_{t+1}(v) = \sigma(W_p \cdot S_t^{(l)}(v) + b_p) \tag{7}$$

where, W_p and b_p are parameters that the model can train. The output is $P_{t+1}(v) \in 0, 1$. This value indicates the degree of confidence that node v can show failure at time $t + 1$. The approach that combines different forms of representation allows the prediction module to work with multiple factors simultaneously. These factors include the pattern that shows how the node develops over time, the changes in local structure that occur after combining information from neighboring nodes, and the changes throughout the entire system that result from updates to the structure that changes over time. This means that the measure of failure probability indicates changes in local conditions. It also shows the cumulative effect of influence that moves across nodes. The spatio-temporal representation already contains information on patterns from previous time periods, information on how structure creates dependencies between elements, and information on the strength of the spreading process across the network. Because of this, the probability that the model predicts can show an increase before the chain-of-failure spreading is complete. This allows the model to identify the initial indicators that appear before the anomalous condition spreads to other parts of the system.

The sequence showing the probability of possible issues provides the basis for finding where problems occur. This approach allows the method using connections between elements to focus on the likely area of spread in the initial stage when issues first appear. This focus reduces the range that requires examination and decreases the time needed to locate the source of the problem. The design that presents information in one form with meanings that remain the same improves how well the combined approach aligns elements in terms of time and structure. This design allows the predicted information to fit naturally into the process that reconstructs how problems spread. The prediction component maintains a structure that is not complex, but responds effectively to spatio-temporal and to features that change over time. This component provides early warning that is stable and timely and that can be interpreted for combined analysis of problems.

2.4 Intelligent Localization Module Based on Graph Reasoning

The main function of this module is to use features that indicate problems from the earlier stage, to apply these features to create the path showing how the problem moves across the network, and to find where the problem starts in a network those changes over time and space. This module uses two sequential stages. The first stage measures the extent to which the problem affects different parts of the network at that particular time using a method that examines relationships between different parts. This method shows the level of impact that moves from one part to another part. The second stage uses this information about impact levels to trace back through the network. This stage applies reasoning to the network structure, showing the chain that the problem follows. This allows the module to find the source of the problem.

2.4.1 Fault Impact Modeling Based on Cross-Node Attention

The form of attention between nodes uses S_t and the mapping of $S_t^{(l)}$ at different levels to show how changes in node states affect other nodes in the same graph structure [28–29]. This approach reveals how influence moves between nodes. To measure the association between nodes that follows from the structure in the matrix A_t, which shows direction and relative importance of relationships, the analysis uses a value for attention between nodes i and j at time t. This value appears as $\alpha_t(i, j)$. The value is derived from differences that occur in the way that nodes appear in representation, from relationships in structure, and from patterns that remain consistent in meaning after the process combines information related to space and time.

Attention weights are constructed from a trainable parameter matrix $W_a \in R^{h \times h}$ as follows:

$$\alpha_t(i, j) = \frac{\exp((S_t^{(l)}(i)W_a) \cdot S_t^{(l)}(j)^\top)}{\sum_{k \in N_t(i)} exp((S_t^{(l)}(i)W_a) \cdot S_t^{(l)}(k)^\top)} \tag{8}$$

where, $N_t(i)$ represents the group of nodes that show structural relationships with node i at time t. The form using dot products indicates the degree to which states match across nodes in the representation domain for combining data, and the value in the denominator provides a means to establish that weights follow a pattern of standardization within the structural context of related nodes. When a weight shows a high value, this indicates that the effect of state changes in node j on node i at the particular time is more substantial, and this pattern reveals the possible direction that problems spread through the system in the representation space that the method examines.

The core of the impact metric lies in its cumulative propagation across time periods. Since S_t has integrated its operational trajectory and structural evolution, the temporal

continuity of attention is expressed through a recursive term as follows:

$$\beta_t(i, j) = \rho\beta_{t-1}(i, j) + (1 - \rho)\alpha_t(i, j) \tag{9}$$

where, $\beta_t(i, j)$ indicates the degree to which one element affects another across time, and $\rho \in 0, 1$ indicates the factor that provides smoothing. This relationship establishes how patterns develop as the system continues over time. The approach preserves trends that occur in the data. Changes that appear for limited periods do not obscure patterns that remain throughout the analysis. Weak relationships over extended intervals do not affect the chains that operate at particular moments in time [30–31].

The measure of influence provides a matrix M_t in the structure that shows relationships, and $M_t(i, j) = \beta_t(i, j)$ shows the influence that one point provides to a different point at time t. This matrix M_t occurs alongside the matrix A_t that shows connections with direction, and this allows relationships from structure and relationships from influence to appear as separate forms in the data. The influence between different points changes in response to changes in the operating conditions of the system, and these changes follow from updates to the combined representation: $S_t^{(l)}$. This shows the direction of the movements of effects and the strength of influence, and the system reveals these patterns without requiring intervention. This provides a weighted basis with temporal continuity and structural sensitivity for subsequent fault propagation chain construction and root cause node determination.

2.4.2 Fault Propagation Chain Reasoning and Root Cause Determination

Fault chain reasoning is based on directed weighted adjacency matrices A_t and influence weight matrices M_t, transforming the propagation directions between nodes into a traceable sequence of paths in the graph domain. The graph reasoning process relies on the propagation structure of the previous moment and the influence relationship of the current moment to form a continuous link, enabling the entire process of fault occurrence and propagation to be presented in a computable manner on the spatiotemporal graph [32–33]. Considering the input influence strength of any node i at time t, it is defined as:

$$\gamma_t(i) = \sum_{j \in N_t(i)} A_t(j, i)M_t(j, i) \tag{10}$$

where, $\gamma_t(i)$ represents the cumulative influence received by node i from its structural neighborhood under the current topological relationship. This metric superimposes structural association and propagation strength on the same scale, thereby clarifying the relative contribution of the fault propagation direction.

