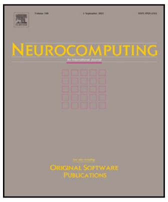
Neurocomputing 619 (2025) 129168

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Enhanced graph diffusion learning with dynamic transformer for anomaly detection in multivariate time series

Rong Gao [a,b], Jiming Wang [a], Yonghong Yu [c,*], Jia Wu [d], Li Zhang [e]

[a] School of Computer Science, Hubei University of Technology, Wuhan, 430068, China
[b] State Key Laboratory for Novel Software Technology at Nanjing University, Nanjing, 210023, China
[c] College of Tongda, Nanjing University of Posts and Telecommunications, Yangzhou, 225127, China
[d] Department of Computing, Faculty of Science and Engineering, Macquarie University, Sydney, NSW 2109, Australia
[e] Department of Computer Science, Royal Holloway, University of London, Surrey TW20 0EX, UK

ARTICLE INFO

Communicated by F.G. Guimaraes

Keywords:
Transformer
Graph diffusion
Anomaly detection
Multivariate time series

ABSTRACT

In recent years, deep learning-based multivariate time series anomaly detection methods have significantly improved detection performance by accurately modeling the spatiotemporal correlations in sensor data. However, they still face the following challenges: the complexity of signals acquired from sensors makes it difficult to capture the true hidden spatial relationships between sensors. Local features and global correlations of time series data are ignored in capturing accurately the temporal correlations of multivariate time series. To address these problems, we propose a **G**raph **D**iffusion **T**ransformed **S**patiotemporal model(**GDTS**) for multivariate time series anomaly detection. Specifically, we first extend a newly developed diffusion probability model to spatial correlation modeling of sensors, which results in a new graph diffusion network. During the diffusion process, we design a novel L2 regularized weighted dot product diffusion function with energy constraints to guide information propagation for accurately capturing the hidden spatial dependencies between sensors. Moreover, we develop a variant of the transformer model with a hybrid sampling strategy to capture the local features and global correlations of time series, which comprehensively describes the temporal correlation of time series data. Subsequently, the multivariate time series features are jointly optimized at the spatio-temporal level for prediction. Finally, the threshold method is used to measure the deviation between the actual observed value and the predicted value to accomplish anomaly detection. Extensive experiments on real-world publicly available datasets demonstrate that GDTS significantly outperforms several state-of-the-art methods.

1. Introduction

Industrial systems use multiple sensors to collect data presented as multivariate time series, reflecting the operational status of the CPS(Cyber-Physical Systems). Timely and accurate multivariate time-series anomaly detection can alert potential events, which is very important for the healthy operation of CPS systems [1]. Therefore, multivariate time-series anomaly detection has become one of the important research tasks for damage detection, data leakage prevention, and security vulnerability identification in the industrial field [2].

Currently, the quantity of sensor data has become increasingly massive and complex. Simple univariate time series anomaly detection methods do not meet the requirements of industrial systems for the healthy operation of information-physical systems. Therefore, the research on multivariate time series anomaly detection methods has gained popularity and has been widely studied. For example, soil moisture detection plays an important role in the sustainable development of ecosystems. Soil moisture anomaly is judged by analyzing the data collected by sensors through anomaly detection algorithm. The data collected from sensors at different locations are often presented in the form of time series data. Generally, changes in vegetation cover data lead to changes in the collected topographic data, resulting in changes in soil moisture. These variations are profoundly affected by variations in the inherent temporal features of time series and complex spatial relationships between sensors. Then, the predictive method is used to predict the normal value of the soil moisture data based on the spatio-temporal features captured by the joint optimization. Ultimately, anomalies are diagnosed by measuring the deviation between

[*] Corresponding author.
E-mail addresses: gaorong@hbut.edu.cn (R. Gao), 102201065@hbut.edu.cn (J. Wang), yuyh@njupt.edu.cn (Y. Yu), jia.wu@mq.edu.au (J. Wu), li.zhang@rhul.ac.uk (L. Zhang).

https://doi.org/10.1016/j.neucom.2024.129168
Received 26 December 2023; Received in revised form 11 November 2024; Accepted 7 December 2024
Available online 16 December 2024

actual observations values and predicted values using a threshold-based method.

Traditional methods such as autoregressive methods [3], and clustering methods [4] are mostly unsatisfactory because most of them do not take advantage of the superior performance coming from modeling data in both spatial and temporal dimensions, as well as distinguishing and dynamically modeling temporal feature long and short pattern interactions. Thanks to the outstanding nonlinear fitting ability of deep learning, methods based on autoencoders [5], recurrent neural networks [6], and generative adversarial networks [7] have made significant progress. Several studies [8–10] have improved anomaly detection performance by simultaneously capturing both hidden spatial relationships between sensors and local–global correlations of time series at the spatio-temporal level. However, these methods still face many difficulties in dealing with realistic scene anomaly detection, which makes it difficult for the model to learn effective spatio-temporal feature representations [11]. In recent years, graph neural networks (GNNs) have demonstrated remarkable capabilities in spatio-temporal modeling and have been widely used in solving various spatio-temporal sequence prediction problems. Some studies have used GNNs to extract spatio-temporal features from sensors for graph modeling and achieved better detection results [12,13]. However, two key challenges remain:

(1) Most methods use the GNN model [10,14] to describe the spatial correlation between nodes, which is often disrupted by the inherent complexity of sensor signal data. Therefore, it is difficult to accurately capture the true spatial hidden dependency relationships among multiple sensors based on graph structures [15]. Some works propose using ϵ-radius [16] or k-nearest neighbor [17] to create relatively simple graph structures or directly introducing undirected graphs for graph modeling, which results in the presence of isolated nodes or redundant structures in the constructed graph structure, leading to non-existent relations being added to the graph structure. Some studies take a different approach by introducing the diffusion idea [18] into the graph construction process and proposing to use diffusion convolutional operation to capture spatial correlations [19]. However, the process of diffusion modeling based on graph structure may lead to systematic bias between the structured representation of data nodes based on graph learning and the real data node dependencies due to the influence of factors such as noise, which makes it impossible to accurately expose the real correlation between the data [20,21].

(2) Different sensors have different characteristics and respond differently to changes in the system state, which makes it difficult to comprehensively consider the impact of time series data. While current RNN/LSTMs for extracting nonlinear temporal features have difficulty in capturing long-term dependencies, transformer [22], which is based on an attentional mechanism, has demonstrated a strong ability to model long-term dependencies in time series [23,24]. However, as the literature [23] points out the traditional transformer architecture can only perform point-by-point level dot product attention on time series data without considering the time lag effect among nodes, which enables neither the effective modeling of local features of the time series nor the comprehensive extraction of global correlation of the time series.

To address the above issues, we propose a novel Graph Diffusion Transformed Spatiotemporal model (GDTS). First, we construct a spatial graph structure based on the current sensor nodes. Then, we fit the dynamic changes of sensor node data at different layers through the diffusion of different nodes on the graph. In the diffusion process, we use a new L2 regularized weighted dot product diffusion function based on an energy-constrained function to guide the direction of information transfer among different sensors, reducing the systematic bias in the graph learning process. At each layer, features are propagated among different nodes, and connected weights are adaptively updated. A newly designed energy function helps the diffusion process to be regularized layer by layer until it reaches a solid internal consistency, which ultimately captures the complex global spatial topology of sensors. Meanwhile, for modeling sensor temporal correlation, we design a transformer variant model called Tformer to capture comprehensively temporal dependencies from both local features and global correlations of the sensors. We begin with key feature extraction of the time series data and encode the information in the temporal dimension through relative positional embedding and value embedding. Then the down-sampling convolutions are employed to capture the local temporal features of the sequence, while a weighted multi-layer transformer architecture models the global correlations of each subsequence's local features. Finally, the integrated local–global features are reconstructed by using up-sampling convolutions for sequence structure. Unlike traditional stacking approaches, we effectively integrate both the local information of time series and their global correlations, significantly improving the performance of temporal correlation modeling. Finally, we predict the anomaly scores and judge the anomalies based on the automatic selection of thresholds. We conduct comprehensive experiments demonstrate the efficiency of GDTS.

The main contributions of this paper are as follows:

(1) We design a novel graph diffusion network model for inter-sensor spatial correlation modeling, which employs a novel L2 regularized weighted dot product diffusion function for inter-sensor information transfer, in which the direction of information transfer from different sensors is steered by introducing energy constraints. Our model reveals the potential of applying diffusion representation learning techniques in capturing sensor spatial dependencies.

(2) We propose a novel transformer variant model for sensor-to-sensor temporal correlation modeling which comprehensively extracts complex time series features by considering local features and global correlations of time series data. A down-sampling convolution strategy is developed to obtain local features of multivariate time series, while the weighted multilayer transformer architecture is employed to extract the global temporal correlation of the series, and finally, adaptive incorporation is achieved based on the up-sampling convolution strategy.

(3) Comprehensive and detailed experimental results based on multiple datasets show that our proposed model can perform anomaly judgment efficiently. For the application scenario in Fig. 1, we performed anomaly detection using the collected soil dataset SMAP. Among them, we designed a novel energy-constrained L2 regularized weighted dot-product diffusion model for graph-structure modeling of multivariate time-series soil moisture data enabling automatic learning of complex spatial dependencies. Meanwhile, we developed a hybrid sampling strategy to extract short-term local temporal dependence patterns among soil moisture time-series data and discover long-term patterns of time-series trends. Then, the spatio-temporal features captured by the above network were fused by GRU. The anomalies predicted by the threshold-based spatio-temporal features were compared with the true values to determine whether the data were anomalous. The experimental results showed that GDTS facilitated the comprehensive extraction of soil spatio-temporal features, where detection of anomalies based on the prediction method increased the sensitivity of the model to non-significant anomalies in soil moisture.

2. Related work

In this section, we review important research works related to the topic of this paper in 2 parts: Multivariate time series anomaly detection and graph and transformer for spatiotemporal modeling.

