Spatial-temporal associations representation and application for process monitoring using graph convolution neural network

Hao Ren[a], Chunhua Yang[a,b], Xiaojun Liang[a,*], Zhiwen Chen[a,b], Weihua Gui[a,b]

[a]the Department of Mathematics and Theories, Peng Cheng Laboratory, Shenzhen, Guangdong, 518055, China.
[b]the School of Automation, Central South University, Changsha, Hunan, 410083, China.

Abstract

Industrial process data reflects the dynamic changes of operation conditions, which mainly refer to the irregular changes in the dynamic associations between different variables in different time. And this related associations knowledge for process monitoring is often implicit in these dynamic monitoring data which always have richer operation condition information and have not been paid enough attention in current research. To this end, a new process monitoring method based on spatial-based graph convolution neural network (SGCN) is proposed to describe the characteristics of the dynamic associations which can be used to represent the operation status over time. Spatia-temporal graphs are firstly defined, which can be used to represent the characteristics of node attributes (dynamic edge features) dynamically changing with time. Then, the associations between monitoring variables at a certain time can be considered as the node attributes to define a snapshot of the static graph network at the certain time. Finally, the snapshot containing graph structure and node attributes is used as model inputs which are processed to implement graph classification by spatial-based convolution graph neural network with aggregate and readout steps. The feasibility and applicability of this proposed method are demonstrated by our experimental results of benchmark and practical case application.

Keywords: Dynamic Associations Analysis; Process Monitoring; Fault Diagnosis; Spatial-based Graph Network.

1. Introduction

Process monitoring becomes increasingly important to safety and product quality assurance, whose practical application is hindered by the increasingly large-scale, complicated and intelligence of modern processes. Currently, process monitoring techniques in current research can be primarily divided into model-based methods and data-based methods. A main limitation of model-based methods mainly refers to the unavailable mechanism models of these processes, while data-based models become increasingly popular with its superiorities on low-dimensional features or representations from high-dimensional process data [1]-[3]. Jung *et al.* [4] focused on residual generators selection to achieve fault detection and isolation, which combined model-based and data-driven methods to find a set of residual generators. Han *et al.* [5] proposed a data-knowledge-based fuzzy neural network, and with the help of multi-layered connections structure, this method can not only make full use of the data from the current scene, but also use the existing knowledge from the source scene to improve learning performance. Thomas *et al.* [6] studied the organization of classified training data, which is the time consumers and the key issue for process monitoring. Luo *et al.* [7] paid their attentions on building easily interpretable process monitoring models by considering the useful process knowledge, which mainly integrated process data with fundamental process knowledge.

Among these methods that had emerged in recent years, a large amount of past efforts focus on the features which can be handled in Euclidean space, such as residuals, length, angle and frequency [8]-[10], while the frequently occurring non-Euclidean spatial features are often ignored in process monitoring, such as networks and structures [11]-[12]. Liao *et al.* [13] utilized the adjacency matrix to represent the similarity

*Corresponding author: Xiaojun Liang; E-mail address: liangxj@pcl.ac.cn (X. J. Liang).

metrics between unknown samples and labeled samples. Tootooni *et al.* [14] tested the hypothesis that the spectral graph-based topological invariants defect incipient process drifts earlier (lower average run length) and with higher fidelity (consistency of detection) when compared with the conventional statistics-based approaches. Chen *et al.* [15] took system topology into account when integrating multiple measurements at different buses to develop a graph convolutional network. Essentially, the aim of graph theoretic methods is to defect the changes (faults) in networks, and one of the most significant fundamental elements of them can be regarded as the adjacency matrix in networks, which are often considered as the associations between nodes.

Associations between nodes can be regarded as one of typical non-Euclidean spatial features, which has attracted increasing scholars attentions around the world. Chen *et al.* [16] used structural analysis to prediagnostic faults and then converted their results into the association graph. Zhang *et al.* [17] focused on the information contained in the relationships between nodes to handle the limitations on learning features from the nodes of input data by conventional deep learning methods. Yuan *et al.* [18] used spatial variable distance to measure similarity and assign weights for historical samples, and this method had been validated that it can be regarded as to effectively handle process time-varying problems of abrupt changes, and they also realized soft sensing with this dynamic time sequences [19] . Chen *et al.* [20] firstly decomposed the whole process into different units, and then the process data in each local unit is used to automatically determined the associations between each pair of units. However, most of the existing association graphs on process monitoring are defined as static graph whose structure or associations between nodes are not changing over time, while the nature graph in practical industrial process is naturally dynamic. Generally, the temporal information (natural dynamics) heavily relies on association features that are related to texture of practical process.

The main characteristics of the temporal information are often changed with that the nodes and edges (associations) often continuously update, insert and delete over time, and furthermore, the attributes of node and edges are also changed over time. For example, the process graph is often affected by various factors, and new interactions and changes are constantly being made between nodes to ensure product quality. In these scenarios, graph model needs to focus not only on graph structure information, but also on the historical information [21]. Due to the importance of consideration on dynamic associations between nodes, it would be an advancement for process monitoring performance to consider the changes in dynamic correlations between monitoring variables in different time. The contributions of this paper can be summarized as follows.

- Spatial-based graph network is addressed to describe the characteristics of the dynamic associations between monitoring variables, which can be used to monitor system operation status and this can be used to improve process monitoring performance.
- An auto- and well-considered dynamic associations analysis for process monitoring is realized in practical industrial process monitoring, and its applicability on enhancing monitoring performance has also been demonstrated in this paper.
- Results on benchmark and practical case application are well performed to illustrate the effectiveness and applicability of this new method, and the results show that this process monitoring can achieve the state-of-art performance.

The rest of this paper is organized as follows. Section II illustrates the preliminaries of this paper. In Section III, the overall approach to process monitoring based on dynamic associations is addressed. The details of the experimental data-sets are described in section IV, including benchmark on Tennessee Eastman (TE) chemical process, case study on cobalt-nickel removal from zinc solution. Finally, conclusion and further works are summarized in last section.

2. Preliminaries

2.1. Spatia-temporal Graphs

More and more process data is generated in non-Euclidean domains and expressed as graphs with complex associations and interdependencies between nodes. The main characteristics of graph data are regarded as that there is a connection between different pair of nodes, which shows the nodes are not completely independent [22]. Dynamic graphs emphasize the sequences of appearance of nodes and edges are strongly coupled to time.

Definition 1: Spatia-temporal Graphs are regarded as an attribute graph in which the attributes of nodes change dynamically over time. This definition can be described as follows:

$$\mathbf{G}^t = (\mathbf{V}^t, \mathbf{E}^t, \mathbf{X}^t) \tag{1}$$

where $\mathbf{X}^t$ represents feature matrix of nodes $\mathbf{V}^t$ and edges $\mathbf{E}^t$ of graphs at time t.

2.2. Static Graph Snapshot

Definition 2: Consider graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ and $\mathbf{E}$ represent nodes-set and edges-set, respectively. And the associations $\mathbf{A}$ can be represented by the adjacent matrix of graph:

$$e_{ij} = \begin{cases} 1, & e_{ij} \in \mathbf{E} \\ 0, & else \end{cases} \tag{2}$$

where $e_{ij} = (v_i, v_j) \in \mathbf{E}$ represents the edge between node v_i and node v_j, where $v_i, v_j \in \mathbf{V}$ are the nodes of nodes-set.

Graph structures and node attributes can be regarded as model inputs, while the output of graph neural network can focus on different graph analysis tasks through one of a variety of different mechanisms [23], such as *node*-level is suitable for graph classification tasks.

2.3. Spatial-based Graph Convolution Neural Network

The nature of knowledge graph can be regarded as the association graph with numbers of nodes and edges which can be easily described by graph data (a kind of ubiquitous data) [24]. The typical idea of convolution graph network can be regarded as three key issues: aggregation, update and readout.

Aggregation can be simplified as to aggregate itself features x_i and its related neighbor features $x_j (j \in N(n_i))$ to generate a new feature vectors, in which $N(n_i)$ represents all the neighbors of node n_i. And this can be modeled as follows, mathematically.

$$\mathbf{AGG}_n^{(k)} = \sum_{i=1}^{k-1} \sum_{u \in N(n)} (\Theta^{(k)})^T \mathbf{h}_u^{(k-1)} \tag{3}$$

where n is the node index of graph. k is current layer number. $N(n)$ are the neighbors of node u. Θ are the parameters that waited to learn. $\mathbf{h}_u^{(k-1)}$ are the aggregate information of the neighbor node u in the (k-1)th layer.

