Dynamic Graph-Based Adaptive Learning for Online Industrial Soft Sensor With Mutable Spatial Coupling Relations

Kun Zhu and Chunhui Zhao, *Senior Member, IEEE*

***Abstract*—The accurate online soft sensor in complex industrial processes remains challenging because underlying spatial coupling relations among process variables have not been effectively mined and exploited. Recently, some deep learning based studies construct static graphs to explicitly represent underlying spatial coupling relations among process variables, but they neglect the fact that spatial coupling relations have mutable characteristics, leading to poor performance in the online soft sensor. Therefore, in this article, we propose a novel deep learning model to address this issue to achieve accurate soft sensors. Specifically, a dynamic graph is proposed to realize adaptive learning and automatic inference for mutable spatial coupling relations so that the proposed model is endowed with the ability to real-timely sense spatial coupling relations in the online industrial soft sensor. Then, based on the dynamic graph, a new multihop attention graph convolutional network is proposed to systematically aggregate various crucial node feature representations during graph convolution processes to capture fine-grained spatial dependence features, thereby achieving effective modeling for variation patterns of process variables. Finally, a new multivariate incremental training algorithm is designed for deep learning models to further improve the prediction performance. The verification study on a coal mill rig demonstrates the feasibility and effectiveness of the proposed model.**

***Index Terms*—Adaptive learning, dynamic graph, multihop attention graph convolutional network (MAGCN), mutable spatial coupling relations, online industrial soft sensor.**

I. Introduction

NOWADAYS, industrial processes have become increasingly complex due to the diversity of process variables and the increased requirements of product quality [30]. To conduct effective process monitoring, process optimization, and quality control, key quality variables should be measured as timely and accurately as possible. However, the quality variables are generally hard to measure online because of remarkable measurement delays or excessive costs [1]. Therefore, a soft sensor, with easy-to-measure process variables as inputs and difficult-to-measure quality variables as outputs, has been developed to construct prediction models to rapidly estimate quality variables [2].

The first principle models [9], [16] have been studied early to implement soft sensor, but it usually requires expert knowledge or wealthy experience, which cannot be satisfied easily because of the increasing complexity of industrial processes [32]. Compared with the first principle models, the data-driven models mainly rely on massive historical data [19], thereby gaining increasing attention during the past decades [25], [34]. The data-driven models can be divided into two categories: traditional machine learning models and deep learning models. The traditional machine learning models include principal component regression [22], partial least squares (PLS) [8], least squares support vector regression (LSSVR), and extreme learning machine (ELM) [10], which can mine valuable nonlinear feature representations from historical data [18].

However, these traditional machine learning models can only extract superficial feature representations, which is inadequate for the accurate soft sensor in complex industrial processes. Therefore, some studies have developed deep learning models to more effectively model complex industrial processes because deep learning models possess strong learning and nonlinear fitting capabilities [17] and can generate deep and abstract feature representations. Recently, various deep learning models have been applied successfully to soft sensor developments and realized promising prediction results. Specifically, Wang et al. [27] proposed a deep belief network (DBN) with an event-triggered learning strategy to improve the efficiency and accuracy of the soft sensor model in the wastewater treatment process. Ke et al. [15] introduced long short-term memory (LSTM) model to effectively handle strong nonlinearity and dynamics of complex industrial processes. Yuan et al. [33] designed a convolutional neural network (CNN) to learn local dynamic feature representations and presented powerful performance.

In practical industrial processes, process variables are actually graph-structured data with non-Euclidean nature [14], [31], which possess underlying and complicated spatial

Manuscript received 30 July 2022; revised 26 September 2022; accepted 12 October 2022. Date of publication 25 October 2022; date of current version 3 April 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62125306, in part by the National Key R&D Program of China under Grant 2021YFC2902701, in part by the Research Project of the State Key Laboratory of Industrial Control Technology, Zhejiang University, China, under Grant ICT2021A15, and in part by the Fundamental Research Funds for the Central Universities Zhejiang University NGICS Platform. *(Corresponding author: Chunhui Zhao.)*

The authors are with the College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China,and also with the Zhejiang University NGICS Platform, Hangzhou 310027, China (e-mail: kunzhu@zju.edu.cn; chhzhao@zju.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIE.2022.3215448.

Digital Object Identifier 10.1109/TIE.2022.3215448

coupling relations that can influence variation patterns of process variables and increase the modeling difficulty of soft sensor [24]. Therefore, it is necessary to effectively capture underlying spatial coupling relations among process variables for achieving the accurate soft sensor in industrial processes. However, the aforementioned deep learning models (e.g., DBN, CNN, and LSTM) ignore the non-Euclidean nature of underlying spatial coupling relations among process variables. They fail to provide specific representations for these spatial coupling relations and directly encapsulate them into abstract features, which hinder the extraction of hidden spatial dependence features.

