# Graph Structure Change-Based Anomaly Detection in Multivariate Time Series of Industrial Processes

Zhen Zhang, Zhiqiang Geng, and Yongming Han, *Member, IEEE*

***Abstract*—Multivariate time series anomaly detection plays an important role for the safe operation of industrial devices and systems. At present, many effective methods have the major limitation that the changes in information propagation between variables are not considered when anomalies occur. Therefore, this article proposes a novel graph structure change-based anomaly detection on multivariate time series (GSC-MAD). First, a stable graph structure under normal conditions is obtained and a single-step prediction for all variables is achieved from a high-dimensional time-series embedding representation learned from the normal data. Then, anomaly detection is achieved by combining the variable behavior deviation reflected by prediction errors and the information propagation deviation between variables reflected by GSC. Extensive experiments on five real-world benchmarks are conducted to demonstrate the effectiveness of the proposed method and compared with current state-of-the-art (SOTA) baselines, a relative improvement of 6.64% on the average F1 is achieved. Moreover, an actual chemical industrial case is provided to verify the effect of the GSC-MAD and a relative improvement of 4.03% is achieved on the F1 metric compared with SOTA baselines. Comparison experiment results show that the proposed method achieves the SOTA results in terms of current baselines. Further experiment analysis shows the good interpretability of the proposed method for detected anomalies.**

***Index Terms*—Anomaly detection, graph neural network (GNN), industrial systems, interpretability analysis, multivariate time series.**

Manuscript received 24 May 2023; revised 16 October 2023 and 3 December 2023; accepted 14 December 2023. Date of publication 11 January 2024; date of current version 4 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62373035 and in part by the Fundamental Research Funds for the Central in China under Grant XK1802-4. Paper no. TII-23-1837. *(Corresponding authors: Yongming Han; Zhiqiang Geng.)*

The authors are with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China, and also with the Engineering Research Center of Intelligent PSE, Ministry of Education in China, Beijing 100029, China (e-mail: zhang_zhen@buct.edu.cn; gengzhiqiang@mail.buct.edu.cn; hanym@mail.buct.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TII.2023.3347000.

Digital Object Identifier 10.1109/TII.2023.3347000

## I. Introduction

With the development of Industry 4.0, the number of industrial devices (entities) and systems is growing rapidly. In addition to their own mechanical abnormalities [1] due to aging, these devices and systems in the industrial Internet may also be subject to external attacks [2], which can cause huge losses if not be detected and dealt with in time. Therefore, it is important to achieve timely and accurate anomaly monitoring for industrial devices or systems to ensure safe industrial operation and avoid economic losses. Moreover, the industrial internet also brings a huge amount of data from different monitoring instruments, which well reflects the operation status of industrial devices or systems. Therefore, the automatic implementation of anomaly monitoring using the industrial data through anomaly detection algorithms has become an active research topic [3].

The monitoring data in the industry are usually presented in time series data, and anomaly detection tends to focus on devices and systems than individual sensors [4], [5]. Moreover, anomaly detection usually takes an unsupervised approach because anomalies in actual industries are rare and labeling the data is costly [6].

At present, many unsupervised multivariate time series anomaly detection methods have been proposed, which can be broadly classified into traditional methods and deep learning-based methods [7]. Traditional methods focus more on mining information in dimensional space [8] or projection space [9] to achieve anomaly detection by means of density [10] or distance [11], [12], but do not consider the inherent temporal dependence of time series, which makes it difficult to make full use of data information. The powerful feature extraction and series representation capabilities in the deep learning have brought promising developments in series modeling and anomaly detection. Recently, prediction [4] or reconstruction-based [5] multivariate time series anomaly detection methods using deep learning have emerged, but these methods tend to ignore the relationships between variables. To address this limitation, the graph neural network-based (GNN) methods capture the relationships between variables in time series modeling, combines the attention mechanism to achieve automatic graph structure learning, and then realizes the anomaly inference by prediction error [2], [13] or reconstruction probability [14]. However, the GNN-based methods still do not consider the changes of data information propagation between variables when an anomaly

occurs, which provides an idea to further improve the performance of anomaly detection. For example, for an actual chemical industry process, under normal conditions, increasing the valve will cause the flow rate to increase, which will cause the tank level to increase. However, when the valve fails, the valve fails are adjusted to achieve the desired changes in flow and level. It is clear that the relationship between the valve, flow rate, and level will change when anomaly occurs, and this change reflected in the graph is the change of the information propagation between the nodes.

Fully considering the changes in the information propagation between variables, this article proposes a new multivariate time series anomaly detection method using the novel graph structure change-based multivariate times series anomaly detection (GSC-MAD). The hidden layer embedding representation of the high-dimensional time series of normal data is learned by GNNs. Then, a single-step prediction of each variable is implemented using the hidden layer representation while learning a stable graph structure. Moreover, an end-to-end training is achieved by joint optimization. Finally, the information deviation of each variable is calculated by combining the variable behavior deviation reflected by the prediction deviation and the information propagation deviation reflected by GSC, which is used to achieve anomaly inference as well as interpretation. The contributions of this article are summarized as follows:

1) A novel GSC-MAD is proposed. Anomaly inference is achieved using the information propagation deviation between variables and prediction deviation of variables.
2) Extensive experiments show that the proposed method achieves the state-of-the-art (SOTA) results across multiple domains on five real-world benchmarks and an actual chemical industrial case compared to current baselines.
3) An anomaly interpretability analysis matrix is provided to help explain the detected anomalies by visualization. The interpretability of the GSC-MAD is demonstrated through an actual real-world water treatment process case study and analysis.

The rest of this article is organized as follows. Section II reviews the related work on multivariate time series anomaly detection. Section III introduces the proposed GSC-MAD method. Section IV analyzes the performance of the GSC-MAD through extensive experiments and gives anomaly interpretability analysis. Finally, Section V concludes this article.

## II. Related Work

Recently, a large number of unsupervised multivariate time series anomaly detection methods has been proposed. This section reviews these methods from the perspective of traditional and deep learning methods.

### A. Traditional Methods for Time Series Anomaly Detection

Traditional multivariate time series anomaly detection methods achieve anomaly detection based on the local density and distance information of data points in high-dimensional space [15]. The local outlier factor (LOF) [10] reflected the local density by calculating the LOF of data at each timestamp, and identified the samples in the region with small local density as anomalies. Isolation forest (IF) [8] determined the local density by continuously dividing the space iteratively, and recognized that samples with few divisions were in low-density space and performed as anomalies. And, similar to the IF, the isolation-based nearest-neighbor ensembles (INNE) [16] utilizes the nearest neighbor method to perform isolation and achieve anomaly detection. The one-class support vector machine (OCSVM) [11] and the support vector data description (SVDD) [12] achieved the segmentation of normal and abnormal data by constructing the optimal classification hyperplane or hypersphere. The principal component analysis (PCA) [9] and the kernel PCA (KPCA) [17] projected the data into a low-dimension space and calculated its statistics at each timestamp to determine whether it is abnormal or not. The histogram-based outlier score (HBOS) [18] assumed independence of the features and achieved a fast anomaly detection method based on the histogram. Recently, both the copula-based outlier detection (COPOD) [19] and the empirical-cumulative-distribution-based outlier detection (ECOD) [20] assumed that anomalies are often the rare events that appear in the tails of a distribution and estimated the tail probabilities per dimension of the data to achieve anomaly detection. However, all these methods ignore the inherent temporal dependence of time series and only use features in spatial dimension to achieve anomaly detection.