Update can be regarded as to utilize the status of related neighbor nodes in current layer to calculate the status of the node in next layer. It can be modeled as follows.

$$\mathbf{h}_n^{(k)} = f((\mathbf{W}^{(k)})^T x_n + \mathbf{AGG}_n^{(k)}) \tag{4}$$

where n is the node index of graph. k is current layer number. $f(\cdot)$ is the activation function or update function, such as sigmoid, Relu, etc. W is the parameters that waited to learn. x_n is the current feature vector of node n. $\mathbf{h}_n^{(k)}$ is the new feature vector of the node n in the kth layer.

Readout integrates the features of all nodes to represent the features of entire graph by *sum* or *average* methods, which can be used to abstract the features of all nodes into hidden features to represent the features of the entire graph. All of these processes can be mathematically modeled as follows.

$$\mathbf{y} = g(\sum_{i=1}^{K} \hbar(\Phi_\upsilon^{(i)} \mathbf{h}_\upsilon^{(i)})), \upsilon = 1, 2, ..., V_i. \tag{5}$$

where K is the number of iterations. $\hbar(\cdot)$ refers to operators which calculates all the node information of each layer, such as *sum*, *max*, etc. $\mathbf{h}_\upsilon^{(i)}$ and $\Phi_\upsilon^{(i)}$ are feature vectors of node υ and corresponding weights that waited to learn on ith layer. V_i is the number of nodes on ith layer. $g(\cdot)$ is the classifier function, such as soft-max regression, etc.

3. Dynamic Associations and Process Monitoring

3.1. Framework of Proposed Method

This process monitoring method is constructed on dynamic associations which should be firstly defined and utilized to represent the operation status over time, whose core of process monitoring can be regarded as definition of node attributes and dynamic associations between nodes. This new proposed process monitoring mainly involves four steps: (i) monitoring variables of interest (definition of node attributions), and (ii) associations calculation and translation operators, and (iii) associations graph treatment by every snapshot, and (iv) graph classification and process monitoring, as shown in Fig.1.

Monitoring variables of interest can be regarded as the definition of nodes and their attributions, which can be used to calculate their associations by translation operators with every graph snapshot. The key to this process monitoring refers to dynamic associations calculation, spatial-based convolution treatment and graph classification, all of whose basic theory will be detailed as follows. Graph classification can be regarded as readout step in spatial-based convolution graph neural network. This new process monitoring framework employs the changes of the associations for nodes over time to implement fault detection and diagnosis. It is worthy note that the dynamic associations for nodes are utilized to achieve process monitoring, such as fault detection, fault diagnosis, etc., which can be considered as the novelty manner on process monitoring (as for as our consideration on process monitoring).

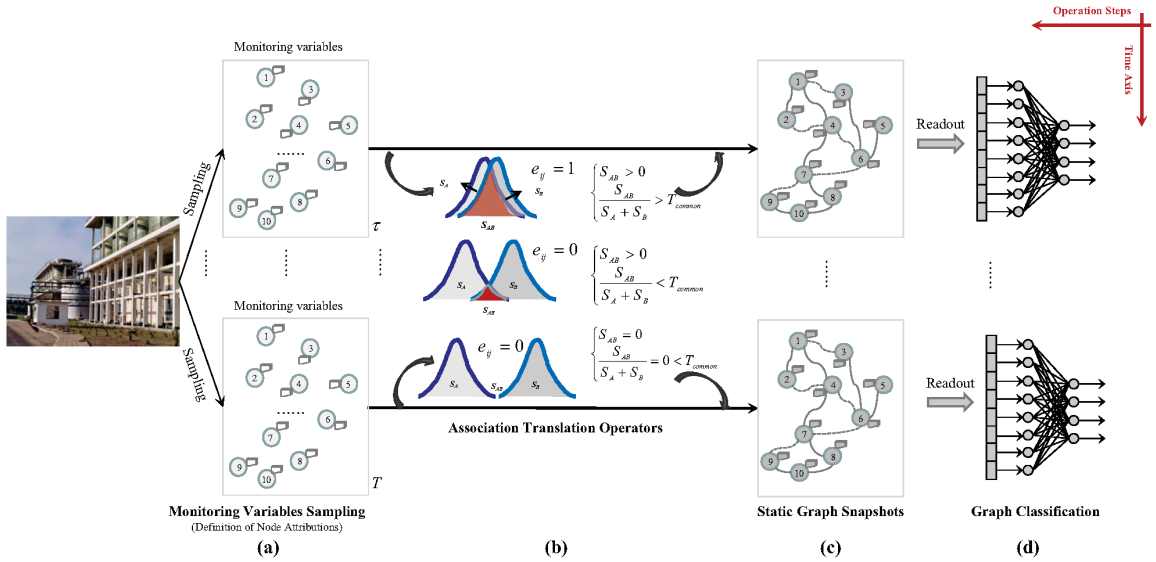


Figure 1: Framework of process monitoring based on dynamic associations with SGCN. (a) Monitoring variables of interest (definition of node attributions); (b) Associations calculation and translation Operators; (c) Association graph treatment by every snapshot; (d) Graph classification and process monitoring.

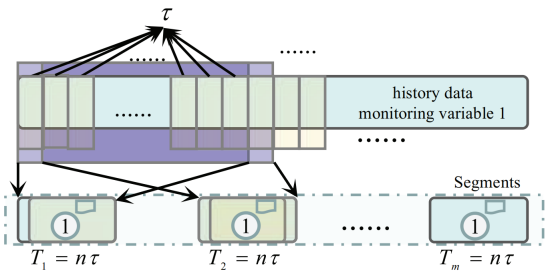


Figure 2: Schematic diagram of segments partition method.

3.2. Definition of Node Attributes

The ever-increasing capabilities of computer, communication technology cannot eliminate the need to extract meaningful information from ever-increasing industrial big data, which often aggravates the complexity of searching for related solutions in fact. To handle this problem, graph theory came into being and its goal is to extract or synthesize some information on given data (it usually represents a complex system) to create a new representation (graph network). Graph theory can be used to describe the relationship structure of complex system, which focuses on the internal structure of a single instance in a simple pattern. The core idea of the description of graph data is the attributes of nodes and edges, especially graphs in industrial big data mainly expressed by adjacency matrix (its rows represent node in data, while an element represents the connection between a pair of nodes).

Definition 3: The graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ for a given plant-wide process can be regarded as an attribute graph, and the attributes of nodes $\mathbf{V}$ can be defined as $F = (C, \mathfrak{D})$, where C is the association matrix between the node of interest and other nodes and $\mathfrak{D}$ is the other attributes of non-interest.

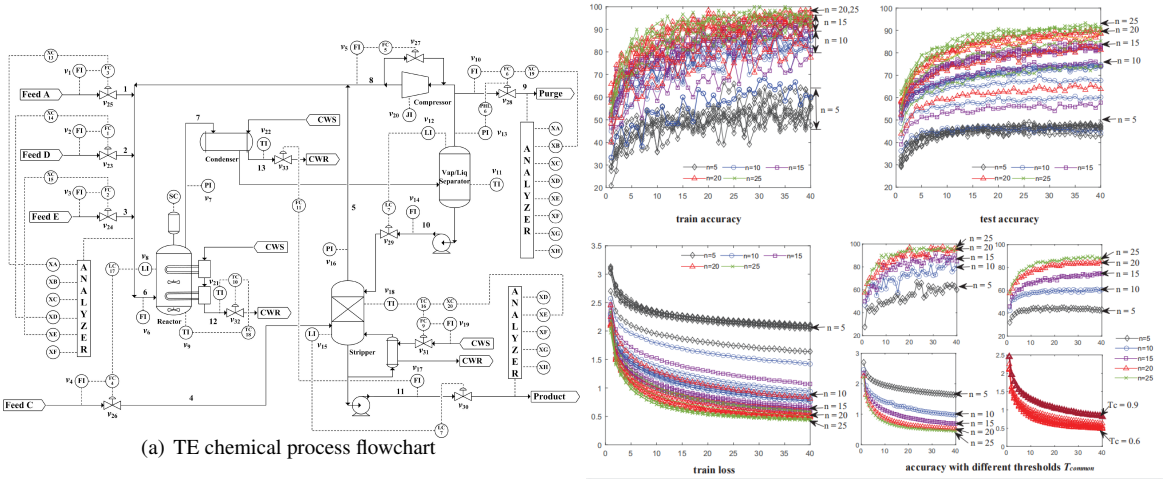
It should be noted that the attributes of nodes in graph can be defined as different entities in physical world, such as name, label, etc. In order to illustrate the basic theory, the attributions of nodes $\mathbf{V}$ are only considered the associations C in this paper.