As the graph convolutional network (GCN) has yielded growing impacts in other fields (Zhu et al., 2021; [3]), some studies have introduced GCN into the soft sensor to explicitly represent underlying spatial coupling relations among process variables. For example, Jia et al. [14] constructed a graph to explicitly express the mutual influence between process variables and achieved excellent prediction results in chemical processes. Huang et al. [13] introduced the self-attention mechanism to learn latent correlations between sensors and leveraged it to generate a graph to reflect the sensor network topology. Wang et al. [28] proposed a spatiotemporal GCN based on a learnable graph and attention mechanism to capture the spatial features of anaerobic digestion processes. Although these GCN-based models consider the non-Euclidean nature of spatial coupling relations among process variables, they only construct the static graphs and ignore the mutable characteristics of spatial coupling relations, making it difficult to effectively model the variation patterns of process variables and leading to unsatisfactory prediction results in the online soft sensor. To the authors' knowledge, mutable characteristics of spatial coupling relations among process variables have not been explored in soft sensor developments. In addition, when deep learning models need to predict multiple quality variables, the conventional training strategy usually uses all quality variables together as the prediction target to train the model parameters, which may make the model unable to find a suitable training starting point and, thus, degrade the prediction performance.

Therefore, a new deep learning model based on the dynamic graph (DGDL) is proposed in this study to overcome the above issues. First, a new dynamic graph is proposed to generalize soft sensor into the non-Euclidean space to explicitly quantify underlying spatial coupling relations among process variables. Moreover, this dynamic graph solves the problem that the static graph only describes fixed spatial coupling relations because it can adaptively learn and automatically infer mutable spatial coupling relations so as to enable DGDL to real-timely sense spatial coupling relations in the online industrial soft sensor. Second, after nonlinear temporal dependence features extraction through a multiscale gated temporal convolutional network (MGTCN), a new multihop attention graph convolutional network (MAGCN) based on the dynamic graph is proposed to systematically aggregate the most important node feature representations from different hops to capture fine-grained spatial dependence features. Then, a new multivariate incremental training (MIT) algorithm is designed to help deep learning models that possess a good training starting point and then gradually learn to predict multiple quality variables simultaneously. Additionally, when we need to predict new emerging quality variables in practical industrial processes, MIT can help deep learning models that use previously well-trained model parameters as the basis and fine tune the model parameters rather than training from scratch, thereby significantly reducing the time and resource consuming.

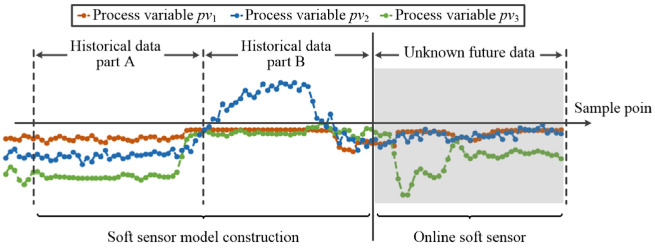

Fig. 1. Variation patterns of three process variables in historical data and unknown future data.

The main contributions of this study are summarized as follows.

1) A new dynamic graph is proposed to endow the proposed model the ability to real-timely sense spatial coupling relations in the online industrial soft sensor. It is the first time that the mutable characteristics of spatial coupling relations are meticulously explored and adaptively learned.
2) A new MAGCN is proposed. Different from the existing GCN-based models that only focus on node feature representations at the last hop, MAGCN can comprehensively consider crucial node feature representations hidden in different hops to mine fine-grained spatial dependence features, thereby more effectively capturing the variation patterns of process variables.
3) A new MIT algorithm is designed for deep learning models to find a good training starting point and learn the difficult soft sensor task (i.e., predicting multiple quality variables together) in a gradual manner, which can enhance prediction performance and help to predict new emerging quality variables efficiently and quickly.

II. Motivation and Problem Statement

A. Motivation of Dynamic Graph

The realization of the soft sensor task is divided into two stages: construction of a soft sensor model using historical data and application of this model for the online soft sensor. The existing GCN-based studies aim to generate fixed representations for spatial coupling relations among process variables from historical data to construct static graph-based models and directly apply these models to the online soft sensor. However, influenced by complex internal and external factors (e.g., operating condition changes, unmeasured interferences, and seasonal variations), spatial coupling relations among process variables possess mutable characteristics (i.e., it changes with time), which cannot be correctly reflected through fixed representations in static graphs.

For example, as shown in Fig. 1, it is assumed that the historical data part A and part B of process variables are used for soft sensor model construction, and unknown future data of process variables are used as input data for online soft sensor.

It can be seen that the variation patterns of process variables $pv1$ and $pv2$ are similar in historical data part A, indicating that they have strong spatial coupling relations. In contrast, their variation patterns are different in historical data part B, indicating that they have weak spatial coupling relations. This phenomenon illustrates that spatial coupling relations among process variables change with time. In addition, according to the variation patterns of process variables $pv1$ and $pv3$, it can be seen that $pv1$ and $pv3$ have strong coupling relations in all historical data (part A and part B) but weak coupling relations in unknown future data. This phenomenon shows that simply using fixed spatial coupling relations to conduct online industrial soft sensor is biased because spatial coupling relations have mutable characteristics, which may make spatial coupling relations in historical data inconsistent with those in unknown future data.

Therefore, to overcome the disadvantage of static graphs, we propose a new dynamic graph to realize adaptive learning and automatic inference for mutable spatial coupling relations among process variables so that the proposed DGDL can real-timely sense spatial coupling relations in online industrial soft sensors, thereby significantly improving soft sensor results.

B. Problem Statement

In this study, we model soft sensor in complex industrial processes from a graph-based perspective. Specifically, we consider process variables as nodes in the graph and describe spatial coupling relations among process variables by the graph adjacency matrix. To further formalize the graph-based soft sensor problem, some key concepts are defined.