### B. Deep Learning Methods for Time Series Anomaly Detection

Based on deep learning and density-based anomaly detection approach, Zong et al. [21] proposed the deep autoencoding gaussian mixture model (DAGMM) by combining deep autoencoder and Gaussian mixture model to achieve anomaly detection through density estimation. Goodge et al. [22] proposed the learnable unified neighborhood-based anomaly ranking method (LUNAR), which learns to use information from the nearest neighbors of each node in a trainable way to find anomalies. Both methods still did not consider the temporal dependence. More generally, considering the inherent temporal dependence of time series, deep learning-based anomaly detection methods were implemented based on prediction or reconstruction methods [7]. A prediction-based method built a time series prediction model from the history data and achieved anomaly inference based on the prediction error [23]. And a reconstruction-based method achieved anomaly inference through the reconstruction error or probability [24]. Hundman et al. [4] used the long short term memory (LSTM) network to implement telemetry data prediction for spacecraft devices, and achieved anomaly detection based on the prediction error and the nonparametric threshold. The generative adversarial network (MAD-GAN) [25] and the autoencoder (AE) [26] were used to model normal time series and achieve the unsupervised anomaly detection (USAD) by reconstruction errors. The Transformer-based anomaly detection method (TranAD) [27] made full use of the temporal dependence of time series by attention mechanism and achieved anomaly inference by series reconstruction. Although the prediction and

reconstruction methods solve the problem in traditional methods with the powerful series modelling ability of deep learning, it does not effectively exploit the relationship of variables.

To address the above limitation, researchers have conducted research on the GNN and verified the ability of the GNN to extract the relationship between variables [28], [29]. Zhao et al. [14] first proposed the multivariate time-series anomaly detection via the graph attention network method (MTAD-GAT), which extracted features in 2-D directions through temporal and spatial graph attention mechanisms, and designed anomaly inference scores by combining prediction error and reconstruction probability. Deng et al. [13] proposed the graph deviation net (GDN) and designed graph deviation scoring to achieve anomaly inference through sensor embedding, graph structure learning, and prediction based on the graph attention mechanism. In addition, Chen et al. [2] used the designed information propagation graph convolution and the transformer architecture to learn graph structures, implemented prediction of the series and realized the anomaly inference by the prediction error.

Most of GNN-based methods utilize the graph attention mechanism to achieve automatic graph structure learning. However, the assumption of full connectivity of a graph [30] or the top-k strategy used for selecting neighborhood nodes [2], random initialization of attention weights [31], and downstream task-based optimization approach may lead to a graph structure that is not practically meaningful or does not have a stable structure. Moreover, although the relationship between variables is effectively utilized in multivariate time series modeling, the way of anomaly inference still utilizes the single prediction or reconstruction information, and does not fully utilize the information of the graph structure reflecting the relationship of variables. Therefore, for graph structure learning, a way of combining similarity metric and graph regularization metric optimizing is used to obtain a stable structure [30], [31]. In anomaly inference, the variable prediction deviation and the information propagation deviation between variables are combined to achieve full utilization of information and anomaly inference.

Moreover, a summary of the related works in form of the table is provided in Appendix A of the Supplementary Material.

## III. Proposed Method

In this work, the GNN-based multivariate time series anomaly detection is focused. To facilitate description and understanding, this section first presents the definition of the multivariate time series anomaly detection task and the graph. Then, a detailed description of the proposed GSC-MAD is given.

### A. Problem Definition

*Definition 1 (Multivariate time series anomaly detection):* Given a multivariate time series $X \in \mathbb{R}^{N\times M}$, where $N$ denotes the length of the series and $M$ denotes the number of features. Anomaly detection task is to algorithmically predict a label $a_t \in \{0, 1\}$ for $x_t \in \mathbb{R}^{M\times 1}$ at each timestamp, where 0 and 1 denotes normal and abnormal, respectively.

*Definition 2 (Graph):* A graph consists of a set of nodes $V$ and a set of edges $E$, denoted $G = (V, E)$. Usually, an edge from node $u$ to node $v$ can be denoted as $e_{u,v} \in E$. For convenience, a graph is usually represented by the adjacency matrix $A \in \mathbb{R}^{|V|\times|V|}$, and $A_{u,v} \in \mathbb{R}$ denotes the weight of the edge $e_{u,v}$. $A_{u,v} = 0$ means there is no edge between the nodes. In GNN-based multivariate time series anomaly detection methods, each variable is considered as a node. And the data for each node is a time series within a window.

### B. Overview

The overview architecture of the GSC-MAD is shown in Fig. 1. The key to the GSC-MAD is to obtain a stable graph structure and achieve anomaly inference by combining the information propagation change in test time series. First, the node embedding representation in the hidden space is learned using a spatial domain GNN, based on which the single-step prediction of all nodes is performed using a full connection neural network. Meanwhile, a new graph structure is obtained by cosine similarity calculation of the embedding representation, edge pruning and weighting with the initial graph, which is used for next computation of node embedding representation. After repeated iterations, a stable graph structure reflecting the normal data can be obtained. In anomaly inference, the anomaly score is obtained by combining the prediction deviation, which reflects the behavior discrepancy and the discrepancy between the new and stable graph structure, which reflects the information propagation change between nodes. Finally, anomaly inference is achieved by the anomaly score and a fixed threshold.

### C. GNN-Based Node Embedding and Forecasting

For multivariate time series, the variables are interdependent and information is propagated to each other. For example, for an actual industrial process, increasing the pressure of the pump will lead to an increase in flow rate, while the level of the tank gradually rises, and under the regulation of the control system, the pump will be counteracted to reduce the pressure appropriately for ensuring the stability of the system. Therefore, it is critical to capture the dependencies and information propagation relationships between variables.

Gilmer et al. [32] proposed a unified framework for the spatial domain GNN. In the GSC-MAD, the information propagation relationships of multivariate time series are extracted using the spatial domain GNN [33], [34] message propagation mechanism to achieve the node embedding in the hidden space. And a nonlinear function is used to perform an activation transformation on the computed results of the network to enhance the nonlinear representation of the network. For each node $\mu$, its embedding representation is calculated by

$$z_u = f\left(w_1 \cdot x_u + W_2 \cdot \sum_{v\in N(u)} e_{v,u} \cdot x_v\right) \tag{1}$$

where $x_u \in \mathbb{R}^{N\times 1}$ denotes the vector of the node $u$ and $z_u \in \mathbb{R}^{d\times 1}$ is the embedding representation of $x_u$ and $d$ is the embedding dimension. $v \in N(u)$ denotes the set of first-order neighbor nodes of node $u$, $e_{v,u}$ denotes the edge weight from node $v$ to node $u$, and $f(\cdot)$ is the nonlinear transformation,

