HFN:Heterogeneous Feature Network for Multivariate Time Series Anomaly Detection

Jun Zhan[a,b,e], Chengkun Wu[b,c,*], Canqun Yang[d], Qiucheng Miao[b], Xiandong Ma[f,**]

[a] *School of Intelligent Manufacturing, Hunan First Normal University, Changsha, 410205, China*

[b] *College of Computer Science, National University of Defense Technology, Changsha, 410073, China*

[c] *State Key Laboratory of High Performance Computing, Changsha, 410073, China*

[d] *National SuperComputing Center in Tianjin, Tianjin, 300000, China*

[e] *Key Laboratory of Industrial Equipment Intelligent Perception and Maintenance Technology in College of Hunan Province, Hunan First Normal University, Changsha, 411201, China*

[f] *School of Engineering, Lancaster University, LA1 4YW, Lancaster, UK*

Abstract

As the key step of anomaly detection for multivariate time-series (MTS) data, learning the relations among different variables has been explored by many approaches. However, most of the existing approaches do not consider the heterogeneity between variables, that is, different types of variables (continuous numerical variables, discrete categorical variables or hybrid variables) may have different and distinctive edge distributions. In this paper, we propose a novel semi-supervised anomaly detection framework based on a heterogeneous feature network (HFN) for MTS. Specifically, we first combine the embedding similarity subgraph generated by sensor embedding and the feature value similarity subgraph generated by sensor values to construct a time-series heterogeneous graph, which fully utilizes the rich heterogeneous mutual information among variables. Then, a prediction model containing nodes and channel attentions is jointly optimized to obtain better time-series

*Corresponding author

**Corresponding author

Email addresses: chengkun_wu@nudt.edu.cn (Chengkun Wu), xiandong.ma@lancaster.ac.uk (Xiandong Ma)

representations. This approach fuses the state-of-the-art technologies of heterogeneous graph structure learning (HGSL) and representation learning. Experiments conducted on four sensor datasets from real-world applications demonstrate that our approach detects the anomalies more accurately than those baseline approaches, thus providing a basis for the rapid positioning of anomalies.

Keywords: Heterogeneous neural network; Anomaly detection; Multi-sensor data; Multivariate time series; Deep learning

---

1. Introduction

As information technology develops, an increasing number of industrial systems are exposed to the internet, posing serious risks to their ability to operate securely [1]. Continuous monitoring the operation data of the system and precisely and effectively identifying potential attacks or the evolution of the equipment condition by using this data is an effective technique to handle these challenges [2]. For instance, an operation and maintenance personnel in a large power plant can quickly identify abnormal sensor behavior using the precise intrusion detection systems, which are developed by massive amounts of data collected by the supervisory control and data acquisition (SCADA) system [3], providing them a possibility to prevent potential system failures before irreversible damage. However, these monitoring data always have complicated structures, high dimensionality, and hard labeling, making manual tasks difficult to handle. Therefore, it is vitally necessary to investigate the semi-supervised or unsupervised time-series anomaly detection approach by utilizing a sizable amount of complicated unlabeled data.

Recently, deep learning technique has been applied successfully in various anomaly detection problems [4, 5, 6]. For high-dimensional MTS analysis, the temporal relations between different timestamps are considered first [7]. Because of their capability of capturing long-term dependency relations, recurrent neural network [8] and temporal convolutional network [9] were demonstrated to achieve better results on the time-series tasks involving single or multiple variables [10]. However, various sensors could be mutually coupled. The capacity of these approaches to detect abnormalities may be constrained by their modeling of solely temporal variables. Therefore, it is crucial to take into account both the temporal features of different timestamps and potential correlations among these variables [11, 12]. Combining the sequential network and the convolution neural network (CNN) is an effective way to achieve this. Cross-correlation among high-dimensional data can be extracted by using the local perception capacity of the convolutional kernel [13]. However, CNN is primarily used to handle Euclid-space data, such as image [14]. There exist some limitations on the MTS with different attributes. In such cases, the graph neural network (GNN) has been successfully applied into the modelling of MTS due to its good structure modelling capability between complex data; the most advanced results are achieved in [11, 15].

With regards to the latent feature modeling of time-series data, the variable attributes from the data are generally seen as homogeneous in the most existing papers; that is, the data types are treated without distinction, such as use of the variational autoencoders [16] and generative adversarial networks [17]. These methods model complex distribution from large-scale high-dimensional datasets. After the training is finished by using the dataset from normal conditions, the similar generative data are viewed as normality, while the dissimilar data are viewed as anomalies. However, there are still fewer works considering the heterogeneity of time-series data, although this kind of data are abundant in practical situations. For instance, in a large-scale water processing system [17], the information, such as flow, pressure and liquid level collected by the sensors in the intermediate process, is collected as the numeric continuous values. However, the signals, such as valve state and location collected by the sensors of the actuator, are generally the categorical discrete values. Inputting the mixed type of heterogeneous data into a deep learning network may cause the useful information to be ignored and therefore satisfied results cannot be obtained. The fundamental reason is that there are totally different edge distributions between the variables with different types [18, 19].

To overcome the limitation of deep learning model in such circumstances, we propose a heterogeneous feature learning network for MTS, and study its abnormal detection capability with the extensive real-world datasets. The overall framework can be divided into three stages: 1) Heterogeneous graph structure learning (HGSL) stage for MTS. We fuse the sensor embedding vector similarity matrix and the feature value similarity matrix of different variable categories to model the heterogeneous structural information. Moreover, we propose a category-based fixed-length approach to replace the widely used meta-path [20] for extracting heterogeneous relation subgraphs. 2) Heterogeneous representation learning stage for MTS. We embed different kinds of variables into vectors for fusion. Distinct from the previous heterogeneous graph attention network [21], we further expand the channel attention on the basis of node attention and semantic attention, so as to achieve a joint optimization training of node embedding representation with different types. 3) Abnormal detection and location stage. By analyzing the deviation between the predicted and real values, we calculate a condition score for each sensor, where the largest condition score is considered as the maximum abnormal probability.

The major contributions of the paper are summarized as follows:

- We propose a novel HGSL approach for MTS, which learns heterogeneous graph structure information between sensor-embedding vectors and category-based feature value vectors simultaneously.
- We propose a heterogeneous feature network (HFN) and apply it to MTS anomaly detection. Our approach successfully learned the dynamic dependency among different variables and timestamps by utilizing two single-level attention mechanisms, namely attention-based node embedding and channel aggregation.
- The extensive experiments indicate that HFN can detect the anomalies from real-world MTS datasets and is proved to outperform the most existing methods. Besides, we analyze the condition scores of MTS, demonstrating that the proposed method has the advantage of locating the anomalies.

The rest of this paper is structured as follows. Section 1 describes the related work of MTS anomaly detection. Section 2 presents the structure and working principle of HFN-based MTS anomaly detection framework in detail. Section 3 show the performance of proposed method on three real-world MTS datasets. Finally, the conclusion and future improvements are given in Section 4.

2. Related work

MTS anomaly detection has extensive application prospects in the fields of industry, financial business, and the Internet of Things. As the key research problem in this paper, we firstly review the related work for MTS anomaly detection, which can generally be categorized as unsupervised, supervised, and semi-supervised. We focus on studying data heterogeneity modeling of MTS, especially heterogeneity representation learning from time-series data, graph structure learning, and heterogeneous graph neural network.

2.1. MTS anomaly detection

MTS anomaly detection is typically regarded as an unsupervised learning problem [22], and algorithms based on clustering [23], such as fuzzy c-means [24], or spatiotemporal clustering [25], are frequently used. By grouping time-series data into various clusters, these techniques can identify anomalies by calculating the similarity or distance between the observed value [26] and the cluster center [27]. However, unsupervised detection methods usually focus more on static data model development. In contrast, a supervised abnormal detection algorithm has a higher detection accuracy. Under the circumstance of high-quality labeling, the indicator accuracy can be approximate to 100% [28]. However, the supervised detection requires that the training set contains correctly both labeled positive and negative samples, which is often not easy. [29]. Fortunately, in the actual cases, we have a chance to obtain a large quantity of data under the normal conditions [17], making the semi-supervised abnormal detection attract wide attentions [30]. In the latest work, Miryam et al. [31] proposes the methods to show the great advantages and extensive application prospects of the semi-supervised algorithm in MTS abnormal detection.

2.2. Modeling for heterogeneous data

The data heterogeneity has been widely concerned such as in the music recommendation system [32], academic network [33] and social platform [34]. The heterogeneous learning method usually focuses on capturing and integrating couplings with multiple variable types at the same or different levels. To learn the embedding representation of heterogeneous data, the matrix decomposition method is traditionally adopted [35, 36]. However, it is usually very expensive and low-efficient in terms of the computation cost of decomposing a large-scale matrix [37]. Moreover, the discretization of continuous features [38] or continuous data [39] are also a typical method; however this transformation may ignore the correlation between variables. To solve these challenges, heterogeneous graph embedding or heterogeneous graph representation learning [40] has been widely studied. Its main goal is to map the input data into low-dimensional space while simultaneously preserving the heterogeneous structure and semantic characteristics of the data [41]. For instance, for the tasks of text classification, Wang et al. [21] proposed a heterogeneous graph attention network (HAN), which aggregates the features of meta-path based neighbors through a hierarchical manner to generate the embedding representation of nodes. Fu et al. [42] proposed a meta-path aggregated graph neural network (MAGNN) by designing multiple candidate encoder functions to extract heterogeneous information from the meta-path. Wang et al. [43] combined the heterogeneous graph neural network with comparison learning, and proposed a self-supervised heterogeneous graph neural network from both heterogeneous network and meta-path for learning node embedding representation. In the social or citation network, in order to capture the dynamic performances of heterogeneous redgraphs, Hu et al. [44] proposed a heterogeneous graph transformer (HGT) by introducing a relative temporal encoding technique for solving the problem where the dynamic result dependence is difficult to capture. Yang et al. [45] proposed a dynamic heterogeneous graph (DyHAN) utilizing structural heterogeneity and time revolution to learn node embedding. In addition, contrastive self-supervised learning has been widely employed to address the limitation of sparse label information in the potential ability of heterogeneous graph neural network models for representation learning. For instance, the HGCL method proposed by Chen et al.[46] effectively utilizes the structural information of heterogeneous graphs to capture relationships between different types of nodes. Zhu et al.[47] combine heterogeneous graph contrastive learning with a structure-enhancement method, proposing the STENCIL method. This approach introduces a novel multi-view contrastive aggregation objective to adaptively distill information from each view. Furthermore, the method enriches the local structural patterns of the underlying heterogeneous graph to better explore true and challenging negative examples in graph contrastive learning.Although the above methods have achieved significant success in their respective application domains, leveraging the structure of heterogeneous graphs to enhance data representation capabilities and demonstrating outstanding performance through representation learning methods, their applicability may be subject to domain specificity and might not necessarily be suitable for other areas such as multivariate time series anomaly detection.