Definition 1: The topology network of process variables is regarded as a graph $G = (V, E, A)$, where V denotes all the graph nodes, indicating N process variables; E is a set of edges, indicating spatial coupling relations among N process variables; $A \in \mathbb{R}^{N\times N}$ denotes the graph adjacency matrix that is a mathematical representation to save the edge weights.

Definition 2: The values of all process variables on G at time point t can be denoted as a graph signal $X_t \in \mathbb{R}^N$. Then, graph signals at T historical time points can be denoted as a graph signal set $X_{(t-T+1):t} \in \mathbb{R}^{N\times T}$, which includes T graph signals $X_{t-T+1}, X_{t-T+2}, \ldots, X_t$. Finally, the goal of the graph-based soft sensor problem is to predict the values of all quality variables at time point t based on G and the graph signal set $X_{(t-T+1):t}$, as shown in the following:

$$\left[X_{(t-T+1):t}; G\right] \xrightarrow{\mathbb{F}} \hat{Y}_t \tag{1}$$

where $\hat{Y}_t \in \mathbb{R}^M$ denotes the predicted values of all quality variables at time point t in which M is the number of quality variables; and $\mathbb{F}$ denotes the mapping function.

III. Methodology

A. Overview of the Proposed DGDL

Fig. 2(a) illustrates the framework of the proposed DGDL, consisting of two main components: spatiotemporal joint learning block (ST-block) and refining module. In the whole proposed model, K ST-blocks are stacked and each ST-block contains two architectures: MGTCN and MAGCN. The corresponding architectures of MGTCN and MAGCN are shown in Fig. 2(b) and (c), respectively. For the MGTCN, multiscale convolution kernels (i.e., a set of convolution kernels with multiple receptive fields) and a gated mechanism are adopted to comprehensively extract nonlinear temporal dependence features. For the MAGCN, two dynamic graphs with opposite directions are automatically updated through model training to realize adaptive learning for mutable characteristics of spatial coupling relations among process variables. Then, the dynamic graph is input into multihop attention aggregation structures to capture fine-grained spatial dependence features. To speed up the convergence rate and solve the gradient vanishing problem, the residual connection is added from the input of MGTCN to the output of MAGCN. Then, the outputs of all MAGCNs are fused via the skip connection to generate the high-level feature representations. Finally, the refining module containing two CNNs and a fully connected layer is constructed to refine high-level feature representations and obtain final prediction results of all quality variables.

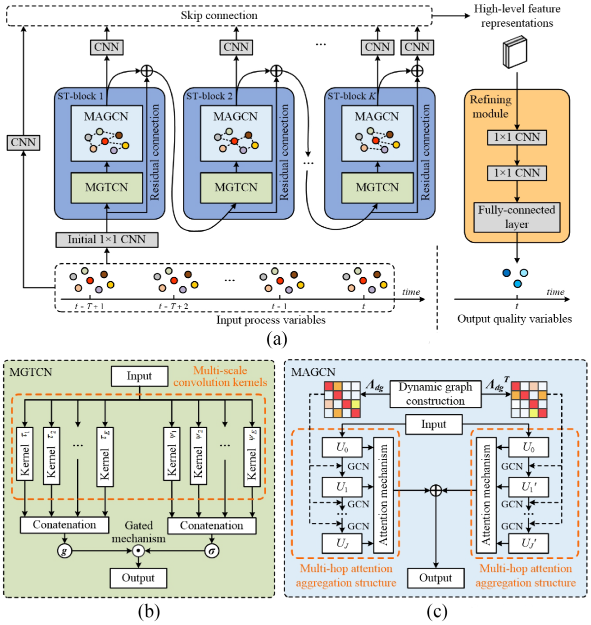

Fig. 2. Overview of the proposed DGDL. (a) Framework of the proposed DGDL. (b) Architecture of MGTCN. (c) Architecture of MAGCN.

B. Multiscale Gated Temporal Convolutional Network

Recently, temporal convolutional network is widely used to extract the temporal features. Different from the traditional temporal convolutional network with a single receptive field, MGTCN has a broader receptive field during the convolution process, thus enabling to comprehensively obtain nonlinear temporal dependence features.

As shown in Fig. 2(b), to possess a broader receptive field during the convolution process, multiscale convolution kernels (i.e., a set of convolution kernels with different sizes) are constructed in MGTCN. In particular, we adopt shared convolution kernels (the size of variable dimension is 1) on each process variable along the temporal dimension. Assume that the current input is

$U_{\text{in}1} \in \mathbb{R}^{N\times S_1\times C_1}$, where N is the variable dimension (equal to the number of process variables), S_1 is the temporal sequence length, and C_1 is the channel dimension; Correspondingly, two sets of multiscale convolution kernels are constructed (each set has E types of convolution kernels), denoted as $\tau_1 \in \mathbb{R}^{1\times h_1\times \frac{C_2}{E}}$, $\tau_2 \in \mathbb{R}^{1\times h_2\times \frac{C_2}{E}}$, ..., $\tau_E \in \mathbb{R}^{1\times h_E\times \frac{C_2}{E}}$ and $\psi_1 \in \mathbb{R}^{1\times h_1\times \frac{C_2}{E}}$, $\psi_2 \in \mathbb{R}^{1\times h_2\times \frac{C_2}{E}}$, ..., $\psi_E \in \mathbb{R}^{1\times h_E\times \frac{C_2}{E}}$, where C_2 is the desired output channel dimension, and $h_1, h_2, \ldots, h_E$ denote the different lengths of convolution kernels along the temporal dimension and h_E is the maximum length.