The chain that results from the process appears as a group of nodes $\{v_1, v_2, \ldots, v_k\}$. Each node in this group shows a relationship that increases across the series over time, $\gamma_{t_1}(v_1) < \gamma_{t_2}(v_2) < \cdots < \gamma_{t_k}(v_k)$. This provides consistency between the direction that the propagation follows and the degree to which impact increases in the time dimension. The inferential process uses the current measure of impact change as the basis for extending links. This generates a sequence of nodes while providing that conditions relating to structure allow reaching other nodes and that impact shows an increase that is consistent. This maintains continuation that is logical and conditions relating to structure that are reasonable as the chain relating to faults develops across time. The formation of the link uses stability of $\gamma_t(i)$ on the axis for time. This prevents short-lived noise from controlling the direction of the propagation, and prevents changes in the link that appear at different points without consistency.

The process of locatingof the root cause uses the starting point of the propagation chain as the candidate region and identifies the source node by combining the back-cumulative relationship of influence [34]. To highlight the initiative of the source node in propagation, the back propagation potential of the node is defined as:

$$\delta_t(i) = \sum_{j \in N_t(i)} A_t(i, j)M_t(i, j) \tag{11}$$

where, $\delta_t(i)$ shows the outward movement effect of node i. Nodes in the fault chain that show $\delta_t(i)$ and $\gamma_t(i)$ with significant imbalance in the early stage of movement are candidates for the root cause, with the imbalance appearing most clearly in the initial stage of the link. The final root cause node is the node that shows $\delta_t(i) > \gamma_t(i)$ and appears earliest in the link time pattern. This node has both the characteristics of the end and beginning of motion in both structural and temporal dimensions, so it can be judged according to the evaluation method of the motion source node.

The approach establishes a reasoning process that allows the chain showing propagation to undergo reconstruction that continues under conditions where structure changes over time. This enables the identification of nodes showing root causes that remain stable within a framework that combines measures of influence, paths in structure, and steps following time. The approach appears to be a reliable means of assessing predictions and determining the connections between elements.

3. EXPERIMENTAL DESIGN AND ANALYSIS

3.1 Experimental Setup

The study uses research data provided by Alibaba, and includes information from operations in the actual production setting that the company maintains. This data contains performance measures for multiple dimensions and also includes information on how tasks receive assignment, and on events that indicate system problems during the operation of a large number of servers that work in combination. Performance data in the study contain measures for: the central processing unit, memory use, disk operations involving input and output, and the network capacity used by the system. The data also contains different types of events that occur in the system, including cases when individual machines stop functioning,

when resources reach limits that prevent further use, and when containers require restarting. These features result in patterns showing considerable variation in terms of different types of problems and also result in data that contains many gaps. The data also provides information on relationships between scheduled tasks and the sharing of resources by various components. This allows a structure to be built that shows connections between elements and that changes over time. This structure provides a means for showing how different machines depend on other machines in a model that captures changes in relationships. The data show large scale, show frequent changes in patterns, and contain complex paths through which problems spread from one location to another location in the system. These features provide strong support for studies examining methods that use structures showing relationships across space and time, for studies examining how problems spread through systems, and for studies attempting to determine sequences through which effects propagate. The link that provides access to the data appears at https://github.com/alibaba/clusterdata.

The study uses Ubuntu 22.04 as the operating system. The system includes two NVIDIA RTX 3090 units that provide twenty-four gigabytes of memory for processing, an Intel Xeon Silver 4314 processor, and two hundred fifty-six gigabytes of memory that allow training of structures with changing connections and processing of large amounts of data. The approach uses Python 3.10, PyTorch 2.1, and DGL 1.1 as the main components. The method for all systems uses an initial rate of one times ten to the negative three for optimization. The study uses a process that decreases the rate following a specific pattern. This process reduces the rate to one times ten to the negative five in later stages to provide more stable results. The approach prevents problems when updating values by using a method that limits changes; this method uses a measure of five point zero for all values. The study uses a value of one times ten to the negative five to reduce effects that create problems, and this allows the system to maintain performance across different conditions. Groups of sixty-four examples were used to provide balance between stability and the use of available resources. The study set the total number of training cycles to one hundred twenty for all systems, thereby providing sufficient time for the different approaches to reach stable performance. The data with connection structures required that the system complete construction of representations, updates of connection information, and alignment of features at each point in time to provide a full representation of changes.

To assess the performance of the proposed method in multiple tasks, analysis was conducted using different baselines that represent different approaches to modeling. In the task that predicts issues in the system, LSTM, GRU, and TCN were selected as baselines that examine patterns over time using different structures. The approaches include models that process information in sequence, models using structures that control information, and models that use specific operations to examine patterns. These baselines measure the method's ability to show patterns and predict outcomes in data from individual points without using information about structure. For baselines that examine structure across space, GCN and GAT were selected to compare how relationships between points are represented when the structure does not change over time. This shows how features of structure affect the ability to predict outcomes.

Using these data and the settings for environment and control model, the study conducted an analysis to examine multiple dimensions. The analysis examined prediction performance and pattern consistency over time. It also examined the process of finding relationships across different points in the structure and how combining factors relating to space and time provides value. The analysis examined consistency in the chains, showing how effects develop through the structure. It examined the ability to find the main source of issues with precision. These different forms of analysis provided systematic evidence that confirmed the model's potential to be applied in systems that distribute processes across multiple components with significant complexity.