2.3. Graph structure learning

MTS usually exists in the form of tabular data [48], lacking of predefined graph structure required for graph neural network [15], which constitutes the challenge for the modelling [49]. Hence, it is extremely vital to learn the links between edges and refine the graph from the existing time-series data [50]. The existing methods can mainly be divided into three categories: metric-based approaches usually implemented by using kernel function [51, 52], cosine similarity [53, 54] or inner product [55] to calculate the similarity between nodes as edge weights. Neural networks-based approaches have generally utilized a complex deep neural network to model the edge weights of the given node features and representations. For instance, Luo et al. [56] proposed a multilayer perception-based graph structure optimization approach, where the edge number of a sparse graph is punished through parameterized network for pruning the edges that are unrelated to the tasks. Zhao et al. [11] proposed a graph structure learning approach with redan attention coefficient, while Sun et al. [57] utilized a dot-product self-attention to model the dynamic connection relations between the nodes. Direct learning approaches, regarding adjacent matrix as a learnable parameter, make associative learning together with the follow-up tasks for optimization. For instance, Gao et al. [58] proposed the graph learning neural networks (GLNNs) utilizing spectral graph theory for graph learning. However, these approaches mostly aim at learning isomorphic graph structure. To enable capture the heterogeneity between the data efficiently, Zhao et al. [41] proposed a heterogeneous graph learning approach utilizing the fusion of feature similarity sub-graph, feature propagation graph and semantic graph, which successfully learns an appropriate graph structure for a heterogeneous graph neural network.

3. Proposed Frameworks

3.1. Problem statement

Generally, we define heterogeneous MTS dataset as a time-series dataset with L variables, N different types of sensors, and T length, which is expressed as $X = \left\{ \boldsymbol{x}_{1:T}^{N} \right\}$, where $N \in \{type^1, \cdots, type^n\}$ denotes the set of data types. Note that the variable number contained in the specified categories may be larger than 1. For instance, for arbitrary data type $type^n$, all time series at the moment t can be denoted as $\boldsymbol{x}_t^{type^n} \in \{x_t^{type_i^n}, for\ i \in \{0, \cdots, d\}\}$, where d represents the number of time-series sequence in this category. In this paper, we adopt the sliding window-based model training approach. At the moment t, we sample a continuous subsequence with the length of ω as the model input, denoted as $S^N(t) = \left[\boldsymbol{x}_{t-w+1}^{N}, \cdots \boldsymbol{x}_t^N \right]$. For the abnormal detection task, our target is to predict the value of all sensors $\boldsymbol{x}_{t+1}^{N}$ at the moment $t+1$ by utilizing the input subsequence $S^N(t)$, and obtain the predicted value $\widehat{\boldsymbol{x}}_{t+1}^N$. The mean square error (MSE) between the predicted value and practical value is used as loss to optimize the model. According to the usual semi-supervised abnormal detection methods, in the training stage, only the data collected from normal conditions are chosen. However, in the testing stage, the deviation between the predicted value and practical value is further used for calculating the condition scores of the data, while the scores of the corresponding data over the threshold are judged as the anomalies, otherwise normal.

Specially, we divide time-series data into three data types, that is $N \in \{C, CD, D\}$:

- Continuous numerical variables C, where the value of data are taken from continuous real number, such as $x_t^{C_i} \in \mathbb{R}$.
- Discrete categorical variables D, where the value of data are taken from a limited set of values, such as $x_t^{D_i} \in \{0,\ 1,\ 2\}$.
- Hybrid variables S_t^{CD} which contain both numerical and categorical variables where the values of the element are taken from the above two categories.

We construct a heterogeneous dynamic graph to model the above MTS. Different time-series variables are viewed as the node in the graph, while their connection relation is seen as the edge. This dynamic graph can be denoted as $\mathbb{G}_{S^N(t)} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ represent node and edge set respectively. We respectively extract categorical feature subgraph $\mathbb{G}_{S^D(t)}$, numerical feature subgraph $\mathbb{G}_{S^C(t)}$, and categorical and numerical mixed subgraph $\mathbb{G}_{S^{CD}(t)}$ for learning heterogeneous information. For the arbitrary subgraph, its adjacent matrix is $A_N \in \mathbb{R}^{|\mathcal{V}_N| \times |\mathcal{V}_N|}$, where $\mathcal{V}_N$ represents the node set with the specific type. If there exist connection relations between two arbitrary nodes in the subgraph, the corresponding element of adjacent matrix is 1. Noted that the final node embedding integrates the node embedding representations of three different subgraphs.

3.2. Model Architecture

Our HFN-based approach aims at learning the complex correlation between different types of time-series data carried by the defined dynamic graph

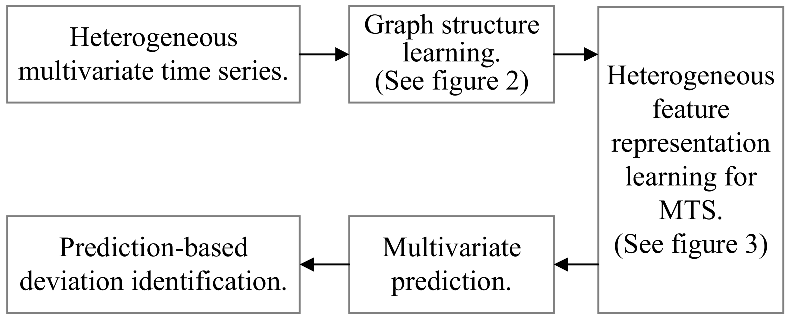

Figure 1: Architecture of HFN-based MTS anomaly detection framework.

above. For each node, the potential temporal correlation is allowed to be considered with a sliding window along the dataset.

Figure 1 shows the proposed HFN-based semi-supervised abnormal detection framework architecture. It can be seen that for a given MTS, we firstly learn a heterogeneous dynamic graph representing the structural information between different variables (as shown in Figure 2), decomposing the time-series data into different graph structures. On this basis, the categorical feature subgraph, the continuous numerical feature subgraph and the hybrid subgraph are extracted and then inputted into the HFN network based on graph attention function to learn the potential embedding representations of each sensor (as shown in Figure 3). Then we predict the future values of each sensor based on these embedding representations. Finally, the deviation between the predicted and practical values is used for measuring and locating the anomalies.

3.3. Graph structure learning pipeline

To learn the complex heterogeneous potential features between different types of sensors, a key process is how to map the variable correlation from MTS into the adjacent matrix of the graph. In the previous studies, all assumed that the constructed graph is the static isomorphic graph, thus resulting in the loss of some key information. For instance, the significance of variables exists great difference at the operating condition of full-load and partial-load of generating equipment [59]. Hence, as shown in Figure 2, we learn the potential heterogeneous graph structure of MTS from the perspectives of global semantic correlation and local feature correlation. For the global semantic correlation, we introduce a learnable embedding vector

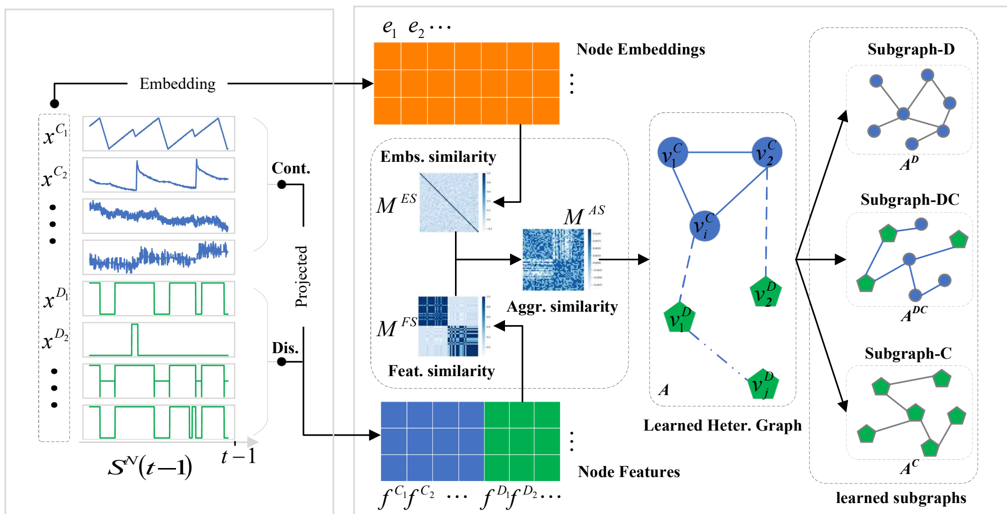

Figure 2: Structure learning of MTS heterogeneous dynamic graph.

for each variable, and denote it as $\boldsymbol{e}_i \in \mathbb{R}^{1\times\omega'}$. For $i \in \{0, \cdots, L\}$, where ω' represents the dimension of embedding vector. This vector can be learned together with subsequent prediction network parameters. For the local feature correlation, we calculate the potential structural information based on the feature values of the variables. We adopt a special mapping network to project different types of input feature vector $S^N(t)$ into a public space. Taking data type C as an example, the projected feature of the arbitrary variable x^{C_i} is denoted as $f^{C_i} \in \mathbb{R}^{1\times\omega'}$:

$$f^{C_i} = SELU\left(x^{C_i} \bullet \boldsymbol{W^C} + \boldsymbol{b^C}\right) \tag{1}$$

where $x^{C_i} \in \mathbb{R}^{1\times\omega}$ is the subset of all continuous numerical variables. ω is the time length of input feature vector. $\boldsymbol{W^C} \in \mathbb{R}^{\boldsymbol{\omega}\times\omega'}$ is learnable weight matrix, and $\boldsymbol{b^C} \in \mathbb{R}^{\boldsymbol{1}\times\omega'}$ is biasing. Similarly, we can calculate and obtain the projected feature representation f^{D_i} of the discrete categorical variables.