These two sets of multiscale convolution kernels are conducted on $U_{\text{in}1}$ to extract two temporal dependence features, respectively, as shown in (2) and (3)

$$tf_1 = (U_{\text{in}1} * \tau_1)||(U_{\text{in}1} * \tau_2)||\ldots||(U_{\text{in}1} * \tau_E) \quad (2)$$

$$tf_2 = (U_{\text{in}1} * \psi_1)||(U_{\text{in}1} * \psi_2)||\ldots||(U_{\text{in}1} * \Psi_E) \quad (3)$$

where $*$ and $\|$ are the convolution operation and concatenation operation, respectively; $(U_{\text{in}1} * \tau_e) \in \mathbb{R}^{N\times(S_1-h_e+1)\times\frac{C_2}{E}}, 1 \leq e \leq E$ denotes the obtained eth feature tensor. Apparently, the feature tensors obtained by convolution kernels with different lengths have different sizes. To facilitate the subsequent concatenation operation, a trim operation is conducted to make all feature tensors have a uniform size $\mathbb{R}^{N\times(S_1-h_E+1)\times\frac{C_2}{E}}$. Then, the trimmed feature tensors are concatenation along the channel dimension to obtain temporal dependence features $tf_1 \in \mathbb{R}^{N\times(S_1-h_E+1)\times C_2}$ and $tf_2 \in \mathbb{R}^{N\times(S_1-h_E+1)\times C_2}$.

Finally, the gated mechanism [5] is adopted to eliminate irrelevant feature information and retain important feature information to obtain the final temporal dependence feature $U_{\text{out}1} \in \mathbb{R}^{N\times(S_1-h_E+1)\times C_2}$, as shown in the following:

$$U_{\text{out}1} = g(tf_1) \odot \sigma(tf_2) \quad (4)$$

where g and σ denote the activation functions tanh and sigmoid, respectively. $\odot$ is the elementwise multiplication operation.

C. Multihop Attention Graph Convolutional Network

1) Dynamic Graph Construction: In this study, to overcome the deficiency of static graphs that only use fixed representations to reflect spatial coupling relations among process variables, we construct a dynamic graph. Inspired by Wu et al. [29], we first construct the node embeddings that can be adaptively updated during the training process, as follows:

$$Q_1 = g(\lambda(W_1 DE_1 + b_1), Q_2 = g(\lambda(W_2 DE_2 + b_2) \quad (5)$$

where $DE_1 \in \mathbb{R}^{N\times EM}$ and $DE_2 \in \mathbb{R}^{N\times EM}$ denote the randomly initialized node embeddings, where N is the variable dimension (equal to the number of process variables) and EM denotes the embedding dimension. $Q_1 \in \mathbb{R}^{N\times EM}$ and $Q_2 \in \mathbb{R}^{N\times EM}$ denote the node embeddings that can be adaptively updated. W_1 and W_2 are the weight parameters. b_1 and b_2 are the biases. g denote the activation function tanh. However, the constructed node embeddings by Wu et al. [29] are static and cannot capture the mutable characteristics of spatial coupling relations among process variables. Therefore, to solve this shortcoming, we build a dynamic kernel to integrate the dynamic information from current input process variables into the dynamic graph, as follows:

$$\Theta = W_\Theta X_{t-T+1:t} + b_\Theta \quad (6)$$

where $X_{t-T+1:t} \in \mathbb{R}^{N\times T}$ denotes the current input process variables. $\Theta \in \mathbb{R}^{N\times EM}$ denote the dynamic kernel, which can help the dynamic graph automatically infer mutable spatial coupling relations based on current input process variables. W_Θ and b_Θ denote the weight parameter and bias, respectively. Finally, we fuse node embeddings that can be adaptively updated and the dynamic kernel to obtain dynamic node embeddings, as follows:

$$DN_1 = Q_1 \odot \Theta, \quad DN_2 = Q_2 \odot \Theta. \quad (7)$$

Based on the above design, dynamic node embeddings $DN_1 \in \mathbb{R}^{N\times EM}$ and $DN_1 \in \mathbb{R}^{N\times EM}$ contain adaptive parameters and the dynamic information from current input process variables so that the adaptive learning for mutable spatial coupling relations can be realized. Then, we use dynamic node embeddings to calculate similarities among all process variables. Similar to Wu et al. [29], we leverage these similarities to generate an asymmetric directed dynamic graph adjacency matrix $A_{dg} \in \mathbb{R}^{N\times N}$ that aims to accurately capture unidirectional spatial coupling relations, as follows:

$$A_{dg} = g'\left(g\left(\lambda\left(DN_1 DN_2^T - DN_2 DN_1^T\right)\right)\right) + I_N \quad (8)$$

where g' denote the activation function ReLU [21]. The subtraction term $(DN_1 DN_2^T - DN_2 DN_1^T)$ makes two elements that are symmetric along the diagonal be opposite to each other and ReLU makes negative elements be zero. These two components ensure the unidirectional property of the dynamic graph. λ denotes the coefficient that can adjust the saturation rate of ReLU. $I_N \in \mathbb{R}^{N\times N}$ denotes the identity matrix, which enables A_{dg} to consider the self-correlation of process variables.