3.2 Overall Performance Evaluation of Collaborative Tasks

In complex distributed systems with multi-source fault scenarios, the response time of models in localization tasks is often influenced by the combined effects of data dependency structure, sequence change patterns, and node association strength. The latency differences exhibited by different architectures under the same fault characterize their adaptability to anomaly propagation mechanisms. In this phase of the study, a unified observation window was set up for five common types of system faults and constructs a cross-model latency performance distribution through parallel evaluations of LSTM, GRU, TCN, GCN, GAT, and the spatio-temporal framework presented in this paper, in order to characterize the response characteristics of various architectures under different fault perturbations. The experimental results are shown in Figure 2.

Figure 2 shows that in the overall distribution of various fault scenarios, the latency level exhibits a stepped difference between recurrent networks, convolutional structures, and graph structures, with numerical fluctuations related to the complexity of the fault impact path. Taking the database write latency scenario as an example, the response times of LSTM and GRU are 12.4 seconds and 11.7 seconds, respectively. The reason for their higher latency is that the sequential structure has a longer processing chain when facing event jumps caused by lock conflicts. TCN, on the other hand, performs poorly in the same scenario, reaching 11.1 seconds, due to the unstable window-capture capability of the convolutional architecture under non-continuing interference. The latency of GCN and GAT further decreases to 10.3 seconds and 9.6 seconds, respectively. They are more likely to identify core nodes in the case of link blockage with obvious structural correlation, making the inference path more compact. The proposed model uses a process combining relationships across time and between positions to limit how patterns move between points in the structure. This approach allows the model to operate with a measure of 7.1 seconds in this context. The measure shows a reduction of 35.6 % on average when compared to other models. Similar

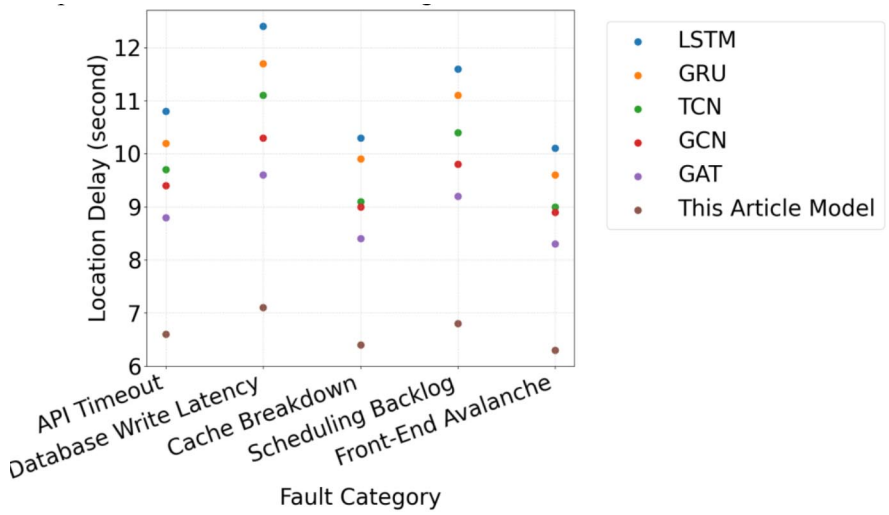


Figure 2 Scatter plot of localization delay for different models under multiple types of faults.

Table 1 Performance comparison of collaborative framework and separated baseline in various failure scenarios.

Fault Category	Collaboration framework MTTR (Second)	Separated baseline MTTR (Second)	Early warning lead time (Second)	MTTR Reduction Ratio
API Timeout	9.4	14.3	3.9	34.3%
Database Write Latency	7.1	11.8	3.5	39.8%
Cache Breakdown	8.3	12.5	3.2	33.6%
Scheduling Backlog	10.5	16.1	4.1	34.8%
Front-End Avalanche	9.8	15.3	4.3	35.9%

relationship patterns appear in different conditions such as when the storage structure shows breakdown or when the sequence of operations shows backlog. The difference in the measure of time indicates how the model responds to the depth of connections between events and to how patterns move between different points. The proposed approach shows consistent measures indicating low values for time across the overall operations.

To further assess the value of combining components, this study compares the proposed framework with a different approach that separates prediction and the finding of locations. This different approach uses the same model for prediction but then uses a separate method to find locations using a static search process. Table 1 shows the performance that the two methods provide in five different scenarios for Mean Time To Repair(MTTR). This comparison indicates differences between the methods.

The results in Table 1 (Collaborative Framework vs. Discrete Baseline) indicate that the collaborative framework outperforms the discrete baseline in terms of average fault resolution time across all five fault scenarios. When looking at the various scenarios, database write latency faults had the largest MTTR reduction (39.8%) due to the fairly clear propagation paths. Front-end avalanche faults, which have larger areas of effect and more complicated responsibility tracing mechanisms, had a slight reduction in MTTR of 35.9% from the collaborative framework's ability to predict and localise the initial spread point (through a predicted 4.3 second early warning to the localising module). These comparisons show beyond any doubt that the collaborative framework presented in this paper makes effective use of prediction time windows and focuses information in optimising the overall timeline from the time of the anomaly occurrence to the identification of the root cause, for different fault classes.

3.3 Performance Comparison Analysis of Sub-Tasks

3.3.1 Fault Prediction Performance Analysis

To compare how different models respond to patterns across time and space, to structure in the form of connections between points, and to changes occurring within individual points in situations involving prediction of system problems, this section provides assessment using five main measures for prediction: accuracy, precision, recall, Area Under Curve (AUC), and Root Mean Squared Error (RMSE). The analysis uses the same data source and the same process for training. This allows examination of how different structures perform when systems show disturbances that occur in actual conditions. The examination places these results in the same framework for analysis. Figure 3 shows the overall performance that each model demonstrates in the task of prediction.

