Learning Hierarchical Spatial-Temporal Graph Representations for Robust Multivariate Industrial Anomaly Detection

Jingyu Yang, and Zuogong Yue

***Abstract*— Multivariate time series anomaly detection is one of the most indispensable yet troublesome links in complex industrial processes. The main challenge lies in discovering the representative patterns for collective or contextual anomalies among interconnected sensory data streams, which has been largely hampered by inefficient spatial-temporal feature extraction and suboptimal decision criteria under the scarcity of positive training samples. This article goes beyond the common limitations of the existing methods, and novelly proposes Hierarchical Spatial-Temporal grAph Representation (HiSTAR). It processes the data with strong structural inductive biases through latent spatial-temporal graph modeling, yet requiring no predefined topological priors for the sensor network. A discriminative decision boundary is constructed by learning hierarchical normality-enclosing hyperspheres on the produced graph-structure representations. In this way, HiSTAR not only presents superior anomaly detection performance, but also provides consistent anomaly localization results. The efficacy of the proposed method is experimentally corroborated through three industrial case studies.**

***Index Terms*— Anomaly Detection, Anomaly Localization, Multivariate Time Series Data, Spatial-Temporal Graph Modeling**

I. INTRODUCTION

ANOMALY detection aims to identify the observations or events that deviate from the expected behaviors of the majority [1]. It plays a fundamental rule in most industrial applications, so as to provide critical health status information and support preventive maintenance for the running systems. Nowadays, with the rapid development of AI technologies in industrial fields, it has been an irresistible trend to discover system anomalies in a data-driven manner. Although plenty of algorithms have been performed on time series anomaly detection, most of them are designed for the univariate case. The ever-growing sensing and computing capabilities have revealed their weakness in scenarios where large-scale sensory data can be collected from different locations of the integrated monitoring systems. It is necessary to develop robust anomaly detectors for multivariate time series data.

This work was supported by the National Natural Science Foundation of China under Grant 92167201. (Corresponding author: Zuogong Yue.)

J. Yang and Z. Yue are with Key Laboratory of Image Processing and Intelligent Control, Ministry of Education and School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan, China (email: {yangjingyu, z_yue}@hust.edu.cn).

The collected monitoring datasets commonly suffer from severe distribution skewness, since mechanical systems are under regular states in most time. It means that the negative (normal) samples far outweigh the positive (anomalous) ones. In fact, most of the anomaly detection approaches fall into the paradigm of unsupervised learning. The challenging central issues are to accurately discover the common patterns underneath normal samples with limited prior knowledge of true anomalies, and to robustly detect the deviation against highly varied and unknown abnormal distributions. The common practice for low-dimensional data is to use time series analytics in combination with shallow machine learning approaches. Wang et al. [2] achieved nonlinear dimensionality reduction via t-distributed stochastic neighbor embedding, and enclosed the attained features within a hypersphere through support vector data description (SVDD). Deng et al. [3] combined tensor Tucker factorization and one class support vector machine (OC-SVM). Ghalyan et al. [4] transformed the time series into a string of symbols and determined abnormality through the state-transition probability vectors. Considering the ever-growing data heterogeneity and nonlinearity in modern industrial datasets, deep learning approaches have been extensively explored. Most of the advanced methods are based on generative models, including variational autoencoder (VAE), generative adversarial networks (GAN) and autoregressive models (ARM). Li et al. [5] utilized VAE for data reconstruction and chose the reconstruction error as the decision criterion. It was assumed that important regularities of normal instances could be learned in a compressed space, while the anomalous ones failed to perform in the same manner. Wu et al. [6] learned a latent feature space based on the combination of VAE and GAN, assuming that the normal data was endowed with higher likelihood and could be better generated from the latent space. Deng [7] aimed at autoregressively forecasting the current data points using the historical trajectories, assuming that normal instances had larger conditional probability and were temporally more predictable than anomalous ones. The common defect of these approaches is that the decision criteria could be suboptimal since the objective functions are designed for certain tasks other than anomaly detection. For example, the objective of VAE may lead to a generic summarization of underlying regularities, which are not optimized for detecting irregularities. Similarly, the underlying objectives of GAN and ARM aim at data synthesis and sequence prediction respectively, rather than anomaly detection. Recent studies [8],

[9] also revealed that generative models could be easily confounded by the background statistics and assigned spuriously high likelihood to anomalies (even higher than normal ones), thus leading to a high rate of false detection. Some recent researches fall into the paradigm of semi-supervised anomaly detection for the circumstance when one may have access to a limited set of labeled anomalous instances. The key issue lies in making the most advantage of the partial labeled information to further enhance the detection robustness. Pu et al. [10] proposed an one-class generative adversarial framework with effective latent knowledge. Xie et al. [11] employed the clustering method to infer the unknown label information in the latent space of generative models. Wang et al. [12] conducted label propagation on a hypergraph structure to calculate the confidence score for a given instance.

Multivariate scenarios present a much more challenging issue for industrial anomaly detection, where the anomalies tend to be contextual and collective. It is an essential prerequisite to accurately extract spatial-temporal features from multivariate time series data, due to the dependencies and interactions among disparate subsystems. Liu et al. [13] proposed a input tensor transformation scheme based on multivariate convolutional neural networks (CNN). Wang et al. [14] constructed variational phase spaces and achieved better separability for different sensor variables. He et al. [15] developed a multi-scale neural network structure composed of two parallel feature extraction modules that were respectively for the temporal and spatial dimension. Yu et al. [16] combined CNN and long short-term memory (LSTM). Although these methods could facilitate the simultaneous discovery for time-scale characteristics and spatial-scale correlations among multiple sensor variables, they only concatenated diverse sources of time series data directly without organization. Recently, some researches have explored much more effective solutions for multivariate time series modeling based on graph neural networks (GNN). They were endowed with structural inductive biases and aimed to take advantage of the topological information to attain organized graph-based representations for the system states. This might benefit efficient spatial-temporal modeling under the insufficiency of positive label information. Khodayar et al. [17] developed a convolutional graph autoencoder based on spectral graph convolution and variational inference. He et al. [18] proposed a topology-aware sequence-to-sequence multivariate anomaly detector by integrating GNN and LSTM into the VAE structure. Deng et al. [19] devised a graph convolutional GAN composed of a spatial-temporal generator and spatial-temporal discriminator. However, what hinders the application of these GNN-based methods lies in the necessity of pre-defined topological priors to construct adjacency matrices, which are not available in most real scenarios. Another important aspect of multivariate time series anomaly detection lies in consistent anomaly localization, which aims to identify the anomalous sensor variables and is important to help experts understand the anomaly detection results. However, most of the existing advanced methods focus mainly on improving anomaly detection performance rather than localizing the anomalies.

Considering the non-negligible deficiencies of the existing methods for robust multivariate anomaly detection in industrial applications, this article novelly proposes Hierarchical Spatial-Temporal grAph Representation (HiSTAR). The contributions of HiSTAR are summarized as follows.

- HiSTAR performs efficient spatial- and temporal-scale feature extraction through a organic integration of deep graph convolution and sequence modeling, as an indispensable premise for the downstream anomaly detection task. It is applicable in most of the multivariate scenarios, since it automatically discovers the latent spatial correlations from data rather than utilizing human-defined topological information.
- HiSTAR goes beyond generative modeling and adopts a discriminative scheme for robust representation learning. The model is trained on a sophisticated anomaly detection based objective, which is designed to learn hierarchical normality-enclosing hyperspheres on the produced spatial-temporal graph representations. In this way, HiSTAR is capable of learning a sufficient data description for normality and largely improving the anomaly detection performance.
- HiSTAR is capable of localizing the anomalous sensor variables, which provides interpretability for the anomaly detection results.
- HiSTAR is applicable in both of the unsupervised and semi-supervised scenarios. It could take good advantage of the precious labeled anomalous training samples to remarkably enhance the detection robustness.

We experimentally corroborate the effectiveness of HiSTAR in three case studies including chemical industrial process monitoring, water treatment security and network intrusion detection. The remainder of this article proceeds as follows: Section II formulates the details of the proposed method. Section III reports the important experiment results and the related discussion. We conclude this article in Section IV.

II. Methodology

In this section, we will first provide some basic notations and a rough outline for the proposed method. The details will be elaborated in the following.

A. Method Overview

Suppose that we have multivariate time series data collected from N spatially-correlated sensors in the monitoring system. The sensor network is represented as a weighted undirected graph $\mathcal{G} \triangleq (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V}$ is the set of unordered vertices/nodes with $|\mathcal{V}| = N$, $\mathcal{E}$ is the set of edges, and $\mathbf{A} \in \mathbb{R}^{N\times N}$ is the weighted adjacency matrix. $\mathbf{A}_{i,j} > 0$ if $v_i, v_j \in \mathcal{V}$ and $(v_i, v_j) \in \mathcal{E}$, which represents the spatial correlation between the i(th) and j(th) sensor variable. Let $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_N$ be the self-looped adjacency matrix, where $\mathbf{I}_N$ denotes N-dimensional identity matrix. Let $\mathbf{D} \in \mathbb{R}^{N\times N}$ be the diagonal degree matrix, with $\mathbf{D}_{i,i} = \sum_j \mathbf{A}_{i,j}$. Since each vertex is associated with a time series data stream, the collected data can be represented by a Γ-timestep N-variable signal $\mathbf{X} \in \mathbb{R}^{\Gamma\times N}$, where $\mathbf{X}_t \in \mathbb{R}^N$ denotes the multivariate signal at the t(th) timestep. We construct the training dataset $\mathcal{D} = \{\boldsymbol{x}_t\}_{t=T+1}^{\Gamma}$ by applying a T-length sliding window on $\mathbf{X}$ in the time dimension, where $\boldsymbol{x}_t = \mathbf{X}_{t-T:t}$ is a sequence