3.3. Dynamic Associations Between Nodes

The monitoring variables are regarded as nodes contained many measurements, which can be used to describe the operation status. Association generally exists on a pair of nodes in each graph snapshot should be calculated by measurements of monitoring variables which are regarded as numerous nodes. Assume a pair of monitoring variables v_A and v_B, the element e_{AB} in adjacent matrix (association) can be calculated as follows.

$$e_{AB} = \begin{cases} 1, & S_{AB}/(S_A + S_B) \geq T_{common} \\ 0, & else \end{cases} \tag{6}$$

where $e_{AB} = 1$ means the connection exists between node v_A and node v_B, else $e_{AB} = 0$. T_{common} is one threshold hyper-parameter which is used to determine the association sparsity (the bigger the sparser).



(a) TE chemical process flowchart

(b) sample numbers n and hyper-parameter threshold T_{common}

Figure 3: TE chemical process and related parameters calculation.

Monitoring variables v_A and v_B are regarded as to obey the Gaussian distribution $N(\mu_A, \sigma_A^2)$ and $N(\mu_B, \sigma_B^2)$, respectively. And the cumulative distribution of monitoring variables v_A and v_B can be described as follows.

$$S_A = \int_{A_{bottom}}^{A_{upper}} N(\mu_A, \sigma_A^2), S_B = \int_{B_{bottom}}^{B_{upper}} N(\mu_B, \sigma_B^2) \quad (7)$$

where μ_A, σ_A^2 and μ_B, σ_B^2 are the mean and variance of monitoring variables v_A and v_B with sliding window τ. $A_{bottom} = \mu_A - 6\sigma_A, A_{upper} = \mu_A + 6\sigma_A$ are the bottom bounds and upper bounds of monitoring variables v_A. $B_{bottom} = \mu_B - 6\sigma_B, B_{upper} = \mu_B + 6\sigma_B$ are the bottom bounds and upper bounds of monitoring variables v_B.

Remark 1: The history data of monitoring variable 1 is regarded as an example, as shown in Fig.2. The number of samples of interest is n, and then the time length of one segment will be $T = n\tau$. As the sliding window τ advances forward, a series of segments will be generated. It should be noted that the samples number n can be considered as one of most important parameters, and if its value is too small, it is more susceptible to noises and uncertainties. Generally, the samples number n should be at least twice the maximum sampling frequency according to Shannon sampling theorem.

Correspondingly, the cross cumulative distribution of monitoring variables v_A and v_B can be represented as follows, mathematically.

$$S_{AB} = \int_{B_{bottom}}^{AB_{middle}} N(\mu_B, \sigma_B^2) + \int_{AB_{middle}}^{A_{upper}} N(\mu_A, \sigma_A^2) \quad (8)$$

where A_{bottom}, A_{upper} and B_{bottom}, B_{upper} are the bottom bounds and upper bounds of monitoring variables v_A and v_B, respectively. AB_{middle} is the cross bound of these two Gaussian distribution.

Consider the monitoring variable (node) v_A of interest, and then the associations between this node v_A and all other monitoring variables can be iteratively calculated to form an association array which can be regarded as a node attribute or feature C. According to associations calculation method, a static graph snapshot at a certain moment can be constructed by looping through all the monitoring variables (nodes) $\mathbf{V}$.

3.4. Spatial-based convolution treatment by every snapshot

The static graph snapshot (association graph) can be represented as $\mathcal{G} = (\mathcal{A}, \mathcal{F})$, where $\mathcal{A}$ is adjacent matrix and $\mathcal{F} \in \mathbb{R}^{n \times n}$ is the node feature vectors with each node owning one-dimensional association array $C \in \mathbb{R}^{1 \times n}$. A series of graph snapshots $\mathcal{D} = \{(\mathcal{G}_1, \mathbf{y}_1), (\mathcal{G}_2, \mathbf{y}_2), ...\}$ where $\mathbf{y}_i \in \mathcal{Y}$ is the label of graph $\mathcal{G}_i \in \mathcal{G}$ with the time i. Obviously, the dynamic refers to that the adjacent matrix $\mathcal{A}$ and the node feature vectors C are changed over time. The goal of graph treatment can be considered as to learn a translation $\Omega : \mathcal{G} \rightarrow \mathcal{Y}$ that maps graphs to the corresponding label-sets. According to the basic theory of SGCN in section II, the association to each graph snapshot can be represented as follows which is to learn useful representations in end-to-end fashion.

$$\begin{aligned} H^{(k)} &= M(\mathcal{A}, H^{(k-1)}; \theta^{(k)}) \\ &= ReLU(\tilde{\mathrm{D}}^{-0.5} \tilde{\mathcal{A}} \tilde{\mathrm{D}}^{-0.5} H^{(k-1)} W^{(k-1)}) \end{aligned} \quad (9)$$

where $\tilde{\mathcal{A}} = \mathcal{A} + I, \tilde{\mathrm{D}} = \sum_j \tilde{\mathcal{A}}_{ij}$ and $W^{(k)} \in \mathbb{R}^{d\times d}$ is the trainable weight matrix. $H^{(k)} \in \mathbb{R}^{n\times n}$ is the node embedding, which can be calculated after k steps of the GCN ($k \in [2, 6]$). The node embedding $H^{(k-1)}$ are calculated from previous step, and the input initial node embedding (iteration $k = 1$) $H^{(0)}$ is the node features on the graph, i.e., $H^{(0)} = \mathcal{F}$. The information aggregation function (as shown in equation (3)-(4)) $M(\cdot)$ is set to *ReLU* in this paper. Generally, the final output node embedding $\mathbb{Z} = H^{(K)} \in \mathbb{R}^{n\times n}$ can be calculated.

The goal of spatial-based convolution graph treatment is to find out a end-to-end differentiable strategy to stack multiple graphs $\mathcal{G}$ in a hierarchical fashion. Considering the graph $\mathcal{G}_i$ at the time i, its output and adjacency matrix are denoted by $\mathbb{Z}$ and A, respectively. Hierarchical coarsening defines a strategy to generate a new coarsened graph with weighted adjacency matrix $A' \in \mathbb{R}^{m\times m}$ and node embedding $Z' \in \mathbb{R}^{m\times 1}$, where $m < n$ is the node number. This new coarsened graph can then be used as the input to another layer, and this iteration can be iterated L times to generate a model with L layers that operates on a series of coarser and coarser versions of the input graph $\mathcal{G}$. Next work becomes the clustering or pooling together nodes using the graph output, which is not to simply cluster the nodes in one graph, but to provide a general recipe to hierarchically pool nodes across a broad set of input graphs.

3.5. Differentiable Pooling for Graph Classification

Graph classification is achieved by the typical differentiable graph pooling (DIFFPOOL) which can generates the assignment matrix $\mathcal{S}^{(l)} \in \mathbb{R}^{n_l\times n_{l+1}}$ [25]. And each row and column of $\mathcal{S}^{(l)}$ correspond to one of the n_l nodes at layer l and one of the n_{l+1} nodes at layer $l + 1$, respectively. The embedding matrices $Z^{(l)}$ at layer l can be calculated as follows, mathematically.

$$\begin{cases} Z^{(l)} = GCN_{l,embeded}(A^{(l)}, X^{(l)}) \\ \mathcal{S}^{(l)} = softmax(GCN_{l,pool}(A^{(l)}, X^{(l)})) \end{cases} \tag{10}$$

where $GCN_{l,embeded}(\cdot)$ and $GCN_{l,pool}(\cdot)$ are the standard GCN and pooling GCN at layer l, respectively. The $softmax$ function is applied in a row-wise fashion. The dimension of the output of $GCN_{l,pool}(\cdot)$ is the predefined maximum number of clusters in layer l, and it can be considered as a hyper-parameter. And the new matrix of embedding $X^{(l)}$ for each nodes in coarsened graph and the new coarsened adjacency matrix $A^{(l)}$ at layer l can be calculated as follows.