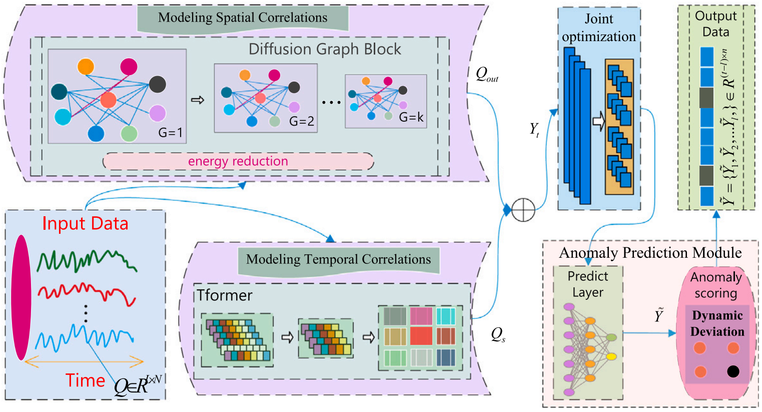


Fig. 1. The framework of GDTS model.

2.1. Multivariate time series anomaly detection

Multivariate time series anomaly detection is a complex task, which is usually modeled in a relatively simple way by these traditional methods for correlation relationships in time series data [25]. However, with the increase in dimensionality resulting in dimensional catastrophe, traditional methods are no longer able to meet the complex and highly nonlinear modeling requirements in the real world. Currently, deep learning-based methods have received widespread attention for their ability to not only improve anomaly detection based on high-dimensional datasets but also to infer correlations between time series. Zhou et al. [26] designed a new deep autoencoder, which not only can discover high-quality nonlinear features but also eliminates outliers and noise. The MST-GAT algorithm proposed by Ding et al. [27] applied a multimodal GAT to detect anomalies. Although these methods are widely used, they do not take into account the correlation dependencies between variables, which can lead to information loss [28]. Aiming at the shortcoming of AutoEncoder's method that is easy to overfit, Du et al. [29] introduced GAN, and accurately captured the distribution of normal data to achieve multivariate time series modeling after generative adversarial training. Zhang et al. [30] introduced a multi-scale convolutional encoder to learn the complex dependencies among temporal multivariate variables, while a convolutional LSTM model incorporating an attention mechanism was also applied to extract the time-domain feature information, and then the anomaly scores were computed based on a reconstruction method.

Deng et al. [12] integrated a structural learning approach with a graph neural network by learning a graph of dependencies between sensors identifying and interpreting deviations from these relationships and providing interpretability of detected anomalies with attentional weights. Miao et al. [31] used a data enhancement technique with geometric distribution masking, integrating a transformer-based self-encoder and a generative adversarial network framework to capture the underlying distributions of normal data and incorporate contrast loss into the discriminator to effectively tune the generative adversarial network and ensure generalization capabilities. Kong et al. [32] proposed a new integrated deep generative model that combines bidirectional long and short-term memory and attentional mechanisms with generative adversarial networks to make the generator and discriminator structures with attentional mechanisms and bidirectional long and short-term memory for anomaly detection. Chen et al. [33] integrated a graph structure learning method based on Gumbel-softmax sampling with a transformer-based architecture to accurately model the correlation and dependency between sensors and spatio-temporal features for anomaly detection. spatio-temporal features for anomaly detection.

2.2. Graph and transformer for spatio-temporal modeling

Due to the strong capability of Graph Neural Networks (GNNs) in handling relational dependencies, many researchers have introduced graph-based learning into the processing of multivariate time series and achieved promising results. Wu et al. [17] proposed a combination of graph convolutional networks and 1D dimensional temporal convolutional networks to extract feature information from both spatio-temporal domains simultaneously, while a course learning-based approach was used to segment the time series and achieve efficient training. Geng et al. [34] argue that temporal changes are influenced by their associated information and therefore devise an approach for spatio-temporal modeling that incorporates LSTM and graphical models. Li et al. [35] proposed a new dynamic graph neural network with an adaptive propagation mechanism for modeling the dynamic relationship between variables in spatio-temporal data, and a gated stacked convolutional network is used to deal with the time dependence. Marisca et al. [36] developed a new reconstruction method to fill in the missing data of the time series, which is to efficiently learn to learn the dynamical graph network node representations in time and space by introducing a sparse spatio-temporal attention mechanism based on inductive biases.

In recent years, the Transformer model has achieved better results in numerous machine-learning tasks [24]. Therefore, many researchers have introduced it in time series modeling tasks with impressive results. By considering the inherent constraints of the transformer, Zhou et al. [37] designed a novel transformer architecture that employs a self-attentive distillation method to efficiently deal with extremely long input sequences and thus effectively deal with long-term dependencies in time series. Xu et al. [38] designed a new multivariate time-series-based transformer structure for anomaly detection, which calculates the a priori and series associations of time nodes separately by association learning method, and then performs anomaly assessment by designing a very large and very small strategy based on the association anomalies of these two associations. Cirstea et al. [39] proposed a new lightweight transformer structure for the multivariate time forecasting problem, which optimizes the architecture of the popular classical transformer to maintain its complexity at a linear level by using a triangle structure based on patch attention while simulating different temporal patterns based on a lightweight approach. Zhang et al. [40] introduced pre-training as an effective method to improve performance, which is used to model long-term and short-term temporal correlations adaptively based on the frequency domain through the combination of self-supervised signals and a pre-training framework that accurately captures temporal dynamic changes. Recently, Wen et al. [41] deeply analyzed the various applications of transformers in the field of time modeling and summarized the existing problems of transformers and the fruitful results achieved by the researchers from multiple perspectives, which further pointed out the future development direction of transformer in the field of time modeling.

The proposed GDTS model is different from the above methods: (1) We introduce a new graph diffusion method for capturing the complex global spatial topology of sensor relationships, which utilizes L2-weighted diffusion functions with energy constraints to guide the direction of information transfer from different sensors, helping the diffusion process with novel energy constraint to regularize layer by layer until it reaches a solid internal consistency. (2) For sensor temporal correlation modeling, we design a novel variant transformer to capture the hidden key information of multivariate time series from both local and global aspects. We use down-sampling convolution to extract local features from the time series, then adopt the multilayer weighted transformer architecture to model the long dependencies by capturing the global correlation in the local features. Ultimately, effective local and global integration is achieved by using up-sampling convolution.

3. Modeling

3.1. Formalization of the problem

First, we define the input sequence $Q = \{Q_1, Q_2, \ldots, Q_L\} \in \mathbb{R}^{L\times N}$, We define the vector $\{y_1, y_2, \ldots, y_L\} \in R^L$ as the output, which $y_T \in \{0, 1\}$ denotes whether or not an anomaly occurs at the timestamp T.

The data $Q = \{Q_1, Q_2, \ldots, Q_L\} \in \mathbb{R}^{L\times N}$ for the i time stamps before the moment T is selected to predict the value $\tilde{Q}_T$ at the moment T. Consistent with the literature [10], we judge the anomalies based on threshold analysis of the relationship between true values and predicted values.

3.2. Overall model architecture

In the area of sensor spatial correlation modeling, we devise a new graph diffusion method to capture the complex global spatial topology of sensor relationships. As far as we know, it is the first to introduce an L2-weighted diffusion model with energy constraints to the field of multivariate time series anomaly detection. This model guides the direction of spatial information transfer between different sensors according to the dynamic idea and helps the diffusion process to be regularized layer by layer a novel energy constraint model, which enables the dynamic propagation and filtering of information, mitigates the interference of noise, and achieves the optimization of spatial sensors' depth features.

In the area of sensor temporal correlation modeling, inspired by CVPR2023's use of a sampling strategy for comprehensive extraction of local–global information in remote sensing images [42], we designed a novel Tformer architecture based on a hybrid sampling strategy, based on a hybrid sampling strategy and a transformer architecture that captures the dependencies of a multivariate time series from both local and global aspects.

The framework of the GDTS model is shown in Fig. 1, which consists of a spatio-temporal feature extraction module and anomaly prediction module. The spatiotemporal feature extraction module consists of three components: spatial correlation modeling, temporal correlation modeling, and joint optimization module. Specifically, in the spatial correlation modeling of sensors, we adopt a new diffusion graph model, which firstly constructs an initial adjacency matrix based on the initial node information and combines it with a diffusion function equipped with energy constraints to obtain an initial diffusion graph of the data. Then, residual connections are introduced to enable multilayer information diffusion, thereby capturing the implicit dependency relationships between sensors.

In the modeling of the temporal correlation of sensors, we design a variant model of transformer modeling. After feature extraction of temporal data, local features of temporal information are extracted by the down-sampling convolution strategy, and the global temporal correlation is captured by weighted multilayer transformer architecture, then the effective integration of global and local is accomplished based on the up-sampling strategy. We input the temporal features and spatial features captured in the above modules into the joint optimization block for feature optimization. In the anomaly prediction module, we further compute anomaly scores based on a thresholding approach to combine predicted and true values, where the predicted values are derived from predictive modeling based on extracted spatio-temporal features.

3.3. Modeling spatial correlations

In recent years, most researchers have preferred to use a dynamic graph model [43,44] to utilize the powerful learning capabilities of the graph model to capture the spatial dependencies between sensors. However, as described in the literature [45], most dynamic graph methods learn deterministic node features, ignoring the uncertainty of the graph structure data and the stochastic generation process of the time series, which is likely to lead to poor model fitting and poor generalization. Therefore, we design a novel Diffusion Graph Block (DGB) which uses an elaborate new multilayer diffusion structure to extract spatial dependencies between sensors. As shown in Fig. 2, the steps of the diffusion graph model are as follows:

First, we construct the initial graph structure on the input data, with sensors as the nodes of the graph. Consequently, for the edges of the graph, we use Euclidean distance to obtain the first k closest to each node for connection. we use an embedding vector $C \in R^{N\times D}$ to embed the features of N sensors and the embedding vector $C \in R^{N\times D}$ will be updated in the subsequent learning. The procedure for constructing the neighbor matrix is as follows:

$$D_{n,m} = F(C_n, C_m) \qquad n, m \in \{1, 2 \ldots N\} \tag{1}$$

$$A_{n,m} = 1 \qquad \{D_{n,m} \in top - k\} \qquad n, m \in \{1, 2 \ldots N\} \tag{2}$$

$$\bar{A} = \sum (A_{n,m}) \qquad n, m \in \{1, 2 \ldots N\} \tag{3}$$

where F denotes the Euclidean distance function, $D_{n,m}$ denotes the distance between sensor nodes. The first k nearest to each node is concatenated to obtain the initial adjacency matrix$\bar{A}$.