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3216006

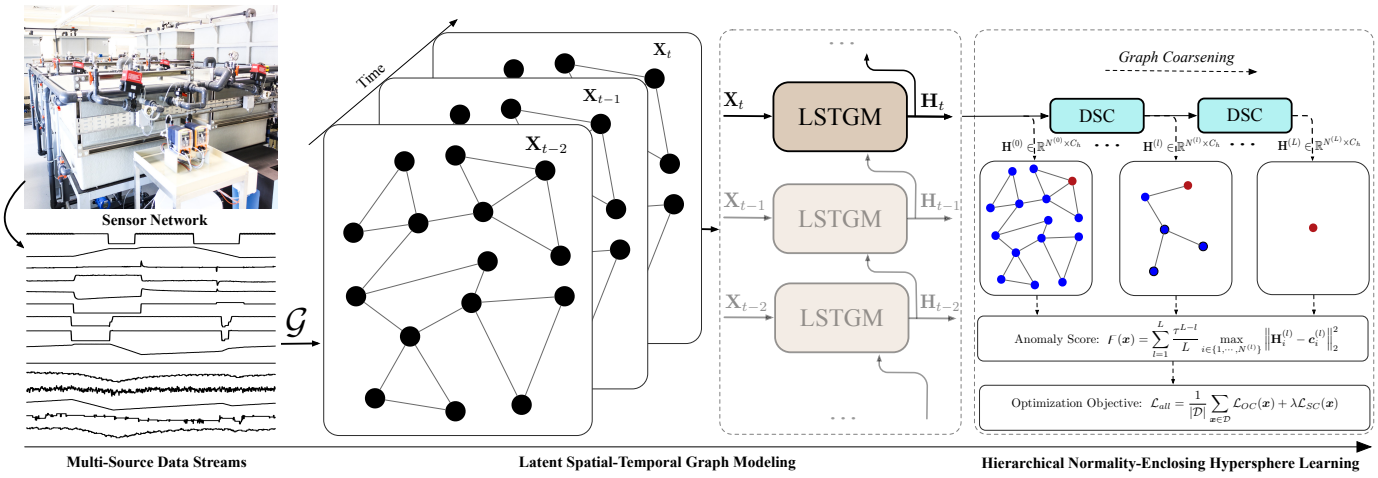


Fig. 1. The outline of the proposed HiSTAR. It performs efficient spatial-temporal feature extraction through latent spatial-temporal graph modeling (LSTGM). Hierarchical graph-structure representations are produced through differentiable spectral clustering (DSC) layers. The model is trained by learning normality-enclosing hyperspheres for the attained hierarchical graph-structure representations.

of observations from the $t-T$(th) timestep to the t(th) timestep. $\mathcal{D}$ can be partitioned into disjoint subsets $\mathcal{D}_n$ and $\mathcal{D}_a$, which include all the negative (normal) instances and positive (anomalous) ones, respectively. In the unsupervised scenario, we assume that most of the training data are normal, i.e., $\mathcal{D} = \mathcal{D}_n$. In the semi-supervised scenario, suppose that we have a very limited set of labeled anomalous instances, i.e., $\mathcal{D} = \mathcal{D}_n \cup \mathcal{D}_a$ and $|\mathcal{D}_n| \gg |\mathcal{D}_a|$.

For each instance $\boldsymbol{x}_t$, the goal is to determine whether an observation $\mathbf{X}_t$ is an anomaly or not, given $T-1$ historical observations $\mathbf{X}_{t-T:t-1}$. Meanwhile, the consistent anomaly localization results should be provided. Supervised learning is ill-suited due to severe data distribution skewness and extreme ambiguity of the potential anomalies. Instead, we consider it as an one-class classification problem, which aims to learn a typical description of the normal data and detect whether new instances conform to it. Our method is developed in the unsupervised paradigm and can be adapted in the semi-supervised scenario, where the precious labeled anomalous instances can be well utilized. It is divided into 3 steps, as outlined in Fig. 1. First, we propose a latent spatial-temporal graph modeling (LSTGM) module, which automatically discovers the implicit graph structure and performs efficient feature extraction from multiple sensory data streams. This process is denoted as $\Phi^{(0)}(\cdot)$ and parameterized by $\Theta_\Phi^{(0)}$. Then, we produce hierarchical graph-structure representations through differentiable spectral clustering (DSC). We denote this process as $\Phi^{(1:L)}$, which is composed of L cascaded DSC layers, i.e., $\Phi^{(1:L)} \triangleq \Phi^{(L)} \circ \cdots \circ \Phi^{(1)}$, and each intermediate layer $\Phi^{(l)}$ is parameterized by $\Theta_\Phi^{(l)}$. The hierarchical graph-structure representations include $\Phi^{(0:l)}(\cdot)$ with l ranging from 0 to L. By contrast, $\Phi^{(0)}$ aims to complement the dynamics of each sensor variable by aggregating information from the correlated ones, while $\Phi^{(0:L)}$ aims to output a unified description for all data streams. Finally, we calculate the anomaly scores and conduct backward optimization for all modules. Other than the commonly adopted deep generative modeling (data compression, data synthesis and sequence prediction), the proposed optimization objective is directly designed for anomaly detection, inspired by [20]. This objective aims to reduce the variation among normal instances in the high-dimensional feature space (i.e., construct normality-enclosing hyperspheres) and increase the discrepancy between normal instances and anomalous ones (i.e., push the anomalous instances away from the hypersphere). Specifically, given a data sample $\boldsymbol{x} \in \mathcal{D}$, the anomaly score measures the distance between each intermediate representation and the corresponding hypersphere center,

$$F(\boldsymbol{x}; \Phi^{(0:L)}) = \sum_{l=0}^{L} \omega^{(l)} \operatorname{dist}\left(\Phi^{(0:l)}(\boldsymbol{x}), \boldsymbol{c}^{(l)}\right), \tag{1}$$

where $\boldsymbol{c}^{(l)} = \mathbb{E}_{\boldsymbol{x} \in \mathcal{D}_n}\left[\Phi^{(0:l)}(\boldsymbol{x})\right]$, and $\{\omega^{(l)}\}$ are weight parameters that discriminate the importances of each intermediate representation. It considers not only the individual contribution of each sensor variable but also their group effect. The optimization objective is formulated as

$$\begin{aligned} \min_{\{\Theta_\Phi^{(l)}\}_{l=0}^{L}} \; & \mathbb{E}_{\boldsymbol{x} \in \mathcal{D}_n}\left[\ell\left(F(\boldsymbol{x}; \Phi^{(0:L)})\right)\right] \\ & - \mathbb{E}_{\boldsymbol{x} \in \mathcal{D}_a}\left[\ell\left(F(\boldsymbol{x}; \Phi^{(0:L)})\right)\right], \end{aligned} \tag{2}$$

where $\ell(\cdot)$ is the pre-programmed loss function. In the testing stage, given a new instance $\boldsymbol{x}_*$, we first calculate the anomaly score based on Eq. (1) and decide the detection result by comparing with a selected threshold ϵ. $F(\boldsymbol{x}_*; \Phi^{(0:L)}) > \epsilon$ represents an abnormal state, otherwise normal state. The anomaly location is attained by focusing on $\Phi^{(0)}(\boldsymbol{x}_*)$ and finding out the sensor variable that contributes the most to $\operatorname{dist}\left(\Phi^{(0)}(\boldsymbol{x}_*), \boldsymbol{c}^{(0)}\right)$. We will specify the details of $\Phi^{(0)}$ and $\Phi^{(1:L)}$ in the following.

B. Latent Spatial-Temporal Graph Modeling

Efficient spatial-temporal feature extraction is an indispensable premise for accurate anomaly detection with multiple spatially-correlated time series. To tackle this issue, we propose latent spatial-temporal graph modeling (LSTGM), which is an organic integration of spatial graph convolution and

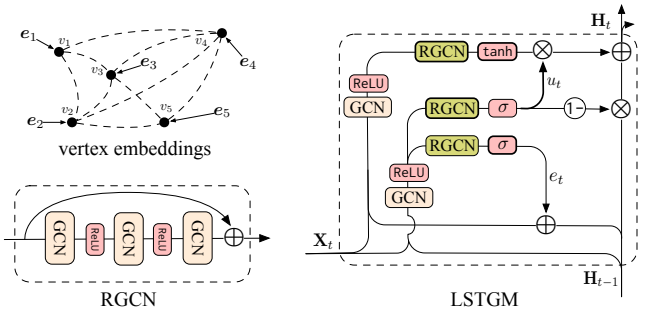


Fig. 2. LSTGM is based on the residual GCN (RGCN) and GRU. The adjacency matrix in graph convolution is attained by learning unique vector embeddings for each vertex and calculating the pairwise similarities.

sequence modeling, yet requiring no prior topological knowledge. Specifically, the spatial-scale features among multiple sensory data variables are extracted through spatial graph convolution. The central issue of convolution operation on graph-structure data lies in aggregating vertex information from the neighborhoods while maintaining the weight-sharing property in a similar way as CNN for grid-like data. We utilize the adjacency matrix to decide the importances of neighbor vertices in spatial graph convolution. Define the neighbor set of v_i as $\mathcal{N}(v_i) = \{v_j \mid \mathbf{A}_{i,j} > 0\}$. As an initial step, the vertex features take a shared linear transformation which is parameterized by $\mathbf{W} \in \mathbb{R}^{C_{in} \times C_{out}}$, with C_{in} and C_{out} denoting the number of input channels and output channels, respectively. Given the input signal $\mathbf{X}_t$, the aggregated feature at v_i is computed as

$$\text{Aggregate}(\mathbf{X}_t, v_i) = \sum_{v_j \in \mathcal{N}(v_i) \cup \{v_i\}} \frac{1}{\varkappa(v_i, v_j)} \tilde{\mathbf{A}}_{i,j} \mathbf{W}^\top \mathbf{X}_{t,j}, \tag{3}$$

where the normalizing item $\varkappa(v_i, v_j)$ is equal to the cardinality of the subset that contains v_i and v_j. It is utilized to balance the contributions of different subsets to the output values. We implement it with the following formula in standard graph convolutional networks (GCN) [21],

$$\text{GCN}(\mathbf{X}_t) = \tilde{\mathbf{\Lambda}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{\Lambda}}^{-\frac{1}{2}} \mathbf{X}_t \mathbf{W}, \tag{4}$$

where $\varkappa(v_i, v_j) = (\tilde{\mathbf{\Lambda}}_{ii} \tilde{\mathbf{\Lambda}}_{jj})^{\frac{1}{2}}$. Furthermore, we aim at a deep GNN structure so that the model is endowed with a large receptive field in a similar style with deep CNN. However, stacking many GNN layers may lead to indistinguishable converged vertex features and vanished updating gradients in back propagation, known as the over-smoothing issue. In fact, most of the advanced GNN models are no deeper than 4 layers. To tackle this issue, the key concept of residual learning is adopted to facilitate a deep GNN structure [22]. A residual GCN block (RGCN) is constructed by performing skipping connections on three cascaded GCN layers with ReLU activation functions, as presented in Fig. 2.