Algorithm 1 Process monitoring by using spatial-based convolution graph neural network.

Off-line Modeling

Step 1: Collect and preprocess history data, $\mathbf{V}_h$.
1.1 Calculate corresponding maximum $\mathbf{x}_{max}$ and minimum $\mathbf{x}_{min}$, and maxmin-normalize these monitoring variables.
1.2 Set a series of sample numbers n ($n = 1, 10, 20, 30$).
1.3 Get corresponding segments $\mathbf{V}_s$ with sliding window τ.

Step 2: Calculation related parameters to segments, $\mathbf{V}_s$.
2.1 Calculate corresponding mean μ and variance σ^2.
2.2 Calculate corresponding bottom and upper bounds $((\star)_{bottom}, (\star)_{upper})$, respectively.
2.3 Calculate corresponding cumulative distributions $\mathbf{S}_{(\star)}$ by equation (7) and the cross cumulative distributions $\mathbf{S}_{(\star)(\star)}$ by equation (8).

Step 3: Association graph snapshot construction, $\mathcal{G}$.
3.1 Set suitable threshold hyper-parameter T_{common}.
3.2 Calculate the adjacent matrix $\mathcal{A}$ in every snapshot by equation (6).
3.3 Set association as each node feature by $definition3$.
3.4 Label the class of this dynamic association graph snapshot to form a sample $(\mathcal{G}, \mathcal{Y})$.

Step 4: Graph snapshot treatment and graph classification.
4.1 Calculate the node embedding $H^{(k)} \in \mathbb{R}^{n\times n}$ in every snapshot by equation (9).
4.2 Differentiable pooling calculation and graph classification by equation (10)-(11).

Step 5: Collect all of relevant parameters, including n, τ, etc.
5.1 Repeat step2 step5, until all the graph snapshots.
5.2 Compare all fault detection and diagnosis results to find the optimal set of parameters.

On-line Monitoring

Step 1: Collect process online operation data, $\mathbf{V}_o$.
1.1 Maxmin-normalize these monitoring variables $\mathbf{V}_o$ with off-line calculation maximum $\mathbf{x}_{max}$ and minimum $\mathbf{x}_{min}$.
1.2 Calculate corresponding mean μ and variance σ^2.
1.3 Calculate corresponding bottom and upper bounds $((\star)_{bottom}, (\star)_{upper})$, respectively.
1.4 Calculate corresponding cumulative distributions $\mathbf{S}_{(\star)}$ by equation (7) and the cross cumulative distributions $\mathbf{S}_{(\star)(\star)}$ by equation (8).

Step 2: Online association graph snapshot construction, $\mathcal{G}$.
2.1 Calculate the adjacent matrix $\mathcal{A}$ in every snapshot by equation (6), and set each node feature in graph snapshot.
2.2 Calculate the node embedding $H^{(k)} \in \mathbb{R}^{n\times n}$ in every snapshot by equation (9).
2.3 Differentiable pooling calculation and graph classification by equation (10)-(11) to online give inference result.

$$\begin{cases} X^{(l)} = \mathcal{S}^{(l-1)^T} Z^{(l-1)} \in \mathbb{R}^{n_l\times 1} \\ A^{(l)} = \mathcal{S}^{(l-1)^T} A^{(l-1)} S^{(l-1)} \in \mathbb{R}^{n_l\times n_l} \end{cases} \tag{11}$$

The parameters of $GCN_{l,embeded}(\cdot)$ and $GCN_{l,pool}(\cdot)$ are different, and the first one is used to generate new embedding for the input nodes at this layer l, while the

Table 1: Parameter configuration details of benchmark experiment (TE chemical process).

P	$\mathbf{x}_{max}$	$\mathbf{x}_{min}$	**P**	$\mathbf{x}_{max}$	$\mathbf{x}_{min}$	**P**	$\mathbf{x}_{max}$	$\mathbf{x}_{min}$	**P**	$\mathbf{x}_{max}$	$\mathbf{x}_{min}$	**P**	$\mathbf{x}_{max}$	$\mathbf{x}_{min}$
v_1	0.3429	0.1808	v_2	3754.7	3559.5	v_3	4600.7	4419.8	v_4	9.5928	9.1286	v_5	27.529	26.280
v_6	42.925	41.659	v_7	2715.8	2690.5	v_8	76.750	73.619	v_9	120.45	120.35	v_{10}	0.3701	0.3051
v_{11}	80.679	79.423	v_{12}	53.178	47.240	v_{13}	2645.0	2619.2	v_{14}	28.660	21.877	v_{15}	52.859	46.962
v_{16}	3113.6	3089.1	v_{17}	24.693	20.812	v_{18}	66.464	64.985	v_{19}	249.82	209.37	v_{20}	344.48	337.77
v_{21}	94.952	94.284	v_{22}	78.046	76.319	v_{23}	64.436	61.296	v_{24}	55.111	52.779	v_{25}	33.682	17.781
v_{26}	65.323	57.903	v_{27}	23.313	20.765	v_{28}	44.575	35.477	v_{29}	47.453	29.977	v_{30}	53.152	39.504
v_{31}	52.124	42.162	v_{32}	42.641	39.510	v_{33}	23.144	14.009	v_{34}	32.863	31.335	v_{35}	9.1496	8.6559
v_{36}	27.181	25.768	v_{37}	7.1773	6.6293	v_{38}	19.355	17.995	v_{39}	1.7219	1.5774	v_{40}	33.779	31.953
v_{41}	14.133	13.538	v_{42}	24.818	23.140	v_{43}	1.5016	0.9956	v_{44}	19.364	17.751	v_{45}	2.3314	2.1854
v_{46}	5.0331	4.6920	v_{47}	2.4307	2.1388	v_{48}	0.0405	-0.002	v_{49}	0.8693	0.7992	v_{50}	0.1242	0.0788
v_{51}	55.057	52.582	v_{52}	44.962	42.190	*num-classes*		–	*num-gc-layers*		3	*dropout*		0.2
assign-ratio		0.1	*num-pool*		1.0	*max-nodes*		1000	*learning-rate*		0.0001	*clip*		2.0
batch-size		20	*num-epochs*		1000	*train-ratio*		0.8	*num-workers*		1	*input-dim*		20
hidden-dim		40	*output-dim*		30	threshold T_{common}		0.6	sliding window τ		1.0	sample numbers		20

second one is to generate a probabilistic assignment of the input nodes to n_l nodes. The first formula of equation (11) takes the node embedding $Z^{(l-1)}$ and aggregates these embedding according to the node assignments $\mathcal{S}^{(l-1)}$ to generate embedding for each of the n_l nodes, while the second equation takes the adjacency matrix $A^{(l-1)}$ to generates a coarsened adjacency matrix to measure the connectivity strength between each pair of nodes. At the the penultimate layer of the deep GCN, all nodes at the final layer L are assigned to a single node to generate a final embedding vector corresponding to the entire graph, which can be used as the feature vectors to feed into a differential classifier (such as soft-max regression in paper). Finally, the associations hierarchical representation method can be trained end-to-end by stochastic gradient descent, or some other methods.

3.6. Main Steps of Process Monitoring based on SGCN

Process monitoring based on SGCN in this paper mainly involves on the dynamic associations between nodes, which are the focus topics in this paper. The implementation of process monitoring method consists of two phases: off-line modeling and online monitoring, as shown in **Algorithm**. Off-line modeling phase of process monitoring based on SGCN mainly involves the configuration of relevant parameters, such as maximum $\mathbf{x}_{max}$, minimum $\mathbf{x}_{min}$, sample numbers n each time, sliding window τ, threshold hyper-parameter T_{common} and etc. Once these relevant parameters are settled down and determined, the model can be easily trained. And on-line monitoring can be easily realized by **Algorithm**.

4. Experimental studies

Benchmark of Tennessee Eastman (TE) chemical process and practical case application studies on cobalt-nickel removal from zinc solution are utilized to verify the effectiveness and applicability of this proposed process monitoring. All of the experimental studies are performed on the Python 3.6.8 software with a general workstation by sixteen CPUs (Intel(R) Xeon(R) Gold 5222 CPU at 3.80GHz 128GB RAM) and one GPU (NVIDIA GeForce RTX 2080 Ti at 11.0GB).