3.4. Similarity Graphs

The main task of graph structure learning is to learn an adjacent matrix representing the mutual connection between nodes in the graph. Therefore, we propose a learning approach based on aggregating cosine similarity. According to the embedding for the variables and the mapping of variable feature vectors, we obtain the global semantic embedding matrix

$\boldsymbol{E} \in \{e_1, \cdots e_L\}$ and local feature vector representation matrix $\boldsymbol{F^N} \in \{f^{N_1}, \cdots, f^{N_L}\}$. Clearly, these obtained matrices from different perspectives contain different information. Specifically, we first calculate cosine similarity between the elements in different matrixes to obtain their connection information. After obtaining the node embedding (NE) similarity matrix $M^{E_s} \in \mathbb{R}^{L\times L}$ and node feature (NF) similarity matrix $M^{F_s} \in \mathbb{R}^{L\times L}$, we fuse them to obtain an aggregating similarity matrix, where the value represents the similarity between the arbitrary two nodes i and j and can be calculated as follows:

$$M^{E_s}[i,j] = \frac{e_i \bullet e_j}{e_i \times e_j} \tag{2}$$

$$M^{F_s}[i,j] = \frac{f^{N_i} \bullet f^{N_j}}{f^{N_i} \times f^{N_j}} \tag{3}$$

$$M^{A_s} = M^{E_s} \circ \boldsymbol{W}^{E_s} + M^{F_s} \circ \boldsymbol{W}^{F_s} \tag{4}$$

where $\circ$ denotes Hadamard product between two matrixes. $\boldsymbol{W}^{E_s} \in \mathbb{R}^{L\times L}$ and $\boldsymbol{W}^{F_s} \in \mathbb{R}^{L\times L}$ are learnable weight matrixes, which weigh the importance of different dimensions of the different similarity matrixes. In M^{A_s}, when the correlation coefficient is larger than a certain threshold, we consider that there exists a connected relation between nodes; otherwise, the connected relation does not exist. To obtain the optimal threshold, we define a learnable parameter $\tau \in \mathbb{R}$ for automatic choice, and obtain the adjacent matrix of aggregating similarity graph through learning, which is denoted as:

$$A_{ij} = \begin{cases} 1 & for\ M^{A_s}[i,j] \geq \tau \\ 0 & for\ M^{A_s}[i,j] < \tau \end{cases} \tag{5}$$

In the heterogeneous dynamic graph, two objects can be connected through different semantic paths, which is called meta-path. However, the selection of meta-path has a strong subjective meaning, which is difficult for complex MTS. Therefore, we propose a classifying-based fixed-length sampled method to replace meta-path for extracting heterogeneous relation subgraphs. Specifically, we divide the aggregating similarity graph into the corresponding classifying subgraphs, including discrete feature subgraph (DFS) $\mathbb{G}_{S^D(t)}$, continuous feature subgraph (CFS) $\mathbb{G}_{S^C(t)}$ and hybrid feature subgraph (HFS) $\mathbb{G}_{S^{CD}(t)}$ according to data types. We further make a random mask operation for the neighboring matrix of the subgraph and obtain the final neighboring matrix with different relations. The transformed heterogeneous graph structure is $A^{'}=\{A^D, A^C, A^{CD}\}$. The random mask is conducive to exchange information between different similarity matrixes in the graph structure learning process, thus improving the accuracy of subsequent tasks and relieving the overfitting problem.

3.5. Graph representation learning for MTS

It can be seen from the learned heterogeneous graph structure that each type of subgraph contains different semantic properties. Hence, to aggregate the node information from different types, we introduce a graph attention-based node embedding network and an attention-based channel aggregating network to construct the HFN for MTS. The structure is shown in Figure 3. Specifically, the obtained three subgraphs A^D, A^C, A^{CD} learned by graph structure learning are inputted into three independent graph attention networks, to learn the importance of different types of nodes for the neighbors in the subgraphs. Moreover, the important neighboring information is aggregated to generate a new node embedding. As shown in Figure 3, taking the continuous numerical variable channel as an example, for the arbitrary node v_i^C and its neighboring node v_j^C in subgraph A^C, we perform self-attention in the nodes. The attention coefficient representing their relation importance can be calculated as:

$$\xi_{ij}=att\left(\boldsymbol{W}f^{C_i},\boldsymbol{W}f^{C_j};A^C\right) \tag{6}$$

where $f^{C_i}\in\mathbb{R}^{1\times\omega^{'}}$ and $f^{C_j}\in\mathbb{R}^{1\times\omega^{'}}$ are mapped node feature vectors, $\boldsymbol{W}\in\mathbb{R}^{\omega^{''}\times\omega^{'}}$ is shared weight matrix. $\omega^{'}$ and $\omega^{''}$ are the calculated node feature vector dimensions before and after the embedding. After obtaining the importance of subgraph-based node pairs, we normalize them via the SoftMax function and obtain weight coefficient α_{ij}:

$$\alpha_{ij}=softmax\left(\sigma_{ij}\right)=\frac{exp\left(\delta\left(\overrightarrow{a}^T\left[\boldsymbol{W}f^{C_i}||\boldsymbol{W}f^{C_j}\right]\right)\right)}{\sum_{\eta\in\mathrm{N}_i^C}exp\left(\delta\left(\overrightarrow{a}^T\left[\boldsymbol{W}f^{C_i}||\boldsymbol{W}f^{C_\eta}\right]\right)\right)} \tag{7}$$

where δ is the activation function, and LeakyReLU function is usually adopted [55]. $\overrightarrow{a}\in\mathbb{R}^{2\omega^{''}}$ is the learnable weight vector, which denotes the information concatenation of the two nodes. Finally, the output of each node can be obtained through aggregating its neighboring nodes. Multi-head attention mechanism is proven to be beneficial in the learning process

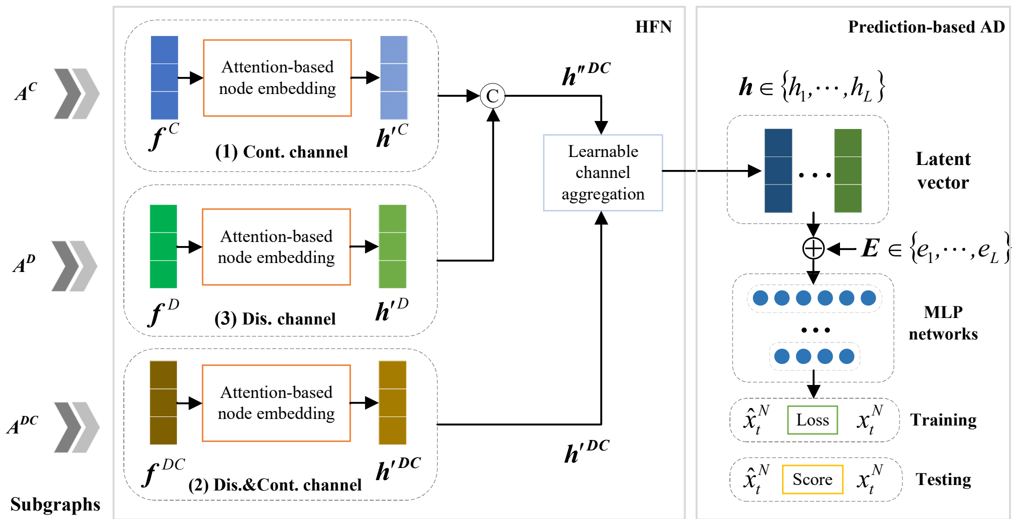

Figure 3: Heterogeneous feature network structure of MTS.

of stabilizing self-attention [60]. To be convenient for training, we perform an average operation to aggregate the results handled by multi-head attention. After the graph attention-based nodes embed into the network, the implicit vector can be represented as:

$$\boldsymbol{h}_i^{\prime C} = \sigma\left(\frac{1}{H}\sum_{h=1}^{H}\sum_{j\in \mathrm{N}_i^C} \alpha_{ij}^h \boldsymbol{W}^h f^{C_j}\right) \tag{8}$$

where N_i^C is the set of nodes i's neighbors in the continuous subgraph. H denotes the number of multi-head attention mechanism head. According to the same computation method, we can obtain the node implicit vectors of discrete subgraph and mixed subgraph represented by $\boldsymbol{h}_i^{\prime D}$ and $\boldsymbol{h}_i^{\prime CD}$.