An informative adjacency matrix is essential for graph convolution but commonly unavailable in most real multivariate scenarios, due to the lack of pre-defined topological priors. Instead, LSTGM discovers the latent spatial correlations among multiple sensor variables by learning vertex embeddings, which capture the unique spatial characteristics of each sensor variable. The adjacency matrix is then attained by calculating the pairwise embedding similarities. Formally, we denote the embedding matrix as $\mathbf{E} = (\boldsymbol{e}_1, \cdots, \boldsymbol{e}_N)^\top$, where $\boldsymbol{e}_i$ is the embedding vector for the vertex v_i, as illustrated in Fig. 2. The learning process of embedding matrix is enforced based on the assumption that sensor variables with similar embedding values should be more likely to be spatially correlated. Then the non-diagonal elements of the adjacency matrix are represented by,

$$\mathbf{A}_{i,j} = \begin{cases} \frac{\boldsymbol{e}_i^\top \boldsymbol{e}_j}{\|\boldsymbol{e}_i\| \cdot \|\boldsymbol{e}_j\|}, & \mathbf{A}_{i,j} \geq \delta \\ 0, & \mathbf{A}_{i,j} < \delta \end{cases}, \tag{5}$$

where $\|\cdot\|$ denotes the ℓ_2 norm. The cosine similarity is adopted, and a predefined positive threshold δ is utilized for sparsity by removing weak or negative connections (e.g., we set $\delta = 0.02$). We note that the vertex similarities do not necessarily represent the physical proximity, but rather indicates quantifiable functional dependencies among sensor variables, e.g, the Granger causality. The structural information can then be interpreted as a set of constraints or instructions for data fusion among multiple time series, thus facilitating discovering the most decisive patterns underneath the data.

To jointly model the spatial-temporal dependency, we replace the matrix multiplications in Gated Recurrent Units (GRU) with RGCN. Then, the details of the spatial-temporal feature extraction process, i.e., $\mathbf{H}_t = \Phi^{(0)}(\boldsymbol{x}_t)$, are formulated as follows,

$$\begin{aligned} e_t &= (\sigma \circ \text{RGCN} \circ \text{ReLU} \circ \text{GCN})\,[\mathbf{X}_t, \mathbf{H}_{t-1}], \\ u_t &= (\sigma \circ \text{RGCN} \circ \text{ReLU} \circ \text{GCN})\,[\mathbf{X}_t, \mathbf{H}_{t-1}], \\ \overline{\mathbf{H}}_t &= (\tanh \circ \text{RGCN})\,[\mathbf{X}_t, e_t \odot \mathbf{H}_{t-1}], \\ \mathbf{H}_t &= (1 - u_t) \otimes \mathbf{H}_{t-1} \oplus u_t \otimes \overline{\mathbf{H}}_t. \end{aligned} \tag{6}$$

$\mathbf{H}_t \in \mathbb{R}^{N \times C_h}$ is the hidden state of LSTGM, with C_h denoting the hidden state size. $\circ$, $\otimes$ and $\oplus$ denote function composition, element-wise multiplication and element-wise addition, respectively. $\sigma(\cdot)$, $\text{ReLU}(\cdot)$ and $\tanh(\cdot)$ denote the sigmoid, ReLU and tanh activation function, respectively. The overall structure of LSTGM is presented in Fig. 2. The additional GCN layers at the first place are used to transform the vertex signals into the hidden state size. For notation ease, the subscript t of all the related variables are omitted in the following unless clearly required.

C. Hierarchical Normality-Enclosing Hyperspheres

We produce hierarchical representations through differentiable spectral clustering (DSC) layers, which could identify communities of vertex features and coarsen the graph layer by layer. It utilizes deep neural networks to perform the spectral clustering task on the target graph-structure feature map. It is based on that the vertices with large connectivity tend to have similar features and vice versa. Therefore, the vertex representations can be partitioned into disjoint clusters, each of which represents a different facet of the original graph-structure feature map. The overall structure of DSC is illustrated in Fig. 3. It is mainly separated into two branches:

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3216006

one is to process the original graph-structure feature map with RGCN, the other is to generate the cluster assignment matrix with a multilayer perceptron (MLP) and row-wise softmax activation function. We explain the details as follows.

Suppose that there are L cascaded DSC layers in the model, denoted by $\Phi^{0:L}(\cdot)$ as aforementioned. Denote the number of target clusters in the l-th DSC layer as $N^{(l)}$, with $N^{(l-1)} > N^{(l)}$. Denote the output of the l-th DSC layer as $\mathbf{H}^{(l)} \in \mathbb{R}^{N^{(l)} \times C_h}$. Let $\mathbf{A}^{(l)} \in \mathbb{R}^{N^{(l)} \times N^{(l)}}$ and $\mathbf{D}^{(l)} \in \mathbb{R}^{N^{(l)} \times N^{(l)}}$ be the output adjacency matrix and degree matrix, respectively. Particularly, we have $N^{(0)} = N$, $\mathbf{H}^{(0)} = \mathbf{H}$, $\mathbf{A}^{(0)} = \mathbf{A}$ and $\mathbf{D}^{(0)} = \mathbf{D}$. Define the cluster assignment matrix as $\mathbf{S}^{(l)} \in \{0,1\}^{N^{(l-1)} \times N^{(l)}}$, where $\mathbf{S}^{(l)}_{i,k} = 1$ if the i(th) vertex is assigned to the k(th) cluster, and 0 otherwise. Therefore the coarsened adjacency matrix and the output feature map are computed as,

$$\mathbf{H}^{(l)} = (\mathbf{S}^{(l)})^\top \mathrm{RGCN}\left(\mathbf{H}^{(l-1)}\right), \mathbf{A}^{(l)} = (\mathbf{S}^{(l)})^\top \mathbf{A}^{(l-1)} \mathbf{S}^{(l)}. \tag{7}$$

The central issue is how to attain $\mathbf{S}^{(l)}$. Considering the standard spectral clustering problem, $\mathbf{S}^{(l)}$ is attained by solving a $N^{(l)}$-way normalized minimum cut problem [23],

$$\begin{aligned} \max_{\mathbf{S}^{(l)}} \quad & \mathrm{tr}\left(\frac{(\mathbf{S}^{(l)})^\top \mathbf{A}^{(l-1)} \mathbf{S}^{(l)}}{(\mathbf{S}^{(l)})^\top \mathbf{D}^{(l-1)} \mathbf{S}^{(l)}}\right), \\ \text{s.t.} \quad & \mathbf{S}^{(l)} \in \{0,1\}^{N^{(l-1)} \times N^{(l)}}, \mathbf{S}^{(l)} \mathbb{1}_{N^{(l)}} = \mathbb{1}_{N^{(l-1)}} \end{aligned} \tag{8}$$

where $\mathbb{1}_{N^{(l)}}$ denotes a $N^{(l)}$-dimension vector of ones. Since Eq. (8) is NP-hard, we implement spectral clustering in a deep learning scheme through DSC layers. It computes a soft cluster assignment matrix as

$$\mathbf{S}^{(l)} = \mathrm{softmax} \circ \mathrm{MLP}\left(\mathbf{H}^{(l-1)}\right). \tag{9}$$

The softmax activation function is applied in a row-wise way to guarantee $\mathbf{S}^{(l)} \mathbb{1}_{N^{(l)}} = \mathbb{1}_{N^{(l-1)}}$. A temperature parameter can be involved in softmax to improve the sharpness of the output assignment probability distribution. An unsupervised loss is utilized to train DSC layers,

$$\begin{aligned} \mathcal{L}_{SC}(\boldsymbol{x}) = \sum_{l=1}^{L} & - \mathrm{tr}\left(\frac{(\mathbf{S}^{(l)})^\top \mathbf{A}^{(l-1)} \mathbf{S}^{(l)}}{(\mathbf{S}^{(l)})^\top \mathbf{D}^{(l-1)} \mathbf{S}^{(l)}}\right) \\ & + \left\| \frac{(\mathbf{S}^{(l)})^\top \mathbf{S}^{(l)}}{\|(\mathbf{S}^{(l)})^\top \mathbf{S}^{(l)}\|_{\mathbf{F}}} - \frac{\mathbf{I}_{N^{(l)}}}{\sqrt{N^{(l)}}} \right\|_{\mathbf{F}}^2, \end{aligned} \tag{10}$$

where $\|\cdot\|_{\mathbf{F}}$ denotes Frobenius norm. The regularization item is added to prevent trivial solutions that all nodes are equally assigned to the clusters, referring to [24].