2) Multihop Attention Aggregation Structure: The existing GCN-based soft sensor models only concentrate on the node feature representation at the last hop, which usually encounters the oversmoothing problem (i.e., the feature vectors of each node in the last hop tend to be consistent and lose crucial node feature information), leading to poor soft sensor performance. Therefore, to solve the oversmoothing problem, we introduce the multihop structure [29] to make full use of the node feature representations from different hops. However, Wu et al. [29] directly combine the node feature representations from different hops, which ignore the fact that the node feature representations from different hops may contain some irrelevant noise information so as to degrade the prediction performance. Therefore, we propose a multihop attention aggregation structure to identify the most important feature information in different hops and systematically aggregate it, thereby extracting fine-grained spatial dependence features.

As shown in Fig. 2(c), to comprehensively exploit the unidirectional coupling relations between pairwise process variables, we construct two dynamic graph adjacency matrices with opposite directions (i.e., A_{dg} and A_{dg}^T) and input them into two multihop attention aggregation structures, respectively. Moreover, the extracted temporal dependence feature $U_{\text{out1}} \in \mathbb{R}^{N\times(S_1-h_E+1)\times C_2}$ from MGTCN is used as the input of multihop attention aggregation structures, denoted as $U_{\text{in2}} \in \mathbb{R}^{N\times S_2\times C_2}$ (S_2 is equal to S_1-h_E+1), where S_2 is the temporal sequence length and C_2 is the channel dimension. The feature extraction process of multihop attention aggregation structure with A_{dg} is defined by the following equations:

$$\tilde{A}_{dg} = D_{dg}^{-1} A_{dg} \tag{9}$$

$$U_j = \varphi U_0 + (1-\varphi)\tilde{A}_{dg} U_{j-1}, 1 \leq j \leq J \tag{10}$$

where $D_{dg} \in \mathbb{R}^{N\times N}$ is the degree matrix of A_{dg}; J is the number of hops; U_0 is equal to the initial input U_{in2} and $U_j \in \mathbb{R}^{N\times S_2\times C_2}$ denotes the extracted feature tensor in the jth hop; φ is the coefficient that can determine the retention rate of the original feature U_0.

Then, to overcome the defect of multihop structures in [29], we propose a channel-based attention mechanism to identify the most important node feature representations from different hops, as shown in the following equations:

$$U_{\text{sum}} = \sum_{j=0}^{J} U_j \tag{11}$$

$$U_{\text{agv}} = \text{GAP}(U_{\text{sum}}) \tag{12}$$

$$\omega_{\text{all}} = W_4 \text{GELU}(W_3 U_{\text{agv}} + b_3) + b_4 \tag{13}$$

$$\text{Split}(\omega_{\text{all}}) = [\omega_0, \omega_1, \ldots, \omega_J] \tag{14}$$

$$\alpha_j = \text{softmax}(\omega_j) = \frac{\exp(\omega_j)}{\sum_{j=0}^{J} \exp(\omega_j)} \tag{15}$$

$$U_{\text{out2}}^{(A_{dg})} = \sum_{j=0}^{J} \alpha_j U_j \tag{16}$$

where $U_{\text{sum}} \in \mathbb{R}^{N\times S_2\times C_2}$ is the summation of all feature tensors and $U_{\text{agv}} \in \mathbb{R}^{C_2}$ is the average feature tensor generated by global average pooling [20]; GELU (Hendrycks and Gimpel, 2020) is the activation function; W_3 and W_4 are the weight parameters; b_3 and b_4 are the biases; $\omega_{\text{all}} \in \mathbb{R}^{((J+1)C_2)}$ is the representation of all attention values; Split() is the operation of splitting ω_{all} into (J+1) attention values ($\omega_0, \omega_1, \ldots, \omega_J \in \mathbb{R}^{C_2}$); softmax is the activation function that can normalize the attention values; $\alpha_j \in \mathbb{R}^{C_2}$ is the final attention score of feature U_j; $U_{\text{out2}}^{(A_{dg})} \in \mathbb{R}^{N\times S_2\times C_2}$ is the output spatial dependence features.

Similarly, $U_{\text{out2}}^{(A_{dg}^T)} \in \mathbb{R}^{N\times S_2\times C_2}$ (the output of multihop attention aggregation structure with A_{dg}^T) can be obtained through the same computational process. Finally, the output of MAGCN $U_{\text{out3}} \in \mathbb{R}^{N\times S_2\times C_2}$ (i.e., fine-grained spatial dependence features) can be gained using elementwise addition, as shown in the following:

$$U_{\text{out3}} = U_{\text{out2}}^{(A_{dg})} + U_{\text{out2}}^{(A_{dg}^T)}. \tag{17}$$

D. Refining Module

As shown in Fig. 2(a), the high-level feature representations can be obtained by using the skip connection to fuse the output of K ST-blocks ($U_{\text{out3}}^{(1)} \in \mathbb{R}^{N\times L_1\times C_2}$, $U_{\text{out3}}^{(2)} \in \mathbb{R}^{N\times L_2\times C_2}$, …, $U_{\text{out3}}^{(K)} \in \mathbb{R}^{N\times L_K\times C_2}$) and current input process variables $X_{t-T+1:t} \in \mathbb{R}^{N\times T}$, as follows:

$$U_{\text{hlf}} = g'\left(X_{t-T+1:t} * \theta_0 + \sum_{k=1}^{K} U_{\text{out3}}^{(k)} * \theta_k\right) \tag{18}$$

where $U_{\text{out3}}^{(k)} \in \mathbb{R}^{N\times L_k\times C_2}$ denotes the output of kth ST-block and L_k denotes its temporal sequence length. $*$ is the convolution operation; $\theta_0 \in \mathbb{R}^{1\times T\times C_3}$ and $\theta_k \in \mathbb{R}^{1\times L_k\times C_3}$, $1 \leq k \leq K$ denote the convolution kernels; $U_{\text{hlf}} \in \mathbb{R}^{N\times 1\times C_3}$ is the high-level feature representations and C_3 is the desired output channel dimension; g' denote the activation function ReLU.