Then, we normalize the embedded node information and use the L2 regularized weighted dot product diffusion function with constraints to measure the similarity as the attention function, which is formulated as follows:

$$\begin{gathered} \bar{C} = norm(C) \\ S^{k,h} = W_S^{k,h} \bar{C}^k, U^{k,h} = W_U^{k,h} \bar{C}^k, V^{k,h} = W_V^{k,h} \bar{C}^k \end{gathered} \tag{4}$$

where $\bar{C}$ denotes C after regularization and $norm()$ denotes regularization. $W_S^{k,h} \in R^{D\times D}$, $W_U^{k,h} \in R^{D\times D}$, $W_V^{k,h} \in R^{D\times D}$ denote the trainable parameters of the h_{th} head of the k_{th} layer. Then we perform L2 normalization energy constraints on the $S^{k,h}$, $U^{k,h}$ directions to obtain $S_L^{k,h}$, $U_L^{k,h}$. The formulas are as follows:

$$S_L^{k,h} = \left[\frac{S_n^{k,h}}{\|S_n^{k,h}\|_2} \right]_{\mathrm{n}=1}^{N}, U_L^{k,h} = \left[\frac{U_n^{k,h}}{\|U_n^{k,h}\|_2} \right]_{n=1}^{N} \tag{5}$$

where $S_n^{k,h}$ denotes the n_{th} row vector of $S_L^{k,h} \in R^{N\times D}$, which realizes the global propagation of the h heads. During the propagation process, we introduce a new energy function, as follows:

$$E(C, k, g) = \|C - C^k\|_F^2 + \upsilon \sum_{n,m} g(\|C_n - C_m\|_2^2) \qquad n, m = \{1, 2 \ldots N\} \tag{6}$$

where υ is the weight parameter. Function g is a monotonically increasing concave function over a specific interval to mitigate the variability between each pair of sensor instances [46]. The first term constrains

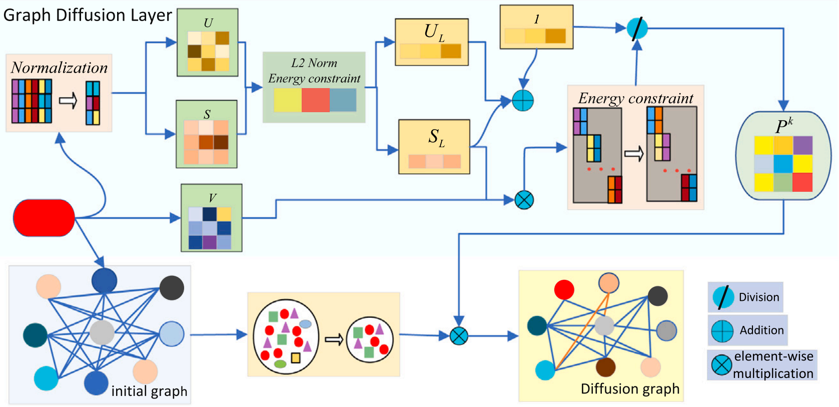


Fig. 2. The framework of diffusion graph model.

the local consistency of each node concerning its current state, and the second term constrains the global consistency concerning other nodes in the system. The direction of the evolution of the node signals during the diffusion process is guided by the constraint-based on energy function to minimize the energy. The formula is as follows:

$$P^k = \frac{1}{h}\sum_{h=1}^{h}\Big\{diag^{-1}(N\mathrm{orm}(U_L^{k,h}((S_L^{k,h})^T 1))) \\ \times \Big[1(1^T V^{k,h}) + U_L^{k,h}((S_L^{k,h})^T V^{k,h})\Big]\Big\} \tag{7}$$

where $1_{N\times 1}$ is an all-1 vector to allow for weighted accumulation operations when computing attention weights. $diag()$ denotes diagonalization and $Norm()$ denotes regularization constraint. We average multiple propagation results to obtain p^k, denoting the diffusion probability among different sensor nodes.

Then, we set $Q \in R^{N\times C_n}$ to be the feature of the input sensor n. The diffusion graph is implemented as follows:

$$G^k = P^k \bar{A}^k \\ Q^{k+1} = (1-\ell G^k)Q^k + \ell G^k Q^k \quad E(Q^{k+1}, k; g) \le E(Q^k, k-1; g) \quad k \ge 1 \tag{8}$$

where G^k denotes the learning mode of graph diffusion. Q^{k+1} denotes the updating process of sensor data. ℓ is a hyperparameter that denotes the probability of retaining the previous layer's features in this layer's features. E denotes the energy constraints, and the final Q_{out} is the output variable of Q^{k+1} after learning through the k layer diffusion graph.

3.4. Modeling temporal correlations

Since sensor data changes continuously over time, when sensor data fluctuates in a certain period the dependencies between sensor data in that period may be masked by complex temporal correlations, which further increases the difficulty in capturing long-term dependencies between time series. Therefore, we designed the Tformer model to consider the relationship between local features and their global correlations, which captures the temporal features more comprehensively by learning the local features of the subsequences and the global correlations among multiple subsequences.

The Tformer model framework is shown in Fig. 3. The model has the following main components: Feature Extraction, which is used to purposefully separate the noise information from the timing data. Feature Encoding, which is used to encode the separated feature information. Local–global module, which uses down-sampling convolution for local feature extraction and weighted multi-layer transformer architecture for global correlation modeling, while reconstructing the data structure using up-sampling convolution, and finally completing the temporal correlation modeling.

3.4.1. Feature extraction

Similar to the literature [47], we design a feature extraction module for purposefully separating the noise information from the time-series data. Through the average pooling method in this way, we obtain more accurate feature information. The process is as follows:

$$x_s = Q - AvgPool(Padding(Q))_k \tag{9}$$

where $Q \in R^{N\times T}$ is the sensor input data, k is the convolutional kernel size, $AvgPool()$ denotes average pooling, and $Padding()$ operation keeps the sequence length constant.

3.4.2. Feature encoding

To avoid redundant computation of the encoder data and to adapt the prediction length, we encode the separated feature information by employing the superposition of two encoding methods to obtain x_{emb}. The specific process is as follows:

$$x_{emb} = sum(B_{VE}(concat(x_s, x_{zero})) + B_{PE}) \tag{10}$$

where B_{VE} is the value encoding and B_{PE} is the position encoding. x_{zero} is an all-zero matrix of which the dimensionality is the same as x_s and which ensures that the dimensionality of both encodings is the same. Next, we import the encoded data into the Local–global module to extract the local features and global correlations.

3.4.3. Local-global modules

We introduce the down-sampling convolution to extract local features of temporal correlation between sensors. Since local features only focus on feature information over a period of time, we use a weighted multilayer transformer architecture to model the global correlation of the down-sampling generation time subsequence. It aims to capture long-term dependencies by modeling correlations between multiple subsequences of sensors. First, the specific local characterization process is as follows:

$$x_{local} = DownConv1d(Avgpool(Padding(x_{emb})_k))_k \tag{11}$$

where x_{local} denotes localized features and $DownConv1d()$ denotes the down-sampling. We set $stride = kernel = k$ as the compression of the local feature.

We then use the weighted multilayer transformer architecture to model the global correlation of the subseries obtained from down-sampling. Since the weighted multilayer transformer aggregation weights are dynamic, all-positive, and normalized, as pointed out in the literature [48] - static, learnable, and unconstrained convolutional aggregation weights help to complement this modeling process. In this regard, we propose a global relevance learning process as follows:

$$x_{global} = Merge((soft\max(\frac{qk^T}{\sqrt{d}}) + b)v + C(v)_k) \tag{12}$$

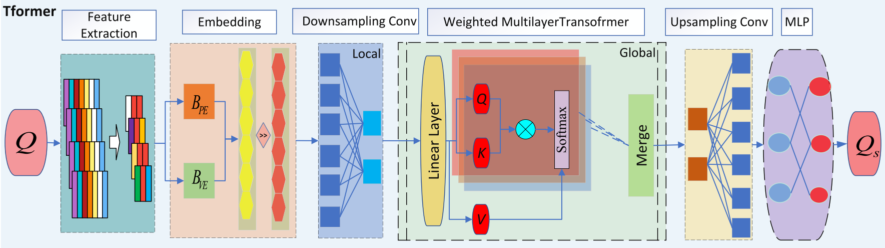


Fig. 3. The framework of Tformer model.

where x_{global} denotes the global feature and b is the relative position bias term. $q = x_{local} \times w_q$, $k = x_{local} \times w_k$, $v = x_{local} \times w_v$. $C(v)_k$ denotes the convolution operation. $softmax()$ denotes the activation function. $Marge()$ represents that multiple layers of attention are fused, while each layer of attention extracts information about different temporal features between sensors.

Then, we integrate the local features with the global correlation to obtain a more accurate feature representation. The specific process is as follows:

$$x' = UpConv1d(Norm(x_{local} + Dropout(\tanh(x_{global})))) \tag{13}$$

$$Q_s = Norm(x' + MLP(x')) \tag{14}$$

where $Norm()$ denotes regularization, $Dropout()$ denotes random discard, and $tanh()$ denotes activation function. The temporal correlation structure is then reconstructed by using $Upcon1d()$ (upsampling). MLP is used to extract a valid feature representation and the final output Q_s is obtained by regularization.

3.5. Joint optimization

In this section, we input both the spatial correlation representation and the temporal correlation representation between sensors obtained in the above modules into the GRU model for joint optimization. The formula is as follows:

$$\begin{aligned} Y_t &= concat(Q_s, Q_{out}) \\ z_t &= sig\bmod(W_z Y_t + O_z P_{t-1}) \\ r_t &= sig\bmod(W_r Y_t + O_r P_{t-1}) \\ \bar{P}_t &= sig\bmod(W Y_t + O(r_t \odot P_{t-1})) \\ P_t &= (1 - z_t)P_{t-1} + z_t \bar{P}_t \end{aligned} \tag{15}$$

Y_t represents the input of the GRU at the moment t and $concat()$ denotes the splicing operation. z_t represents the update gate, The reset gate r_t functions similarly to the update gate. $\bar{P}_t$ denotes the candidate's hidden layer state. P_t represents the candidate activation state of the GRU at the moment.