With L DSC layers for graph coarsening, we take into account the intermediate graph-structure feature maps and propose a sophisticated anomaly score function, which measures the Euclidean distances between the attained graph-structure feature maps and the hierarchical normality-enclosing hypersphere centers,

$$\mathcal{F}(\boldsymbol{x}; \Phi^{(0:L)}) = \sum_{l=1}^{L} \frac{\tau^{L-l}}{L} \max_{i \in \{1, \cdots, N^{(l)}\}} \left\| \mathbf{H}_i^{(l)} - \boldsymbol{c}_i^{(l)} \right\|_2^2, \tag{11}$$

where $\mathbf{H}^{(l)} = \Phi^{(0:l)}(\boldsymbol{x})$ and $\boldsymbol{c}^{(l)} = \mathbb{E}_{\boldsymbol{x} \in \mathcal{D}_n}\left[\mathbf{H}^{(l)}\right]$ (refer to Eq. (1)). $\mathbf{H}_i^{(l)}$ is the i(th) entry of $\mathbf{H}^{(l)}$ and represents the i(th) unique facet of the feature map. The max operation is utilized to enforce the model to focus on the most prominent facet of each intermediate graph-structure feature map. This decision criterion considers not only the group effect of the sensory data streams, but also the individual contributions to the decision criteria. Therefore, it can help to localize the anomalous sensor variables while performing high-performance anomaly detection. A hyperparameter τ is introduced to distinguish the importances of DSC layers in different depth to the final scores. We organize the loss function of our one-class classifier by referring to the binary scenario. The decision region in standard binary deep classifiers is constructed with sigmoid activation function and cross-entropy loss. In our case, we substitute the sigmoid function with a radial basis function in order to construct a spherical decision boundary, known as hypersphere classification [25]. It encourages centering the mapped normal data in a hypersphere and taking good advantage of anomalous training samples. The loss function for a given instance $\boldsymbol{x}$ is formulated as

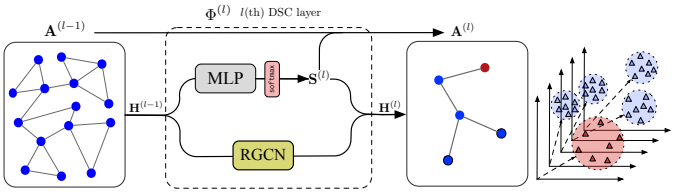


Fig. 3. A differentiable spectral clustering (DSC) layer aims to coarsen the input feature map into a smaller scale. Each entry of the coarsened feature map represents a unique facet of the original one. The produced feature maps from normal instances are enclsoed into multiple hyperspheres.

$$\mathcal{L}_{OC}(\boldsymbol{x}) = \begin{cases} \mathcal{F}(\boldsymbol{x}), & \boldsymbol{x} \in \mathcal{D}_n \\ -\log(1 - \exp(-\mathcal{F}(\boldsymbol{x}))), & \text{otherwise} \end{cases}. \tag{12}$$

The compositive loss function that should be minimized is written as

$$\mathcal{L}_{all} = \frac{1}{|\mathcal{D}|} \sum_{\boldsymbol{x} \in \mathcal{D}} \mathcal{L}_{OC}(\boldsymbol{x}) + \lambda \mathcal{L}_{SC}(\boldsymbol{x}), \tag{13}$$

where λ is the weight hyperparameter for $\mathcal{L}_{SC}$.

D. Comprehensive Algorithm of HiSTAR

Two significant technical points should be figured out for practical implementation of HiSTAR, including the pretraining procedure for LSTGM and the prevention of hypersphere collapse issue. It is worth recalling that HiSTAR is endowed with strong structural inductive biases to constrain the information fusion among multiple sensory data streams, thus facilitating discovering the most decisive patterns. The spatial correlations among sensor variables are represented by learning vertex embeddings $\mathbf{E}$ and calculating pairwise similarities. To advance vertex embedding learning, the LSTGM module is first pretrained to perform autoregressive prediction task, as illustrated in Fig. 4. More specifically, given the historical data, an additional instrumental MLP is constructed to forward the concatenation of the hidden state and embedding matrix of LSTGM and predict the data at the next step, i.e.,

$$\mathbf{H}_t = \mathrm{LSTGM}(\mathbf{X}_{t-T:t}), \quad \hat{\mathbf{X}}_t = \mathrm{MLP}([\mathbf{H}_t, \mathbf{E}]).$$

Algorithm 1 Fwd (Forward process of HiSTAR)

Input: Data sample $\boldsymbol{x}$, LSTGM and L DSC layers;
Output: $\{\mathbf{H}^{(l)}\}$,$\{\mathbf{S}^{(l)}\}$,$\{\mathbf{A}^{(l)}\}$
$\mathbf{H}, \mathbf{A} \leftarrow \mathrm{LSTGM}(\boldsymbol{x})$
$\mathbf{H}^{(0)} \leftarrow \mathbf{H}$, $\mathbf{A}^{(0)} \leftarrow \mathbf{A}$;
for $l \in \{1, \cdots, L\}$ **do**
  $\mathbf{H}^{(l)}, \mathbf{S}^{(l)}, \mathbf{A}^{(l)} \leftarrow \mathrm{DSC}\left(\mathbf{H}^{(l-1)}, \mathbf{A}^{(l-1)}\right)$;
end for

Then the LSTGM is pretrained by minimizing the mean squared error between predictions and ground truths,

$$\min_{\Theta_{\Phi}^{(0)}} \quad \mathbf{E}_{\boldsymbol{x}\in\mathcal{D}} \left\|\mathbf{X}_{t+1} - \hat{\mathbf{X}}_{t+1}\right\|_2^2. \tag{14}$$

On the other hand, we need to prevent the issue of hypersphere collapse. Looking back into Eq. (2), the contrastive component in this objective takes no effect in the unsupervised scenario (i.e., $\mathcal{D}_a = \varnothing$). It is ill-posed since there exists an optimal where the network parameters are all zero, and all the data samples degenerate into one point in the high-dimensional space. [20] mitigated this issue by using none-zero biases and fixing the hypersphere centers as predefined values, which might largely limit the model expressiveness. Instead, we get inspiration from [26] and inject data sample noises to guarantee the training robustness. The core idea is to generate synthetic anomalies from normal instances based on the manifold assumption that the typical normal data lie on a low-dimensional manifold and can be well sampled with a large-scale training dataset. Therefore, it is straightforward to assume that one point can be regarded as an anomaly once it is off the manifold of typical training points. We constrain the normal data distribution within the union of tiny Euclidean balls around the training data. Synthetic anomalous samples are generated by imposing tiny perturbations to the normal ones so that they are outside the union balls. Fig. 5(a) provides a simple illustration for the synthetic anomalies. Specifically, given a normal instance $\boldsymbol{x} \in \mathcal{D}_n$, the produced synthetic anomaly $\tilde{\boldsymbol{x}}$ satisfies

$$r \leq \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_{\kappa} \leq \gamma r, \tag{15}$$

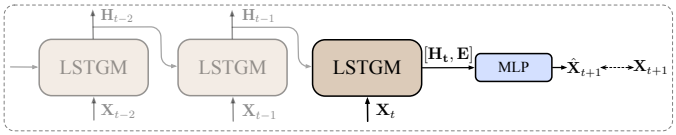
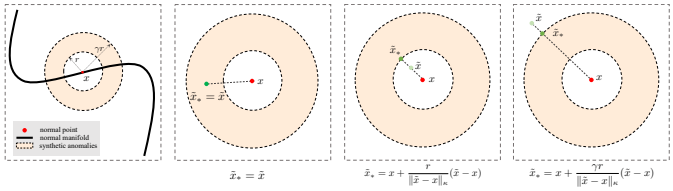


Fig. 4. LSTGM is pretrained to perform the autoregressive prediction task, so as to advance the learning process for vertex embeddings.



(a) (b) (c) (d)

Fig. 5. Illustration for synthetic anomalies based on the manifold assumption.

Algorithm 2 HiSTAR (Train)

Input: Dataset $\mathcal{D} = \mathcal{D}_n \cup \mathcal{D}_a$, pretrained LSTGM, randomly initialized DSC layers
Output: Post-trained LSTGM and DSC layers, hypersphere centers $\{\boldsymbol{c}_i^{(l)}\}$
iteration$\leftarrow$1;
while iteration$\leq$ epochs **do**
  $***$ *calculate the hypersphere centers* $***$
  for $\boldsymbol{x} \in \mathcal{D}$ **do**
    $\{\mathbf{H}^{(l)}\}, \_, \_ \leftarrow \mathrm{Fwd}(\boldsymbol{x}, \mathrm{LSTGM}, \mathrm{DSC})$;
  end for
  for $l \in \{0, \cdots, L\}$ **do**
    for $i \in \{1, \cdots, N^{(l)}\}$ **do**
      $\boldsymbol{c}_i^{(l)} \leftarrow \mathbb{E}_{\boldsymbol{x}\in\mathcal{D}_n}\left[\mathbf{H}_i^{(l)}; \boldsymbol{x}\right]$;
    end for
  end for
  $***$ *mini-batch optimization* $***$
  for $\mathcal{B} \subset \mathcal{D}$ **do**
    Augment $\mathcal{B}$ with synthetic anomalies via Eq. 18;
    for $\boldsymbol{x} \in \mathcal{B}$ **do**
      $\{\mathbf{H}^{(l)}\}, \{\mathbf{S}^{(l)}\}, \{\mathbf{A}^{(l)}\} \leftarrow \mathrm{Fwd}(\boldsymbol{x}, \mathrm{LSTGM}, \mathrm{DSC})$;
      Calculate $\mathcal{L}_{SC}(\boldsymbol{x})$ via Eq. (10);
      Calculate $F(\boldsymbol{x})$ via Eq. (11);
      Calculate $\mathcal{L}_{OC}$ via Eq. (12);
    end for
    Calculate $\mathcal{L}_{all}$ via Eq. (13);
    AdamOptimizer($\mathcal{L}_{all}$, learning rate);
  end for
  iteration$\leftarrow$iteration+1
end while

Algorithm 3 HiSTAR (Detection and Localization)