To address the node semantic importance of different types in the heterogeneous graph, we put forward an attention-based multi-channel node embedding aggregating network. We can clearly see from Figure 3 that the node implicit vectors $\boldsymbol{h}^{\prime C}$ and $\boldsymbol{h}^{\prime D}$ singly from a continuous channel and discrete channel first are concatenated in feature dimension to obtain the global node implicit vector $\boldsymbol{h}^{\prime\prime DC}$. The main purpose is to achieve the joint embedding representation learning of all nodes simultaneously. Then $\boldsymbol{h}^{\prime\prime DC}$ and the node implicit vector $\boldsymbol{h}^{\prime CD}$ from the mixed channel are sent to the multi-channel node embedding aggregating network for aggregating their heterogeneous information. The aggregating network automatically learns the importance degree β of the embedding vectors between different channel node implicit vectors, which can be explained as the contribution of the node correlation due to the different types of variables. The final embedding vector is computed as follows:

$$\boldsymbol{h}=\beta\left(\boldsymbol{h}^{'C}||\boldsymbol{h}^{'D}\right)+(1-\beta)\,\boldsymbol{h}^{'CD} \tag{9}$$

where $\beta\in\mathbb{R}$ is a learnable parameter representing the importance degree of the embedding vectors between different channel node implicit vectors, and $||$ is a concatenation operation.

3.6. Prediction-based anomaly detection pipeline

From the above node heterogeneous feature learning network, we obtain new embedding representations of all nodes. Finally, as shown in Figure 3, we input the embedding data fused with $\boldsymbol{h}$ and embedding vector $\boldsymbol{E}$ into the MLP layer to have the predicted value $\widehat{\boldsymbol{x}}_t^N$ of all sensors at the moment t:

$$\widehat{\boldsymbol{x}}_t^N=SeLU\left(f\left(\boldsymbol{h}\oplus\boldsymbol{E}\right)\right) \tag{10}$$

where $f\left(\cdot\right)$ is multiple layers of MLP output layer. $SeLU$ is activation function, and $\oplus$ is addition operation.

At the training stage, we adopt MSE as the loss function of the model:

$$\mathcal{L}_{mse}=\frac{1}{L}\sum_{i}^{L}\left(\boldsymbol{x}_t^N-\widehat{\boldsymbol{x}}_t^N\right)^2 \tag{11}$$

After the training is finished, we apply the network to perform real-time abnormal detection tasks. By comparing the predicted and original values of the input, we calculate the condition scores of each sample in time-series data. We define the difference between the original value and predicted value as the condition scores. To eliminate the effect of different variable dimensions, we normalize the condition scores. Finally, the condition score is computed as follows:

$$Score_i=\frac{\left|\boldsymbol{x}_t^{N_i}-\widehat{\boldsymbol{x}}_t^{N_i}\right|-IQR_i}{\mu_i+1} \tag{12}$$

where IQR_i denotes an interquartile range of the predicted value of the ith variable, μ_i is its median. To achieve the anomaly positioning, we take the largest value of $Score_i$ as the condition score of overall record data at the moment t, as denoted by $Score{=}max\left(Score_i\right)$. Finally, if the $Score$ is larger than the threshold, this record is judged as an anomaly. However, because the threshold selection refers to complicated domain knowledge and the selection methods are various depending on the applications [61], this paper will not further explore the selection method for the threshold. The experiment in the subsequent section will report the optimal value of each evaluating metric (see details in Section 3.3).

3.7. Training

Following the application of the components introduced in the preceding sections, predictions for multivariate time series can be acquired. The fundamental concept of our approach centers on maximizing the utilization of diverse sensor data types within the time series, enhancing prediction accuracy, and identifying anomalies based on prediction errors. To accomplish this, we collaboratively optimize a heterogeneous feature network across multiple channels to update the parameters of the entire network. Throughout the training process, the comprehensive forward propagation procedure is delineated in Algorithm 1.

Algorithm 1 HFN training procedure

Input: Heterogeneous multivariate time series training dataset $S^N\left(t-1\right)=\left[\boldsymbol{x}_{t-w}^N,\cdots\boldsymbol{x}_{t-1}^N\right]$, Batch Size $\mathcal{B}$,Number of Epochs $\mathcal{E}$

Output: Predicted values $\widehat{\boldsymbol{x}}_t^N$

```
for epoch=1:ℰ do
    Calculate the projected feature f^{C_i} and node embedding vector e_i;
    Calculate similarity matrix M^{E_s}, M^{F_s} and M^{A_s} with Eq. (2),Eq. (3) and Eq. (4);
    Calculate adjacent matrix A_N with Eq. (5);
    Extract subgraph features to obtain the node implicit vectors h_i'^C ,h_i'^D and h_i'^CD with Eq. (8);
    Calculate the final embedding vector h with Eq. (9);
    Calculate the predicted value x̂_t^N with Eq. (10);
    Calculate the loss ℒ_mse with Eq. (11);
    Update parameters.
end for
```

Table 1: Statistics of the datasets.

Items	SWaT	WADI	WTD
Time series (C/D)	51 (25/26)	123 (68/55)	37(31/6)
Training dataset	496800	784571	1000000
Testing dataset	449919	172803	940000
Anomaly Rate (%)	11.97%	5.99%	20.64%
Sampling Rate	$1Hz$	$1Hz$	$1Hz$

4. Experiments

We employ extensive experiments on two open and one private real-world datasets to answer the following research questions: (1) Whether the proposed model is more optimal than the baseline models? (2) How each component of the model affects the model? (3) How the proposed approach detects anomalies? (4) How the detection results locate anomalies?

4.1. Benchmark datasets

The selected three datasets contain two datasets (SWaT and WADI) based on water treatment simulator testbed and a real-world dataset from a large-scale wind farm (WTD). The statistical data of the datasets are given in Table 1:

Secure Water Treatment (SWaT) Dataset [62]. This dataset was collected from a six-stage Secure Water Treatment (SWaT) testbed. SWaT represents a scaled-down version of a real-world industrial water treatment plant. It took 11 days for the data collection process, which ran with normal operation mode during the first seven days, and constituted a training dataset. During the later four days, the testbed was implemented by intermittent network and physical attacks, which constituted the labeled testing dataset. The data were collected once every second, containing 51 time-series features, including 25 continuous features and 26 discrete categorical features. We chose this dataset for case study, and the primary sensors or actuators involved are shown in the Table 2 below.

Water Distribution (WADI) Dataset [63]. This dataset was collected from a water distribution testbed (WADI). It took 16 days for the data

Table 2: Statistics of the datasets.

No.	Name	Type	Description
1	FIT-401	Sensor (continuous)	Flow transmitter to control the UV dechlorinator.
2	UV-401	Actuator (discrete)	Dechlorinator to remove the chlorine from water.
3	FIT-504	Sensor (continuous)	Flow meter, a RO re-circulation flow meter.
4	P-501	Actuator (discrete)	Pump to pump the dechlorinated water to RO.
5	LIT-401	Sensor (continuous)	Level transmitter to regulate the RO feed water tank level.
6	LIT-101	Sensor (continuous)	Level transmitter to regulate the raw water tank level.
7	FIT-601	Sensor (continuous)	Flow meter a UF backwash flow meter.
8	AIT-504	Sensor (continuous)	RO permeate conductivity analyzer to measure the NaCl level.
9	AIT-201	Sensor (continuous)	Conductivity analyzer to measure the NaCl level.

collection process. During the last two days, the attack was launched to the testbed with different intentions and time intervals, and the duration of the attack lasted between 1.5 to 30 minutes to acquire the abnormal operating data. The data were collected once every second, containing 123 time-series features, including 68 continuous features and 55 categorical features.

Wind Turbine Dataset (WTD). This dataset was collected from a large-scale wind farm [64]. It lasted 1 to 2 years for the data collection process. At the training stage, there are no abnormal operating data since only the time-based maintenance process was arranged for the wind turbines, while at the testing stage, the abnormal operating data were detected in the repairing process. All data have been labeled by the experts. The data were collected once every 10 minutes, containing 37 time-series features, including 31 continuous features and 6 categorical features.

It is noteworthy that in this paper, the time scales of the time series datasets are uniform, with all datasets adhering to a fixed time scale of 1 second. However, it is crucial to recognize that the time scale, or the sampling rate of the data, can impact the identification results in time series analysis. The uniformity in time scales across the datasets employed in the paper ensures the effective facilitation of direct comparisons between different methods.

4.2. Baseline models

We first compare the FHN model with the most advanced approaches including LSTM-VAE [65], USAD [66], MAD-GAN [17], graph network based MTAD-GAT [11] and GDN [54]. These approaches are extensively concerned with the cross-time and cross-sequence correlation of MTS. The approaches based on sequence reconstruction or prediction are used to learn the representations of the whole time series. Moreover, the anomalies are judged by the reconstructing or predicting errors.

Furthermore, we compare the proposed approach with those classic shallow anomaly detection approaches, including PCA [67], Isolation Forest (IF) [68] and LightGBM [69]. These shallow detection methods are regarded as the relatively direct abnormal detection methods, which usually can directly locate the outlier. Moreover, to complete the anomaly detection in temporally related contexts has also attracted the interests of the researchers, such as LSTM-NDT [1]. The idea underlying this method is to model the temporal features of the data, predict the corresponding values, and then judge whether the anomalies occur by comparing the deviation between the real value and the predicted value.

In addition, we also conducted comparisons with the latest methods based on transformer and spatiotemporal graph approaches. These include: TranAD [70], an anomaly detection and diagnostic model based on deep transformer networks. It employs attention-based sequence encoders for rapid inference, possessing knowledge of broader temporal trends in the data; FuSAGNet [71], which combines sparse autoencoder and graph neural network. The latter predicts future time series behavior from sparse latent representations learned by the former, along with graph structures learned through recurrent feature embedding; MAD-SGCN [72], which effectively captures the spatiotemporal correlations of input sequences using long short-term memory networks (LSTMs) and spectral-based graph convolutional networks (GCNs).