3.6. Anomaly prediction module

In this section, we accomplish the anomaly detection task based on 2steps: prediction and detection.

3.6.1. Prediction layer

In this section, we obtain the final prediction $\tilde{Y} = \{\tilde{Y}_t, \tilde{Y}_2, \dots \tilde{Y}_l\} \in R^{(t-l)\times n}$ by feeding the output of the GRU into the prediction model consisting of two fully connected layers, as follows:

$$\tilde{Y} = (Y_{gru}h_a + j_a)h_b + j_b \tag{16}$$

The loss function is as follows:

$$\varsigma_{loss} = \sqrt{\sum_{i=1}^{n} (\tilde{Y}_{t,i} - \bar{Y}_{t,i})^2} \tag{17}$$

where $\tilde{Y}_{t,i}$ denotes the predicted value of the i_{th} sensor at time t, and $\bar{Y}_{t,i}$ shows the true value of the i_{th} sensor at time t. ς_{loss} is represented by the root-mean-square deviation of the true and predicted values [49].

3.6.2. Anomaly scoring

Next, we calculate the anomaly analysis by analyzing the squared difference between predicted value $\tilde{Y}_t$ and the true value $\bar{Y}_t$ [10]:

$$e = \sum_{t=1}^{n} (\tilde{Y}_t - \bar{Y}_t)^2 \tag{18}$$

We used an automatic selection thresholding (SPOT) method [8] to analyze the anomalies. The sample mean and variance, as well as other parameters, are calculated using the following formulas.

$$K = \sum_{j=1}^{Z} \frac{Y_j}{Z} \tag{19}$$

$$G^2 = \sum_{j=1}^{Z} \frac{(Y_j - K)^2}{Z - 1} \tag{20}$$

where K denotes the sample mean and G^2 denotes the sample variance. Y_t is the sample point of $t < G$, and Z is the number of Y_t. The final shape parameter a and the final scale parameter b of the GPD for SPOT are calculated as follows:

$$\tilde{a} = \frac{K}{2}(1 + \frac{K^2}{G^2}), \tilde{b} = \frac{1}{2}(1 - \frac{K^2}{G^2}) \tag{21}$$

$$\xi_{end} = \zeta + \frac{\tilde{a}}{\tilde{b}}((\frac{\omega n}{Z})^{-\tilde{b}} - 1) \tag{22}$$

where ξ_{end} denotes the final threshold, ω represents the risk factor for determining anomalies.

3.6.3. Time complexity analysis

The time complexity of GDTS models mainly consists of graph diffusion and Tformer architecture in spatio-temporal modeling. The time complexity of graph diffusion mainly comes from graph construction and diffusion rate. The computational complexity of graph construction is O(N^2), where N denotes the number of nodes (sensors). The diffusion rate is based on the subspace embedding and multi-head mechanism, which generates h keys and queries, and the time complexity of the diffusion rate is $O(h \bullet {D_c}^2)$, where D_c denotes the vector dimension of the subspace embedding, and h denotes how many heads there are. Finally, the information between nodes is updated and multilayer diffusion is performed making the time complexity to be $O(h \bullet k \bullet {D_c}^2)$, where k denotes the number of layers of graph diffusion. In Tformer, feature extraction, feature coding and downsampling and all require traversing the original sequence with a time complexity of $O(L)$, where L denotes the sequence length. The time complexity of the global transformer architecture consists of multilayer attention and convolution, with a time complexity of $O(L^2)$ for each self-attention mechanism. Taken together, the total time complexity of the Tformer architecture is $O(L + I \bullet L^2)$, where I denotes the number of layers of attention. Finally, both are passed into GRU optimized and predicted for anomaly detection.

In summary, the GDTS modeling process does perform expensive computations and consumes a large amount of storage space, but with the full support of the High-Performance Computing Center of the Hubei University of Technology, we have successfully solved this problem to provide good results for the GDTS model. The complete training and inference procedure of GDTS is shown in Algorithm 1.

Algorithm 1 The learning algorithm for the GDTS model

```
Input: Sensor data $Q \in \mathbb{R}^{L \times N}$
Output: Detection of anomalies
Step 1: Train spatial-temporal feature $Q_s, Q_{out}$.
for each epoch do
    $Loss \leftarrow 0$
    for each $Q_s$ do
        $\bar{A} \leftarrow \sum(D_{n,m})$
        $P^k \leftarrow E(S^{k,h}, U^{k,h}, V^{k,h})\mathrm{A}$
        $Q^{k+1} \leftarrow G^k$
    end for
    for each $Q_{out}$ do
        $x_{emb} \leftarrow \mathrm{sum}(B_{VE}, B_{PE})$
        $x_{local} \leftarrow \mathrm{DownConv1d}((x_{emb})_k)$
        $x_{global} \leftarrow \mathrm{Merge}((\mathrm{softmax}(\frac{qk^T}{\sqrt{d}}) + b)v + C(v)_k)$
        $x' \leftarrow \mathrm{UpConv1d}(x_{local}, x_{global})$
    end for
end for
Step 2: Apply $\xi_{\mathrm{end}}$ threshold
for predicting and scoring do
    prediction $\leftarrow 1_{(th_{bestf_1} < \bar{Y})}$
end for
```

4. Experiments

In this section, we conduct experiments based on multiple real data to answer the following questions:

Q1: Does the GDTS model outperform the baseline algorithm in terms of model performance?

Q2: How do different modules affect the performance of model?

Q3: How does the setting of hyperparameters affect the performance of model?

4.1. Experimental setup

we adopt a similar approach as the literature [50] for data processing, the method combining spectral residuals and convolutional neural networks for data cleaning.

4.2. Dataset

In this paper, we conduct comprehensive experiments using five publicly available datasets MSL,[1] SMAP,[2] SMD,[3] SWAT,[4]and WADI.[5] SMAP and MSL are two public datasets published by NASA [51]. SMD [8] is a dataset collected from a large internet company. The SWAT dataset [52] is a scaled down version of a real industrial water treatment plant that produces filtered water. The collected dataset contains 11 consecutive days of operation, of which 7 days for the normal operation dataset and 4 days for the attack scenario dataset. The WADI dataset [53,54] is collected from the WADI testbed, which is an extension of the SWAT testbed. It contains 16 days of continuous runs, of which 14 days are collected for normal runs and 2 days for attack scenarios. Their statistical results are shown in Table 1.

Table 1
Dataset introduction.

Dataset	Feature	Train	Test	Anomaly (%)
SMAP	25	135 183	427 617	10.72
MSL	55	73 729	73 729	13.13
SMD	38	708 405	708 420	4.16
SWAT	51	449 919	449 919	11.98
WADI	123	1 048 571	172 801	5.99

We selected some of the data collected in the five datasets for display and labeled the anomalous data with a red vertical line, as shown in Fig. 4:

The MSL dataset and the SMAP dataset contain point anomalies as well as overall change trend anomalies. For the SMD dataset, we consider trend changes that are inconsistent with the normal pattern as anomalous data. For the SWAT dataset and WADI data, we treat their attack scenario datasets as anomalous data.

4.3. Evaluation metrics

We evaluate the performance of various methods by the most metrics in anomaly detection, Precision (P), Recall (R), F1-score (F1), and AUC.

$$\begin{aligned} P &= \frac{T_P}{(T_P + F_P)} \\ R &= \frac{T_P}{(T_P + F_N)} \\ F1 &= \frac{2 \times P \times R}{P + R} \\ AUC &= \frac{T_P + T_N}{T_P + F_P + F_N + T_N} \end{aligned} \tag{23}$$

In anomaly detection studies in the time series domain, a number of researchers [8,55] have adopted the $F1_{PA}$ metric as a baseline measure of anomaly performance. However, Kim et al. [1] suggested some limitations where $F1_{PA}$ is likely to be overestimated, and they provided empirical evidence that randomized anomaly measurements outperform state-of-the-art methods on almost all baseline datasets. For this reason, they propose an alternative assessment metric called PA%K. It compensates for the overestimation of $F1_{PA}$ and the underestimation of F1 with the following formula:

$$\hat{y}_t = \begin{cases} 1, if\, A(x_{t-w:t}) > \zeta or \\ t \in S_m \text{ and } \frac{|\{t' | t' \in S_m, A(x_{t'-w:t'}) > \zeta\}|}{|S_m|} > K \\ 0, otherwise \end{cases} \tag{24}$$

where $\hat{y}_t$ is the predicted label, ζ is the threshold, $A(.)$ is the input anomaly metric, $|.|$ is the base of the set, $K \in [0, 1]$ is the ratio. In our experiments $K = 1$, i.e., the F1 point-wise scores are computed based on the actual output of the algorithm without any adjustment.

4.4. Experimental design

We answer the above questions based on the following aspects of experiments to fully validate the performance of GDTS:

1. Performance Comparison. We demonstrate the advancement of our model by analyzing the results in comparison with several state-of-the-art methods.

2. Ablation experiments. We show the necessity of spatial modeling based on graph diffusion and temporal modeling of the model based on transformer variables, as well as the influence of the individual components regarding the performance of the model.

3. Hyper-parametric experiments. We experimentally analyze several hyper-parameters to further demonstrate the performance of GDTS.

4.4.1. Baselines

In this paper, we have selected seven state-of-the-art group recommendation algorithms for comparison to demonstrate the validity of GDTS.