Figure 3 shows significant differences in the performance of each model across various metrics. LSTM performed poorly in multiple evaluations, reflecting the predictive fluctuations of a simple temporal structure under abnormal disturbances. GRU's RMSE decreased to 0.164, indicating

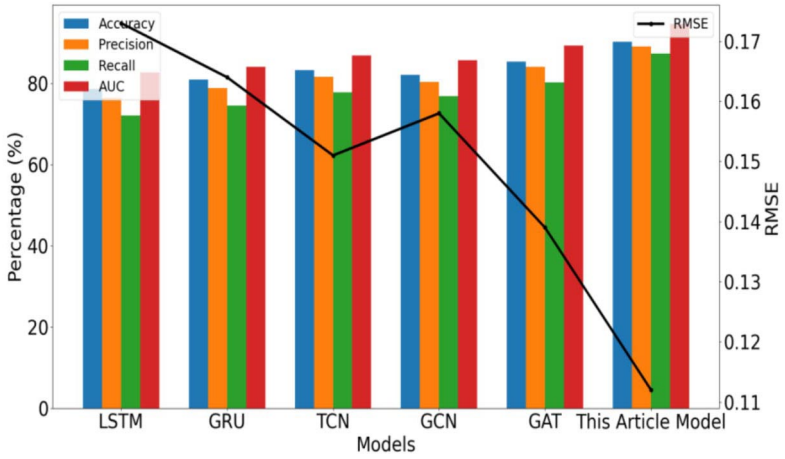


Figure 3 Comprehensive performance chart of multiple indicators for fault prediction.

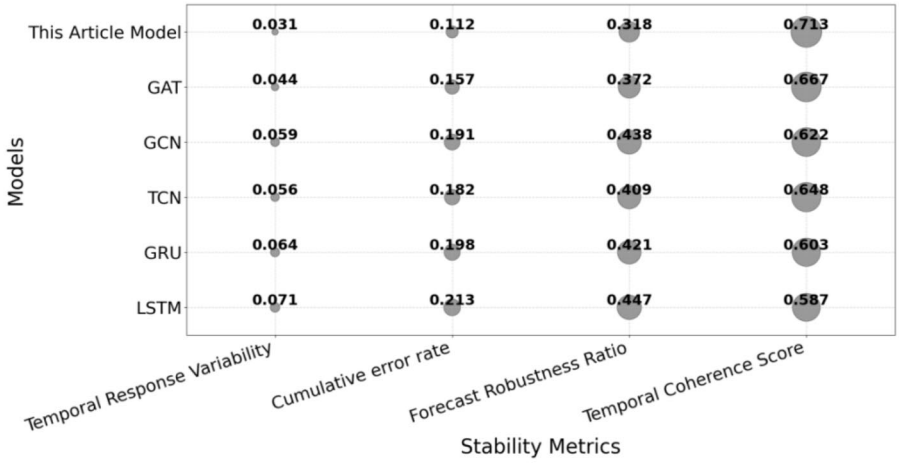


Figure 4 Comparison of multi-model stability indices.

that the overall error of the gating mechanism remained at a high level. TCN performed more stably than the previous two groups, reflecting the stability of convolutional structures in long-term dependency extraction. GCN's overall error was lower than that of the recurrent model but still lower than that of temporal convolution. GAT outperformed GCN in multiple metrics, forming a more concentrated and stable performance range. The model presented in this paper achieved an accuracy of 90.2%, a precision of 89.1%, a recall of 87.3%, and an AUC of 94.6%, while decreasing the RMSE to 0.112. It demonstrated consistent advantages in prediction strength, recognition range, and error control, resulting in the most stable structural characteristics in overall prediction performance.

The evaluation examines stability in different models that provide predictions when system conditions change. This analysis measures differences in performance for models that face changes in operations, changes in relationships between factors, and changes in features of the data. All models use the same data and the same conditions for structure and the same approach for development. The study examines how models show stability, examining patterns that change over time and examining relationships between structural elements. The evaluation uses a single system that allows comparison of different structures. Figure 4 shows performance for different models.

Figure 4 shows differences in patterns of stability for different models. These differences relate to factors that include control of changes, response to errors, and consistency of structure. Results for LSTM indicate particular issues. The structure that follows across time in this model appears vulnerable to increases in changes. This vulnerability produces effects in predictions. Errors show patterns of increase across multiple steps. These patterns of increase result in early indication of limits to stability in the overall model. GRU has certain advantages in suppressing short-term fluctuations, but its lack of modeling of cross-node dependencies limits the improvement in stability. TCN has a significant advantage in reducing the impact of high-frequency noise, resulting in a relatively compact distribution of stability. Under spatial structure constraints, GCN and GAT achieve relative temporal volatility of 0.059 and 0.044, error accumulation rates of 0.191 and 0.157, prediction robustness ratios of 0.438 and 0.372, and temporal consis-

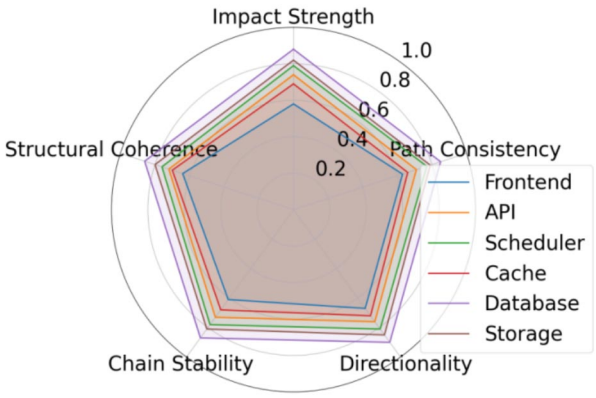


Figure 5 Multidimensional radar analysis of the fault propagation chain reconstruction effect.

tency scores of 0.622 and 0.667, respectively. Structural aggregation ability plays a significant role in improving stability, showing a trend of increasing consistency in temporal consistency scores, reflecting the contribution of cross-node dependencies to error diffusion suppression. The model in this study scored 0.031, 0.112, 0.318 and 0.713 on the four indicators, respectively. The relative temporal volatility and error accumulation rate were kept at the lowest, while the temporal consistency score reached the highest level. This indicates that the spatiotemporal coupling structure has established a more stable cooperative constraint among noise disturbance, structural change and historical dependence, so that the overall stability maintains the strongest consistency characteristics under dynamic conditions.