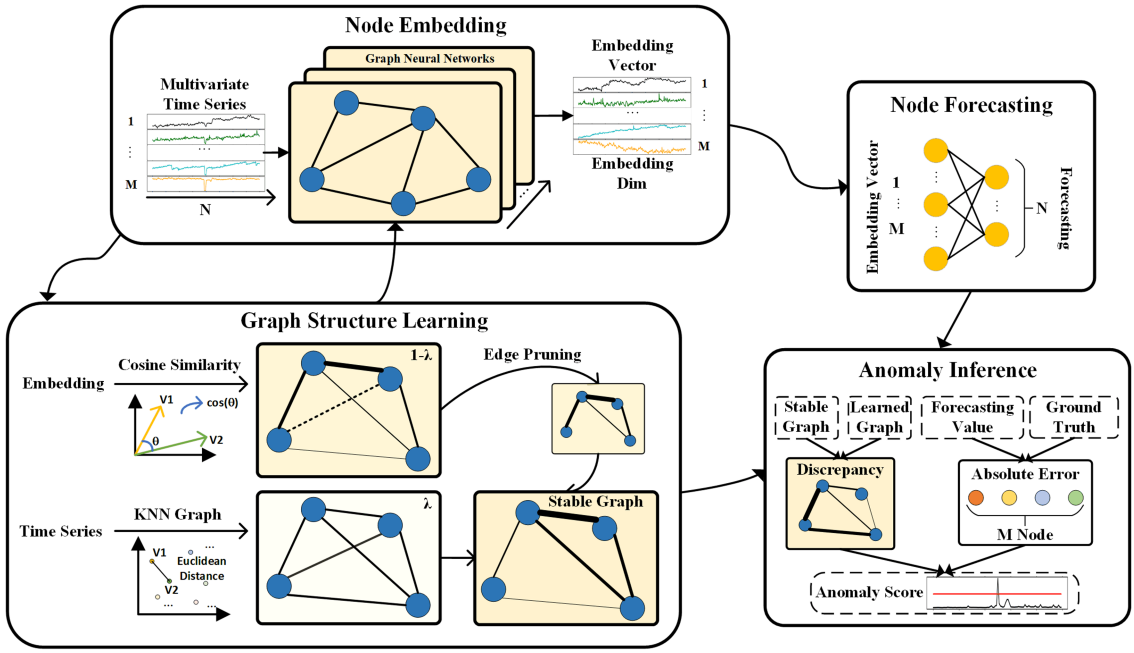


Fig. 1. Overview of the proposed GSC-MAD method. The thickness of the connection line indicates the relationship degree of the variables in the graph structure learning module.

and the ReLU function is used here. The above computation approach captures the relationships between each variable and its directly related variables. Thus, by stacking the above layer, each variable can capture the relationships between itself and multiple-order neighboring nodes (i.e., directly and indirectly related variables).

In addition, combining the inherent temporal dependence of time series (i.e., the future value at each timestamp is based on its history data), a single-step prediction for all nodes is achieved by a full connection neural network using the node embedding, which facilitates anomaly inference and interpretation

$$\hat{y} = Z \times W^T + b \tag{2}$$

where $\hat{y} \in \mathbb{R}^{M\times 1}$ and $Z \in \mathbb{R}^{M\times d}$ denote prediction values and embedding representation of all nodes, respectively.

### D. Graph Structure Learning

Both the MTAD-GAT and the GDN use the graph attention mechanism to achieve graph structure learning. However, the attention-based graph structure learning approach may have some limitations: 1) The assumption of full connection of a graph may affect the graph construction between nodes with real relationships, along with high computation complexity [30]. 2) Random initialization of attention weights may result in a learned graph structure that is not practically meaningful [31]. 3) The downstream task-based optimization approach may not yield a stable graph structure. Therefore, the similarity metric combined with optimizing the graph regularization metric is used to achieve graph structure learning.

A KNN graph $A^0$ is initially established with Euclidean distance, where the edge weight from node $u$ to node $v$ is

$$A^0_{u,v} = \|x_u - x_v\|_2. \tag{3}$$

The node embedding representation obtained through the GNN reflects the implicit information of the variables in the hidden space, and the similarity metric in the hidden space can be calculated by cosine similarity for updating the graph structure

$$s_{u,v} = \cos(z_u, z_v) = \frac{z_u \cdot z_v}{\|z_u\| \cdot \|z_v\|} \tag{4}$$

where $z_u$ and $z_v$ denote the node embedding representation of node $u$ and node $v$, respectively, $\cos(\cdot)$ denotes the cosine calculation, and $s_{u,v}$ denotes the similarity in hidden space between nodes. Furthermore, the similarity values should be non-negative and nodes with minimal similarity do not have ability to influence each other. Therefore, edge pruning is used to prune off the non-significant edges to obtain a similarity matrix for updating the graph structure

$$S_{u,v} = \begin{cases} s_{u,v}, s_{u,v} > \varepsilon \\ 0, \text{otherwise.} \end{cases} \tag{5}$$

Moreover, the initial graph structure reflects the information between nodes in the original space [32]. Thus, both the initial graph structure and the similarity matrix in the hidden space are considered when updating the graph structure

$$A^{\text{iter}} = \lambda \cdot A^{\text{iter}-1} + (1-\lambda) \cdot S \tag{6}$$

where $A^{\text{iter}-1}$ denotes the graph structure at the previous iteration, and when $\text{iter} = 1$ is the initial graph structure $A^0$. Therefore, the information of the initial and each learned graph structure is included in every subsequent iteration. $\lambda$ is used to

control the proportion of information retained about the previous graph at each iteration.

### E. Joint Optimization

As mentioned previously, the proposed GSC-MAD performs both prediction task and graph structure learning task using the node embedding representation in the hidden space, so the node embedding needs to contain the information required by both modules. Therefore, joint optimization is used here.

*Node Forecasting Module:* The task of the prediction module is to predict the value of the next timestamp for all nodes. Therefore, the optimization objective is to minimize the discrepancy between the ground truth and predicted values, which is achieved by minimizing the mean square error (MSE)

$$L_F = \frac{1}{M}\sum_{u=1}^{M}\left(y_u(t) - \widehat{y_u}(t)\right)^2 \tag{7}$$

where $y_u(t)$ and $\widehat{y_u}(t)$ denote the predicted value and ground truth of node $u$ at timestamp $t$, respectively.

*Graph Structure Learning Module:* A stable graph structure can be obtained more efficiently by direct optimization. Based on the target that a good graph structure tends to be low-rank, sparse, and smooth, the graph regularization theory is used to construct the objective function [35]. According to the assumption of graph homogeneity, similar nodes are inclined to build edges, while the variation between similar nodes is smooth. Therefore, a smoothness constraint [36] is imposed as follows:

$$\begin{aligned} L_{sm}(Z, A) &= \frac{1}{2}\alpha \sum_{u,v=1}^{M} A_{u,v}\left(z_u - z_v\right)^2 \\ &= \alpha \cdot tr\left(Z^T L Z\right) \end{aligned} \tag{8}$$

where $Z$ is the hidden embedding representation, $A$ is the adjacency matrix, $L = D - A$ denotes the graph Laplace matrix, where $D = A \cdot \mathbf{1}$ denotes the degree matrix of $A$, $tr(\cdot)$ denotes the trace of the matrix, and $\alpha$ is hyperparameter. Furthermore, since a full connection graph usually has noisy and irrelevantly connection, a sparsity constraint [37] is imposed to force it to become sparse, while avoiding optimizing the graph to $\mathbf{0}$