The refining module contains two 1 × 1 CNNs and a fully connected layer. Specifically, two 1 × 1 CNNs are used to refine the feature information in U_{hlf} and the fully connected layer is used to obtain final prediction results of all quality variables at time point t $\hat{Y}_t \in \mathbb{R}^M$, as shown in the following:

$$\hat{Y}_t = W_{fc}\left((g'(U_{\text{hlf}} * \zeta_1)) * \zeta_2\right) + b_{fc} \tag{19}$$

where g' denote the activation function ReLU; $\zeta_1 \in \mathbb{R}^{1\times 1\times C_4}$ and $\zeta_2 \in \mathbb{R}^{1\times 1\times M}$ are two convolution kernels in which C_4 is the output channel dimension of the first 1 × 1 CNN and M is the number of quality variables; W_{fc} and b_{fc} are the weight parameter and bias, respectively.

E. MIT Algorithm

At the beginning of training, the conventional training strategy, directly using all the quality variables together as the prediction target, may make deep learning models unable to find a suitable training starting point because predicting multiple quality variables together is more difficult than predicting a single quality variable in industrial processes.

Therefore, we proposed a new training strategy MIT to decrease the training difficulty of deep learning models to improve the prediction performance. As shown in Algorithm 1, to help the deep learning model have a good training starting point, MIT starts by solving the simple soft sensor task (i.e., using only one quality variable as the prediction target). As the number of iterations increases in the training process, new quality variables are introduced into the prediction target one by one until all variables are included. Based on this gradual learning mechanism, the model can effectively learn difficult soft sensor tasks (i.e., predicting multiple quality variables together) and achieve good prediction performance.

Moreover, in practical industrial processes, we may need to predict new emerging quality variables. MIT does not need to use new and old quality variables together as the prediction target to train the deep learning model from scratch because it can use previously well-trained model parameters as the basis and fine tune the model parameters by gradually adding new quality variables to the prediction target so as to significantly decrease time and resource consuming.

Algorithm 1: MIT.

```
Input: The number of training samples $ns$, the training set
  ($X \in \mathbb{R}^{ns \times N \times T}$, $Y \in \mathbb{R}^{ns \times M}$), batch size $bs$, the number
  of epochs $ne$, the number of MIT step $ms$, learning rate $\eta$
Output: Final model's parameters $\phi_{\text{final}}$
Initialize model's parameters $\phi$ and define index = 1,
  $v = 1$
for $i$ in 0, 1, …, $ne$-1 do
  for $j$ in 0, 1, …, ($ns/bs$) do
    Select a batch ($X_{(j)} \in \mathbb{R}^{bs \times N \times T}, Y_{(j)} \in \mathbb{R}^{bs \times M}$)
    from training set
    $\hat{Y}_{(j)} \leftarrow \phi(X_{(j)})$
    gradient$_\phi \leftarrow \nabla_\phi[\text{loss}(Y_{(j)}[:, :v], \hat{Y}_{(j)}[:, :v])]$
    $\phi \leftarrow \phi - \eta \times \text{Adam}(\phi, \text{gradient}_\phi)$
    if index % $ms$ == 0 and $v \leq M$ then
      $v = v + 1$
    end if
    index = index + 1
  end for
end for
```

IV. Verification Study

In this section, a verification study was carried out based on the real experimental data recorded from the coal mill rig. A computer with NVIDIA GeForce RTX 2080 GPU was used as the experimental platform, and all deep learning models were implemented with Pytorch.

All the data were normalized with Z-score normalization before training. Mean absolute error (MAE), root-mean-squared error (RMSE), and mean absolute percentage error (MAPE) were employed to evaluate soft sensor models, which can be defined as follows:

$$\text{MAE} = \frac{1}{V}\sum_{v=1}^{V}|\hat{y}_v - y_v| \tag{20}$$

$$\text{RMSE} = \sqrt{\frac{1}{V}\sum\nolimits_{v=1}^{V}(\hat{y}_v - y_v)^2} \tag{21}$$

$$\text{MAPE} = \frac{1}{V}\sum_{v=1}^{V}\left|\frac{\hat{y}_v - y_v}{y_v}\right| \tag{22}$$

where V is the number of test samples, $\hat{y}_v$ and y_v are the predicted value and real value of the vth sample, respectively.

A. Description of Coal Mill Rig and Experimental Data

A coal mill rig, one of the significant machines in the thermal power process, is employed as the test bed to conduct the soft sensor performance evaluation. Its schematic is shown in Fig. 3 [7], [35]. Specifically, the raw coal is sent to the coal mill rig and is pulverized on the grinding table. Then, the pulverized coal is dried through the primary air and is delivered into the dynamic classifier, where the qualified coal powder is delivered into the boiler but the unqualified coal powder falls back to the grinding table for regrinding. Owing to mutable operating conditions, intricate operation environment, and coupled control loops, coal grinding processes are extremely complex and the recorded process variables show complicated spatial coupling relations, which gives a good platform to verify the online soft sensor performance.