3.3.2 Root Cause Localization Performance

To examine relationships between problems in systems that include multiple parts with connections, this work develops a method to measure across different aspects. This method allows assessment of how the approach describes features relating to diffusion across space and time from various perspectives. The method follows patterns that appear in six types of components: components for the front-end, components for Application Programming Interface (API), components for scheduling, components for caching, components for the database, and components for storage. These components show particular patterns while problems spread through the system. The work then measures the strength of impact, the consistency of the path, the degree of directionality, the stability of the chain, and the coherence of structure for each component in the relationship chain that the method produces through analysis. Figure 5 shows the relative performance relating to structure for different components in the chain that reconstruction provides.

The pattern in Figure 5 shows that the Database maintains high measures across all features. The measures for influence in spreading, direction in spreading, and features of structure are 0.88, 0.90, and 0.86. These measures indicate that this node occupies a main position in the path for spreading. The Storage shows measures that differ from the Database, with measures for direction in spreading and stability in connection of 0.85 and 0.81. This suggests that while these nodes have the same level in function, the main roles differ. The Scheduler shows measures for consistency in path and direction in spreading of 0.77 and 0.81. These measures reveal the role this node has in connections between processes. The API shows distribution across the five measures that appears more dispersed, with measures for direction in spreading and features of structure of 0.76 and 0.72. This indicates the role that supports that the components in the middle layer have in patterns of spreading. The Cache shows measures that range from 0.66 to 0.72. This suggests differences in the degree that contribution these nodes provide to the path for spreading. The Frontend has the distribution that appears lowest overall. These nodes show 0.58 in measures of impact in spreading and 0.63 in measures of consistency in path. These measures reveal the function that occurs near the interface for users. This function relates to the structure in design that the system has, where the layer at the front provides response to factors that are external. Overall, the structure that is hierarchical relates to the features in the path for spreading that the system in use shows.

Assessment of system stability requires extracting the main features that show distribution of faults from how multiple modules operate in practice. This allows for establishing how different units that provide functions differ in sensitivity when problems develop, spread across the system, and build up over time. The study provides a summary using data that describes events showing problems from modules including the component handling frontend operations, the API, the scheduler, the cache, and the database, using records that include multiple categories of faults. The results that combine this analysis appear in Figure 6.

Figure 6 shows 528 events involving performance reduction in the interface layer. This indicates that the layer experienced continuous high demand during conditions of high simultaneous activity. Events involving resource depletion at the data storage level reached 449. This reflects that operations with high frequency of reading and writing data produce resource consumption at the level that provides data access. Events involving resource depletion in the storage component reached 387. This indicates that storage capacity also shows strong limitation effects in conditions of high demand. Events involving scheduled restarts of the

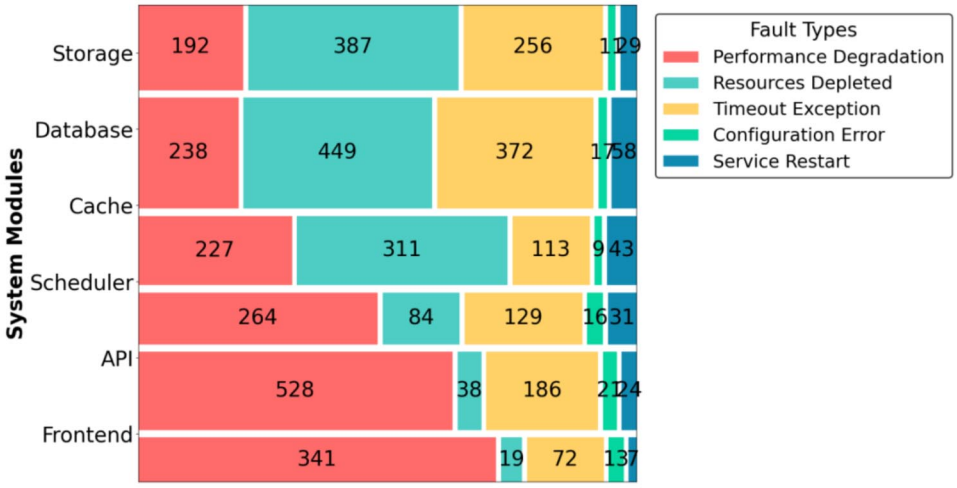


Figure 6 Visualization of system fault type distribution.