$$L_{sp}(A) = -\beta \mathbf{1}^T log(D) + \gamma \|A\|_F \tag{9}$$

where $\beta$ and $\gamma$ are hyperparameters, and $\|\cdot\|_F$ denotes the Frobenius norm. The logarithmic barrier forces a graph to become sparse, but does not prevent $A$ from converging to $\mathbf{0}$, so the second term is used to balance the first term to avoid the graph from becoming too sparse. The optimization objective of the graph structure learning module is shown as follows:

$$L_G = L_{sm}(Z, A) + L_{sp}(A). \tag{10}$$

Since both the graph structure learning and the prediction modules are based on the results of node embedding, and the learned graph structure is used in the next iteration of the node embedding calculation, the optimization objective of the GSC-MAD can be obtained by summing the optimization objectives of two modules

$$L = L_F + L_G. \tag{11}$$

### F. Anomaly Inference

For the normal data, the relationships between variables are stable and do not change significantly, but when the test time series contains anomaly, the relationships will change significantly, and the edges that change significantly indicate that the connected nodes may have anomalies. In addition, the GDN points out that variables with large prediction deviations tend to have anomalies [13]. Therefore, the two deviations are combined by multiplying to obtain the information deviation $\mathrm{Err}_{\mathrm{Information}} \in \mathbb{R}^{M\times 1}$ for each node

$$\begin{aligned} \mathrm{Err}_{\mathrm{Information}}(t) = &\left|A^{\mathrm{test}}(t) - A^{\mathrm{steady}}\right|^T \\ &\times \left|\hat{y}(t) - y(t)\right| \end{aligned} \tag{12}$$

where $A^{\mathrm{steady}}$ and $A^{\mathrm{test}}(t)$ denote the stable graph and the graph learned from test series, respectively. $\hat{y}(t)$ and $y(t)$ denote ground truth and predicted values of all variables, respectively. And $|\cdot|$ denotes the absolute error.

Theoretically, $\mathrm{Err}_{\mathrm{Information}}$ calculates the weighted sum of the neighboring edge weight deviations of nodes on their own predicted deviations, and contains both the information propagation deviation between variables and the behavior deviation of the variables themselves, which can better reflect the status of the variables. More specifically, on the one hand, if anomalies exist, the target node deviation can be further amplified by weighting; on the other hand, if no anomalies exist, the target node deviation can be smoothed by neighboring nodes weighting, so as to increase the discrepancy between normal and anomaly. Furthermore, the max value in $\mathrm{Err}_{\mathrm{Information}}$ at timestamp $t$ reflects the device or system status, because if the anomaly exists, the deviation is smaller for unrelated nodes and larger for nodes with anomalies, so the max value is used. Therefore, the anomaly score at timestamp $t$ is obtained using the maximization function

$$\mathrm{Anomaly\_Score(t)} = \max \mathrm{Err}_{\mathrm{Information}}(\mathrm{t}). \tag{13}$$

Finally, anomaly inference is achieved by the $\mathrm{Anomaly\_Score(t)}$ and a predefined threshold. Iif the $\mathrm{Anomaly\_Score(t)}$ at timestamp $t$ exceeds a predefined threshold, the state is marked as anomaly

$$a(t) = \begin{cases} 1, \mathrm{Anomaly\_Score(t)} > \mathrm{Threshold} \\ 0, \mathrm{Anomaly\_Score(t)} \le \mathrm{Threshold} \end{cases} \tag{14}$$

The algorithm flow of the overall method is detailed in Algorithm 1 in Appendix B of the Supplementary Material.

## IV. Experiments

### A. Benchmarks and Evaluation Metrics

The main experiments are conducted on five real-world multivariate time-series anomaly detection benchmarks: 1) Mars science laboratory (MSL) and soil moisture active passive (SMAP) released by NASA [4]; 2) server machine dataset (SMD) published by Tsinghua University [5] and pooled server metrics

TABLE I
THE DETAILS OF BENCHMARK DATASETS

| Dataset | Category | Dimension | Train | Test | Anomaly Ratio |
|---|---|---|---|---|---|
| MSL | Spacecraft (Device) | 55 | 58317 | 73729 | 0.105 |
| SMAP | Spacecraft (Device) | 25 | 135183 | 427617 | 0.128 |
| SMD | Server (Device) | 38 | 708405 | 708420 | 0.042 |
| PSM | Server (Device) | 25 | 132481 | 87841 | 0.278 |
| SWaT | Water Treatment (Process) | 51 | 495000 | 449919 | 0.121 |

(PSM) published by eBay [1]; 3) Secure Water Treatment (SWaT) published by Singapore University of Technology and Design. The details of these datasets are shown in Table I.

Standard evaluation metrics in anomaly detection task are used: Precision (P), Recall (R) and F1 score (F1), which can be calculated by

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{15}$$

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{16}$$

$$F1 = \frac{2 \times P \times R}{P + R} \tag{17}$$

where TP, FP denote correctly and incorrectly detected anomalies, respectively. And TN, FN denote correctly and incorrectly labeled normal, respectively. For the anomaly detection task with highly unbalanced labels, $F1$ is more suitable for evaluating the performance of a detection model. In addition, the performance evaluation metrics are calculated using the currently commonly used point adjustment strategy [4]: for a real-world anomaly detection task, each alarm needs to be treated carefully, so once a consecutive anomaly is detected that overlaps with the ground truth, the entire segment of anomalies is considered to be detected. False alarms are counted and treated in a usual way. And for a fair comparison, the best $F1$ score which is searched for by a grid search on all possible anomaly thresholds is used to analyze the performance of the GSC-MAD and the baselines.

### B. Baselines

Extensive comparison experiments are conducted between the proposed GSC-MAD and 20 different types of baselines for multivariate time series anomaly detection: 1) Traditional methods: the KPCA [17], the OCSVM [11], the IF [8], the HBOS [18], the INNE [16], the COPOD [19] and the ECOD [20]; 2) deep learning and density-based methods: the DAGMM [21], the DeepSVDD [38] and the LUNAR [22]; 3) Prediction-based method: the LSTM [4]; 4) Reconstruction-based methods: the MAD-GAN [25], the OmniAnomaly [5], the USAD [26], the TranAD [27], the CAE-M [3] and the BeatGAN [6]; 5) GNN-based methods: the MTAD-GAT [14], the GDN [13], and the GTA [2]. The GTA is the latest SOTA GNN-based method.

### C. Comparisons on Benchmarks

The implementation details are described in Appendix C of the Supplementary Material. And the performance metrics of the GSC-MAD and other baselines on the five benchmark datasets are shown in Table II. For an unbalanced dataset, the results of $F1_{\text{best}}$ are more meaningful, and the results are analyzed here mainly for $F1_{\text{best}}$. The results show that the GSC-MAD outperforms the current baselines in general and achieves 6.64% improvement on the average $F1_{\text{best}}$. Specifically, it outperforms the current SOTA results by relative 6.32% (93.01→98.89), 2.77% (91.11→93.63), 3.71% (90.41→93.76), 4.19% (89.57→93.32) and 5.00% (91.34→95.91) on five benchmarks of the PSM, the MSL, the SMAP, the SMD, and the SWaT, respectively.

For the OCSVM, the IF, the HBOS, the INNE, the COPOD, the ECOD, the DeepSVDD, the DAGMM and the LUNAR, these methods implement modeling in dimensional space, but do not fully utilize the inherent temporal dependence of time series, which is not conducive to multivariate time series anomaly detection in the absence of partial data information. For the LSTM, the MAD-GAN, the OmniAnomaly, the USAD, the TranAD, the CAE-M, and the BeatGAN-CNN, although the temporal dependence is considered, there are some limitations in capturing the relationships between variables.