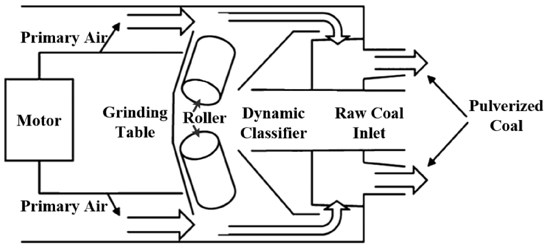

Fig. 3. Schematic of coal mill rig.

TABLE I
Variable Description for Coal Mill

Variable name	Type	Variable name	Type
Coal feed rate	*PV*	Inlet air pressure	*PV*
Power signal	*PV*	Cold air door opening	*PV*
Environment temperature	*PV*	Sealed air pressure	*PV*
Rotary separator motor	*PV*	Mill differential pressure	*PV*
Rotary separator bearing	*PV*	Hot air door opening	*PV*
Outlet temperature	*PV*	Motor coil temperature	*QV*
Outlet pressure	*PV*	Motor bearing temperature	*QV*
Inlet air volume	*PV*	Lubricating oil	*QV*
Inlet air temperature 1	*PV*	Planetary gear bearing	*QV*
Inlet air temperature 2	*PV*	Rotary separator bearing	*QV*
……	*PV*	Oil tank temperature	*QV*

* *PV* denotes the process variables and *QV* denotes the quality variables.

A total of 36 variables recorded by sensors were used in this experiment. For simplicity, only 21 variables were shown in Table I. In particular, six temperature indicators [i.e., motor coil temperature (Mct), motor bearing temperature (Mbt), lubricating oil temperature (Lot), planetary gear bearing temperature (Pbt), rotary separator bearing temperature (Rbt), and oil tank temperature (Ott)] were selected as output quality variables because too low temperatures in key units of the coal mill may affect the grinding efficiency, while too excessive temperatures may cause the dangerous accidents. Then, the remaining 30 variables were selected as input process.

The number of historical time points T was set to 40; thus, a 40-step time window was used to serialize the dataset to obtain 19 961 sequential sample pairs (each sequential sample pair contains input $X_{(t-40+1):t} \in \mathbb{R}^{20\times 40}$ and output $Y_t \in \mathbb{R}^4$). Then, these sequential sample pairs were split into three parts: 80% as training set (15 969 samples), 10% as validation set (1996 samples), and 10% as test set (1996 samples).

B. Baseline Models

To verify the superior performance of the proposed DGDL, some baseline soft sensor models were selected for result comparison, as shown in Table II.

TABLE II
BASELINE MODELS SELECTED IN THIS STUDY

Traditional inference model	
PLS [8]	The maximum number of iterations is 500 and the number of retained principal components is 10.
Shallow machine learning models	
LSSVR [26]	The penalty parameter is 0.03.
ELM [12]	It has 500 neurons and uses the sigmoid as the activation function.
Deep learning models	
Multichannel CNN (MCNN) [33]	It has four input channels and uses three CNN layers. The first two layers both have 128 convolution kernels with size 7×7, the third layer has 16 convolution kernels with size 3×3. The final prediction results are obtained bythe fully connected layer.
LSTM [15]	It has 3 LSTM layers and each has 128 neurons. The final prediction results are obtained by the fully connected layer.
ConvLSTM [23]	It has 2 ConvLSTM layers and each has 32 neurons. The size of convolution kernels is 7×7. The final prediction results are obtained by the fully-connected layer.
GCN [5]	It has 2 GCN layers with 128 neurons and 6 neurons, respectively, and obtains the final prediction results by the fully connected layer. The order of Chebyshev polynomial is 1. In particular, it uses grey relational analysis [6] (Deng 1989) to calculate the spatial coupling relations among process variables to generate the static graph with predefined node relations.

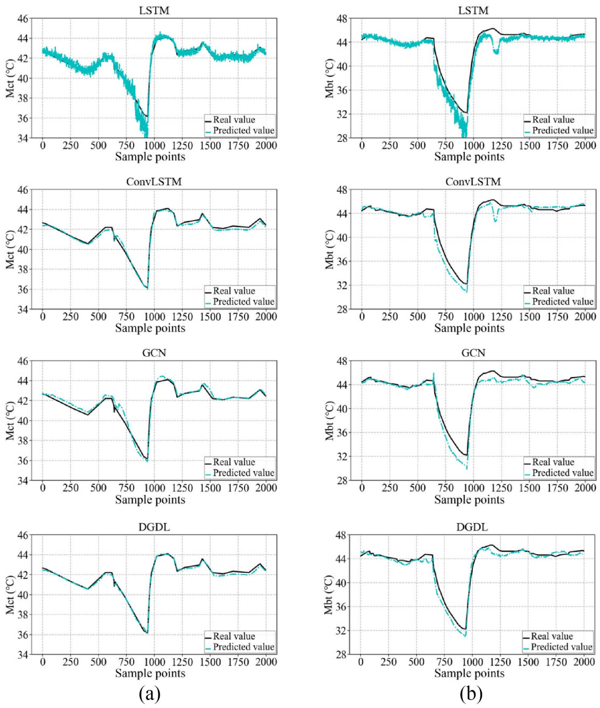

Fig. 4. Predicted and real values of four deep learning models for coal mill. (a) Comparison on Mct. (b) Comparison on Mbt.