Input: Data sample $\boldsymbol{x}$, post-trained LSTGM and L DSC layers, hypersphere centers $\{\boldsymbol{c}_i^{(l)}\}$, threshold ϵ;
Output: IsAnomaly, Location
$\{\mathbf{H}^{(l)}\}, \_, \_ \leftarrow \mathrm{Fwd}(\boldsymbol{x}, \mathrm{LSTGM}, \mathrm{DSC})$;
Calculate $F(\boldsymbol{x})$ via Eq. (11);
IsAnomaly $\leftarrow$ False, Location $\leftarrow$ None;
if $F(\boldsymbol{x}) > \epsilon$ **then**
  IsAnomaly $\leftarrow$ True;
  Location $\leftarrow \mathrm{argmax}_{i\in\{1,\cdots,N\}} \left\|\mathbf{H}_i^{(0)} - \boldsymbol{c}_i^{(0)}\right\|_2^2$;
end if

where $\|\cdot\|_{\kappa}$ is the Mahalanobis distance, $r > 0$ represents the tolerance that a normal point can be off the manifold, and $\gamma > 1$ is a regularization hyperparameter. γr is a distance upper bound in case that the synthetic anomalies far away from one normal point may be close to other normal ones. Mahalanobis distance is adopted considering that the data measurements from different spatial and temporal locations are of different importance to the decision criteria and should be rescaled by κ. Specifically,

$$\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_{\kappa}^2 = \sum_{t=1}^{T}\sum_{j=1}^{N} \kappa_{t,j} \cdot (\boldsymbol{x}_{t,j} - \tilde{\boldsymbol{x}}_{t,j})^2. \tag{16}$$

This article has been accepted for publication in IEEE Transactions on Industrial Informatics. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TII.2022.3216006

$\kappa_{t,j}$ measures the importance of $x_{t,j}$ to the anomaly score function, calculated as $\kappa_{t,j} = \partial F(\boldsymbol{x})/\partial \boldsymbol{x}_{t,j}$. Denote the perturbation on $\boldsymbol{x}$ as $\Delta\boldsymbol{x}$, which is initialized by randomly sampling from a multivariate Gaussian distribution. The synthetic anomaly candidate is then initialized as $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \Delta\boldsymbol{x}$, and updated through a gradient ascent phase,

$$\Delta\boldsymbol{x} \leftarrow \Delta\boldsymbol{x} + \mu \nabla_{\tilde{\boldsymbol{x}}} F(\tilde{\boldsymbol{x}}), \tilde{\boldsymbol{x}} \leftarrow \boldsymbol{x} + \Delta\boldsymbol{x}, \tag{17}$$

where μ is the updating rate. The updating process is conducted for several iterations. Considering that it may not yield ideal points exactly satisfying the constraint in Eq. (15), the adjustments on $\tilde{\boldsymbol{x}}$ are necessary, as illustrated in Fig. 5(b)-(c). It is solved as an one-dimensional optimization problem. Given the candidate $\tilde{\boldsymbol{x}}$ originating from $\boldsymbol{x}$, we attain the optimal synthetic anomalous sample $\tilde{\boldsymbol{x}}_*$ by solving the optimization problem: $\min_{\tilde{\boldsymbol{x}}_*} \quad \|\tilde{\boldsymbol{x}}_* - \tilde{\boldsymbol{x}}\|_\kappa^2, \quad \text{s.t. } r^2 \le \|\tilde{\boldsymbol{x}}_* - \boldsymbol{x}\|_\kappa^2 \le \gamma^2 r^2$. The solution is formulated as

$$\tilde{\boldsymbol{x}}_* = \begin{cases} \boldsymbol{x} + \frac{r}{\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_\kappa}(\tilde{\boldsymbol{x}} - \boldsymbol{x}), \text{ if } \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_\kappa^2 < r^2 \\ \boldsymbol{x} + \frac{\gamma r}{\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_\kappa}(\tilde{\boldsymbol{x}} - \boldsymbol{x}), \text{ if } \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_\kappa^2 > \gamma^2 r^2 \\ \tilde{\boldsymbol{x}}, \text{ if } r^2 \le \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_\kappa^2 \le \gamma^2 r^2 \end{cases} \tag{18}$$

A brief proof for Eq. (18) is provided in Appendix.

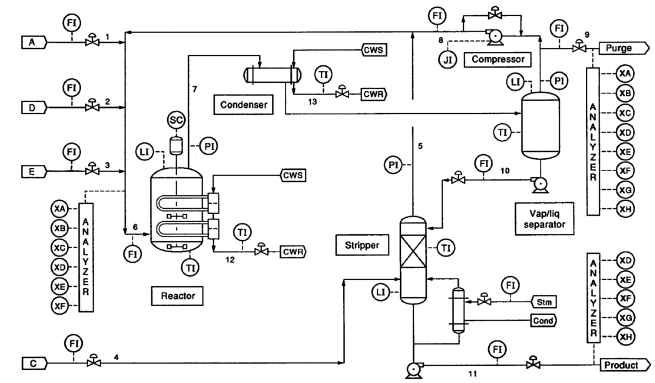
Based on the aforementioned modules and technical points, the comprehensive framework of HiSTAR is summarized in Algorithm 1-3. Algorithm 1 is the basic forward process of HiSTAR, which involves a LSTGM module and L DSC layers. Algorithm 2 outlines the training procedure of HiSTAR, in which the LSTGM is first pretrained to perform the autoregressive prediction task, the calculation of hypersphere centers and the backward optimization are performed alternately, and synthetic anomalous instances are involved to prevent the hypersphere collapse issue. Algorithm 3 describes the anomaly detection and localization procedure.

III. Case Studies

We verify the effectiveness of HiSTAR in multivariate time series anomaly detection in different industrial applications, including the Tennessee Eastman process (TEP) [27] for chemical industrial process monitoring, Water Distribution (WADI) for water treatment security [28], and UNSW-NB15 [29] for network intrusion detection. We first briefly introduce the main properties of the used datasets and then provide the technical results in the following.

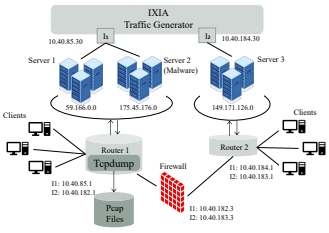
A. Dataset Description

1) Tennessee Eastman Process: The TEP is an important simulated chemical industrial process proposed by [27] to benchmark process monitoring techniques. It is involved with 8 main chemical components labeled from A to H, and dominated by 5 main process units including a reactor, condenser, separator, compressor and stripper. An overview of the TEP is given in Fig. 6(a). Specifically, four gaseous reactants (A, C, D and E) and the inert (B) were fed into the reactor, and produced two liquid products (G and H). The gaseous product stream from the reactor was cooled down in the condenser. The gaseous and liquid products from the condenser were separated in the separator. The remaining gaseous stream was flowed back into the reactor by a centrifugal compressor. The products were separated from unreacted feed components by a stripper.



(a) TEP

(b) WADI



(c) UNSW-NB15

Fig. 6. Platform overview of Tennessee Eastman Process (TEP) [27], Water Distribution (WADI) [28] and UNSW-NB15 [29].

A purge was utilized to remove the inert and the byproduct (F) in the whole process. We use the extended dataset generated by [30], which contains 500 different sets of simulation results. In each set, the TEP was conducted for 21 runs representing a normal state and 20 pre-programmed fault states. Each simulation run produced a training subset and a testing subset, which consisted of 500 and 960 samples respectively. These samples were collected sequentially every three minutes, each of which contained a total of 52 observed variables, including 41 process measurements and 11 manipulated variables. Under each fault state, the faults were introduced in one hour for the training data and in eight hours for the testing data. Finally, the whole TEP dataset is comprised of a fault-free training data file (250k samples), a fault-free testing data file (480k samples), a faulty training data (5000k samples) and a faulty testing data (9600k samples). We train the models with the fault-free training data in the unsupervised scenario. A very limited subset of the faulty training data is additionally utilized in the semi-supervised scenario. The evaluation dataset is constructed with the faulty-free testing data and 5% of the faulty testing data (960k data points with 41.67% anomalies).

2) Water Distribution: This case study involves the dataset collected on the WADI testbed towards water treatment security, as illustrated in Fig. 6(b). WADI integrated digital and physical components for system control and monitoring, and represented a scale-down version of a realistic modern industrial cyber-physical system. The water distribution process in WADI was comprised of a primary grid for water reservation, a secondary grid for water supply to six consumer tanks, and a return-water grid for excess water draining. Each subprocess was controlled by a pair of Programmable Logic Controllers (PLCs), which communicated with the corresponding sensors and actuators through a local network. The monitoring

data streams were read from the PLCs, transmitted to the SCADA system and recorded by the Historian through a star network. The testbed worked non-stop from its initial state to the operational state amount to 16 days in two different scenarios, allowing data collection under normal or attacked behavior modes. Normal operations were performed for the first 14 days, and 15 cyber-physical attacks were launched in the remaining 2 days at intervals. Therefore, the collected dataset could be naturally separated into the training dataset and evaluation dataset based on the date. The attacks were generated by disrupting the sensor readings or overturning the operation states of actuators, with duration time ranging from 2 to 30 minutes. The process variables were collected at a rate of one measurement per second, including 121 sensor readings or actuator states. Finally, nearly 785k anomaly-free records were collected for the training dataset, and 173k (with 5.85% anomalies) for the evaluation dataset.

3) UNSW-NB15: UNSW-NB15 is a comprehensive network traffic record of 41 variables under a hybrid of real-world normal and synthesized attack behaviors. The abnormal traffic was simulated with the IXIA PerfectStorm tool and represented nine different types of attacks, including Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode and Worms. An overview of the testbed is provided in Fig. 6(c). Specifically, the IXIA traffic generator was configured with three servers, with Server 1 and Server 3 for normal traffic spread, and Server 2 for abnormal activities. Two interfaces were utilized for the intercommunication among these servers. The connection of the servers to hosts was established via two routers connected to a firewall unit. Router 1 was installed with a tcpdump tool to capture the Pcap files. We convert the categorical variables in the provided training and testing source file to numerical values as [31] did. Finally, we construct the training dataset with all the normal traffic in the training file (nearly 37k anomaly-free data points). A limited subset of the abnormal traffic in the training file is additionally utilized in the semi-supervised scenario. All the records in the testing file are utilized for performance evaluation (175k data points with 68.06% anomalies).