For the same GNN-based anomaly detection methods, the MTAD-GAT, the GDN, and the GTA fully capture the relationships between variables. However, in the design of anomaly inference, the GDN and the GTA utilizes variable prediction deviation, and the MTAD-GAT chooses to utilize both reconstruction and prediction deviation, but all ignore the information propagation deviation between variables, i.e., the graph structure change. The GSC-MAD method utilizes the prediction deviation while considering the effect from the graph structure change, and the experiment results show that utilizing both deviations is beneficial to improve the anomaly detection performance.

In addition, the feasibility study with the parameter sensitivity, the computation time and the threshold selection on the benchmarks is detailed in Appendix D of the Supplementary Material, the convergence analysis of the GSL module is detailed in Appendix E of the Supplementary Material and the similarity metric analysis in the GSL module is detailed in Appendix F of the Supplementary Material.

### D. The Real-World Chemical Industrial Case

Fluid catalytic cracking (FCC) is a key technology for secondary processing of crude oil and consists of three systems: reaction regeneration system (RRS), fractionation system, and absorption stabilization system. Among them, the RRS is the core of the FCC, which contains the reaction process and the regeneration process, both of which operate continuously. Its topology is shown in Fig. 2.

The preheated high-boiling petroleum feedstock is mixed with recycle oil slurry, and injected into the riser and reactor to contact and mix with the catalyst from the regenerator, where a catalytic cracking reaction occurs to generate small molecule mixed gases. These produced gasses are discharged from the top of the reactor, while the catalyst particles settle to the bottom through the settler at the top of the reactor. The catalyst coke located at the bottom of the reactor is blown to the regenerator through the pending generation pipe for combustion, regeneration, and recovery of activity. Most of the heat generated by combustion is taken to the external heater to heat the feedstock

TABLE II
PERFORMANCE ($F1_{BEST\%}$) OF THE GSC-MAD AND BASELINES

| Model | | PSM | | | MSL | | | SMAP | | | SMD | | | SWaT | | | F1_avg |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | |
| OCSVM | (2004) | 68.27 | 89.99 | 77.64 | 82.31 | 89.21 | 85.62 | 99.13 | 53.80 | 69.75 | - | - | - | - | - | - | 77.67 |
| KPCA | (2007) | 96.11 | 90.11 | 93.01 | 91.31 | 85.42 | 88.27 | 94.16 | 55.41 | 69.77 | 76.18 | 79.75 | 77.92 | 90.19 | 79.59 | 84.56 | 82.71 |
| IF | (2008) | 95.37 | 86.00 | 90.44 | 80.88 | 85.72 | 83.23 | 94.59 | 55.94 | 70.30 | 84.21 | 51.56 | 63.96 | 86.01 | 83.46 | 84.71 | 78.53 |
| HBOS | (2012) | 95.89 | 85.81 | 90.57 | 91.77 | 87.72 | 89.70 | 96.14 | 55.76 | 70.58 | 66.20 | 50.10 | 57.04 | 97.00 | 76.94 | 85.81 | 78.74 |
| INNE | (2018) | 67.74 | 92.39 | 78.17 | 77.43 | 89.35 | 82.96 | 99.19 | 53.14 | 69.20 | 83.39 | 76.14 | 79.60 | 22.32 | 94.87 | 36.13 | 69.21 |
| COPOD | (2020) | 99.48 | 85.73 | 92.10 | 90.92 | 85.95 | 88.36 | 97.03 | 55.11 | 70.29 | 68.58 | 51.47 | 58.81 | 95.96 | 77.59 | 85.80 | 79.07 |
| ECOD | (2022) | 96.63 | 86.33 | 91.19 | 86.70 | 88.41 | 87.55 | 96.09 | 55.35 | 70.24 | 65.71 | 48.96 | 56.11 | 93.22 | 78.67 | 85.33 | 78.08 |
| LUNAR | (2022) | 70.83 | 88.63 | 78.74 | 82.83 | 55.46 | 66.44 | 95.62 | 54.59 | 69.50 | 66.86 | 70.97 | 68.85 | 54.34 | 75.93 | 63.35 | 69.37 |
| DeepSVDD | (2018) | 89.20 | 86.51 | 87.83 | 88.32 | 86.93 | 87.62 | 99.99 | 35.10 | 51.96 | 91.48 | 57.02 | 70.26 | 99.40 | 72.56 | 83.88 | 76.31 |
| DAGMM | (2018) | 68.24 | 90.39 | 77.77 | 54.12 | 99.34 | 70.07 | 93.15 | 56.53 | 70.36 | 51.51 | 53.77 | 52.61 | 90.52 | 78.34 | 83.99 | 70.96 |
| LSTM | (2018) | 99.99 | 68.21 | 81.10 | 94.35 | 72.89 | 82.24 | 87.18 | 59.07 | 70.43 | 95.60 | 84.25 | 89.57 | 94.10 | 77.00 | 84.70 | 81.61 |
| MAD_GAN | (2019) | 68.00 | 90.07 | 77.50 | 54.03 | 87.79 | 66.89 | 94.17 | 55.26 | 69.65 | 57.75 | 83.14 | 68.16 | 90.09 | 78.34 | 83.80 | 73.20 |
| OmniAnomaly | (2019) | 67.90 | 90.14 | 77.46 | 85.38 | 72.70 | 78.53 | 95.38 | 54.82 | 69.62 | 49.13 | 57.75 | 53.09 | 96.35 | 74.47 | 84.01 | 72.54 |
| USAD | (2020) | 67.70 | 90.12 | 77.32 | 54.78 | 87.79 | 67.46 | 94.30 | 55.26 | 69.68 | 54.81 | 55.13 | 54.97 | 90.01 | 78.34 | 83.77 | 70.64 |
| TranAD | (2022) | 90.05 | 85.70 | 87.82 | 53.98 | 83.02 | 65.42 | 91.90 | 55.61 | 69.29 | 64.04 | 50.48 | 56.46 | 89.01 | 78.34 | 83.33 | 72.47 |
| CAE_M | (2023) | 60.09 | 88.81 | 71.68 | 75.06 | 79.05 | 77.00 | 66.33 | 49.99 | 57.01 | 72.76 | 75.32 | 74.02 | 91.17 | 77.82 | 83.97 | 72.74 |
| BeatGAN | (2023) | 65.93 | 89.23 | 75.83 | 58.89 | 69.68 | 63.83 | 96.58 | 53.80 | 69.10 | 60.38 | 74.30 | 66.62 | 96.27 | 74.99 | 84.31 | 71.94 |
| MTAD_GAT | (2020) | 73.68 | 89.78 | 80.94 | 87.54 | 94.40 | 90.84 | 89.06 | 91.23 | 90.13 | 67.35 | 66.39 | 66.87 | 95.62 | 76.10 | 84.75 | 82.71 |
| GDN | (2021) | 99.93 | 71.45 | 83.32 | 87.80 | 89.05 | 88.42 | 75.62 | 80.97 | 78.20 | 60.25 | 55.52 | 57.79 | 99.35 | 68.12 | 80.82 | 77.71 |
| GTA | (2022) | 91.74 | 82.99 | 87.15 | 91.04 | 91.17 | 91.11 | 89.11 | 91.76 | 90.41 | 88.72 | 83.24 | 85.89 | 94.83 | 88.10 | 91.34 | 89.18 |
| GSC_MAD | (2023) | 97.97 | 99.84 | **98.89** | 94.19 | 93.09 | **93.63** | 89.57 | 98.35 | **93.76** | 92.25 | 94.42 | **93.32** | 96.73 | 95.11 | **95.91** | **95.10** |

Bold indicates the best performance, underline indicates the second-best performance.
- The cost time of the OCSVM exceeds 12 hours for SMD and SWaT datasets.