4.3. Evaluation

4.3.1. Metrics

We select precision, recall and F1 as the evaluating metrics of the model, where $Precision = \frac{TP}{TP+FP}$, $Recall = \frac{TP}{TP+FN}$,$F1 = \frac{2\times Precision\times Recall}{Precision+Recall}$. TP, FP and FN refer to true positives, false positives, and false negatives, respectively. These metrics are required to be obtained with a certain threshold. Hence, due to the different threshold selection methods among different tasks, there exist large differences in the metric values. Therefore, to avoid introducing additional hyperparameters, we report the evaluation metrics values when the optimal F1 value is obtained. The threshold value is determined by traversing between the maximum and the minimum scores of the testing dataset.

We calculate the condition scores that decide the abnormal degree of the overall dataset based on eq. (12). Noted that in unsupervised anomaly detection for MTS (USAD) [66] and temporal hierarchical one-class network (THOC) [73], the authors applied a specific evaluation method, called point adjust, making F1 value higher and close to 1. It has been proved that the capability of the model may be highly evaluated [74]. Hence, for the comparison, we apply the open-source code of USAD, and utilize the same parameters of the model in this paper to calculate the performance metrics without adjustment.

4.3.2. Setup

We use Pytorch to achieve the HFN and its variants. Moreover, the model is trained on a server with Intel(R) Xeon(R) Gold 5218R CPU @ 2.1GHz and NVIDIA GeForce RTX 3090 graphics cards. We select Adam optimizer to train the model. Meanwhile, we adopt early stopping to relieve overfitting. The maximum training epoch is set to be 100. If the loss is less than 0.0001 after 10 epochs, the training stops automatically and the optimal model is saved.

The proposed HFN method and the compared baseline models have strived to maintain a similar level of complexity in parameter settings, ensuring a fair comparison. For classical anomaly detection models, including PCA and Isolation Forest, we have maintained the parameter settings at a relatively standard level. The 'contamination' parameter for PCA has been set to 0.05. In Isolation Forest, we opted for 100 isolation trees, each trained using all features. In LightGBM, the 'num_boost_round' parameter has been set to 1000 to ensure the model has a sufficient number of epochs for training.

Regarding deep learning models, in LSTM-NDT, we employed a 4-layer LSTM network, with each layer having 128 hidden nodes. Similarly, in LSTM-VAE, a 4-layer LSTM network was used with 128 nodes in each hidden layer and a latent space dimension of 32. The parameter settings for the DAGMM model align with those specified by the authors in the open-source code, utilizing a Gaussian Mixture Model composed of four individual Gaussian models. For the USAD model, a window length of 15 and a latent space dimension of 40 were set. In the MTAD-GAT model, a convolutional kernel size of 7 was chosen, and the hidden dimensions for the temporal and spatial graph attention networks were set to 150. The prediction and reconstruction networks comprise a 4-layer GRU network. The GDN model has a hidden layer dimension of 128, an output layer with 64 hidden nodes, and a graph network with 4 layers.

Finally, for the proposed HFN method, we selected a structure with a hidden layer dimension of 64 and 4 layers in the graph network to ensure consistency with other deep learning models. This configuration aims to provide each model with similar capabilities in learning data representations, facilitating a more equitable evaluation of their performance in anomaly detection tasks.

Table 3: Precision, recall and F1 values of HFN and all baseline methods on different datasets.

Model	**SWaT**			**WADI**			**WTD**		
	Pre.	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**	**Pre.**	**Rec.**	**F1**
PCA	0.249	0.216	0.230	0.395	0.056	0.100	0.160	0.513	0.244
IF	0.951	0.588	0.727	0.299	0.158	0.207	0.278	<u>0.953</u>	0.430
LightGBM	0.783	0.666	0.719	<u>0.989</u>	0.153	0.270	0.237	0.602	0.340
LSTM-NDT	0.982	0.688	0.809	0.758	0.328	0.457	0.365	0.736	0.497
LSTM-VAE	0.962	0.599	0.740	0.878	0.145	0.250	0.165	0.550	0.254
DAGMM	0.470	0.666	0.551	0.544	0.267	0.360	0.164	0.242	0.195
OmniAnomaly	0.983	0.650	0.782	0.995	0.130	0.230	-	-	-
USAD	0.985	0.661	0.792	0.995	0.132	0.233	0.157	0.417	0.228
MAD-GAN	0.990	0.637	0.770	0.414	0.339	0.370	-	-	-
MTAD-GAT	0.991	0.633	0.772	0.988	0.153	0.265	0.128	0.397	0.193
GDN	<u>0.994</u>	0.681	0.810	0.975	0.402	0.570	0.385	0.937	0.546
TranAD	0.976	0.699	0.815	0.353	0.829	0.495	0.305	0.715	0.428
FuSAGNet	0.988	0.726	0.837	0.830	0.479	<u>0.607</u>	-	-	-
MAD-SGCN	0.986	0.690	0.823	0.564	0.399	0.552	0.416	0.688	0.518
HFN	0.973	**0.758**	**0.852**	0.827	**0.413**	0.551	**0.505**	0.837	**0.630**

4.4. Experimental analysis

The optimal metric values are shown in bold in Table 3. For the datasets SWaT and WADI, we refer to the results in USAD [66] and graph deviation network (GDN) [54]. For WTD dataset, to guarantee the objectivity of the results, we only report the metrics from the obtained open code approaches.

4.4.1. Performance comparison of anomaly detection

To demonstrate the performance of the proposed model, we evaluated the precision, recall, and F1 of all methods on the test set. We can observe from Table 3 that HFN shows a good abnormal detection capability with remarkable performance improvements on SWaT and WTD. The improvement range of the proposed approach is 5% to 14%, as compared with the optimal baseline models. The optimal baseline GDN outperforms our approach in terms of F1; however, our approach has a more optimal recall rate. It is acceptable in real scenarios because we hope to detect more anomalies. In short, HFN outperforms the selected baselines in terms of the overall performances, because it not only concerns with the traditional spatial-temporal correlation, but also obtains its heterogeneous attributes from different types of data, making the model more robust. Moreover, we observe that prediction-based algorithms such as HFN, GDN and LSTM-NDT outperform the reconstruction-based algorithms such as LSTM-VAE and USAD on these datasets, indicating that the prediction-based models have an advantage in the streaming abnormal detection tasks with a single-timestamp value as the target. The temporal information is also very vital in the tasks for MTS abnormal detection. The results of LSTM-NDT show that HFN outperforms all baselines except GDN. The PCA result is dissatisfactory, because it gives more attentions to the point anomalies without spatial-temporal correlation being considered.

Specifically, among these abnormal detection approaches, GDN, MTAD-GAT and HFN adopt the graph attention network to capture the temporal and feature correlations. Therefore, these types of models achieve good results on all datasets. GDN approach recodes multidimensional data at each moment, and utilizes its strong structural learning capability of graph attention network to learn coupling relations between different sensors. However, it does not consider the heterogeneity of data. MTAD-GAT approach also captures time-dimension information through an attention mechanism. Although it considers the spatial-temporal correlation of MTS, it requires a configuration of hyper-parameters for fusing the prediction-based and reconstruction-based condition scores, leading to the evident differences in results when this approach is applied to different datasets. Compared to the recently introduced transformer-based TranAD, as well as the spatial-temporal graph networks FuSAGNet and MAD-SGCN, HFN continues to exhibit superior performance on the SWaT and WTD datasets. However, the most recent experimental outcomes suggest that FuSAGNet achieved the top results on the WADI dataset. Nonetheless, our attempts to reproduce this outcome using the authors' open-sourced code were unsuccessful.

Furthermore, although we processed different types of data separately, our optimization efforts were predominantly concentrated on enhancing the network structure without introducing a significant increase in complexity. Consequently, the processing time did not exhibit a substantial increase when handling the same amount of data.

4.4.2. Ablation experiment

We utilize SWaT and WADI datasets to study the necessity of five components of our approach, namely, node embedding similarity matrix (NE), node feature similarity matrix (NF), discrete feature subgraph (DFS), continuous feature subgraph (CFS), and hybrid feature subgraph (HFS). As shown in Figure 4, we successively exclude the corresponding component from the experiments to observe its effect on the model performance. The key idea of our approach is to learn the potential steady representations from heterogeneous MTS. Hence, first, we exclude NE or NF to study whether the heterogeneous information is learned. Second, we discuss the anomaly detection performance when we only use HFS or DFS and CFS. Specifically:

- Excluding NF (expressed as "-NF") degrades the overall performance of the approach and has a great influence on WADI dataset. This indicates that NF is in favor of feature extraction with high-dimensional dataset for the model; however, NF is not the key factor to determine the model performance.

- Excluding NE (expressed as "-NE") degrades the performances clearly, which implies that NE has an evident advantage in the graph structure learning process.