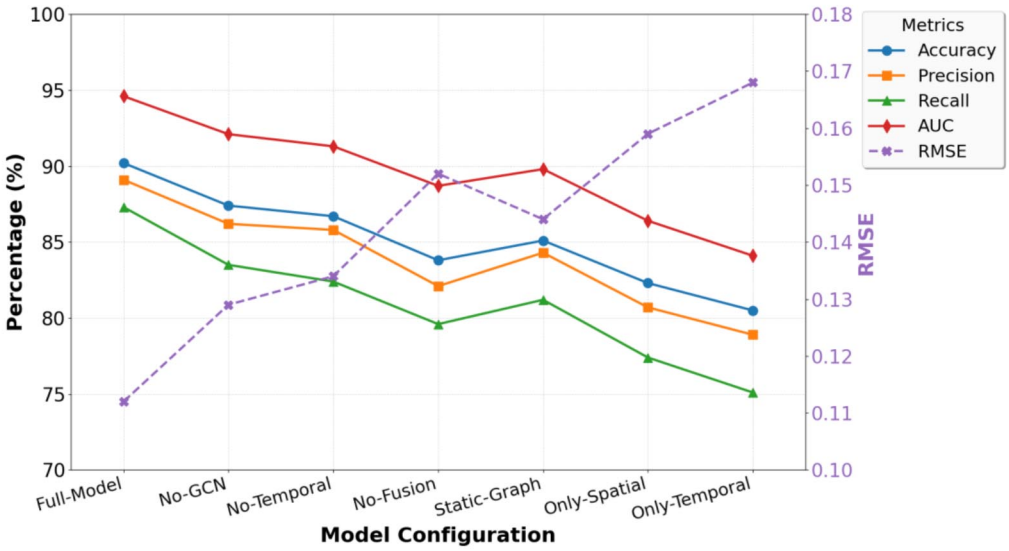


Figure 7 Performance Variation Curves of Module Ablation Experiment.

service layer numbered 31. This indicates that the middle section of the processing sequence shows greater response to accumulated requests in the connection path. The overall total of configuration errors remained relatively low. This reflects that changes to underlying configuration settings produce greater effects on system consistency. Overall, the pattern of different types of failures across components shows a clear structure organized by levels.

3.4 Analysis of Ablation and Contribution of Key Components

In the overall framework of spatiotemporal feature modeling, the roles of each structural module in prediction performance differ, and different functional units participate in structural dependency extraction, temporal state representation, and feature fusion in different ways. Therefore, it is necessary to observe the performance changes when key modules are removed under unified data conditions to determine the relative contributions of spatial encoding, temporal recursion, spatiotemporal fusion, and dynamic graph structure updates in the model's working mechanism. This study constructs multiple control configurations based on the complete STGNN architecture. By gradually disabling different components while maintaining a consistent training process, performance fluctuations can reflect the structural impact of module loss, thereby obtaining the natural performance differences of different sub-modules in the prediction task. The results are shown in Figure 7.

Figure 7 shows that after removing the graph convolutional network, the accuracy dropped to 87.4%, the recall to 83.5%, and the AUC to only 92.1%. This indicates that spatial dependency plays a key role in the anomalous propagation pattern. With the weakening of the correlation between nodes, the model's perception depth of cross-node perturbations decreases. When the temporal recursive unit is disabled, the accuracy drops to 86.7%, the recall to 82.4%, and the AUC to 91.3%, a slightly greater performance degradation than the result of removing spatial dependency.

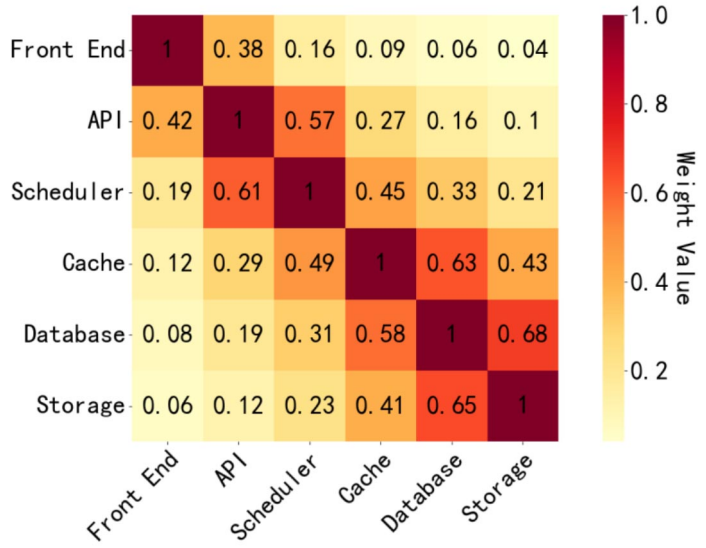


Figure 8 Visualization of cross-node anomaly correlations.

This indicates that the impact of temporal continuity on state preservation is demonstrated in the robust expression of the running trend, while the constraint on structural propagation is insufficient to maintain higher detection coverage. When the fusion module is removed, the accuracy drops to 83.8%, the precision to 82.1%, and the AUC to 88.7%. Different feature streams cannot form stable coupling in a unified semantic space, leading to a weakening of the synergistic ability of spatiotemporal dependencies. When the dynamic graph structure was replaced with a static topology, the accuracy dropped to 85.1%, the AUC was 89.8%, and the RMSE rose to the 0.144 range. After the dynamic adjustment dependent on strength disappeared, the propagation path became fixed. The two models using a single structure show the lowest analytical results. The model examining only those features relating to space shows that the accuracy decreases to 82.3%. The model examining only those features relating to time shows that the accuracy decreases to 80.5%. In these models, the error increases to 0.159 and 0.168, respectively. This indicates that the use of a single dimension cannot adequately indicate the actual behavior of the system as it develops. The findings show that a combination of the space and time features has a significant effect on performance. Also, when the structure changes over time, this also has a significant effect on performance. Other components in the model provide support within their respective dimensions. These components form a source of performance that remains consistent across different conditions in the analysis.