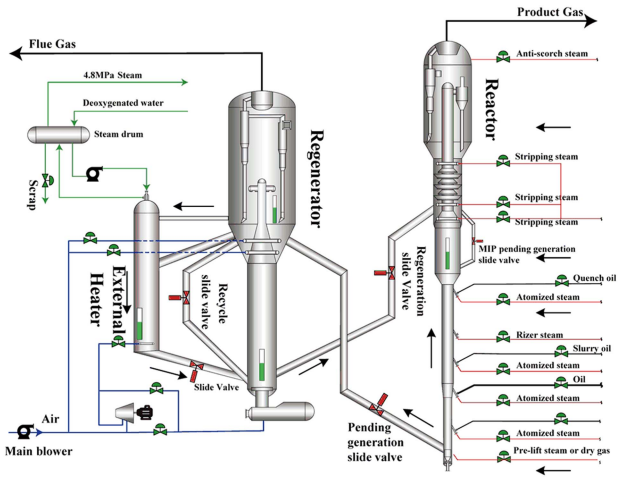


Fig. 2. RRS topology of the FCC.

required for the regenerator and the air blown by the main blower. The above processes are cycled through the reactor and regenerator, constituting the reaction regeneration process of the FCC.

In order to verify the effectiveness of the proposed method for the practical application, anomaly monitoring was implemented using the GSC-MAD for an actual chemical process with the RRS. The experiment dataset (FCC-RRS) was collected from the DCS of an actual chemical industrial production process from 2019 to 2021 (with a sampling period of 3 min) and contains 14 key variables of the RRS: reactor top pressure, regenerator top pressure, external heater ladle pressure, circulating slide valve pressure drop, pending generation slide valve level, recycle slide valve level, reactor material level, regenerator material level, lift pipe outlet temperature, reactor collector chamber temperature, coke tank middle temperature, regenerator bottom temperature, regenerator outlet flue gas temperature, and main blower flow. The FCC-RRS consists of a training set with the normal data and a test set with anomalies, where the length of the training set is 30 000, which is divided into training (80%) and validation set (20%) following standard protocol [2], and the length of the test set is 297522, and the anomaly ratio is 0.027.

From the FCC-RRS anomaly detection results shown in Table III, compared with the existing time series anomaly detection methods, the GSC-MAD improved by relatively 394.44%, 5.59%, 4.30%, 6.30%, 722.38%, 4.89%, 5.59%, 5.63%, 5.62%, 5.74%, 5.74%, 6.70%, 6.68%, 6.69%, 5.73%, 6.68%, 9.49%, 4.03%, 5.88%, and 7.03% than that of the OCSVM, the KPCA, the IF, the HBOS, the INNE, the COPOD, the ECOD, the DeepSVDD, the DAGMM, the LUNAR, the LSTM, the MAD_GAN, the OmniAnomaly, the USAD, the TranAD, the CAE_M, the BeatGAN, the MTAD_GAT, the GDN, and the GTA. Meanwhile, the GSC-MAD is the only method that can detect all anomalies while ensuring a low false alarm rate and a high precision, which is crucial for the actual chemical production process. The experimental results of the actual catalytic cracking RRS show that the proposed method has practical significance in ensuring production safety and avoiding economic losses.

## E. Ablation Study

In order to illustrate the necessity of each component of the GSC-MAD, the anomaly detection performance variation on five datasets is analyzed by stepwise excluding its components. First, the importance of $\text{Err}_{\text{information}}$ for anomaly inference is investigated by directly using the prediction deviation. And the significant of the graph regularization constrain is studied by removing the $L_G$, and the $\text{Err}_{\text{information}}$ is reserved in this study.

TABLE III
PERFORMANCE ($F1_{\text{BEST}\%}$) OF THE GSC-MAD AND BASELINES ON THE FCC-RRS

| Model | Metric | GSC-MAD | OC SVM | KPCA | IF | HBOS | INNE | COP OD | ECOD | LU NAR | Deep SVDD | DA GMM | LSTM | MAD-GAN | Omni Anomaly | USAD | Tran AD | CAE_M | Beat GAN | MTAD-GAT | GDN | GTA |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FCC-RRS | P | 98.34 | 99.73 | 99.11 | 99.32 | 98.84 | 6.42 | 97.68 | 96.80 | **100.00** | **100.00** | 97.89 | **100.00** | 94.22 | 94.26 | 94.23 | 96.69 | 94.26 | 95.16 | **100.00** | 99.73 | **100.00** |
| | R | **100.00** | 11.15 | 89.24 | 89.07 | 88.33 | **100.00** | 91.60 | 93.41 | 88.28 | 88.47 | 90.20 | 88.28 | 91.69 | 91.69 | 91.69 | 91.07 | 91.69 | 86.40 | 91.07 | 88.28 | 86.32 |
| | F1 | **99.16** | 20.06 | 93.91 | 93.91 | 93.29 | 12.06 | 94.54 | 95.08 | 93.78 | 93.88 | 93.89 | 93.78 | 92.94 | 92.96 | 92.95 | 93.80 | 92.96 | 90.57 | 95.32 | 93.66 | 92.66 |

The bold values indicate the best performance.

TABLE IV
ANOMALY DETECTION PERFORMANCE OF THE PROPOSED GSC-MAD AND ITS VARIANTS ($F1_{\text{BEST}}$ %)

| Model | MSL | SMAP | PSM | SMD | SWaT |
|---|---|---|---|---|---|
| **GSC-MAD** | **93.63** | **93.76** | **98.89** | **93.32** | **95.91** |
| - ErrInformation | 89.68 | 82.97 | 97.56 | 91.33 | 92.58 |
| - $L_G$ | 88.94 | 84.99 | 95.54 | 89.72 | 95.41 |
| - ErrInformation | 88.83 | 82.05 | 95.68 | 89.31 | 95.38 |
| - GSL | 85.39 | 73.05 | 94.48 | 79.01 | 83.25 |

The bold values indicate the best performance.

Then, based on the model without $L_G$, only the prediction deviation is used to further investigate the contribution of the $\text{Err}_{\text{information}}$. Furthermore, the importance of graph structure learning is investigated by removing the graph structure learning module, utilizing the initial KNN graph to achieve the modeling and prediction, and also using the prediction deviation.