4.1. Benchmark experiment

4.1.1. ***Process description***

Tennessee Eastman (TE) chemical process had been regarded as a benchmark of chemical process, and it has widely employed to demonstrate the feasibility and effectiveness of some new proposed methods in related research area. This process is utilized to produce G and H liquid products under one by-product F by feeding four reactants A, D, C, E. It is worthy noting that five major operation units (reactor, product condenser, vapor-liquid separator, recycle compressor, and product stripper) are collaboratively effected each other. Associations between nodes are very obvious in this process, which is much more suitable for demonstrating this new proposed method. This process involves 52 monitoring variables (including 22 process measurement variables (v_1-v_{22}), 12 control parameters (v_{23}-v_{33}), and 19 indicator variables (v_{34}-v_{52})) which can be regarded as nodes in a graph snapshot, as shown in Fig.3(a). 21 type faults and one normal status has been simulated, which can be used to verify the feasibility and effectiveness of fault detection and diagnosis on process monitoring. More details of process description have been elaborated in [26]-[30].

Table 2: Fault diagnosis performance (*Acc*) of Case-1 (%).

method	F^4	F^9	F^{11}	Average
FDA	75.5	8.5	53.0	45.7
LFDA	57.0	25.5	48.0	43.5
SLFDA	94.0	8.5	40.5	47.7
SKLFDA	81.0	**100.0**	81.0	87.3
SGCN-based	**99.4**	98.9	**94.4**	**98.1**

Table 3: Fault diagnosis performance (*Acc*) of Case-2 (%).

method	nor.	F^2	F^6	F^7	F^{14}	Ave.
ICA+NB	95.5	95.0	75.0	**100.0**	92.5	91.6
DICA+NB	97.0	98.5	**100.0**	91.5	89.5	95.3
ICA+CDMWNB	**99.5**	94.5	**100.0**	**100.0**	95.5	97.9
DICA+CDMWNB	97.5	98.0	**100.0**	**100.0**	99.5	99.0
SGCN-based	97.5	**100.0**	**100.0**	99.9	**100.0**	**99.5**

Table 4: Fault diagnosis performance (*Acc*) of Case-3 (%).

Fault	SOM	CCA-SOM	B-LSTM	SGCN-based
normal	60.0	**100.0**	54.9	95.7
F^1	92.5	97.5	97.8	**99.6**
F^2	97.5	97.5	97.3	**99.9**
F^4	65.0	97.5	57.9	**99.5**
F^5	60.0	97.5	82.9	91.5
F^6	97.5	**100.0**	97.6	99.2
F^7	42.5	95.0	96.6	**99.3**
Average	73.6	**97.9**	83.6	**97.8**

Table 5: Fault diagnosis performance (*Acc*) of Case-4 (%).

Fault	FDA	MRDA	dynamic MRDA	SGCN-based
F^3	54.0	97.1	93.6	**98.7**
F^4	86.9	98.6	98.6	94.8
F^{11}	41.1	87.2	96.0	**97.1**
Average	60.7	94.3	96.1	**96.9**
F^1	97.0	93.8	93.0	**97.7**
F^3	57.2	74.5	86.9	**97.1**
F^4	95.0	92.5	98.0	95.3
F^{11}	53.4	43.0	79.6	**93.9**
F^{13}	41.4	94.6	81.3	84.4
F^{14}	45.8	95.9	93.9	**99.8**
Average	65.0	77.4	88.8	**94.7**

4.1.2. *Parameter configuration*

Dynamic association graph is constructed on a series of parameter configuration, such as maximum matrix $\mathbf{x}_{max}$, minimum matrix $\mathbf{x}_{min}$, sample numbers n in each step, sliding window τ, threshold hyper-parameter T_{common}, and so on. Table 1 lists the details of parameter configuration. The maximum matrix $\mathbf{x}_{max}$ and minimum matrix $\mathbf{x}_{min}$ are calculated from normal data. It should be noted that the maximum sampling frequency is $1/(15min)$ while the minimum sampling frequency is $1/(3min)$, then the sample numbers can be configured as $(15min) \times 4/(3min)$ with four times the maximum sampling frequency. As shown in Fig.3(b), train and test accuracy are following the segments length (samples number n) to increase, such as the test accuracy is increasing from 42% in $n = 5$ to 92% in $n = 25$. And this conclusion can be verified by train loss, while the hyper-parameter thresholds T_{common} also impacts the test accuracy inversely with optimal threshold $T_{common} = 0.6$. The determinations of samples number n and hyper-parameter threshold T_{common} are in line with remark.1. One normal and 21 type faults states (22 classifications) with mixing the train and test sets are considered to assemble the new sample collection, and then this new sample collection has been randomly divided into new train and test sets by split-ratio 0.8. And finally the train accuracy, test accuracy, train loss and accuracy with different thresholds T_{common} can be obtained, as shown in Fig.3(b). The reason for this assembly and re-division is to exclude the impact of the difference between original train and test samples itself which may lead to the specificity of parameter selection on subsequent experiments.

4.1.3. *Results and Comparison*

Process monitoring in this paper mainly refers to fault diagnosis to illustrate the effectiveness of this proposed method (fault detection can be regarded as two-classification task while fault diagnosis can be considered as multi-classification task). Fault diagnosis accuracy (Acc) can be regarded as the main indicator to evaluate the diagnosis performance [31]-[32]. And it can be defined as $Acc(i) = 100 \times m_i/n_i$, where $Acc(i)$ is the classification accuracy of class $F^i, i = 1, 2, ..., 21$, m_i, n_i are the number of correct classified samples and total samples in class F^i, respectively. All the results of fault diagnosis are achieved by the configuration on hyper-parameters of this proposed method, as shown in Table 1. The fault diagnosis results are obtained on employing the original train-set to train the model, while the last 750 samples of original test-set are used to calculate the accuracy. The reason of employing the last 750 samples instead of employing all samples of original test-set is to exclude the uncertainty in the transition from normal to fault states [33]. For the convenience of comparison and discussion, this paper employs three cases studies to illustrate the outstanding performance of this proposed process monitoring method.

Case-1: classification fault-4,9,11. These three faults are selected by [34] to prove the effectiveness and feasibility of complex processes fault diagnosis using sparse kernel local fisher discriminant analysis

Table 6: The definition of 8 conditions involved on practical case application.

Classes	FV1715	FV1717	FV1718	FV1719	TV1704	WICQ17141	WICQ17151	WICQ17161	WICQ17171	WICQ17181
Con^1	23.2-25.2	68.4-75.5	46.4-51.5	29.2-33.0	40.0	120.0	100.0	80.0	61.0	25.0
Con^2	22.7-23.5	42.4-46.5	25.5-28.5	24.8-28.7	50.0	97.5	90.0	85.3	missing	25.0
Con^3	22.8-23.5	41.4-48.1	33.8-38.4	30.0-28.7	50.0	105.0	90.0	85.3	missing	30.0
Con^4	26.2-28.2	36.9-46.1	34.1-36.0	21.7-23.7	35.5	130.0	125.0	80.0	60.0	30.0
Con^5	25.0-26.8	34.9-45.3	34.9-37.4	25.7-27.9	40.0	130.0	100.0	80.0	60.0	28.25
Con^6	22.6-24.1	34.9-41.3	38.3-42.2	29.8-35.6	45.0	115.0	110.0	80.0	60.0	27.0
Con^7	23.1-24.6	41.1-49.9	52.3-57.2	34.5-37.4	45.0	145.0	122.5	80.0	60.0	27.0
Con^8	22.5-23.8	42.6-47.7	22.5-39.3	28.1-33.1	52.5	100.0	95.0	80.0	57.75	27.0

(SKLFDA). This paper employs their related methods to illustrate the advancement of our proposed methods, such as FDA (fisher discriminant analysis), LFDA (local fisher discriminant analysis), SLFDA (sparse local fisher discriminant analysis) and SKLFDA (results are from [34]). The average fault diagnosis accuracy of SGCN-based in Table 2 is better than other four methods, and this result indicates that this SGCN-based method can be used to diagnose faults by achieving considerable accuracy. Especially, the diagnosis accuracy of fault-9 can be obtained 98.9% which is equal to that using SKLFDA.