- Excluding DFS and CFS (expressed as "-DFS" and "-CFS") degrades the model performance; however, the descend range of model performance is less than that of NE. This approach is actually degenerated

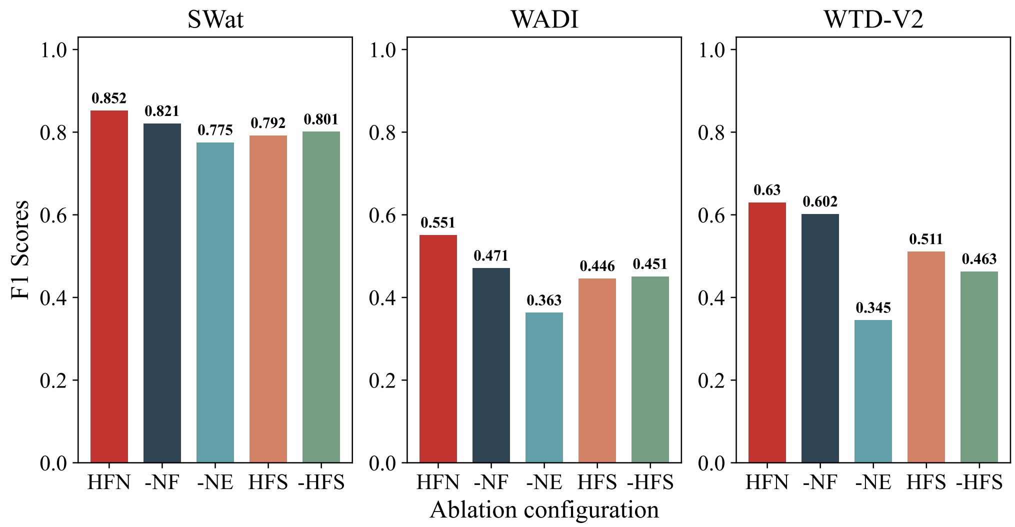

Figure 4: Effects of different HFN components on anomaly detection performance.

to the processing of isomorphic graphs, leading to the loss of heterogeneous information.

- Excluding HFS (expressed as ”-HFS”) degrades the model performance; however, it is superior to the cases when DFS and CFS are totally excluded. This indicates that the interaction between different types of sensors in the hybrid subgraph plays a complementary role in extracting the follow-up HFN heterogeneous information.

To sum up, it is necessary to extract heterogeneous structure information in the MTS datasets. The heterogeneous information can present different weights in the model according to the attention mechanism, which helps to improve the abnormal detection performance.

4.4.3. Case study

(1) Anomaly detection analysis

Figure 5 shows the abnormal detection results on SWaT testing dataset, where Figure 5(a) represents the actual data anomalies on this dataset, including network and physical attacks directed at the Secure Water Treatment (SWaT) testbed within the continuous four days. The data are labeled as 1 if the system is attacked at a certain timestamp; otherwise, it is labeled as 0. Figure 5(b) represents the results of HFN anomaly detection, where

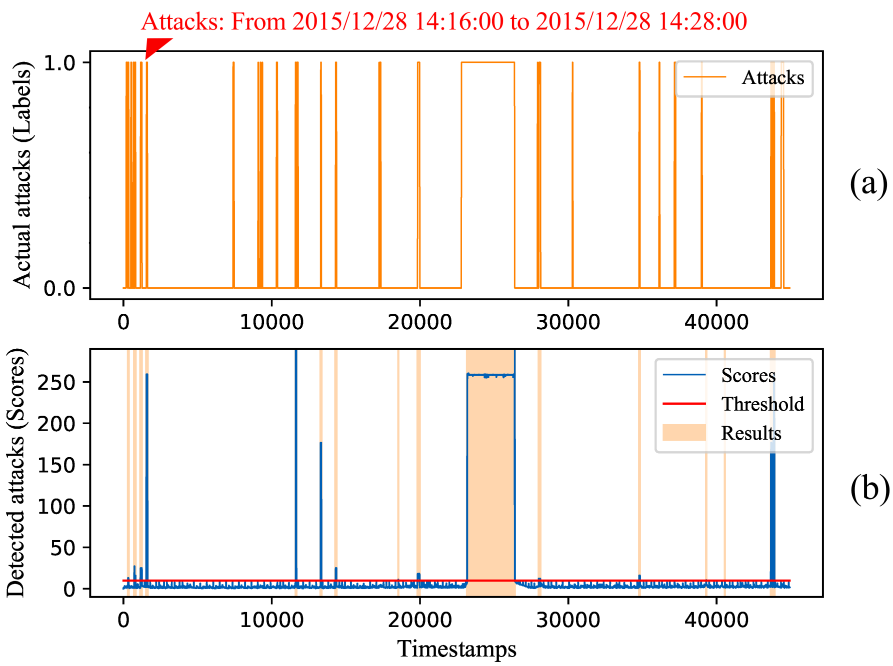

Figure 5: SWaT dataset anomaly detection results.

the orange shadow represents the detected anomalies, the blue curve represents the condition scores calculated as described in Section 3.6, and the red straight line represents the threshold when the optimal F1 is obtained on the testing dataset. It can be seen from Figure 5(b) that, aside from a few anomalies that are very difficult to distinguish possibly due to labeling errors, our approach accurately identifies the most anomalies. According to the instructions provided by SWAT dataset [62], we select an attack case to further interpret the abnormal detection capability of HFN. As shown in Figure 5(a), the attack starts from 14:16:00 28/12/2015 to 14:28:00 28/12/2015 against FIT401, UV401 and P501, where FIT401 is the flow transmitter for measuring the flow of UV de-chlorinator, UV401 is de-chlorinator for removing chlorine from water, and P501 is pump actuator for pumping the dechlorinated water to reverse osmosis. During the attack, as shown in Figure 6, the flow value (continuous value) of FIT401 is set twice to the value deviating from the normal mode. Meanwhile, the actuators UV401 and P501 (discrete value), which should be kept to an open state, are forcefully closed.

Figure 6 shows the curves of actual and predicted values of attack-related sensors and actuators and the HFN anomaly detection results. In order to reduce the influence of data dimensions and accelerate the convergence of the model, we have standardized the values of the dataset by min-max normalization. It is worth noting that we used the same normalized parameters for both the training dataset and the testing dataset, which is why the normalized data of the testing data shown in Figure 6 has negative values. This was done to reduce the impact of testing data information leakage on the model performance. In the real water treatment process, the unit of the flow sensors values are gallons per minute (GPM), while the actuators have two conditions: 0 means turn on and - 1 means turn off.

It can be seen from Figure 6(a), (c) and (d) that before the attack, the predicted values of HFN are consistent with the actual values, where the prediction for both continuous variables and discrete variables achieves good results. In the attack process, the flow variation arises from the prediction result of FIT401 and UV401 simultaneously. This is due to the interaction among these variables in the actual water treatment system. A larger deviation between the predicted value and the actual value would provide a better basis for abnormal detection. Note that although the experiment personnel did not launch the attack on FIT504 sensor in the attack process, we can see from Figure 6(b) and (d) that the value changes of FIT504 are still detected, which is due to being abnormally closed caused by the attack on P501. We can observe from the detection results in Figure 6(e) that the proposed approach shows a good detection capability of such complex anomalies. These anomalies have been resulted from attacks to different types of sensors, including continuity, discreteness and their correlation, which represent real scenarios.

(2) Anomaly localization analysis

From the above analysis, we can see that our method can successfully detect the occurrence of anomalies. However, we cannot assume that all the variables in a real complicated system are of the same significance. In other words, the variables associated with a particular system component will be impacted to varying degrees of operation when that component is attacked or behaves abnormally. Therefore, it is necessary to locate variables that have been strongly impacted by the attack, thus helping system maintenance personnel to rapidly find and solve the problems. We use the prediction error of each time-series sensor to represent the condition score of the sequence where the sensor with the maximum score times is considered

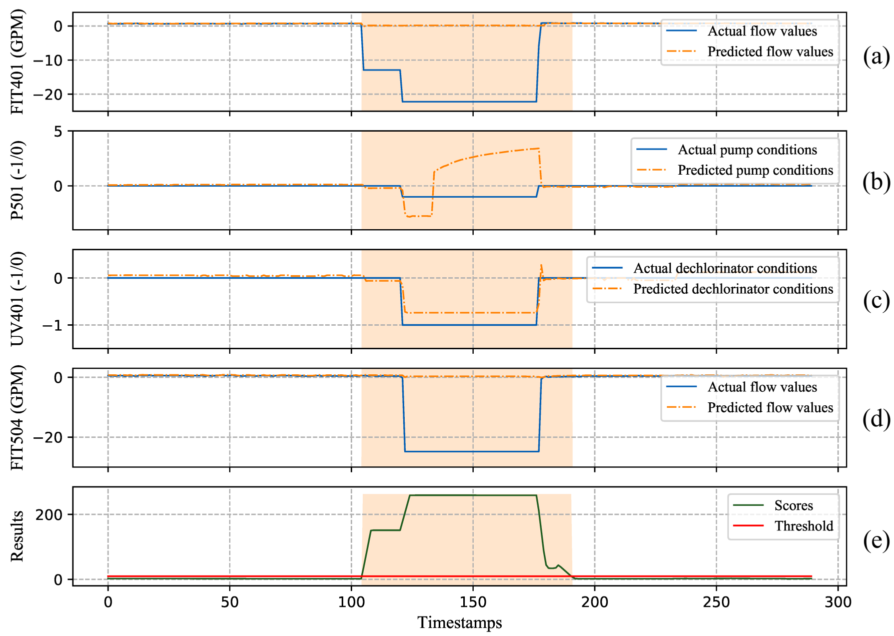

Figure 6: Abnormal detection case. The orange shadow represents the detected anomalies. The blue curve denotes the actual value of sensor or actuator. The orange dotted line represents the predicted value. The green and red curves in Fig. 6(e) represent condition score and the threshold of the optimal F1, respectively.