The experiment results are shown in Table IV, and the following analysis and conclusions can be drawn as follows:

1) Without using $\text{Err}_{\text{information}}$, the performance decreases by relatively 4.2%, 11.4%, 1.4%, 2.1%, and 3.5% for datasets including the MSL, the SMAP, the PSM, the SMD, and the SWaT, respectively. This indicates that the information propagation deviation between variables has a positive effect on the performance improvement for anomaly inference. Moreover, the GSC-MAD is also superior to the GDN with the same use of prediction deviation, which further demonstrates advantages of the GSC-MAD and the necessity of graph structure learning strategy adopted by the GSC-MAD.
2) Without using $L_G$, the performance decreases by relatively 5.0%, 9.4%, 3.4%, 3.8%, and 0.5%, respectively. This sheds light on that besides the GSL module, the graph regularization constrain is also necessary for anomaly detection performance of the GSC-MAD. And a further small decrease in anomaly detection performance occurs with further use of prediction deviation, suggesting that the graph regularization constraint is critical for obtaining a graph structure to further detect anomalies.
3) With further removal of the GSL module, the performance decreases significantly by relatively 8.8%, 22.1%, 4.5%, 15.3%, and 13.2%, respectively. This implies that the graph structure considering only the explicit original multivariate space can hardly express the relationships between variables adequately, and further combining the information of the implicit space is positive to realize relationship representation. Moreover, under the condition of stable graph structure, it is more favorable to achieve anomaly detection by using $\text{Err}_{\text{information}}$.

The above analysis shows that each component of the GSC-MAD is important and has positive effect that can explain the powerful anomaly detection performance.

### F. Interpretation of Anomaly Detection

Another important significance of an anomaly detection method in the real industry is to provide good interpretation for detected anomalies [39]. The interpretability of the GSC-MAD is introduced and analyzed based on several key points: 1) The variable with the max $\text{Err}_{\text{Infromation}}$ is usually the source of the anomaly or with a large anomaly impact, and variables connected to it are noteworthy variables. 2) The graph learned from the test series reflects the relationships between variables. 3) The prediction deviations of variables also have implications for the interpretation of anomalies. Combining the above points, the anomaly interpretability (graph) matrix can be calculated

$$\text{Graph}_{\text{AnoInter}}(t) = A^T_{\text{test}}(t) \odot |\hat{y}(t) - y(t)| \tag{18}$$

where $A_{\text{test}}(t)$ denotes the graph learned by GSC-MAD at timestamp $t$, $|\hat{y}(t) - y(t)|$ denotes the absolute error of the prediction, $\odot$ denotes the element-by-element multiplication. $\text{Graph}_{\text{AnoInter}}$ is the anomaly interpretability matrix, which reflects the influence of each node affected by the neighboring nodes weighted by prediction deviations. By visualizing $\text{Graph}_{\text{AnoInter}}$ using a force-directed layout [40], the relationships between nodes with anomalies and neighboring nodes will be represented in a radial form, which can provide an intuitive explanation of anomalies.

The SWaT dataset is a water treatment process dataset with a detailed report of each anomaly. Therefore, an actual case study of attack 30 with a known cause on the SWaT dataset is used to illustrate the interpretability of anomalies. The details of the attack are described as follows. The system's P-101, MV-101 is off and LIT-101 is set in normal range. The attack sets P-101 and MV-101 to be continuously open and LIT-101 is set to surpass the high limit. The attack finally caused the tank T-101 to overflow. The above attack targets process 1, and the piping and instrumentation diagram (P&ID) for process 1 is shown in Fig. 3.

The variable with the max information deviation is LIT101, which is indeed one of the attack sources according to attack details. In addition, Fig. 4(a) shows the visualization of $\text{Graph}_{\text{AnoInter}}$ for this anomaly. It can be seen that all the variables are distributed in a radial form around LIT101, and the variables with large connection weights are in the same process (P1) and next process (P2), while the other two anomaly sources MV101,

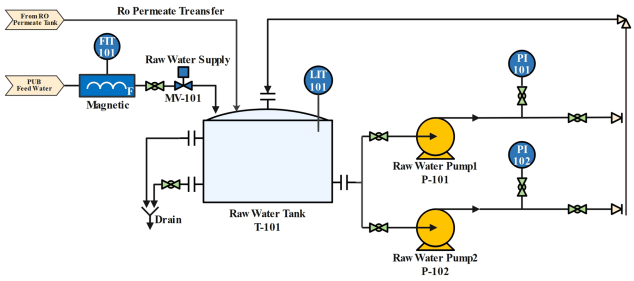


Fig. 3. P&ID of the water treatment process 1.

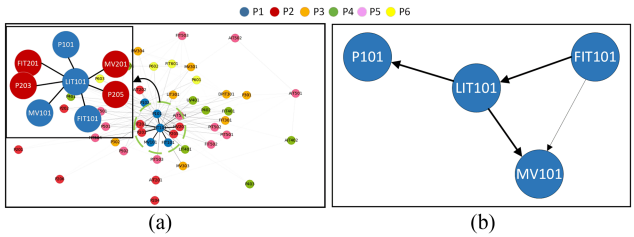


Fig. 4. Anomaly interpretability analysis. SWaT consists of six processes, and nodes of different colors are used to indicate different processes. The thickness of the connection line indicates the relationship degree of the variables, and the distance between the nodes is determined by the relationship degree which is represented by connection weight.

P101 both have strong connection weights with LIT101, which can give good explanations and suggestions to the operator.

In addition, a relationship graph of all variables (except P102) of process 1 in $A_{\text{test}}$ is visualized for this anomaly. As shown in Fig. 4(b), LIT101 affects the states of P101 and MV101, FIT101, and LIT101 jointly affect MV101, and FIT101 has an effect on LIT101. According to the details of the attack and Fig. 3, the LIT101 setting value exceeds the high limit, and the control system will increase the flow rate of FIT101 to achieve the setting condition of LIT101. At the same time, the pump P101 will be adjusted appropriately to safeguard the level LIT101 of the tank T101. MV101 is located between T101 and FIT101 and is subject to both LIT101 and FIT101 in the control system. Therefore, the GSC-MAD effectively learns the relationships between variables. In summary, the GSC-MAD can provide good anomaly interpretation.

## V. Conclusion

In this article, a novel multivariate time series anomaly detection method for industrial processes with the GSC-MAD is proposed. The graph structure learning is suggested to implemented by similarity metric and optimizing graph regularization metrics. In addition, the information propagation deviations between variables is fully considered to achieve anomaly inference combining the prediction deviations. Extensive experiments on five real-world datasets and an actual chemical industrial case demonstrate that the anomaly detection performance of the GSC-MAD outperforms the current SOTA baselines in terms of F1-score metric. And a way to explain anomalies is provided, illustrating the reasonable interpretability of anomalies through a real-world case.

Finally, although the GSC-MAD achieves excellent anomaly detection performance, in the same way as the current extensive researches, it still requires a pre-defined threshold for anomaly detection, which is a threat to practical applications. Therefore, in future work, we will investigate the work related to adaptive threshold setting.

## References

[1] A. Abdulaal, Z. Liu, and T. Lancewicki, "Practical approach to asynchronous multivariate time series anomaly detection and localization," in *Proc. 27th ACM Int. Conf. Knowl. Discov. Data Mining*, 2021, pp. 2485–2494.

[2] Z. Chen, D. Chen, X. Zhang, Z. Yuan, and X. Cheng, "Learning graph structures with transformer for multivariate time-series anomaly detection in IoT," *IEEE Internet Things.*, vol. 9, no. 12, pp. 9179–9189, Jun. 2022.