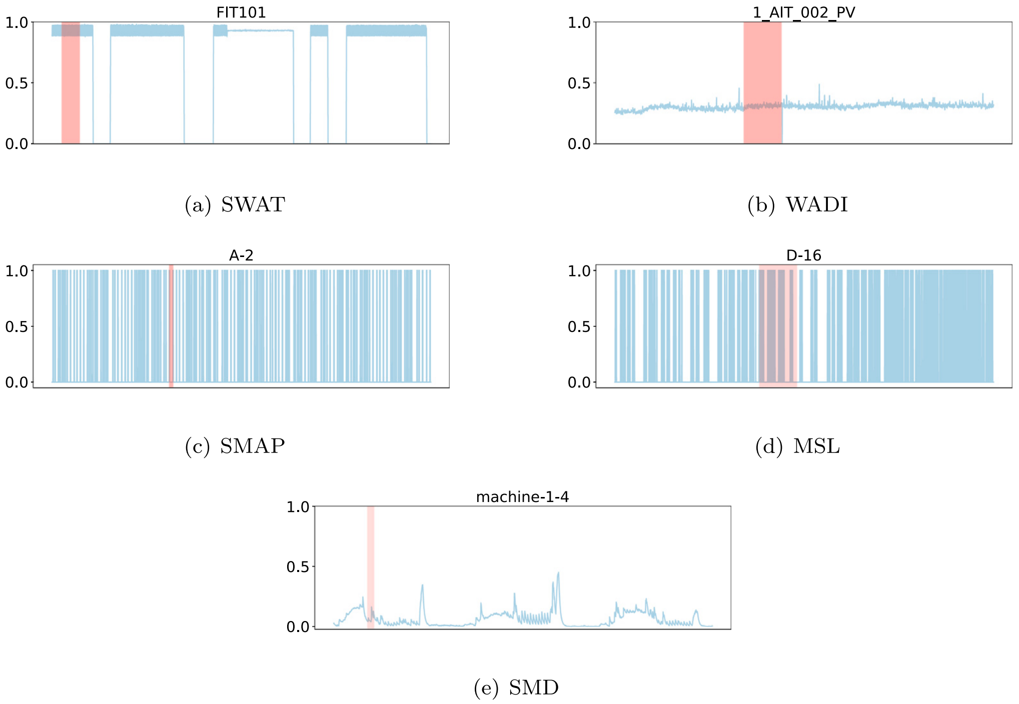
[1] https://drive.google.com/drive/folders/1RaIJQ8esoWuhyphhmMaH-VCDh-WIluRR.

[2] https://drive.google.com/drive/folders/1RaIJQ8esoWuhyphhmMaH-VCDh-WIluRR.

[3] https://drive.google.com/drive/folders/1RaIJQ8esoWuhyphhmMaH-VCDh-WIluRR.

[4] https://drive.google.com/drive/folders/1ABZKdclka3e2NXBSxS9z2YF59p7g2Y5I.

[5] https://drive.google.com/drive/folders/1ABZKdclka3e2NXBSxS9z2YF59p7g2Y5I.



(a) SWAT

(b) WADI

(c) SMAP

(d) MSL

(e) SMD

Fig. 4. Partial fragments of each dataset.

(1) LSTM-NDT [51]: It uses LSTM to capture the nonlinear temporal relationships and spatial depth features among spacecraft telemetry data, and then proposes a new dynamic threshold-based reconstruction method for multivariate temporal unsupervised anomaly detection judgment.

(2) MSCRED [30]: It proposes a joint multi-scale convolutional network and LSTM model for anomaly prediction, which uses a multi-scale signature matrix to represent the multistage system state with different time steps, encodes and reconstructs the data with a convolutional encoder–decoder architecture, then captures the time pattern based on attention combined with LSTM, and finally uses a residual feature matrix for anomaly detection and diagnosis.

(3) Omni-Anomaly [8]: It designs a reconstruction-based anomaly detection method oriented to spatio-temporal feature learning. On the one hand, it utilizes GRU to capture complex temporal correlations, and on the other hand, it proposes the random variable approach of joint VAE to learn the complex coupling properties among spatial-level observations.

(4) USAD [9]: It proposes an unsupervised joint autoencoder and GAN approach to anomaly detection. It introduces the generative adversarial idea and exploits the robust representation learning capability of the autoencoder on time series data, while extracting correlation features between multivariate time series and reconstructing complex data distributions thus further improving the diagnostic capability of the model to detect and interpret anomalies when anomalous events occur.

(5) MTAD-GAT [10]: It proposes an anomaly detection method based on a dual graph attention model. It models the unfolding along the temporal and spatial dimensions using the graph attention model respectively, then analyses the captured spatiotemporal features for anomaly prediction, and finally combines prediction and reconstruction for anomaly assessment.

(6) CAE-M [56]: It proposes a spatiotemporal anomaly detection method that mixes deep convolutional networks and memory networks. It reduces the problem of noise and overfitting in temporal signal learning through deep convolutional networks while capturing temporal dependencies by constructing LSTM-based memory networks.

(7) GDN [12]: It proposes an attention-based anomaly detection method for graph neural networks. It combines structural learning methods with graph neural networks by learning the bias of inter-sensor dependencies and providing interpretability for detected anomalies with attention weights.

(8) GTA [33]: It proposes a multivariate temporal anomaly detection with a graph neural network incorporating a transformer. It utilizes a joint multi-scale extended convolutional as well as graph convolutional network and transformer as a spatio-temporal learning module for the model.

(9) TranAD [57]: It proposes a multivariate temporal anomaly detection based on the combination of reconstructed transformer and AE. It also introduces a two-stage generative adversarial technique to train the model, which ultimately achieves robust multivariate feature extraction.

(10) DuoGAT [55]: It proposes anomaly detection using a dual time-oriented graph attention network. The method utilizes a weighted directed graph to model interrelationships between variables by pointing only from past events to future events and assigning higher weights to edges of neighboring events. In addition, it models the smoothness of time series using a time-oriented graph that captures the changes in the series by differencing.

(11) ImDiffusion [58]: It proposes an anomaly detection method that utilizes the information of neighboring values in a time series to achieve accurate modeling of temporal and interrelated dependencies, and which further utilizes a diffusion model to accurately capture complex sensor dependencies through stepwise denoising.

(12) AD-NEv [59]: It proposes a new method for collaborative optimization of feature subspaces, which optimizes the architecture of a single anomaly detection model and allows for non gradient fine-tuning of network weights.

4.4.2. Parameter setting

For baseline models, we refer to their optimal parameter settings reported in the original paper. Regarding GDTS, the experimental setup is as follows: For the SMAP dataset, the number of training sessions was 10, the size of each batch was 128, the initial learning rate was 0.0001, and the number of graph diffusion layers was 3. For the MSL dataset,

Table 2
Comparison of model performance.

Method	SMAP		MSL		SMD		SWAT		WADI	
	F1 point-wise	F1	F1 point-wise	F1	F1 point-wise	F1	F1 point-wise	F1	F1 point-wise	F1
LSTM-NDT	0.2204	0.7879	0.2142	0.7721	0.4265	0.6037	0.7246	0.7721	0.2071	0.2238
MSCRED	0.1721	0.8664	0.2512	0.9363	0.3854	0.8781	0.7484	0.8926	0.1226	0.2414
Omni-Anomaly	0.2414	0.8728	0.2071	0.8765	0.4150	0.9401	0.7530	0.8965	0.2596	0.3285
USAD	0.2272	0.8419	0.2101	0.8822	0.4243	0.9363	0.7628	0.8880	0.2328	0.3717
MTAD-GAT	0.2623	0.8880	0.2864	0.8768	0.4121	0.8683	0.7741	0.8768	0.4373	0.3715
CAE-M	0.2425	0.9422	0.2485	0.8733	0.4264	0.9376	0.7431	0.8733	0.3212	0.4671
GDN	0.2521	0.8515	0.2178	0.9491	**0.5161**	0.8342	0.8080	0.8891	0.5702	0.4874
GTA	0.2316	0.9041	0.2182	0.9111	0.3520	0.9290	0.7610	0.9111	0.5028	0.5217
TranAD	0.2414	0.8915	0.2511	0.9494	0.4582	0.9405	0.7291	0.9094	0.4150	0.5574
DuoGAT	0.2461	0.9267	0.2193	0.9403	0.4195	0.9057	0.7610	0.9103	0.3956	**0.6428**
ImDiffusion	0.2613	0.9175	0.2781	0.8779	0.4365	**0.9488**	0.7623	0.9109	0.2545	0.5209
AD-NEv	**0.77**	–	**0.57**	–	–	–	**0.82**	–	**0.62**	–
GDTS	0.2653	**0.9658**	0.2884	**0.9510**	0.4589	0.9486	0.7682	**0.9147**	0.2684	0.3401

Table 3
Comparison of model performance.

Method	SMAP			MSL			SMD		
	P	R	AUC	P	R	AUC	P	R	AUC
LSTM-NDT	0.8523	0.7326	0.9577	0.6288	**1.0000**	0.9239	0.5684	0.6438	0.9648
MSCRED	0.8175	0.9216	0.9695	0.8912	0.9862	0.9835	0.9103	0.9914	0.9855
Omni-Anomaly	0.8130	0.9419	0.9705	0.7848	0.9924	0.9646	0.8881	**0.9985**	0.9847
USAD	0.7480	0.9627	0.9612	0.7949	0.9912	0.9665	0.9314	0.9617	0.9854
MTAD-GAT	0.7991	0.9991	0.9730	0.7917	0.9824	0.9649	0.8210	0.9215	0.9783
CAE-M	0.8733	0.8193	0.9679	0.7751	0.9999	0.9631	0.9082	0.9671	0.9845
GDN	0.7480	0.9891	0.9864	0.9308	0.9892	0.9814	0.7170	0.9974	0.9824
GTA	0.8911	0.9176	0.9791	0.9104	0.9117	0.9778	0.9826	0.9317	0.9868
TranAD	0.8043	0.9999	0.9739	0.9038	0.9999	0.9872	0.9262	0.9974	0.9874
DuoGAT	0.8634	**0.9999**	0.9830	0.9271	0.9538	0.9841	0.9142	0.9228	0.9782
ImDiffusion	0.8771	0.9618	0.9828	0.8930	0.8638	0.9685	0.9520	0.9509	0.9882
GDTS	**0.9613**	0.9703	**0.9912**	**0.9571**	0.9451	**0.9898**	**0.9538**	0.9501	**0.9894**

Table 4
Comparison of model performance.