These three datasets are preprocessed with normalization to mitigate the value scale differences among feature variables.

B. Hyperparameter Selection

Table I summarizes the hyperparameters used in our experiments. We assume that the graph scale is decreased by 2/3 in spectral clustering. Then, the number of DSC layers is empirically decided as $\lceil \log_3 N \rceil + 1$ (3 for TEP, 4 for WADI and 3 for UNSW-NB15). The sliding window length T, the hidden state size C_h, the learning rate, the batch size and the training epochs bring limited variation to the evaluation results if chosen reasonably in our experiments. The MLP structures in the pretraining procedures and DSC layers are taken as 3 hidden layers, with 128, 256 and 128 units, respectively. τ and λ are two important hyperparameters to balance the contribution of each loss item in the proposed method. We conduct sensitivity analysis for these two hyperparameters in the unsupervised scenario by varying one while freezing the other. The average F1 scores are reported in Fig. 7. The results show that the model performance is insensitive to variation in these parameters within a reasonable range.

TABLE I
THE SELECTED HYPERPARAMETERS FOR EACH DATASET.

Hyperparameters	TEP	WADI	UNSW-NB15
L	3	4	3
$\{N^{(l)}\}_{l=0}^{L}$	{52,17,5,1}	{121,40,13,4,1}	{41,13,4,1}
τ	1.0	1.4	0.7
λ	1.1e-2	9e-3	1.3e-2
T	32		
C_h	64		
MLP hidden units	[128, 256, 128]		
learning rate	1e-4		
batch size	128		
epoch	200		

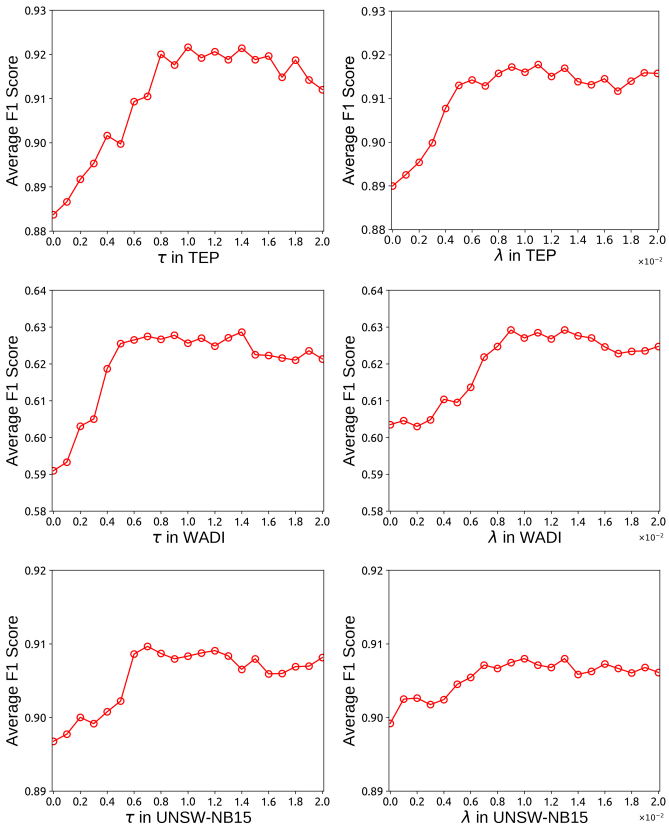


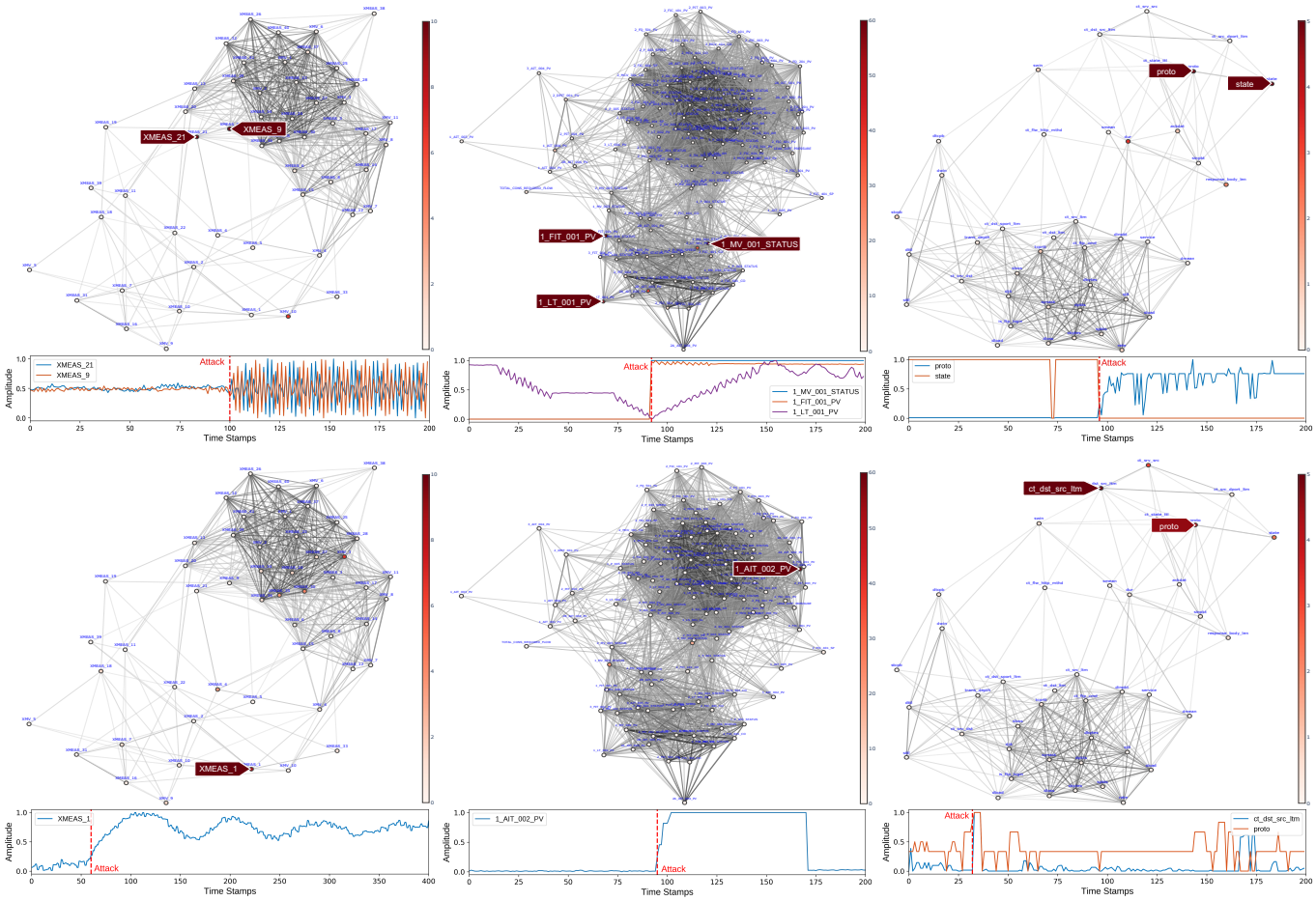
Fig. 7. Sensitivity analysis for two important hyperparameters τ and λ in HiSTAR.

We construct our model with Pytorch and PyTorch Geometric Library on a PC in the operating system of Ubuntu 16.04 equipped with Intel Core i5-9400F CPU @ 2.90GHz×6, 16 GB RAM and GeForce GTX 1660 GPU. We use the Adam solver and set $\beta_1 = 0.9$, $\beta_2 = 0.99$ and the weight decay rate as 10^{-4}. Then the whole model is trained for up to 200 epochs with the mini-batch size 256. The learning rate is fixed as 10^{-4} in the first 100 epochs, and decays to 10^{-5} linearly in the following 100 epochs.

TABLE II

PERFORMANCE RESULTS OF HISTAR AND THE BASELINES ON THE EVALUATION DATASETS. ABLATION STUDIES FOR EACH COMPONENT OF HISTAR ARE ALSO INCLUDED.

	TEP (41.67% anomaly ratio)				WADI (5.85% anomaly ratio)				UNSW-NB15 (68.06% anomaly ratio)			
	Pr	Re	F1	Acc	Pr	Re	F1	Acc	Pr	Re	F1	Acc
OCSVM	0.7484	0.5806	0.6539	0.7439	0.3981	0.3115	0.3495	0.9322	0.7309	0.9561	0.8285	0.7305
IsolationForest	0.6196	0.6395	0.6294	0.6862	0.3719	0.2861	0.3234	0.9300	0.6987	0.9909	0.8195	0.7030
LOF	0.7959	0.5764	0.6686	0.7619	0.1523	0.2643	0.1932	0.8710	0.7498	0.9367	0.8329	0.7442
VAR	0.7292	0.7431	0.7361	0.7780	0.3911	0.3722	0.3814	0.9294	0.7368	0.9215	0.8189	0.7225
DeepSVDD	0.8033	0.7762	0.7895	0.8276	0.4236	0.4663	0.4439	0.9317	0.7545	0.9522	0.8419	0.7566
AE	0.7875	0.5653	0.6582	0.7553	0.3286	0.3853	0.3547	0.9180	0.7412	0.9607	0.8368	0.7449
DAGMM	0.8035	0.6248	0.7030	0.7800	0.3389	0.4772	0.3963	0.9150	0.7498	0.9653	0.8440	0.7571
LSTM-VAE	0.8579	0.8262	0.8418	0.8706	0.4762	0.5494	0.5102	0.9383	0.8051	0.9120	0.8552	0.7898
OmniAnomaly	0.8793	0.8346	0.8564	0.8833	0.4524	0.6186	0.5226	0.9339	0.7864	0.9217	0.8487	0.7763
AnoGAN	0.7531	0.6176	0.6787	0.7563	0.3291	0.5564	0.4136	0.9078	0.7560	0.9341	0.8357	0.7500
BeatGAN	0.8264	0.8353	0.8308	0.8583	0.4387	0.6732	0.5312	0.9305	0.8236	0.9023	0.8612	0.8020
MAD-GAN	0.8613	0.8711	0.8662	0.8878	0.5083	0.6245	0.5604	0.9427	0.8452	0.8948	0.8693	0.8169
MTAD-GAT	0.8558	0.8940	0.8745	0.8931	0.4987	0.6647	0.5699	0.9413	0.8873	0.8676	0.8773	0.8349
GDN	0.8726	0.8958	0.8840	0.9021	0.5186	0.6582	0.5801	0.9443	0.8923	0.8783	0.8852	0.8450
GANF	0.8813	0.8804	0.8808	0.9008	0.5225	0.6451	0.5774	0.9448	0.8978	0.8834	0.8905	0.8522
VGCRN	0.8838	0.9182	0.9007	0.9156	0.5489	0.6749	0.6054	0.9486	0.9112	0.8903	0.9006	0.8663
HiSTAR	0.9133	0.9454	**0.9291**	**0.9399**	0.5921	0.6923	**0.6383**	**0.9541**	0.9212	0.9034	**0.9122**	**0.8817**
w/o HH	0.8694	0.9142	0.8912	0.9070	0.5414	0.6743	0.6006	0.9476	0.9119	0.8871	0.8993	0.8648
w/o GM	0.8414	0.8547	0.8480	0.8723	0.5117	0.6138	0.5581	0.9432	0.8552	0.8675	0.8613	0.8098