Case-2: classification normal and fault-2,6,7,14. These four type faults are selected by [35] to prove the effectiveness and feasibility of class-specific distributed monitoring weighted naïve Bayes(CDMWNB) method. This paper employs the ICA+NB (independent component analysis and naïve Bayes), DICA+NB (dynamic independent component analysis and naïve Bayes), ICA+CDMWNB and DICA+CDMWNB (results are from [35]) to illustrate the advancement of our proposed methods. The diagnosis accuracy of five classes is shown in Table 3. The average fault diagnosis accuracy of SGCN-based in Table 3 is better than other methods, and this result indicates that this SGCN-based process monitoring method can extract more useful characteristics to obtain the state-of-art performance among these process monitoring models.

Case-3: classification normal and fault-1,2,4,5,6,7. Fault-1,2,4,5,6,7 are all step-type ones which are considered by [36]-[37] to demonstrate the good performance of their fault diagnosis methods. Similarly, this paper employs the SOM (self-organizing map, results are from [37]), CCA-SOM (canonical correlation analysis and self-organizing map, results are from [37]) and B-LSTM (Bidirectional Long Short Term Memory, results are from [36]) to demonstrate the effectiveness of this SGCN-based fault diagnosis, and the diagnostic results of 7 classes are presented in Table 4. Most diagnosis accuracy of these 7 classes can be reached to above 99.2% except normal and fault-5. All of these results indicate that this SGCN-based method can be used to diagnose faults with superiority accuracy.

Case-4: performance when classes increasing. Practical industrial processes exist various fault types, it is necessary to analyze the fault diagnosis performance when fault classes are increasing. This problem is also handled by [38]. The fault classes are increasing from fault-3,4,11 (3 classes) to fault-1,3,4,11,13,14 (6 classes), and the methods adopted consist of FDA (fisher discriminant analysis), MRDA (maximized ratio divergence analysis), dynamic MRDA. All the diagnosis accuracy results are from [38]. The diagnostic results are shown in Table 5. The average fault diagnosis accuracies of SGCN-based in case-4 are 96.9% on 3 classes instance and 94.7% on 6 classes instance, which is better than other methods in both 3 classes and 6 classes. Experimental results in Table 5 indicate that this SGCN-based method can be used to achieve superiority accuracy with hardly changes when fault classes increasing.

4.2. Practical case application

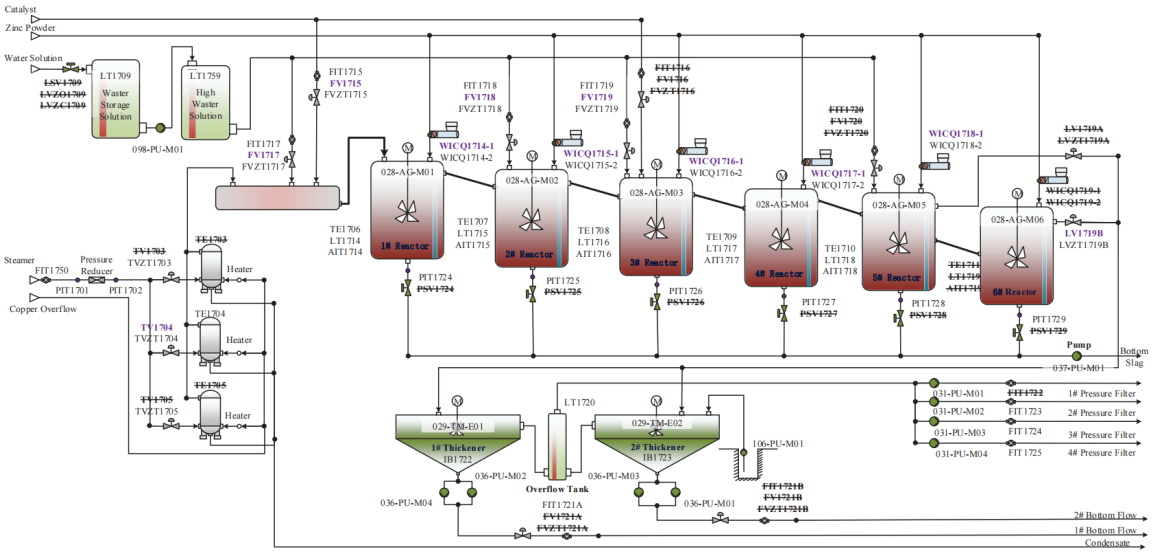
*4.2.1. **Process description***

Zinc is one of the most used non-ferrous metal, and zinc hydrometallurgy accounts for more than 85% zinc production. Cobalt-Nickel removal from zinc solution is to add the impurity removal agent (zinc elemental powder) and catalyst to reactors, and then complex redox reaction happens with impurity metal ions, and finally alloy or metal compound precipitation can be formed to gradually reduce the concentration of impurity ions until make it tend to the range of technical indicators [39], as shown in Fig.4. On this application study, 52 variables are selected to describe the system operation states, include pressure, temperature, feedback of valve open degree, etc. In fact, 11 control variables (purple words in Fig.4) are employed to change the process conditions correspondingly to meet the practical production demands. It should be noted that although some other control and monitor variables had also been designed, they had not been employed in practical process control (no corresponding monitors data had been

Table 7: Parameter configuration details of benchmark experiment (Practical case application).

P	range	P	range	P	range	P	range
TE1704	[0,120]^oC	TE1706	[0,120]^oC	TE1707	[0,120]^oC	TE1708	[0,120]^oC
TE1709	[0,120]^oC	TE1710	[0,120]^oC	TE1711	[0,120]^oC	PIT1701	[0,1.2]MPa
PIT1702	[0,0.6]MPa	PIT1724	[0,0.4]MPa	PIT1725	[0,0.4]MPa	PIT1726	[0,0.4]MPa
PIT1727	[0,0.4]MPa	PIT1728	[0,0.4]MPa	FIT1715	[0,8]m^3/h	FIT1717	[0,15]m^3/h
FIT1718	[0,15]m^3/h	FIT1719	[0,15]m^3/h	FIT1721A	[0,150]m^3/h	FIT1723	[0,250]m^3/h
FIT1724	[0,250]m^3/h	FIT1725	[0,250]m^3/h	FIT1726	[0,70]m^3/h	FIT1750	[0,20]m^3/h
LT1709	[0,4]m	LT1714	[0,11.25]m	LT1715	[0,11.25]m	LT1716	[0,11.25]m
LT1717	[0,11.25]m	LT1718	[0,11.25]m	LT1720	[0,8]m	LT1759	[0,2]m
AIT1714	-[700,400]mv	AIT1715	-[700,400]mv	AIT1716	-[700,400]mv	AIT1717	-[700,400]mv
AIT1718	-[700,400]mv	TVZT1703	[0,100]%	TVZT1704	[0,100]%	TVZT1705	[0,100]%
FVZT1717	[0,100]%	FVZT1718	[0,100]%	FVZT1719	[0,100]%	LVZT1719B	[0,100]%
WICQ17142	[0.1,1]t/h	WICQ17152	[0.1,1]t/h	WICQ17162	[0.1,1]t/h	WICQ17172	[0.1,1]t/h
WICQ17182	[0.1,1]t/h	*num-classes*	8	*num-gc-layers*	3	*dropout*	0.2
assign-ratio	0.1	*num-pool*	1.0	*max-nodes*	1000	*learning-rate*	0.0001
clip	2.0	*batch-size*	20	*num-epochs*	1000	*num-workers*	1
input-dim	20	*hidden-dim*	40	*output-dim*	30	threshold T_{common}	0.8
sliding window τ	10s	sample numbers	4				

(a) Practical production site of Cobalt-Nickel removal from zinc solution



(b) The schematic diagram of Cobalt-Nickel removal process

Figure 4: The process description of practical Cobalt-Nickel removal from zinc solution.

obtained) and these variables have not been considered in some application experiment. According to these 11 control variables (the value of LV1719B is always slightly changed, so it is not listed in Table 6), 8 conditions are defined and identified to demonstrate the effectiveness and applicability of this proposed SGCN-based process monitoring method, as shown in Table 6.