In the stage combining analysis across nodes relating to patterns that differ from the usual, it is necessary to test the model. The model shows ability to find possible relationships between nodes in structures that present complexity in the system. The levels that indicate when patterns differ from usual occur together for the six main nodes within a time window that provides unity. These levels change to form a measure showing relationships. The measure shows relationships between nodes. This measure uses information that is mutual and also uses measures from continuous data that monitor the system. These measures involve finding relationships across observations and also use a process that makes values comparable. Figure 8 shows the particular way these relationships distribute across the analysis.

Figure 8 shows a structure with different levels that relate to one another. The front part of the system has a measure of 0.06 for the database component and 0.04 for storage. This indicates a relationship with limited strength indicating that the front part that provides the interaction does not relate in a direct way to the part that provides resources. The component for system interaction has a measure of 0.57 for the scheduling function, and this function has a measure of 0.61 for the component. These measures indicate a relationship with considerable strength and show that these parts in the middle of the system organize tasks and organize factors that one task requires from other tasks, and manage system activity. This pattern suggests that problems in one part can affect another related part. The storage mechanism that provides rapid data has a measure of 0.63 for the database, and the database has a measure of 0.68 for storage. These measures indicate patterns in the back part of the system in which problems occur that are similar to those in other parts. This is due to factors that involve maintaining consistency in data and factors that involve using resources. This suggests that problems that occur in one part tend to occur in other parts at this level in the system. Overall, the abnormal relationships in the system exhibit a structural gradient that gradually strengthens from the front-end to the back-end. The correlation patterns shown in this heatmap conform to the hierarchical dependency characteristics of distributed systems.

4. CONCLUSION

This work proposes an approach that uses a structure showing relationships and changes over time. The approach examines relationships between elements using analysis of connec-

tions, and it follows changes across time using methods that consider previous conditions. The analysis produces representations that combine information from relationships and time patterns. Using this representation, the work identifies chains showing how problems develop, and determines the sources of problems by examining patterns across different elements and analyzing the structure. This approach helps to anticipate problems before they occur, and to find the locations of problems within a single framework that combines relationship and time information. The process that identifies problems before occurrence provides measures of unusual patterns that limit the time range examined by the process that identifies locations. The pinpointing of these locations shows the chains of development, confirming whether or not the results obtained by the problem-identification process have reasonable structures. This movement of information in both directions increases the reliability of results in situations where operational and management conditions change over time. The data analysis shows that the proposed approach provides significant benefits across multiple categories that were examined. A total of 528 occurrences of decreasing performance were observed in the interface layer of the system, suggesting a strong need to evaluate the method in conditions where demands are high. The results show that the approach maintains consistent performance when examining associations between different elements, when responding to changes in structures over time, and when following chains of problems that persist. Current methods are still unable to deal adequately with the demands of constructing structures representing relationships and from changes in conditions across long sequences when applied to large situations. Changes to structures showing relationships over time have difficulty reaching stable conditions when the structures are extremely dense. Subsequent work will examine mechanisms that respond to structures in more detail, representations that can be used in different domains, and analysis that combines different types of operation-related data to facilitate the identification of problems in systems that are distributed across locations in conditions that change substantially over time.

FUNDING

This work was supported by the Guizhou Power Grid Technology project, distributed collaborative automated operation and maintenance control technology research and system development (GZKJXM20220023).

DECLARATION OF INTERESTS

All authors declare that they have no conflicts of interest.

REFERENCES

1. Yang, J., Gao, T., Yan, G., Yang, C., & Li, G. (2023). A fault location method based on ensemble complex spatio-temporal attention network for complex systems under fluctuating operating conditions. Applied Soft Computing 144 (1), 110489.
2. Dwivedi, A., & Tajer, A. (2023). GRNN-based real-time fault chain prediction. IEEE Transactions on Power Systems, 39(1), 934–946.
3. Leite, D., Andrade, E., Rativa, D., & Maciel, AM (2024). Fault detection and diagnosis in industry 4.0: a review on challenges and opportunities. Sensors (Basel, Switzerland) 25(1), 60.
4. Shao, Q., Fan, S., Zhang, Z., Liu, F., Fu, Z., Lv, P., & Mu, Z. (2025). Artificial intelligence in cable fault detection and localization: Recent advances and research challenges. Energies 18(14), 3662.
5. Wang, Z., Huang, B., Zhou, B., Chen, J., & Wang, Y. (2024). An Enhanced Fault Localization Technique for Distribution Networks Utilizing Cost-Sensitive Graph Neural Networks. Processes, 12(11), 2312.
6. Ma, M., Hua, X., Zhang, Y., & Zhai, Z. (2024). Spatiotemporal polynomial graph neural network for anomaly detection of complex systems. Measurement 235 (1), 115035.
7. Zhao, Z., Xiao, Z., & Tao, J. (2024). MSDG: Multi-scale dynamic graph neural network for industrial time series anomaly detection. Sensors, 24(22), 7218.
8. Qiao, L., Li, X., Wang, X., & Peng, K. (2024). Root cause diagnosis and fault propagation path identification for complex industrial processes based on data space. Measurement 226(1), 114219.
9. Jiang, W., & Bai, Y. (2023). APGNN: Alarm Propagation Graph Neural Network for fault detection and alarm root cause analysis. Computer Networks 220 (1), 109485.
10. Liu, Y., Ren, S., Wang, X., & Zhou, M. (2024). Temporal logical attention network for log-based anomaly detection in distributed systems. Sensors, 24(24), 7949.
11. Song, P., Zhao, C., Huang, B., & Ding, J. (2023). Explicit representation and customized fault isolation framework for learning temporal and spatial dependencies in industrial processes. IEEE Transactions on Neural Networks and Learning Systems, 35(3), 2997–3011.
12. Zhang, J., Wang, X., Yang, Y., Miao, H., & Yang, S. (2025). A novel anomaly detection method for multivariate time series based on spatial-temporal graph learning. Journal of King Saud University Computer and Information Sciences, 37(1), 9.
13. Baz, M. (2024). TVGeAN: Tensor Visibility Graph-Enhanced Attention Network for Versatile Multivariant Time Series Learning Tasks. Mathematics 12(21), 3320.
14. Vrahatis, AG, Lazaros, K., &Kotsiantis, S. (2024). Graph attention networks: a comprehensive review of methods and applications. Future Internet 16(9), 318.
15. Ponzi, V., & Napoli, C. (2025). Graph Neural Networks: Architectures, Applications, and Future Directions. IEEE Access, 13(1), 62870 - 62891.
16. Guo, L., Shi, H., Tan, S., Song, B., & Tao, Y. (2024). A knowledge-driven spatial-temporal graph neural network for quality-related fault detection. Process Safety and Environmental Protection, 184(1), 1512–1524.
17. Wang, Y., Peng, H., Wang, G., Tang, X., Wang, X., & Liu, C. (2023). Monitoring industrial control systems via spatio-temporal graph neural networks. Engineering Applications of Artificial Intelligence, 122(1), 106144.
18. Xue, Z. (2024). Graph learning framework for precise anomaly localization in distributed microservice environments. Journal of Computer Technology and Software, 3(4), 1–7.
19. Zhu, J., Li, B., Zhang, Z., Zhao, L., & Li, H. (2022). High-order topology-enhanced graph convolutional networks for dynamic graphs. Symmetry, 14(10), 2218.
20. Yu, Y., Jiang, X., Huang, D., Li, Y., Yue, M., & Zhao, T. (2024). PIDGeuN: Graph neural network-enabled transient dynamics prediction of networked microgrids through full-field measurement. IEEE Access, 12(1), 49464–49475.