Method	SWAT			WADI		
	P	R	AUC	P	R	AUC
LSTM-NDT	0.7778	0.5109	0.8436	0.1636	0.7669	0.5912
MSCRED	**0.9992**	0.6770	0.8433	0.2513	0.7319	0.8412
Omni-Anomaly	0.9782	0.6957	0.8467	0.3158	0.6541	0.8198
USAD	0.9977	0.6879	0.8460	0.1873	0.8296	0.8723
MTAD-GAT	0.9718	0.6957	0.8464	0.2818	0.8012	0.8821
CAE-M	0.9697	0.6957	0.8464	0.2782	0.7918	0.8728
GDN	0.9697	0.6957	0.8462	0.2912	0.7931	0.8777
GTA	0.9783	0.6987	0.8460	0.3391	0.8361	0.8778
TranAD	0.9712	0.6997	0.8491	0.3529	0.8296	0.8968
DuoGAT	0.8802	**0.8241**	**0.8830**	0.4942	**0.8797**	**0.8999**
ImDiffusion	0.9588	0.7465	0.8528	0.3446	0.8223	0.8789
GDTS	0.9117	0.7545	0.8613	**0.5609**	0.1504	0.8841

the number of training sessions was 30, the size of each batch was 128, the initial learning rate was 0.0005, and the number of graph diffusion layers was 1. For the SMD dataset, the number of training sessions was 20 times, the size of each batch was 128, the initial learning rate was 0.003, and the number of graph diffusion layers was 2. For the sliding window size, the sizes of SMAP, MSL, and SMD were 60. In addition, we also used a grid-searching methodology to select the optimal parameter configurations for the different datasets to obtain the model's optimal performance.

4.4.3. Performance comparison

As shown in Tables 2, 3 and 4, We observe:

1. LSTM-NDT has the highest R score in MSL, but all other performances are the lowest. The possible reason is that LSTM-NDT only considers univariate time series and ignores the interaction and dependency between sensor data.

2. MSCRED achieved very good results at that time, with F1 values over 85% and AUC over 98%, but its performance on the low time-dimensional SMAP was relatively weak, probably because it failed to model the relationships within the sequence well.

3. OmniAnomaly captures normal patterns in multivariate time series by learning robust representations of multivariate time series and captures anomalies by reconstruction, and in general, its methodology has achieved some success, with the highest R-value of 99.85% in SMD, and scores of more than 80% in evaluation metrics in both SMAP and SMD, but it is not effective in capturing time dependencies in MSL with high time dimensions.

4. USAD, CAE-M, MTAD-GAT, and TranAD are all optimized by using the reconstruction method for the model, with TranAD having the best performance, followed by CAE-M, and MTAD-GAT the worst. Some of the reasons that contribute to this possibility are: MTAD-GAT uses two graph attention layers to learn the complex dependencies of the multivariate time series in the time. USAD and CAE-M separate the anomalies through the encoder–decoder structure, which does not employ an attention mechanism making it not very good at the time of time series temporal. TranAD is still slightly inferior to GDTS, due to the fact that multivariate time series spatial dependencies are not considered. Moreover, all these methods reconstruct normal data, which may reduce the accuracy of anomaly detection [60].

5. GTA utilized graph structures, graph convolution, and a transformer-based architecture to model temporal dependencies. It achieves an overall performance of approximately 90%, with the highest P-value of 98.26% in the SMD dataset. However, other evaluation metrics remain stable and do not fully demonstrate its performance. This may be attributed to a sampling bias in the construction of the graph, leading to a decrease in overall performance.

6. GDN utilizes graph structure learning and attention architecture to model temporal dependencies. Its anomaly detection performance is greatly improved compared to previous models, and its R score is the second highest in the MSL dataset. However, the F1 score is

average across the baseline, which may be attributed to the failure to model the sampling bias in the graph structure well, resulting in a poor improvement in the detection accuracy for outliers.

7. The DuoGAT model is similar to the GDN, but it uses weighted directed graphs to model interrelationships between variables by pointing from past events only to future events and assigning higher weights to edges of neighboring events. Its performance on the WADI dataset is very impressive, with its F1 value being the highest against all baselines, and it is also strong on the SWAT dataset and the MSL dataset against other baselines, with the second highest F1 value. However, its performance on the other two datasets is mediocre. It is probably due to the fact that smooth modeling of the time series through differencing leads to loss of information in the time series.

8. The ImDiffusion model achieves good results in case F1 but does not outperform most baseline models in terms of F1 point-wise score on the SMD and WADI datasets. This could be attributed to its reliance on diffusion methods, which struggle to effectively reduce noise in the data.

Comparing the F1 scores and F1 point-wise scores from different baselines, F1 point-wise scores are lower than the F1 scores, which coincides with the literature's point [61]. In the datasets, most of the baselines have low F1 point-wise scores. Compared with the baselines, the anomaly detection performance of GDN is greatly improved. Among the WADI and SWAT data, the GDN model has the second-highest F1 point-wise score, which may be attributed to the fact that GDN focuses on using powerful graph neural networks to fully learn the interrelationship between variables and captures trends in the time series, thus performing anomaly detection effectively. In addition, the performance of GDN in such as the WADI dataset proves its effectiveness in dealing with unbalanced and high-dimensional data scenarios, which provides comprehensive model interpretability through sensor embedding vectors and learned graph structures [12]. In the WADI dataset, the GTA model has the third-highest F1 point-wise score, which may be due to the fact that it accurately models time dependence by exploiting the information of neighboring values in the time series. In the SMAP dataset, the F1 point-wise of the MTAD-GAT model is the third-highest, which may be due to the fact that the model implements multivariable feature extraction based on graph concerns, effectively distinguishing anomalies. After the latest AD-NEv model appeared, its F1 point-wise surpassed all baselines (including ours) on four datasets, which is a major breakthrough in the field of anomaly detection.

In terms of F1, GDTS performs well across most datasets, achieving the highest or second-highest results compared to the baselines. In the SMAP and MSL datasets, the GDTS model shows overwhelming performance in both the F1 score and the AUC score. For the SWAT dataset, the GDTS model achieves a narrow victory. In the WADI dataset, the detection performance of all baselines is unsatisfactory, and our model fared poorly. This poor performance could be attributed to the large volume of data, extensive feature dimensionality, and a scarcity of outliers in the WADI dataset compared to other datasets. For the F1 point-wise, the GDTS model outperforms most baselines on all datasets except WADI. The reasons for this are as follows: (1) In modeling the spatial correlation of sensors, we use a newly designed graph diffusion model to extract the correlation between neighboring sensors, and the spatial structure of sensors can be learned more accurately through the newly designed energy-constrained diffusion. (2) To model the temporal correlation of sensors, we use a novel combination of a hybrid sampling strategy and a transformer module, which can be used to model the correlation between sensors and the energy-constrained diffusion of the sensors, by taking into account the temporal sequence's local features and global correlations to trap the long-term temporal correlation of the time series more comprehensively.

4.4.4. Ablation study

To evaluate the impact of the graphical diffusion module and the Tformer module concerning the GDTS model, we conducted ablation experiments. We define the different models as follows.

GDTS-1: We remove the graph diffusion module in the spatial correlation modeling.

GDTS-2: We remove the Tformer module in temporal correlation modeling.

GDTS-3: We use the basic graph convolution instead of the graph diffusion module for spatial correlation modeling, and the temporal correlation modeling is consistent with the GDTS model.

GDTS-4: we use the basic transformer instead of the Tformer module to perform the acquisition of long-term temporal dependence on time series and use graph diffusion for the capture of complex topological relationships of sensors for anomaly detection.

The results are shown in Fig. 5. We get the following conclusions:

1. GDTS-1 is the lowest. When the graph diffusion module is removed, the F1 score decreases by more than 0.5, which is also the lowest compared to the performance of the other models that employ the structural features of the sensors. This suggests that extracting hidden spatial correlations between sensor data is important.

2. GDTS-2 has the second-highest performance score among the five models. When the Tformer module is removed and anomaly detection is performed only through the graph diffusion module, the overall performance of the model is degraded due to its inability to adequately fit the information of the temporal data in the time dimension. This suggests that how the temporal correlation between sensor data is obtained is also important for the success of multivariate temporal anomaly detection.

3. GDTS-3 is the third among the five models. After replacing our designed graph diffusion with GCN, non-existing relationships may be introduced into the graph structure without eliminating the differences between the information of individual sensors, which will introduce errors in subsequent information transfer and lead to degradation of the model performance.

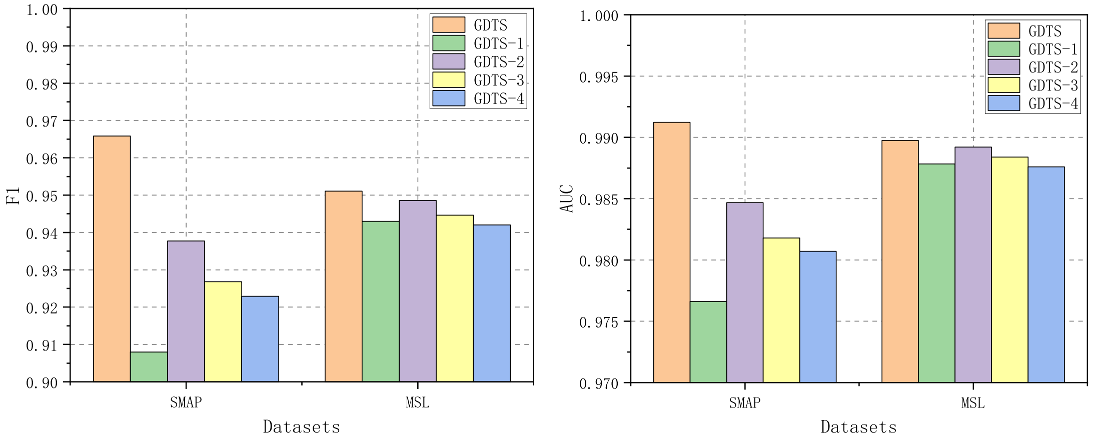
4. GDTS-4 is the second lowest among the five models. The reason is that replacing the Tformer module of our design with a conventional transformer causes the model to be unable to sense the occurrence of anomalies in a timely manner because it cannot obtain local information about the temporal dimension of the sensor data. Obviously, the powerful self-attention mechanism combined with the sampling strategy constructs connections between global key points in the time series, thus more accurately characterizing the contextual information in the time series, which is important for improving the accurate modeling of temporal correlation of time series using Tformer.