4.2.2. *Parameter configuration*

Due to the different conditions of practical process on, the numerous parameter configurations are different to obtained by maximum and minimum. Under these circumstances, the Min-Max Normalization of monitor variables is realized by employing their measurement ranges, as shown in Table 7. It should be noted that the maximum sampling frequency is 10 seconds (all monitoring variables), and then the sample numbers can be configured as $(10sec) \times 4/(10sec)$ with four times the maximum sampling frequency (it has been proved in last experiment). 8 conditions (8 classes) are employed to handled this demonstration by about 400 samples of test-set and more than 700 samples of train-set. The details of parameter configuration is in Table 7.

Table 8: Conditions classification performance (*Acc*) of practical case application (%).

Class	LDA	KNN	LR	DT	NB	SGCN-based
Con^1	100.0	100.0	100.0	100.0	100.0	99.24
Con^2	100.0	95.75	99.75	99.75	95.25	98.23
Con^3	100.0	100.0	100.0	100.0	100.0	99.75
Con^4	100.0	100.0	100.0	98.75	100.0	99.49
Con^5	100.0	99.25	100.0	100.0	100.0	99.24
Con^6	100.0	67.50	68.00	99.75	100.0	81.06
Con^7	100.0	97.00	100.0	100.0	100.0	98.74
Con^8	70.75	93.50	70.75	42.25	70.50	99.75
Average	96.34	94.13	92.31	92.56	95.72	96.94

4.2.3. *Results and Comparison*

Process monitoring in this practical case application is mainly employed to consider conditions classification (multi-classification task), and this conditions classification study is the first step for many process safety and control. The effectiveness and applicability of this proposed method can be demonstrated by the results of conditions classification. Similarly, classification accuracy (Acc) is the main indicator to evaluate the performance of this proposed method. The configuration on hyper-parameters of this proposed method are shown in Table 7.

For the convenience of comparison and discussion, five related methods are employed to illustrate the advancement of the proposed method, such as LDA (linear discriminant analysis), KNN (k-nearest neighbors), LR (logistic regression classifier), DT (decision tree classifier) and NB (naïve bayes classifier), and the best results are shown in Table 8. Most accuracies of conditions classification can reach to more than 98% except condition-6 (Con^6). However, this lowest classification accuracy (81.06%) of Con^6 is still better than those of KNN (67.50%) and LR (68.00%), which means that some inaccessible features cannot be described in this spatial associations between numerous variables. According to the consideration of equivalent performance within 1.0%, the performances of this proposed method for Con^1, Con^2, Con^3, Con^4, Con^5, Con^7 are as good as other five methods. It is worthy noted that the accuracy of this method for Con^8 reaches to 99.75% which violently exceeds the corresponding performances of other methods. Finally, the average condition classification accuracy of SGCN-based method (96.94%) in Table 8 is better than other methods (96.34%, 94.13%, 92.31%, 92.56%, 95.72%). All the results indicate that this proposed SGCN-based method can be used to extract effective associations between monitor variables to cover the defect that traditional methods not paying enough consideration, and these useful dynamic associations can be used to achieve the state-of-art performance among these selected process monitoring methods.

5. Conclusion

This paper represents a new process monitoring method based on spatial-based graph convolution neural network to describe the dynamic associations between monitoring variables, which transform the process monitoring problem into multi-tasks classification. The dynamic associations mining and its application on process monitoring can be regarded as the main contributions. The highlights of this paper can be addressed as four issues: (i)SGCN-based process monitoring is proposed to describe the system status change, and this can be used as a basis to broaden the research area; (ii) dynamic associations between monitoring are auto- and well-considered to enhance the monitoring performance; (iii) information of node itself and its neighbors is performed during the calculation of graph convolutional network to reduce the disturbance or noise impact; and (iv) the effectiveness and applicability of SGCN-based method are demonstrated by benchmark and practical-case-application.

Staying awake will promote our research advance. Although this proposed method can be employed to considered the dynamic associations to achieve better performance, there are still some problems need to be solved, such as the specific and detailed mathematical

theory of this graph-based process monitoring. The false alarm rates should also pay more attention, and this problem has not been deeply studied. Overall, future works need to promote the deep integration of industrial intelligence and the real economy on the basis of the full-processes monitoring of typical process industries.

Acknowledgment

This work was supported in part by The Major Key Project of PCL under Grants PCL2021A09, and it was also supported in part by the National Natural Science Foundation of China under Grants 62103207, 61988101 and 62173349.

References

References

[1] H. Luo, H. Zhao, S. Yin, "Data-Driven Design of Fog-Computing-Aided Process Monitoring System for Large-Scale Industrial Processes," *IEEE T. IND. INFORM.*, vol. 14, no. 10, pp. 4631-4641, Oct. 2018, doi: 10.1109/TII.2018.2843124.

[2] Z. W. Chen, K. T. Liang, Steven X. Ding, *et al.*, "A Comparative Study of Deep Neural Network-Aided Canonical Correlation Analysis-Based Process Monitoring and Fault Detection Methods," *IEEE T. NEUR. NET. LEAR.*, Early Access, pp. 1-15, Mar. 2021, doi: 10.1109/TNNLS.2021.3072491.

[3] K. K. Huang, Y. M. Wu, C. Wang, *et al.*, "A Projective and Discriminative Dictionary Learning for High-Dimensional Process Monitoring With Industrial Applications," *IEEE T. IND. INFORM.*, vol. 17, no. 1, pp. 558-568, Jan. 2021, doi: 10.1109/TII.2020.2992728.

[4] D. Jung, C. sundström, "A Combined Data-Driven and Model-Based Residual Selection Algorithm for Fault Detection and Isolation," *IEEE T. CONTR. SYST. T.*, vol. 27, no. 2, pp. 616-630, Mar. 2019, doi: 10.1109/TCST.2017.2773514.

[5] X. L. Wu, H. G. Han, Z. Liu, *et al.*, "Data-Knowledge-Based Fuzzy Neural Network for Nonlinear System Identification," *IEEE T. FUZZY. SYST.*, vol. 28, no. 9, pp. 2209-2221, Sept. 2020, doi: 10.1109/TFUZZ.2019.2931870.

[6] M. C. Thomas, W. B. Zhu, J. A. Romagnoli, "Data mining and clustering in chemical process databases for monitoring and knowledge discovery," *J. PROCESS. CONTR.*, vol. 67, pp. 160-175, Feb. 2017, doi: 10.1016/j.jprocont.2017.02.006.

[7] L. J. Luo, S. Y. Bao, "Knowledge-data-integrated sparse modeling for batch process monitoring," *CHEM. ENG. SCI.*, vol. 189, pp. 221-232, May. 2018, doi: 10.1016/j.ces.2018.05.055.

[8] Y. Zhuo, Z. Q. Ge, "Gaussian Discriminative Analysis aided GAN for imbalanced big data augmentation and fault classification," *J. PROCESS. CONTR.*, vol. 92, pp. 271-287, Jul. 2020, doi: 10.1016/j.jprocont.2020.06.014.

[9] Y. Zhuo, Z. Q. Ge, "Auxiliary Information-Guided Industrial Data Augmentation for Any-Shot Fault Learning and Diagnosis," *IEEE T. IND. INFORM.*, vol. 17, no. 11, pp. 7535-7545, Jan. 2021, doi: 10.1109/TII.2021.3053106.

[10] L. Yao, Z. Q. Ge, "Industrial Big Data Modeling and Monitoring Framework for Plant-Wide Processes," *IEEE T. IND. INFORM.*, vol. 17, no. 9, pp. 6399-6408, Jul. 2020, doi: 10.1109/TII.2020.3010562.

[11] K. K. Huang, S. Li, P. L. Dai, *et al.*, "SDARE: A stacked denoising autoencoder method for game dynamics network structure reconstruction," *Neural Networks*, vol. 126, pp. 143-152, Mar. 2020, doi: 10.1016/j.neunet.2020.03.008.

[12] L. Q. Zeng, Z. Q. Ge, "Bayesian network for dynamic variable structure learning and transfer modeling of probabilistic soft sensor," *J. PROCESS. CONTR.*, vol. 100, pp. 20-29, Mar. 2021, doi: 10.1016/j.jprocont.2021.02.004.