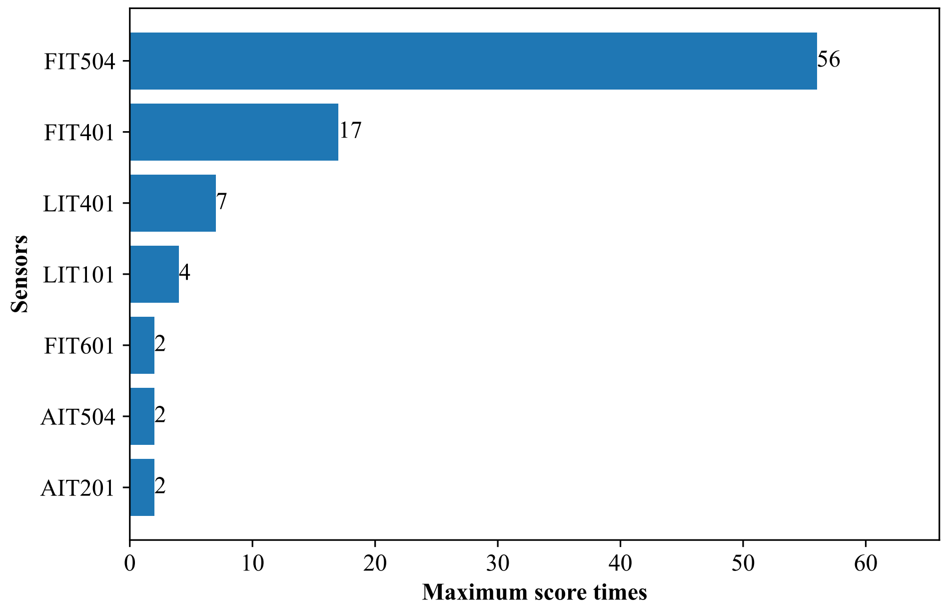

Figure 7: Maximum number of sensor scores in the abnormal dataset.

to have the possibility of the biggest anomalies. Figure 7 shows the number of times when the condition scores are above the threshold for different sensors within the attack period in the case analysis. It can be known from the figure that the sensors FIT504 and FIT401 have the maximum score times, which is consistent with the attacks where the experiment personnel made to the sensor FIT401 and pump actuator P501 during the tests. The turn-off attacks on P501 caused a sharp drop in the FIT504 flow values, as shown in Figure 6(d), since they are physically connected. On the contrary, we can also speculate which component of the system has been attacked or abnormal according to the maximum score times. In this case, during real operation and maintenance, particular attention should be paid to and checks should be made on the locations relating to FIT504, FIT401, and LIT401.

4.5. Feasibility analysis

To further illustrate how the heterogeneous relation in time series is learned and takes effect on the abnormal detection, we explain it through the similarity matrix before and after the anomalies due to the attacks. Figure 8 and Figure 9 represent different similarity matrices before and after the attack on SWaT, respectively. Its similarity value range is [-1,1], and

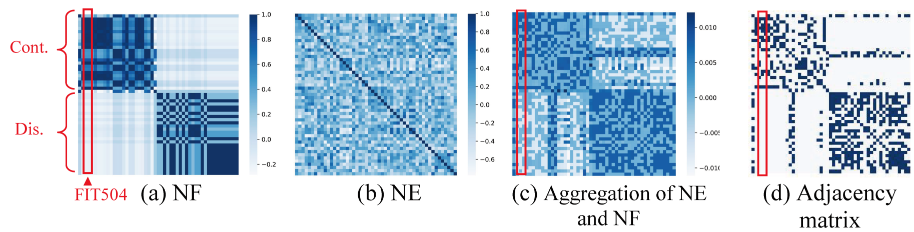

Figure 8: Example of similarity subgraphs under normal conditions.

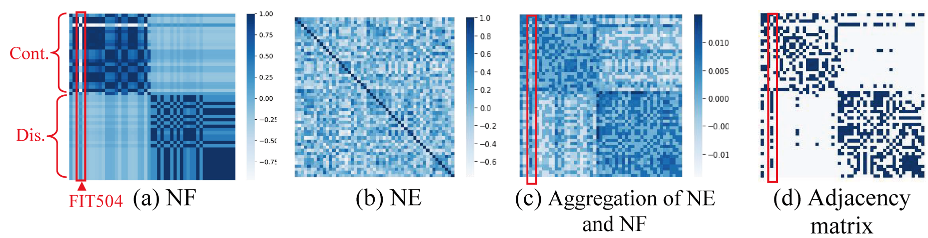

Figure 9: Example of similarity subgraphs under attack conditions.

the closer to 1, the stronger the similarity is. Overall, HFN aggregates the similarities of sensor signals from different perspectives to represent its heterogeneous information. Embedding similarity matrix learns the structural information among different sensors globally from the training data. Hence, similar features are shown in Figure 8(b) and Figure 9(b) under abnormal and normal states. However, concerning the feature similarity, we can see clearly that there exist significant differences in feature similarity between Figure 8(a) and Figure 9(a) at different timestamps, because the data vary with time. Ignoring this part of information always degrades the abnormal detection performance.

Specifically, as shown in Figure 8(a), before the attacks on FIT401, UV401 and P501, the similarity values of FIT504 flow value and other continuous variable sensor values are close to 1. However, after the attack, we can see from Figure 9(a) that the similarity value varies to -0.75. The sudden change indicates that the sensor anomalies occur, while there are slight variations in the embedding similarity. After comparing the adjacent matrix before and after the attack in Figure 8(d) and Figure 9(d), we can find that the changes in feature similarity cause the changes in the connection relation to improve the ability of the algorithm in capturing dynamic feature correlation. This further demonstrates that HFN, by aggregating the global data learning-based embedding similarity matrix and the feature similarity matrix at a specific timestamp, can better capture the normal and abnormal conditions in MTS.

5. Conclusions

In this paper, we propose a novel heterogeneous feature network for MTS anomaly detection. This approach is able to learn the complex heterogeneous structural information and temporal information between MTS data. Therefore, it is suitable for abnormal detection in real scenarios where the dataset comprises continuous numerical variables and discrete categorical variables simultaneously. The extensive experiments indicate that our approach outperforms the baseline models by assessing two open datasets from water treatment plants and a private dataset from a wind power plant. Particularly noteworthy is its significant performance improvement on the SWaT and WTD-V2 datasets, where the F1 score increased by 5% and 14%, respectively, compared to the best baseline. Furthermore, our approach demonstrates a good abnormal interpretability and can help operation and maintenance personnel rapidly discover and locate the anomalies.

In the future, we will continue to explore various avenues to enhance the proposed algorithm. We plan to extend its capabilities by incorporating more real and complex heterogeneous datasets, encompassing combined time series data and textual information. This expansion aims to boost the accuracy and practicality of the approach. While our method excels in diverse data handling, potential challenges in computational efficiency may arise with larger datasets. Future efforts will be directed towards optimizing the algorithm for improved scalability, especially in scenarios involving more extensive network scales. Additionally, we aim to investigate the impact of varying sampling intervals on our method across different datasets, thereby broadening its applicability.

Acknowledgement

The work is supported by National Natural Science Foundation of China (62006236), NUDT Research Project (ZK20-10), National Key Research and Development Program of China (2020YFA0709803), Hunan Provincial Natural Science Foundation (2020JJ5673), National Science Foundation of China

(U1811462), National Key R&D project by Ministry of Science and Technology of China (2018YFB1003203), and Autonomous Project of HPCL (201901-11, 202101-15).

References

[1] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, T. Soderstrom, Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding.

[2] H. Zhou, K. Yu, X. Zhang, G. Wu, A. Yazidi, Contrastive autoencoder for anomaly detection in multivariate time series, Information Sciences 610 (2022) 266–280.

[3] J. Zhan, R. Wang, L. Yi, Y. Wang, Z. Xie, Health assessment methods for wind turbines based on power prediction and mahalanobis distance, International Journal of Pattern Recognition and Artificial Intelligence 33 (02) (2019) 1951001.

[4] G. Pang, C. Shen, L. Cao, A. V. D. Hengel, Deep learning for anomaly detection: A review, ACM Computing Surveys (CSUR) 54 (2) (2021) 1–38.

[5] H. Mamdouh Farghaly, M. Y. Shams, T. Abd El-Hafeez, Hepatitis c virus prediction based on machine learning framework: a real-world case study in egypt, Knowledge and Information Systems 65 (6) (2023) 2595–2617.

[6] P. D. Rosero-Montalvo, Z. István, P. Tözün, W. Hernandez, Hybrid anomaly detection model on trusted iot devices, IEEE Internet of Things Journal (2023).

[7] P. Wu, J. Liu, Learning causal temporal relation and feature discrimination for anomaly detection, IEEE Transactions on Image Processing 30 (2021) 3513–3527.

[8] B. Lindemann, B. Maschler, N. Sahlab, M. Weyrich, A survey on anomaly detection for technical systems using lstm networks, Computers in Industry 131 (2021) 103498.

[9] J. Yang, L. Zhang, C. Chen, Y. Li, R. Li, G. Wang, S. Jiang, Z. Zeng, A hierarchical deep convolutional neural network and gated recurrent unit framework for structural damage detection, Information Sciences 540 (2020) 117–130.

[10] A. Blázquez-García, A. Conde, U. Mori, J. A. Lozano, A review on outlier/anomaly detection in time series data, ACM Computing Surveys (CSUR) 54 (3) (2021) 1–33.

[11] H. Zhao, Y. Wang, J. Duan, C. Huang, Q. Zhang, Multivariate time-series anomaly detection via graph attention network (2020) 841–850.

[12] S. Du, T. Li, Y. Yang, S.-J. Horng, Multivariate time series forecasting via attention-based encoder–decoder framework, Neurocomputing 388 (2020) 269–279.

[13] T.-Y. Kim, S.-B. Cho, Predicting residential energy consumption using cnn-lstm neural networks, Energy 182 (2019) 72–81.

[14] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition (2016) 770–778.

[15] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, C. Zhang, Connecting the dots: Multivariate time series forecasting with graph neural networks 753–763.

[16] Y. Guo, W. Liao, Q. Wang, L. Yu, T. Ji, P. Li, Multidimensional time series anomaly detection: A gru-based gaussian mixture variational autoencoder approach 97–112.