4.4.5. Hyperparameter analysis

This section investigates the effect of two hyper-parameters regarding the GDTS. One is the number of graph diffusion layers, and the other is the learning rate.

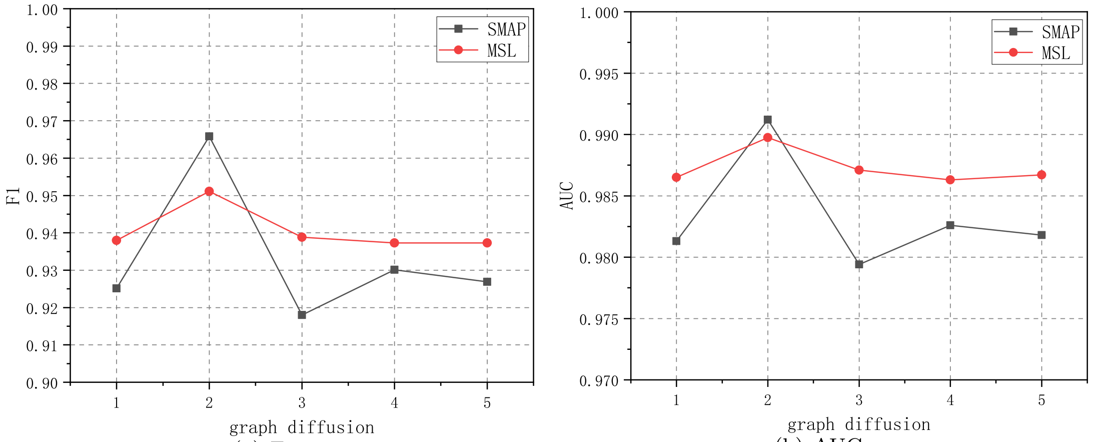
Fig. 6 shows the results for five different graph diffusion layer numbers. When the graph diffusion layers are 2, the SMAP dataset and MSL dataset have better detection results compared to the other results. This is due to its ability to fully capture the complex relationships of the graph structure. If the number of graph diffusion layers is too large, the anomalies are hidden and are not easily detected by the model. However, if the number of graph diffusion layers is too small, the model cannot capture the complex relationships in the graph structure well and it is difficult to perform inference.

Fig. 7 shows the detection results for five different learning rates. When the learning rate size is 0.001, the SMAP dataset has better detection results compared to the other learning rate sizes. When the learning rate is 0.0001, the MSL dataset has better detection results compared to the other learning rate sizes. The possible reasons for this situation are: Firstly, the feature information dimensions of the two datasets are different, which leads to the learning rate of the two datasets being inconsistent. Second, a learning rate that is too large leads to a model that fails to converge, while a learning rate that is too small leads to a model that converges slowly or fails to learn.



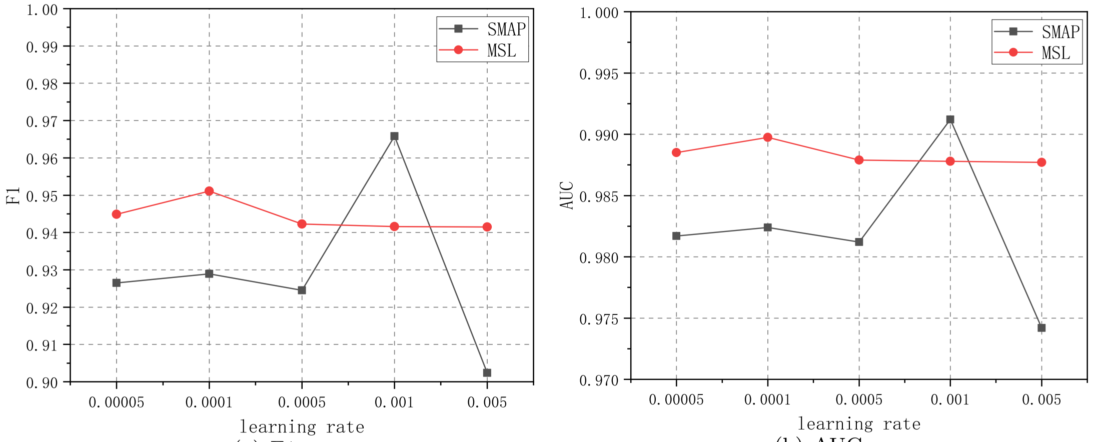
(a) F1 (b) AUC

Fig. 5. The performance of different models.



(a) F1 (b) AUC

Fig. 6. Effect of graph diffusion.



(a) F1 (b) AUC

Fig. 7. Effect of learning rate.

5. Conclusion

In this paper, a hybrid spatio-temporal neural network-based model is proposed for multivariate timing detection. The model explores the energy constraint-based graph diffusion convolutional transform network and sampling strategy-based transformer to model the sensor data from both temporal and spatial dimensions simultaneously. For spatial correlation modeling, we design a novel graph diffusion network that introduces a new information propagation method based on energy function constraints to capture the complex dependencies between sensors and make them stable through multi-layer diffusion. In addition, we develop a network that combines the up-and-down sampling convolution strategy and the weighted multilayer transformer to comprehensively describe the temporal correlation of sensors. Finally, experiments on five real-world time series datasets show that GDTS shows a more significant improvement.

CRediT authorship contribution statement

Rong Gao: Writing – review & editing, Writing – original draft, Validation, Methodology, Formal analysis, Data curation, Conceptualization. **Jiming Wang:** Writing – review & editing, Writing – original draft, Software, Resources, Methodology, Investigation, Data curation. **Yonghong Yu:** Supervision, Funding acquisition. **Jia Wu:** Writing – review & editing, Validation, Supervision. **Li Zhang:** Visualization, Validation, Supervision, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is funded by the National Natural Science Foundation of China (No. 62376089). This work was supported in part by the Chunhui Plan Collaborative Research Project, Ministry of Education, China (HZKY20220350). This work was supported in part by the Future Network Scientific Research Fund Project under Grant FNSRFP-2021-YB-54 and in part by the Qing Lan Project of Jiangsu Province.

Data availability

The datasets used are all publicly available.

References

[1] S. Kim, K. Choi, H. Choi, et al., Towxivds a rigorous evaluation of time-series anomaly detection, in: Proceedings of the 36th AAAI Conference on Artificial Intelligence, 2022, pp. 7194–7201.
[2] A. Cook, G. Mısırlı, Z. Fan, Anomaly detection for IoT time-series data: A survey, IEEE Internet Things J. 7 (7) (2019) 6481–6494.
[3] I. Melnyk, A. Banerjee, B. Matthews, et al., Semi-Markov switching vector autoregressive model-based anomaly detection in aviation systems, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1065–1074.
[4] J. Li, H. Izakian, W. Pedrycz, et al., Clustering-based anomaly detection in multivariate time series data, Appl. Soft Comput. 100 (2021) 106919.
[5] J. Chow, Z. Su, J. Wu, et al., Anomaly detection of defects on concrete structures with the convolutional autoencoder, Adv. Eng. Inform. 45 (2020) 101105.
[6] B. Lindemann, B. Maschler, N. Sahlab, et al., A survey on anomaly detection for technical systems using LSTM networks, Comput. Ind. 131 (2021) 103498.
[7] X. Xia, X. Pan, N. Li, et al., GAN-based anomaly detection: A review, Neurocomputing 493 (2022) 497–535.
[8] Y. Su, Y. Zhao, C. Niu, et al., Robust anomaly detection for multivariate time series through stochastic recurrent neural network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2828–2837.
[9] J. Audibert, P. Michiardi, F. Guyard, et al., Usad: Unsupervised anomaly detection on multivariate time series, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 3395–3404.
[10] H. Zhao, Y. Wang, J. Duan, et al., Multivariate time-series anomaly detection via graph attention network, in: Proceedings of the 20th IEEE International Conference on Data Mining, ICDM, 2020, pp. 841–850.
[11] G. Pang, C. Shen, L. Cao, et al., Deep learning for anomaly detection: A review, ACM Comput. Surv. 54 (2) (2021) 1–38.
[12] A. Deng, B. Hooi, Graph neural network-based anomaly detection in multivariate time series, in: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2021, pp. 4027–4035.
[13] T. Pourhabibi, K. Ong, B. Kam, et al., Fraud detection: A systematic literature review of graph-based anomaly detection approaches, Decis. Support Syst. 133 (2020) 113303.
[14] L. Zhao, Y. Song, C. Zhang, et al., T-gcn: A temporal graph convolutional network for traffic prediction, IEEE Trans. Intell. Transp. Syst. 21 (9) (2019) 3848–3858.
[15] S. Erfani, S. Rajasegarar, S. Karunasekera, et al., High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning, Pattern Recognit. 58 (2019) 121–134.
[16] M. Khodayar, J. Wang, Spatio-temporal graph deep neural network for short-term wind speed forecasting, IEEE Trans. Sustain. Energy 10 (2) (2018) 670–681.
[17] Z. Wu, S. Pan, G. Long, et al., Connecting the dots: Multivariate time series forecasting with graph neural networks, in: Proceedings of 26th International Conference on Knowledge Discovery and Data Mining, KDD, 2020, pp. 753–763.
[18] A. Venkitaraman, P. Frossard, Annihilation filter approach for estimating graph dynamics from diffusion processes, in: Proceedings of the 47th IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2022, pp. 5583–5587.
[19] A. Venkitaraman, P. Frossard, Diffusion convolutional recurrent neural network with rank influence learning for traffic forecasting, in: Proceedings of the 18th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, 2019, pp. 678–685.
[20] Z. Li, G. Zhang, J. Yu, et al., Dynamic graph structure learning for multivariate time series forecasting, Pattern Recognit. 138 (2023) 109423.
[21] Z. Shao, Z. Zhang, F. Wang, et al., Pre-training enhanced spatial-temporal graph neural network for multivariate time series forecasting, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 1567–1577.
[22] A. Vaswani, N. Shazeer, N. Parmar, et al., Attention is all you need, in: Proceedings of the 31st Conference on Neural Information Processing Systems, NIPS 2017, 2017.
[23] S. Li, X. Jin, Y. Xuan, et al., Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting, in: Proceedings of the 33rd Conference on Neural Information Processing Systems, NeurIPS 2019, 2019.
[24] G. Zerveas, S. Jayaraman, D. Patel, et al., A transformer-based framework for multivariate time series representation learning, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, 2021, pp. 2114–2124.
[25] V. Chandola, A. Banerjee, V. Kumar, Anomaly detection: A survey, ACM Comput. Surv. 41 (3) (2009) 1–58.
[26] C. Zhou, R. Paffenroth, Anomaly detection with robust deep autoencoders, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 665–674.
[27] C. Ding, S. Sun, J. Zhao, MST-GAT: A multimodal spatial–temporal graph attention network for time series anomaly detection, Inf. Fusion 89 (2023) 527–536.
[28] A. Blázquez, García, A. Conde, MST-GAT: A review on outlier/anomaly detection in time series data, ACM Comput. Surv. 54 (3) (2021) 1–33.
[29] B. Du, X. Sun, J. Ye, et al., GAN-based anomaly detection for multivariate time series using polluted training set, IEEE Trans. Knowl. Data Eng. (2021) 1–1.
[30] C. Zhang, D. Song, Y. Chen, et al., A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data, in: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, 2019, pp. 1409–1416.
[31] J. Miao, H. Tao, H. Xie, et al., Reconstruction-based anomaly detection for multivariate time series using contrastive generative adversarial networks, Inf. Process. Manage. 61 (1) (2024) 103569.
[32] F. Kong, J. Li, B. Jiang, et al., Integrated generative model for industrial anomaly detection via bidirectional LSTM and attention mechanism, IEEE Trans. Ind. Inform. 19 (1) (2021) 541–550.
[33] Z. Chen, D. Chen, X. Zhang, et al., Learning graph structures with transformer for multivariate time-series anomaly detection in IoT, IEEE Internet Things J. 9 (12) (2021) 9179–9189.
[34] X. Geng, X. He, L. Xu, et al., Graph correlated attention recurrent neural network for multivariate time series forecasting, Inform. Sci. 606 (2022) 126–142.
[35] Z. Li, J. Yu, G. Zhang, et al., Dynamic spatio-temporal graph network with adaptive propagation mechanism for multivariate time series forecasting, Expert Syst. Appl. 216 (2023) 119374.