(a) TEP (b) WADI (c) UNSW-NB15

Fig. 8. Visualization for the learned graph structure and illustration for the capability of HiSTAR in anomaly localization. The variable values are normalized for better visualization.

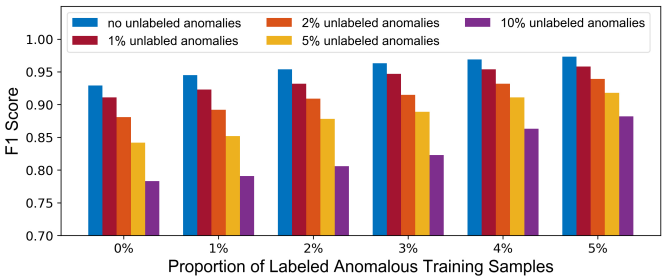
C. Results and Discussion

We adopt F1 score and accuracy as the performance evaluation metrics. Specifically, F1 score is a harmonic mean of precision (Pr) and recall (Re), and accuracy (Acc) measures the ratio of correct predictions on the evaluation dataset,

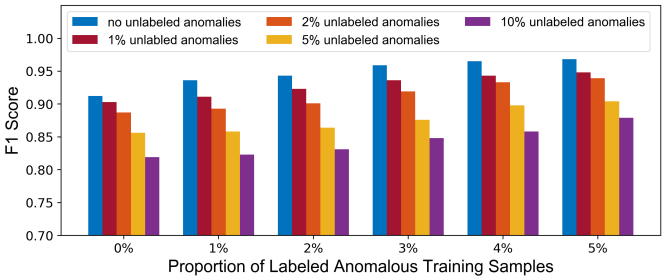
$$\begin{aligned} \mathrm{Pr} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \mathrm{Re} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \\ \mathrm{F1} &= \frac{2\times \mathrm{Pr}\times \mathrm{Re}}{\mathrm{Pr}+\mathrm{Re}}, \mathrm{Acc} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{FN}+\mathrm{TN}+\mathrm{FP}}, \end{aligned} \tag{19}$$

where TP, FN, TN and FP denote true positive, false negative, true negative and false positive, respectively. The anomaly score thresholds are selected as the values that lead to the highest F1 scores. Comparison experiments are conducted between HiSTAR and several baseline methods in the unsupervised scenario, including OCSVM [32], Isolation Forest [33], LOF [34], VAR [35], Deep SVDD [20], AE [36], DAGMM [37], LSTM-VAE [38], OmniAnomaly [39], AnoGAN [40], BeatGAN [41] and MAD-GAN [42], MTAD-GAT [43], GDN [7], GANF [44] and VGCRN [45]. OCSVM, Isolation Forest, LOF and VAR are classical shallow models, while the others are based on deep neural networks. MTAD-GAT, GDN, GANF, and VGCRN are advanced GNN-based anomaly detectors. The proposed HiSTAR is distinguished by constructing hierarchical decision boundaries in a discriminative way, while these methods belong to the family of generative models. For fair comparison, the baselines are adjusted with the corresponding optimal hyperparameters. The evaluation results are listed in Table II. It shows that deep learning methods achieve much better performance than shallow ones. Among them, GNN-based methods tend to present remarkable improvements compared with others, verifying that graph modeling is important to enhance robustness of anomaly detectors. Based on hierarchical spatial-temporal graph modeling, the proposed HiSTAR exhibits a prominent performance superiority over the baselines, with 0.9291 F1 score/0.9399 accuracy on TEP, 0.6383/0.9541 on WADI and 0.9122/0.8817 on UNSW-NB15. We also figure out the effect of each component in HiSTAR through ablation studies. In general, our method has two essential components, including graph modeling (GM) and hierarchical hyperspheres (HH). We gradually exclude each component and observe how the model performance degrades. HH can be removed by setting τ as a very small value, such as $1e-5$. Based on that, GM can be removed by substituting the LSTGM with the standard GRU and substituting all the DSC modules with MLP. By excluding HH and GM, HiSTAR degenerates into a temporal extension of DeepSVDD via GRU. It can be observed that the model performance of HiSTAR is significantly decreased by removing these two components, from 0.9291/0.9399 to 0.8480/0.8723 for TEP, from 0.6383/0.9541 to 0.5581/0.9432 for WADI and from 0.9122/0/8817 to 0.8613/0.8098 for UNSW-NB15.

We further show that the limited prior information about anomalies can help to further improve our model performance. The positive training samples in semi-supervised scenarios can be separated into unlabeled ones and labeled ones. The unlabeled positive training samples are roughly treated as normal in training, which will inevitably bring noise to the training process and cause an adverse effect on the model performance. We set up 30 different semi-supervised scenarios for each of TEP and UNSW-NB15 by selecting the unlabeled anomaly ratio from 0%, 1%, 2%, 5% and 10% and ranging the labeled anomaly ratio in 0%, 1%, 2%, 3%, 4% and 5%. The evaluation F1 scores are illustrated in Fig. 9. It shows that the F1 score is increased from 0.9291 to 0.9732 for TEP and from 0.9122 to 0.9683 for UNSW-NB15 with merely 5% labeled anomalous training samples in the noise-free scenario. Even in the most contaminated scenario where the unlabeled anomaly ratio reached 10%, the F1 score is increased from 0.7832 to 0.8814 for TEP and from 0.8187 to 0.8793 for UNSW-NB15. It can be concluded that HiSTAR could take good advantage of the precious labeled positive training samples so as to remarkably enhance the detection robustness.



(a) TEP



(b) UNSW-NB15

Fig. 9. F1 scores of HiSTAR with a small amount of anomalous training samples in TEP and UNSW-NB15. The results show that a small fraction of prior information about anomalies could remarkably enhance the detection performance of HiSTAR.

We also demonstrate that the proposed HiSTAR is capable of consistent anomaly localization towards interpretability of the anomaly detection results. We extract the intermediate feature map $\mathbf{H}^{(0)}$ and calculate the anomaly scores for each vertex, as presented in Algorithm 3. The learned graph structure and the visualization results for anomaly localization are given in Fig. 8. It shows that HiSTAR could accurately identify the attacked sensor variables. We take examples with the first row of Fig. 8. It is the ground truth that the reactor cooling water valve is sticking in TEP, and our model reasonably identifies XMEAS_9 and XMEAS_21 as the attacked variables, which actually represent the reactor temperature and reactor cooling water outlet temperature, respectively. The three variables including 1_MV_001_STATUS, 1_FIT_01_PV and 1_LT_001_PV are identified as attacked variables in WADI, consistent with the ground truth that the motorized valve 1_MV_001 is maliciously turned on and causes an overflow

on the primary tank reflected on 1LT001 and 1FIT001. The abnormal sudden changes of the state and the dependent protocol in UNSW-NB15 are detected by our model, when the Backdoor attack is launched as the ground truth.

IV. Conclusion

This article proposes HiSTAR, a novel and powerful method for robust industrial anomaly detection with multivariate time-series sensory data. It is highlighted by learning hierarchical normality representations through organized latent spatial-temporal graph modeling, without requiring any human-defined topological information. Although developed in the unsupervised scenario, it can effectively utilize the precious labeled anomalous training samples to further enhance the detection robustness. We also demonstrate that the proposed method can help localize the anomalies, thus providing interpretability for the detection results. The experiment results in three different industrial applications confirm the effectiveness of the proposed method. Our future research will consider the dynamic graph structure in graph modeling in a sense that the vertices or edges may change over time. This may be beneficial to a broader application of our method in multivariate industrial anomaly detection.