21. Bloemheuvel, S., van den Hoogen, J., &Atzmueller, M. (2024). Graph construction on complex spatiotemporal data for enhancing graph neural network-based approaches. International Journal of Data Science and Analytics, 18(2), 157–174.
22. Islam, M., Chowdhury, M., Rahman, M., & Khan, Z. (2024). Time-to-approach-based model predictive controller for autonomous vehicle safety during unexpected wheelchairs on the path. Cyber-Physical Systems, 10(4), 318–339.
23. Wang, Y. (2023). Advances in spatiotemporal graph neural network prediction research. International Journal of Digital Earth, 16(1), 2034–2066.
24. He, Z., Xu, L., Yu, J., & Wu, X. (2024). Dynamic multi-fusion spatio-temporal graph neural network for multivariate time series forecasting. Expert Systems with Applications, 241(1), 122729.
25. Xu, S., & He, Q. (2024). Spatio-Temporal Dynamic Graph Attention Network-Based Detector for Sea-Surface Small Targets. IEEE Transactions on Aerospace and Electronic Systems, 60(6), 8010–8021.
26. Qian, S., & Zhu, X. (2025). MeshHSTGT: hierarchical spatio-temporal fusion for mesh network traffic forecasting. Scientific Reports 15(1), 20411.
27. Qu, Y., Jia, X., Guo, J., Zhu, H., & Wu, W. (2024). MSSTGNN: Multi-scaled Spatio-temporal graph neural networks for short- and long-term traffic prediction. Knowledge-Based Systems 306(1), 112716.
28. Nguyen, BL, Vu, TV, Nguyen, TT, Panwar, M., &Hovsapian, R. (2023). Spatial-temporal recurrent graph neural networks for fault diagnostics in power distribution systems. IEEE Access, 11(1), 46039–46050.
29. Li, Z., Yu, J., Zhang, G., & Xu, L. (2023). Dynamic spatio-temporal graph network with adaptive propagation mechanism for multivariate time series forecasting. Expert Systems with Applications, 216(1), 119374.
30. Liu, J., Yang, D., Zhang, K., Gao, H., & Li, J. (2023). Anomaly and change point detection for time series with concept drift. World Wide Web 26(5), 3229–3252.
31. Gupta, M., Wadhvani, R., & Rasool, A. (2024). Comprehensive analysis of change-point dynamics detection in time series data: A review. Expert Systems with Applications, 248(1), 123342.
32. Wang, P., & Wen, Z. (2024). A spatio-temporal graph wavelet neural network (ST-GWNN) for association mining in timely social media data. Scientific Reports 14(1), 31155.
33. Nguyen, B., &Chioua, M. (2024). A propagation path-based interpretable neural network model for fault detection and diagnosis in chemical process systems. Control Engineering Practice 150(1), 105988.
34. Chen, Y., Xu, D., Chen, N., & Wu, X. (2023). FRL-MFPG: Propagation-aware fault root cause location for microservice intelligent operation and maintenance. Information and Software Technology, 153(1), 107083.
35. Li, J., Zhang, S., Xiong, H., & Zhou, H. (2022). Autost: Towards the universal modeling of spatio-temporal sequences. Advances in Neural Information Processing Systems, 35(1), 20498–20510.
36. Zhou, Q., Lai, Y., Yu, H., Zhang, R., Jing, X., & Luo, L. (2023). Multi-modal fusion for millimeter-wave communication systems: A spatio-temporal enabled approach. Neurocomputing 555(1), 126604.
37. Li, Z. & Hu, J. (2021). Smart Grid Fault Diagnosis, Positioning Method and Monitoring System Based on Wireless Sensor Network. International Journal of Engineering Intelligent Systems, 29(6), 363–369.
38. Ma, B., Jin, X. & Li, Z. (2023). Power Equipment Fault Diagnosis and Prevention Based on Comprehensive Feature Quantity Evaluation. International Journal of Engineering Intelligent Systems, 31(6),473–482.