[3] Y. Zhang, Y. Chen, J. Wang, and Z. Pan, "Unsupervised deep anomaly detection for multi-sensor time-series signals," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 2, pp. 2118–2132, Feb. 2023.

[4] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proc. 24th ACM Int. Conf. Knowl. Discov.*, New York, NY, USA, 2018, pp. 387–395.

[5] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proc. 25th ACM Int. Conf. Knowl. Discov.*, New York, NY, USA, 2019, pp. 2828–2837.

[6] S. Liu et al., "Time series anomaly detection with adversarial reconstruction networks," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 4293–4306, Apr. 2023.

[7] A. A. Cook, G. Mısırlı, and Z. Fan, "Anomaly detection for IoT time-series data: A survey," *IEEE Internet Things.*, vol. 7, no. 7, pp. 6481–6494, Jul. 2020.

[8] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining*, 2008, pp. 413–422.

[9] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang, "A novel anomaly detection scheme based on principal component classifier," in *Proc. IEEE Found. Directions Data Mining Workshop*, 2003, pp. 172–179.

[10] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," *SIGMOD Rec.*, vol. 29, no. 2, pp. 93–104, May 2000.

[11] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.

[12] D. M. Tax and R. P. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, 2004.

[13] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 5, pp. 4027–4035.

[14] H. Zhao et al., "Multivariate time-series anomaly detection via graph attention network," in *Proc. IEEE Int. Conf. Data Mining*, 2020, pp. 841–850.

[15] S. Schmidl, P. Wenig, and T. Papenbrock, "Anomaly detection in time series: A comprehensive evaluation," *Proc. VLDB Endow.*, vol. 15, no. 9, pp. 1779–1797, May 2022.

[16] T. R. Bandaragoda, K. M. Ting, D. Albrecht, F. T. Liu, Y. Zhu, and J. R. Wells, "Isolation-based anomaly detection using nearest-neighbor ensembles," *Comput. Intell.*, vol. 34, no. 4, pp. 968–998, 2018.

[17] H. Hoffmann, "Kernel PCA for novelty detection," *Pattern Recognit.*, vol. 40, no. 3, pp. 863–874, 2007.

[18] M. Goldstein and A. Dengel, "Histogram-based outlier score (Hbos): A fast unsupervised anomaly detection algorithm," *KI-2012: Poster Demo Track*, vol. 1, pp. 59–63, 2012.

[19] Z. Li, Y. Zhao, N. Botta, C. Ionescu, and X. Hu, "Copod: Copula-based outlier detection," in *Proc. IEEE Int. Conf. Data Mining*, 2020, pp. 1118–1123.

[20] Z. Li, Y. Zhao, X. Hu, N. Botta, C. Ionescu, and G. Chen, "Ecod: Unsupervised outlier detection using empirical cumulative distribution functions," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 12, pp. 12181–12193, Dec. 2023.

[21] B. Zong et al., "Deep autoencoding Gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Representations*, 2018.
[22] A. Goodge, B. Hooi, S.-K. Ng, and W. S. Ng, "Lunar: Unifying local outlier detection methods via graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2022 vol. 36, no. 6, pp. 6737–6745.
[23] Y. Wang, X. Du, Z. Lu, Q. Duan, and J. Wu, "Improved LSTM-based time-series anomaly detection in rail transit operation environments," *IEEE Trans. Ind. Inf.*, vol. 18, no. 12, pp. 9027–9036, Dec. 2022.
[24] L. Li, J. Yan, H. Wang, and Y. Jin, "Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 3, pp. 1177–1191, Mar. 2021.
[25] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "Mad-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," in *Proc. Int. Conf. Artif. Neural Netw.*, Springer, 2019, pp. 703–716.
[26] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. A. Zuluaga, "USAD: Unsupervised anomaly detection on multivariate time series," in *Proc. 26th ACM Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 3395–3404.
[27] S. Tuli, G. Casale, and N. R. Jennings, "Tranad: Deep transformer networks for anomaly detection in multivariate time series data," *Proc. VLDB Endow.*, vol. 15, no. 6, pp. 1201–1214, Feb. 2022.
[28] X. Pan, X. Cai, K. Song, T. Baker, T. R. Gadekallu, and X. Yuan, "Location recommendation based on mobility graph with individual and group influences," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8409–8420, Aug. 2023.
[29] Y. Qi, J. Wu, A. K. Bashir, X. Lin, W. Yang, and M. D. Alshehri, "Privacy-preserving cross-area traffic forecasting in ITS: A transferable spatial-temporal graph neural network approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 12, pp. 15499–15512, Dec. 2023.
[30] Y. Chen, L. Wu, and M. J. Zaki, "Deep iterative and adaptive learning for graph neural networks," 2019, *arXiv:1912.07832*.
[31] B. Fatemi, L. El Asri, and S. M. Kazemi, "Slaps: Self-supervision improves structure learning for graph neural networks," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34, Red Hook, NY, USA: Curran Associates, Inc., 2021, pp. 22667–22681.
[32] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learn. Ser. Mach. Learn. Res.*, 2017, pp. 1263–1272.
[33] C. Morris et al., "Weisfeiler and leman go neural: Higher-order graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2019, vol. 33, no. 01, pp. 4602–4609.
[34] Y. Wang et al., "Contrastive GNN-based traffic anomaly analysis against imbalanced dataset in IoT-based its," in *Proc. IEEE Glob. Commun. Conf.*, 2022, pp. 3557–3562.
[35] Y. Zhu et al., "A survey on graph structure learning: Progress and opportunities," 2021, *arXiv:2103.03036*.
[36] M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering," in *Advances in Neural Information Processing Systems*, T. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14, Cambridge, MA, USA: MIT Press, 2001.
[37] V. Kalofolias, "How to learn a graph from smooth signals," in *Proc. 19th Int. Conf. Artif. Intell. Statist. ser. Mach. Learn. Res.*, 2016, pp. 920–929.
[38] L. Ruff et al., "Deep one-class classification," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 4393–4402.
[39] G. Srivastava et al., "Xai for cybersecurity: State of the art, challenges, open issues and future directions," 2022, *arXiv:2206.03585*.
[40] S. G. Kobourov, "Spring embedders and force directed graph drawing algorithms," 2012, *arXiv:1201.3011*.

**Zhen Zhang** received the B.E. degree in measurement and control technology and instrumentation, in 2018, and the M.S degree in control science and engineering, in 2021, from Beijing University of Chemical Technology, Beijing, where he is currently woorking toward the Ph.D. degree in control science and engineering.

His research interests include time series anomaly detection, forecasting, and industrial process monitoring.

**Zhiqiang Geng** received the B.E degree in process equipment and control engineering and the M.E degree in chemical process machinery from Zhengzhou University, China, in 1997 and 2002, respectively, and the Ph.D. degree in control science and engineering from Beijing University of Chemical Technology, China, in 2005.

He is currently a Professor with the College of Information Science & Technology. Beijing University of Chemical Technology. His research interests include neural networks, intelligent computing, data mining, knowledge management and process modeling.

**Yongming Han** (Member, IEEE) received the B.E degree in computer science and technology and the Ph.D. degree in control science and engineering from Beijing University of Chemical Technology, China, in 2009 and 2014, respectively.

He is currently a Professor with the College of Information Science and Technology, Beijing University of Chemical Technology. His current research interests include power system modeling, neural networks, intelligent computing, data mining and intrusion detection.