[13] W. L. Liao, D. C. Yang, Y. S. Wang, *et al.*, "Fault Diagnosis of Power Transformers Using Graph Convolutional Network," *CSEE. J. POWER. ENERGY.*, vol. 7, no. 2, pp. 241-249, Mar. 2021, doi: 10.17775/CSEEJPES.2020.04120.

[14] M. S. Tootooni, P. K. Rao, C. Chou, *et al.*, "A Spectral Graph Theoretic Approach for Monitoring Multivariate Time Series Data From Complex Dynamical Processes," *IEEE T. AUTOM. SCI. ENG.*, vol. 15, no. 1, pp. 127-144, Jan. 2018, doi: 10.1109/TASE.2016.2598094.

[15] K. Chen, J. Hu, Y. Zhang, *et al.*, "Fault Location in Power Distribution Systems via Deep Graph Convolutional Networks," *IEEE J. SEL. AREA. COMM.*, vol. 38, no. 1, pp. 119-131, Jan. 2020, doi: 10.1109/JSAC.2019.2951964.

[16] Z. W. Chen, J. M. Xu, T. Peng, *et al.*, "Graph Convolutional Network-Based Method for Fault Diagnosis Using a Hybrid of Measurement and Prior Knowledge," *IEEE T. CYBERNETICS.*, Early Access, pp. 1-13, MaR. 2021, doi: 10.1109/TCYB.2021.3059002.

[17] D. C. Zhang, E. Stewart, Y. L. Wang, "A spatial-temporal LW-PLS for adaptive soft sensor modeling and its application for an industrial hydrocracking process," *MEASUREMENT*, vol. 156, 107585, Feb. 2020, doi: 10.1016/j.measurement.2020.107585.

[18] X. F. Yuan, J. Zhou, M. Entezami, *et al.*, "Intelligent acoustic-based fault diagnosis of roller bearings using a deep graph convolutional network," *CHEMOMETR. INTELL. LAB.*, vol. 197, 103921, Jan. 2020, doi: 10.1016/j.chemolab.2019.103921.

[19] X. F. Yuan, L. Li, K. Wang, *et al.*, "Sampling-Interval-Aware LSTM for Industrial Process Soft Sensing of Dynamic Time Sequences With Irregular Sampling Measurements," *IEEE SENS. J.*, vol. 21, no. 9, pp. 10787-10795,, May. 2021, doi: 10.1109/JSEN.2021.3056210.

[20] G. J. Chen, Z. Q. Ge, "Hierarchical Bayesian Network Modeling Framework for Large-Scale Process Monitoring and Decision Making," *IEEE T. CONTR. SYST. T.*, vol. 28, no. 2, pp. 671-679, Mar. 2020, doi: 10.1109/TCST.2018.2882562.

[21] D. Tomasz, S. Przemysław, T. Jacek, *et al.*, "Spatial Graph Convolutional Networks," *cs.LG*, Jul. 2020, doi: arXiv:1909.05310v2.

[22] Z. J. Ma, G. Mei, E. Prezioso, *et al.*, "A deep learning approach using graph convolutional networks for slope deformation prediction based on time-series displacement data," *NEURAL. COMPUT. APPL.*, May. 2021, doi: 10.1007/s00521-021-06084-6.

[23] S. Khatib, P. Daoutidis, "Application of graph theory and filter based variable selection methods in the design of a distributed data-driven monitoring system," *CHEM. ENG. SCI.*, vol. 143, Sept. 2020, doi: 10.1016/j.compchemeng.2020.107098.

[24] J. T. Liu, F. Schmid, K. P. Li, *et al.*, "A knowledge graph-based approach for exploring railway operational accidents," *RELIAB. ENG. SYST. SAFE.*, vol. 207, 107352, Mar. 2021, doi: 10.1016/j.ress.2020.107352.

[25] R. Ying, J. You, C. Morris, *et al.*, "Hierarchical Graph Representation Learning with Differentiable Pooling," presented at the Proceedings of the *32nd International Conference on Neural Information Processing Systems (NeurIPS 2018),* Montréal, Canada, 4805-4815, 2018.

[26] P. Hajihosseini, M. M. Anzehaee, B. Behnam, "Fault detection

and isolation in the challenging Tennessee Eastman process by using image processing techniques," *ISA T.*, vol. 79, pp. 137-146, Aug. 2018, doi: 10.1016/j.isatra.2018.05.002.

[27] Y. L. Wang, Z. F. Pan, X. F. Yuan, *et al.*, "A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network," *ISA T.*, vol. 96, pp. 457-467, Jan. 2020, doi: 10.1016/j.isatra.2019.07.001.

[28] Y. Tian, H. Yao, Z. Q. Li, "Plant-wide process monitoring by using weighted copula-correlation based multiblock principal component analysis approach and online-horizon Bayesian method," *ISA T.*, vol. 96, pp. 24-36, Jan. 2020, doi: 10.1016/j.isatra.2019.06.002.

[29] L. Q. Zeng, Z. Q. Ge, "Improved Population-Based Incremental Learning of Bayesian Networks with partly known structure and parallel computing," *ENG. APPL. ARTIF. INTEL.*, vol. 95, 103920, Sep. 2020, doi: 10.1016/j.engappai.2020.103920.

[30] Z. Y. Yang, Z. Q. Ge, "Monitoring and prediction of big process data with deep latent variable models and parallel computing," *J. PROCESS. CONTR.*, vol. 92, pp. 19-34, May. 2020, doi: 10.1016/j.jprocont.2020.05.010.

[31] Y. Liu, J. S. Zeng, L. Xie, *et al.*, "An improved mixture robust probabilistic linear discriminant analyzer for fault classification," *ISA T.*, vol. 98, pp. 227-236, Aug. 2019, doi: 10.1016/j.isatra.2019.08.037.

[32] H. T. Chen, Z. W. Chen, Z. Chai, *et al.*, "A Single-Side Neural Network-Aided Canonical Correlation Analysis With Applications to Fault Diagnosis," *IEEE T. CYBERNETICS.*, Early Access, pp. 1-13, Feb. 2021, doi: 10.1109/TCYB.2021.3060766.

[33] Z. P. Gao, M. X. Jia, Z. Z. Mao, *et al.*, "Transitional phase modeling and monitoring with respect to the effect of its neighboring phases," *CHEM. ENG. RES. DES.*, vol. 145, pp. 288-299, Mar. 2019, doi: 10.1016/j.cherd.2019.03.023.

[34] K. Zhong, M. Han, T. Qiu, *et al.*, "Fault Diagnosis of Complex Processes Using Sparse Kernel Local Fisher Discriminant Analysis," *IEEE T. NEUR. NET. LEAR.*, vol. 31, no.5, pp. 1581-1591, May. 2020, doi: 10.1109/TNNLS.2019.2920903.

[35] Y. L. Wang, Y. C. Ma, Y. Xu, *et al.*, "Fault Diagnosis Using Novel Class-Specific Distributed Monitoring Weighted Naıve Bayes: Applications to Process Industry," *IND. ENG. CHEM. RES.*, vol. 59, pp. 9593-9603, Apr. 2020, doi: 10.1021/acs.iecr.0c01071.

[36] G. S. Chadha, A. Panambilly, A. Schwung, *et al.*, "Bidirectional deep recurrent neural networks for process fault classification," *ISA T.*, vol. 106, pp. 330-342, Nov. 2020, doi: 10.1016/j.isatra.2020.07.011.

[37] Z. C. Li, L. Tian, Q. C. Jiang, *et al.*, "Fault Diagnostic Method Based on Deep Learning and Multimodel Feature Fusion for Complex Industrial Processes," *IND. ENG. CHEM. RES.*, vol. 59, pp. 18061-18069, Sep. 2020, doi: 10.1021/acs.iecr.0c03082.

[38] Q. G. Lu, B. B. Jiang, E. Harinath, "Fault Diagnosis in Industrial Processes by Maximizing Pairwise Kullback–Leibler Divergence," *IEEE T. CONTR. SYST. T.*, vol. 29, no.2, pp. 780-785, Mar. 2021, doi: 10.1109/TCST.2019.2950403.

[39] B. Sun, C. H. Yang, H. Q. Zhu, *et al.*, "Modeling, Optimization, and Control of Solution Purification Process in Zinc Hydrometallurgy," *IEEE-CAA J. AUTOMATIC.*, vol. 5, no. 2, pp. 564-576, Mar. 2018, 10.1109/JAS.2017.7510844.