V. Appendix

Proof: [Proof for Eq (18)] Let $\alpha_1 \geq 0$ and $\alpha_2 \geq 0$ be the Lagrangian multipliers, then the Lagrangian function of the above problem is written as

$$\begin{aligned} L(\tilde{\boldsymbol{x}}, \alpha_1, \alpha_2) = \|\tilde{\boldsymbol{x}}_* - \tilde{\boldsymbol{x}}\|_\kappa^2 - \alpha_1(\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_\kappa^2 - r^2) \\ + \alpha_2(\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_\kappa^2 - \gamma^2 r^2) \end{aligned} \tag{20}$$

The optimal solution $\tilde{\boldsymbol{x}}_*$ satisfies the following Karush–Kuhn–Tucker (KKT) conditions,

$$\begin{cases} \partial L(\tilde{\boldsymbol{x}}_*, \alpha_1, \alpha_2)/\partial \tilde{\boldsymbol{x}}_* = 0 \\ \|\tilde{\boldsymbol{x}}_* - \boldsymbol{x}\|_\kappa^2 \geq r^2, \|\tilde{\boldsymbol{x}}_* - \boldsymbol{x}\|_\kappa^2 \leq \gamma^2 r^2, \\ \alpha_1 \geq 0, \alpha_2 \geq 0, \\ \alpha_1(\|\tilde{\boldsymbol{x}}_* - \boldsymbol{x}\|_\kappa^2 - r^2) = 0, \\ \alpha_2(\|\tilde{\boldsymbol{x}}_* - \boldsymbol{x}\|_\kappa^2 - \gamma^2 r^2) = 0. \end{cases} \tag{21}$$

It should be discussed under three circumstances:

- $\alpha_1 = \alpha_2 = 0$. Both constraints are inactive, i.e., $r^2 \leq \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_\kappa^2 \leq \gamma^2 r^2$. Then we attain the interior solution $\tilde{\boldsymbol{x}}_* = \tilde{\boldsymbol{x}}$.
- $\alpha_1 > 0$ and $\alpha_2 = 0$. Only the constraint of $\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_\kappa^2 < r^2$ is active. Then we attain the boundary solution when $\|\tilde{\boldsymbol{x}}_* - \boldsymbol{x}\|_\kappa^2 = r^2$, i.e., $\tilde{\boldsymbol{x}}_* = \boldsymbol{x} + \frac{r}{\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_\kappa}(\tilde{\boldsymbol{x}} - \boldsymbol{x})$
- $\alpha_1 = 0$ and $\alpha_2 > 0$. Only the constraint $\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_\kappa^2 > \gamma^2 r^2$ is active. Then we attain the boundary solution when $\|\tilde{\boldsymbol{x}}_* - \boldsymbol{x}\|_\kappa^2 = \gamma^2 r^2$, i.e., $\tilde{\boldsymbol{x}}_* = \boldsymbol{x} + \frac{\gamma r}{\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_\kappa}(\tilde{\boldsymbol{x}} - \boldsymbol{x})$.

■

References

[1] G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, "Deep learning for anomaly detection: A review," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[2] B. Wang, Z. Li, Z. Dai, N. Lawrence, and X. Yan, "Data-driven mode identification and unsupervised fault detection for nonlinear multimode processes," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 3651–3661, 2019.

[3] X. Deng, P. Jiang, X. Peng, and C. Mi, "An intelligent outlier detection method with one class support tucker machine and genetic algorithm toward big sensor data in internet of things," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 6, pp. 4672–4683, 2018.

[4] I. F. Ghalyan, N. Ghalyan, and A. Ray, "Optimal window-symbolic time series analysis for pattern classification and anomaly detection," *IEEE Transactions on Industrial Informatics*, 2021.

[5] L. Li, J. Yan, H. Wang, and Y. Jin, "Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 3, pp. 1177–1191, 2020.

[6] J. Wu, Z. Zhao, C. Sun, R. Yan, and X. Chen, "Fault-attention generative probabilistic adversarial autoencoder for machine anomaly detection," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 12, pp. 7479–7488, 2020.

[7] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4027–4035.

[8] Z. Xiao, Q. Yan, and Y. Amit, "Likelihood regret: An out-of-distribution detection score for variational auto-encoder," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[9] J. Ren, P. J. Liu, E. Fertig, J. Snoek, R. Poplin, M. Depristo, J. Dillon, and B. Lakshminarayanan, "Likelihood ratios for out-of-distribution detection," *Advances in Neural Information Processing Systems*, vol. 32, pp. 14 707–14 718, 2019.

[10] Z. Pu, D. Cabrera, Y. Bai, and C. Li, "A one-class generative adversarial detection framework for multifunctional fault diagnoses," *IEEE Transactions on Industrial Electronics*, 2021.

[11] Q. Xie, P. Zhang, B. Yu, and J. Choi, "Semisupervised training of deep generative models for high-dimensional anomaly detection," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[12] N. Wang, Z. Zhang, X. Zhao, Q. Miao, R. Ji, and Y. Gao, "Exploring high-order correlations for industry anomaly detection," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9682–9691, 2019.

[13] C.-L. Liu, W.-H. Hsaio, and Y.-C. Tu, "Time series classification with multivariate convolutional neural network," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 6, pp. 4788–4797, 2018.

[14] X. Wang, S. Si, and Y. Li, "Variational embedding multiscale diversity entropy for fault diagnosis of large-scale machinery," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 3, pp. 3109–3119, 2021.

[15] Q. He, Y. Pang, G. Jiang, and P. Xie, "A spatio-temporal multiscale neural network approach for wind turbine fault diagnosis with imbalanced scada data," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 10, pp. 6875–6884, 2020.

[16] J. Yu, Y. Song, D. Tang, D. Han, and J. Dai, "Telemetry data-based spacecraft anomaly detection with spatial–temporal generative adversarial networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–9, 2021.

[17] M. Khodayar, S. Mohammadi, M. E. Khodayar, J. Wang, and G. Liu, "Convolutional graph autoencoder: A generative deep neural network for probabilistic spatio-temporal solar irradiance forecasting," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 2, pp. 571–583, 2019.

[18] Z. He, P. Chen, X. Li, Y. Wang, G. Yu, C. Chen, X. Li, and Z. Zheng, "A spatiotemporal deep learning approach for unsupervised anomaly detection in cloud systems," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[19] L. Deng, D. Lian, Z. Huang, and E. Chen, "Graph convolutional adversarial networks for spatiotemporal anomaly detection," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[20] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4393–4402.

[21] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations*, 2017.

[22] G. Li, M. Müller, G. Qian, I. C. D. Perez, A. Abualshour, A. K. Thabet, and B. Ghanem, "Deepgcns: Making gcns go as deep as cnns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[23] X. Y. Stella and J. Shi, "Multiclass spectral clustering," in *Computer Vision, IEEE International Conference on*, vol. 2. IEEE Computer Society, 2003, pp. 313–313.

[24] F. M. Bianchi, D. Grattarola, and C. Alippi, "Spectral clustering with graph neural networks for graph pooling," in *International Conference on Machine Learning*. PMLR, 2020, pp. 874–883.

[25] L. Ruff, R. A. Vandermeulen, B. J. Franks, K.-R. Müller, and M. Kloft, "Rethinking assumptions in deep anomaly detection," in *ICML 2021 Workshop on Uncertainty & Robustness in Deep Learning*, 2021.
[26] S. Goyal, A. Raghunathan, M. Jain, H. V. Simhadri, and P. Jain, "Drocc: Deep robust one-class classification," in *International Conference on Machine Learning*. PMLR, 2020, pp. 3711–3721.
[27] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Computers & chemical engineering*, vol. 17, no. 3, pp. 245–255, 1993.
[28] C. M. Ahmed, V. R. Palleti, and A. P. Mathur, "Wadi: a water distribution testbed for research in the design of secure cyber physical systems," in *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*, 2017, pp. 25–28.
[29] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*. IEEE, 2015, pp. 1–6.
[30] C. A. Rieth, B. D. Amsel, R. Tran, and M. B. Cook, "Issues and advances in anomaly detection evaluation for joint human-automated systems," in *International Conference on Applied Human Factors and Ergonomics*. Springer, 2017, pp. 52–63.
[31] F. H. Botes, L. Leenen, and R. De La Harpe, "Ant colony induced decision trees for intrusion detection," in *16th European Conference on Cyber Warfare and Security*. ACPI, 2017, pp. 53–62.
[32] B. Schölkopf, R. C. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection," *Advances in neural information processing systems*, vol. 12, 1999.
[33] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.
[34] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: identifying density-based local outliers," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, 2000, pp. 93–104.
[35] H. Lütkepohl, *Introduction to multiple time series analysis*. Springer Science & Business Media, 2013.
[36] C. C. Aggarwal, "An introduction to outlier analysis," in *Outlier analysis*. Springer, 2017, pp. 1–34.
[37] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International Conference on Learning Representations*, 2018.
[38] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1544–1551, 2018.
[39] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. Pei, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 2828–2837.
[40] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *International Conference on Information Processing in Medical Imaging*. Springer, 2017, pp. 146–157.
[41] B. Zhou, S. Liu, B. Hooi, X. Cheng, and J. Ye, "Beatgan: Anomalous rhythm detection using adversarially generated time series." in *IJCAI*, 2019, pp. 4433–4439.
[42] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. Ng, "Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks," in *International Conference on Artificial Neural Networks*. Springer, 2019, pp. 703–716.
[43] H. Zhao, Y. Wang, J. Duan, C. Huang, D. Cao, Y. Tong, B. Xu, J. Bai, J. Tong, and Q. Zhang, "Multivariate time-series anomaly detection via graph attention network," in *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2020, pp. 841–850.
[44] E. Dai and J. Chen, "Graph-augmented normalizing flows for anomaly detection of multiple time series," in *International Conference on Learning Representations*, 2022.
[45] W. Chen, L. Tian, B. Chen, L. Dai, Z. Duan, and M. Zhou, "Deep variational graph convolutional recurrent network for multivariate time series anomaly detection," in *International Conference on Machine Learning*. PMLR, 2022, pp. 3621–3633.

Jingyu Yang received the B.S. degree in mechanical engineering from the School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2019. He is currently working toward the Ph.D. degree in control science and technology in the School of Artificial Intelligence and Automation, Huazhong University of Science and Technology. His research interests include signal processing and machine learning with applications to industrial fault detection.

Zuogong Yue is currently an Assistant Professor at Huazhong University of Science and Technology. He received the B.Sc. degree in mechatronics engineering from Zhejiang University, China, in 2011; the M.Phil. degree in mechanical engineering from the Hong Kong University of Science and Technology, China, in 2013; and the Ph.D. degree in engineering science from the University of Luxembourg, Luxembourg, in 2018. He worked as a postdoctoral fellow at the Luxembourg Centre for Systems Biomedicine in 2018 and at the University of New South Wales in 2019-2021. His research interests include system/network identification and control, signal processing and learning.