[17] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, S.-K. Ng, Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks (2019) 703–716.

[18] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, G. Kasneci, Deep neural networks and tabular data: A survey, arXiv preprint arXiv:2110.01889 (2021).

[19] A. Nazabal, P. M. Olmos, Z. Ghahramani, I. Valera, Handling incomplete heterogeneous data using vaes, Pattern Recognition 107 (2020) 107501.

[20] Y. Sun, J. Han, X. Yan, P. S. Yu, T. Wu, Pathsim: Meta path-based top-k similarity search in heterogeneous information networks, Proceedings of the VLDB Endowment 4 (11) (2011) 992–1003.

[21] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, P. S. Yu, Heterogeneous graph attention network 2022–2032.

[22] M. Lngkvist, L. Karlsson, A. Loutfi, A review of unsupervised feature learning and deep learning for time-series modeling, Pattern Recognition Letters 42 (2014) 11–24.

[23] J. Li, H. Izakian, W. Pedrycz, I. Jamal, Clustering-based anomaly detection in multivariate time series data, Applied Soft Computing 100 (4) (2020) 106919.

[24] J. Zhao, K. Liu, W. Wang, Y. Liu, Adaptive fuzzy clustering based anomaly data detection in energy system of steel industry, Information Sciences 259 (2014) 335–345.

[25] M. Jones, D. Nikovski, M. Imamura, T. Hirata, Anomaly detection in real-valued multidimensional time series.

[26] E. G. S. Nascimento, O. de Lira Tavares, A. F. De Souza, A cluster-based algorithm for anomaly detection in time series using mahalanobis distance 622.

[27] G. Pu, L. Wang, J. Shen, F. Dong, A hybrid unsupervised clustering-based anomaly detection method, Tsinghua Science and Technology 26 (2) (2020) 146–153.

[28] W. Jia, R. M. Shukla, S. Sengupta, Anomaly detection using supervised learning and multiple statistical methods (2019) 1291–1297.

[29] N. Görnitz, M. Kloft, K. Rieck, U. Brefeld, Toward supervised anomaly detection, Journal of Artificial Intelligence Research 46 (2013) 235–262.

[30] L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K.-R. Müller, M. Kloft, Deep semi-supervised anomaly detection, arXiv preprint arXiv:1906.02694 (2019).

[31] M. E. Villa-Pérez, M. A. Alvarez-Carmona, O. Loyola-González, M. A. Medina-Pérez, J. C. Velazco-Rossell, K.-K. R. Choo, Semi-supervised anomaly detection algorithms: A comparative summary and future research directions, Knowledge-Based Systems 218 (2021) 106878.

[32] R. Wang, X. Ma, C. Jiang, Y. Ye, Y. Zhang, Heterogeneous information network-based music recommendation system in mobile networks, Computer Communications 150 (2020) 429–437.

[33] X. Liang, Y. Ma, G. Cheng, C. Fan, Y. Yang, Z. Liu, Meta-path-based heterogeneous graph neural networks in academic network, International Journal of Machine Learning and Cybernetics 13 (6) (2022) 1553–1569.

[34] X. Deng, F. Long, B. Li, D. Cao, Y. Pan, An influence model based on heterogeneous online social network for influence maximization, IEEE Transactions on Network Science and Engineering 7 (2) (2019) 737–749.

[35] X. Chen, J. Yin, J. Qu, L. Huang, Mdhgi: matrix decomposition and heterogeneous graph inference for mirna-disease association prediction, PLoS computational biology 14 (8) (2018) e1006418.

[36] Y. Sun, J. Gao, X. Hong, B. Mishra, B. Yin, Heterogeneous tensor decomposition for clustering via manifold optimization, IEEE transactions on pattern analysis and machine intelligence 38 (3) (2015) 476–489.

[37] Y. Liu, X. Luo, X. Yang, Semantics and structure based recommendation of similar legal cases 388–395.

[38] C. Wang, C. H. Chi, Z. Wei, R. Wong, Coupled interdependent attribute analysis on mixed data (2015).

[39] S. Jian, L. Cao, G. Pang, L. Kai, G. Hang, Embedding-based representation of categorical data by hierarchical value coupling learning (2017).

[40] H. Cai, V. W. Zheng, K. C.-C. Chang, A comprehensive survey of graph embedding: Problems, techniques, and applications, IEEE Transactions on Knowledge and Data Engineering 30 (9) (2018) 1616–1637.

[41] J. Zhao, X. Wang, C. Shi, B. Hu, G. Song, Y. Ye, Heterogeneous graph structure learning for graph neural networks 35 (5) (2021) 4697–4705.

[42] X. Fu, J. Zhang, Z. Meng, I. King, Magnn: Metapath aggregated graph neural network for heterogeneous graph embedding 2331–2341.

[43] X. Wang, N. Liu, H. Han, C. Shi, Self-supervised heterogeneous graph neural network with co-contrastive learning 1726–1736.

[44] Z. Hu, Y. Dong, K. Wang, Y. Sun, Heterogeneous graph transformer 2704–2710.

[45] L. Yang, Z. Xiao, W. Jiang, Y. Wei, Y. Hu, H. Wang, Dynamic heterogeneous graph embedding using hierarchical attentions (2020) 425–432.

[46] M. Chen, C. Huang, L. Xia, W. Wei, Y. Xu, R. Luo, Heterogeneous graph contrastive learning for recommendation (2023) 544–552.

[47] Y. Zhu, Y. Xu, H. Cui, C. Yang, Q. Liu, S. Wu, Structure-enhanced heterogeneous graph contrastive learning (2022) 82–90.

[48] P. Bloomfield, Fourier analysis of time series: an introduction, John Wiley & Sons, 2004.

[49] R. Shwartz-Ziv, A. Armon, Tabular data: Deep learning is not all you need, Information Fusion 81 (2022) 84–90.

[50] Y. Zhu, W. Xu, J. Zhang, Q. Liu, S. Wu, L. Wang, Deep graph structure learning for robust representations: A survey, arXiv preprint arXiv:2103.03036 (2021).

[51] R. Li, S. Wang, F. Zhu, J. Huang, Adaptive graph convolutional neural networks 32.

[52] X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, J. Pei, Am-gcn: Adaptive multi-channel graph convolutional networks 1243–1253.

[53] Y. Chen, L. Wu, M. Zaki, Iterative deep graph learning for graph neural networks: Better and robust node embeddings, Advances in Neural Information Processing Systems 33 (2020) 19314–19326.

[54] A. Deng;, B. Hooi., Graph neural network-based anomaly detection in multivariate time series, aaai2021 (2021).

[55] D. Yu, R. Zhang, Z. Jiang, Y. Wu, Y. Yang, Graph-revised convolutional network 378–393.

[56] D. Luo, W. Cheng, W. Yu, B. Zong, J. Ni, H. Chen, X. Zhang, Learning to drop: Robust graph neural network via topological denoising 779–787.

[57] Q. Sun, J. Li, H. Peng, J. Wu, X. Fu, C. Ji, P. S. Yu, Graph structure learning with variational information bottleneck, arXiv preprint arXiv:2112.08903 (2021).

[58] X. Gao, W. Hu, Z. Guo, Exploring structure-adaptive graph learning for robust semi-supervised classification 1–6.

[59] P. Kaewprapha, P. Prempaneerach, V. Singh, T. Tinikul, N. Intarangsi, T. Kijkanjanarat, Predicting full load, partial load efficiency of a combined cycle power plant using machine learning methods 11–16.

[60] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, arXiv (2017).

[61] H. Ren, B. Xu, Y. Wang, C. Yi, C. Huang, X. Kou, T. Xing, M. Yang, J. Tong, Q. Zhang, Time-series anomaly detection service at microsoft 3009–3017.

[62] J. Goh, S. Adepu, K. N. Junejo, A. Mathur, A dataset to support research in the design of secure water treatment systems (2017) 88–99.

[63] C. M. Ahmed, V. R. Palleti, A. P. Mathur, Wadi: a water distribution testbed for research in the design of secure cyber physical systems.

[64] J. Zhan, S. Wang, X. Ma, C. Wu, C. Yang, D. Zeng, S. Wang, Stgat-mad: Spatial-temporal graph attention network for multivariate time series anomaly detection 3568–3572.

[65] D. Park, Y. Hoshi, C. C. Kemp, A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder, IEEE Robotics and Automation Letters PP (99) (2017).

[66] J. Audibert, P. Michiardi, F. Guyard, S. Marti, M. A. Zuluaga, Usad: Unsupervised anomaly detection on multivariate time series 3395–3404.

[67] J. Camacho, A. Pérez-Villegas, P. García-Teodoro, G. Maciá-Fernández, Pca-based multivariate statistical network monitoring for anomaly detection, Computers & Security 59 (2016) 118–137.

[68] T. L. Fei, M. T. Kai, Z. H. Zhou, Isolation forest.

[69] M. Qi, Lightgbm: A highly efficient gradient boosting decision tree.

[70] S. Tuli, G. Casale, N. R. Jennings, Tranad: Deep transformer networks for anomaly detection in multivariate time series data, arXiv preprint arXiv:2201.07284 (2022).

[71] S. Han, S. S. Woo, Learning sparse latent graph representations for anomaly detection in multivariate time series (2022) 2977–2986.

[72] P. Qi, D. Li, S.-K. Ng, Mad-sgcn: Multivariate anomaly detection with self-learning graph convolutional networks (2022) 1232–1244.

[73] L. Shen, Z. Li, J. Kwok, Timeseries anomaly detection using temporal hierarchical one-class network, Advances in Neural Information Processing Systems 33 (2020) 13016–13026.

[74] S. Kim, K. Choi, H. S. Choi, B. Lee, S. Yoon, Towards a rigorous evaluation of time-series anomaly detection (2021).