[36] I. Marisca, A. Cini, C. Alippi, et al., Learning to reconstruct missing data from spatiotemporal graphs with sparse observations, in: Proceedings of the 36th Conference on Neural Information Processing Systems, 2022, pp. 32069–32082.

[37] H. Zhou, S. Zhang, J. Peng, et al., Informer: Beyond efficient transformer for long sequence time-series forecasting, in: Proceedings of the 35th AAAI Conference on Artificial Intelligence, 2021, pp. 11106–11115.

[38] J. Xu, H. Wu, J. Wang, et al., Anomaly transformer: Time series anomaly detection with association discrepancy, in: Proceedings of the 10th International Conference on Learning Representations, 2022.

[39] R. Cirstea, C. Guo, B. Yang, et al., Triformer: Triangular, variable-specific attentions for long sequence multivariate time series forecasting–full version, in: Proceedings of the 31st International Joint Conference on Artificial Intelligence, 2022, pp. 1994–2001.

[40] X. Zhang, Z. Zhao, T. Tsiligkaridis, et al., Self-supervised contrastive pre-training for time series via time-frequency consistency, Adv. Neural Inf. Process. Syst. 35 (2022) 3988–4003.

[41] Q. Wen, T. Zhou, C. Zhang, et al., Transformers in time series: A survey, in: Proceedings of the 32nd International Joint Conference on Artificial Intelligence, 2023, pp. 6778–6786.

[42] Z. Zhu, M.Z. X. Cao, J. Huang, et al., Probability-based global cross-modal upsampling for pansharpening, in: Proceedings of the 22nd IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 14039–14048.

[43] W. Hu, Y. Yang, Z. Cheng, et al., Time-series event prediction with evolutionary state graph, in: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, 2021, pp. 580–588.

[44] Y. Zhou, H. Ren, Z. Li, et al., Multivariate time series forecasting with dynamic graph neural odes, IEEE Trans. Knowl. Data Eng. 35 (9) (2021) 107153.

[45] G. Liang, P. Tiwari, S. Nowaczyk, et al., Dynamic causal explanation based diffusion-variational graph neural network for spatio-temporal forecasting, 2023, Arxiv abs/2305.09703.

[46] Y. Yang, T. Liu, Y. Wang, et al., Graph neural networks inspired by classical iterative algorithms, in: Proceedings of the 38th International Conference on Machine Learning, 2021, pp. 11773–11783.

[47] H. Wu, J. Xu, J. Wang, et al., Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, Adv. Neural Inf. Process. Syst. 34 (2021) 22419–22430.

[48] Y. Song, Z. He, H. Qian, et al., Vision transformers for single image dehazing, IEEE Trans. Image Process. 32 (2023) 1927–1941.

[49] H. Zhou, K. Yu, X. Zhang, et al., Contrastive autoencoder for anomaly detection in multivariate time series, IEEE Trans. Image Process. 610 (2022) 266–280.

[50] H. Ren, B. Xu, Y. Wang, et al., Time-series anomaly detection service at microsoft, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 3009–3017.

[51] K. Hundman, V. Constantinou, C. Laporte, et al., Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, pp. 387–395.

[52] K. Faber, M. Pietron, D. Zurek, Ensemble neuroevolution-based approach for multivariate time series anomaly detection, Entropy 23 (11) (2021) 1466.

[53] A. Garg, W. Zhang, J. Samaran, et al., An evaluation of anomaly detection and diagnosis in multivariate time series, IEEE Trans. Neural Netw. Learn. Syst. 33 (6) (2021) 2508–2517.

[54] L. Shen, Z. Li, J. Kwok, Timeseries anomaly detection using temporal hierarchical one-class network, Adv. Neural Inf. Process. Syst. 33 (2020) 13016–13026.

[55] J. Lee, B. Park, D. Chae, Duogat: Dual time-oriented graph attention networks for accurate, efficient and explainable anomaly detection on time-series, in: Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, 2023, pp. 1188–1197.

[56] Y. Zhang, Y. Chen, J. Wang, et al., Unsupervised deep anomaly detection for multi-sensor time-series signals, IEEE Trans. Knowl. Data Eng. 35 (2) (2021) 2118–2132.

[57] S. Tuli, G. Casale, N. Jennings, Tranad: Deep transformer networks for anomaly detection in multivariate time series data, in: Proceedings of the 48th International Conference on Very Large Databases, 2022.

[58] Y. Chen, C. Zhang, M. Ma, et al., Imdiffusion: Imputed diffusion models for multivariate time series anomaly detection, in: Proceedings of the 50th International Conference on Very Large Databases, 2024.

[59] M. Pietroń, D. Żurek, K. Faber, et al., AD-NEv: A scalable multilevel neuroevolution framework for multivariate anomaly detection, IEEE Trans. Neural Netw. Learn. Syst. (2024) 1–15.

[60] Y. Liang, J. Zhang, S. Zhao, et al., Omni-frequency channel-selection representations for unsupervised anomaly detection, IEEE Trans. Image Process. 32 (2023) 4327–4340.

[61] C. Lai, F. Sun, Z. Gao, et al., Nominality score conditioned time series anomaly detection by point/sequential reconstruction, in: Proceedings of the 36th Neural Information Processing Systems, 2023.

Rong Gao received the Ph.D. degree from Wuhan University, Wuhan, China, in 2018. He is currently an assistant professor in the School of Computer Science, Hubei University of Technology, Wuhan, China. His research interests include machine Learning and data mining.

Jiming Wang received a B.Sc. degree in Data Science and Big Data Technology from Hubei University of Engineering in Hubei, China, in 2022. He is currently pursuing a Master's degree in Computer Science at Hubei University of Technology. His research interests include machine learning, and data mining.

Yonghong Yu is a professor in Nanjing University of Posts and Telecommunications. He received the Ph.D from Nanjing University. His main research interests include machine learning and data mining.

Jia Wu (M'16) received the Ph.D. degree in computer science from the University of Technology Sydney, Ultimo, NSW, Australia. He is currently an ARC DECRA Fellow in the Department of Computing, Macquarie University, Sydney and a Visiting Professor in School of Computer Science, Wuhan University, China. Prior to that, he was with the Centre for Artificial Intelligence, University of Technology Sydney. His current research interests include data mining and machine learning. Since 2009, he has published 100+ refereed journal and conference papers, including IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), IEEE Transactions on Knowledge and Data Engineering (TKDE), IEEE Transactions on Neural Networks and Learning Systems (TNNLS), ACM Transactions on Knowledge Discovery from Data (TKDD), IEEE Transactions on Industrial Informatics (TII), International Joint Conference on Artificial Intelligence (IJCAI), AAAI Conference on Artificial Intelligence (AAAI), IEEE International Conference on Data Mining (ICDM), and SIAM International Conference on Data Mining (SDM). Dr Wu was the recipient of SDM'18 Best Paper Award in Data Science Track, IJCNN'17 Best Student Paper Award, and ICDM'14 Best Paper Candidate Award. He is the Associate Editor of the ACM Transactions on Knowledge Discovery from Data (TKDD), Journal of Network and Computer Applications (JNCA) and Neural Networks (NN).

Li Zhang is an Associate Professor & Reader in University of London, UK and also serving as an Honorary Research Fellow in the University of Birmingham, UK. Dr Zhang holds expertise in artificial intelligence, machine learning, intelligent robotics and affective computing. She also gained her Ph.D. and postdoctoral experience from University of Birmingham previously. She has served as a programme co-chair and IPC member for international conferences and as an associate editor for Decision Support Systems. Dr Zhang is a member of IEEE.