C. Parameter Settings of DGDL

The number of multivariate incremental step was 1800 and the number of ST-blocks was 4. The grid search was adopted to generate the combinations of hyperparameters in DGDL and the validation set was used to determine the optimal hyperparameters. Finally, the hyperparameters of DGDL were selected as follows: the output channel dimensions of initial 1 × 1 CNN, MAGCN, and MGTCN were all 16. The output channel dimension of the skip connection layer was 64. For MAGCN, λ was 3, φ was 0.05, the node embedding dimension EM was 40, and the number of hops was 2. For MGTCN, there were four types of convolution kernels with different sizes 1 × 2, 1 × 3, 1 × 6, and 1 × 7. For the refining module, the output channel dimensions of the first 1 × 1 CNN and second 1 × 1 CNN were 64 and 6, respectively, and the fully connected layer had 6 neurons. Besides, for training settings, we used Adam optimizer with gradient clip 5 to train the DGDL for 50 epochs by minimizing MAE between the predicted quality variables and real quality variables. The batch size, learning rate, and L2 regularization penalty were 64, 0.001, and 0.0001, respectively.

D. Experimental Results

1) Comparison With Baseline Models:

Table III presents the prediction errors of different models. Some observed results can be summarized as follows.

Traditional models with simpler structures (i.e., PLS, LSSVR, and ELM) are inferior to deep learning models (i.e., MCNN, LSTM, ConvLSTM, and DGDL), indicating that deep learning models have the stronger nonlinear fitting ability and can achieve more accurate soft sensor of industrial processes.

In addition, DGDL is inferior to MCNN in RMSE of the quality variable Ott and is inferior to ConvLSTM in MAE, RMSE, and MAPE of the quality variable Pbt. Although DGDL cannot gain the lowest prediction errors in these cases, the prediction errors of DGDL are much close to the best ones. For example, for the quality variable Pbt, the MAEs obtained by ConvLSTM and DGDL are 0.10 and 0.20, respectively. In other cases, DGDL presents superior prediction errors than the other baseline models, which demonstrates its feasibility and effectiveness for the online industrial soft sensor. In summary, DGDL outperforms other deep learning models in most metrics because only DGDL can use the dynamic graph to real-timely sense spatial coupling relations among process variables, thereby realizing more accurate soft sensor results.

Fig. 4 exhibits the predicted values of the proposed DGDL and the other three deep learning models (i.e., LSTM, ConvLSTM, and GCN). It can be seen that DGDL possesses the best-fitting capability for quality variables. Moreover, DGDL can generate smoother predicted values when quality variables frequently fluctuate. Therefore, it is concluded that DGDL owns better online soft sensor performance in industrial processes.

2) Visualization Analysis of Different Graphs:

To further illustrate the effectiveness of the dynamic graph in online industrial soft sensor, two variants of DGDL were constructed as follows.

1) w/o DG-1: DGDL without the dynamic graph. The dynamic graph is replaced by the static graph with predefined node relations in GCN [5].

TABLE III
PREDICTION ERRORS OF DIFFERENT MODELS FOR COAL MILL

Quality	Metric	PLS	LSSVR	ELM	MCNN	LSTM	ConvLSTM	GCN	**DGDL**
Mct	MAE	0.19 (±0.00)	0.29 (±0.00)	0.35 (±0.03)	0.22 (±0.04)	0.28 (±0.04)	0.15 (±0.02)	0.24 (±0.02)	**0.13** (±0.02)
	RMSE	0.26 (±0.00)	0.38 (±0.00)	0.45 (±0.04)	0.31 (±0.07)	0.39 (±0.06)	0.18 (±0.03)	0.35 (±0.04)	**0.16** (±0.03)
	MAPE	0.46 (±0.00)	0.70 (±0.00)	0.85 (±0.08)	0.54 (±0.10)	0.68 (±0.10)	0.35 (±0.06)	0.59 (±0.06)	**0.31** (±0.06)
Mbt	MAE	0.70 (±0.00)	0.74 (±0.00)	0.86 (±0.12)	0.83 (±0.12)	0.66 (±0.07)	0.61 (±0.04)	0.61 (±0.05)	**0.58** (±0.05)
	RMSE	0.88 (±0.00)	0.85 (±0.00)	1.13 (±0.12)	1.07 (±0.12)	1.01 (±0.11)	0.84 (±0.04)	0.81 (±0.08)	**0.73** (±0.04)
	MAPE	1.70 (±0.00)	1.71 (±0.00)	2.03 (±0.28)	1.94 (±0.27)	1.61 (±0.18)	1.46 (±0.08)	1.49 (±0.14)	**1.38** (±0.09)
Lot	MAE	0.39 (±0.00)	0.45 (±0.00)	0.44 (±0.09)	0.29 (±0.03)	0.30 (±0.04)	0.27 (±0.02)	0.26 (±0.03)	**0.23** (±0.01)
	RMSE	0.48 (±0.00)	0.51 (±0.00)	0.50 (±0.08)	0.34 (±0.03)	0.38 (±0.05)	0.31 (±0.02)	0.34 (±0.04)	**0.29** (±0.01)
	MAPE	1.05 (±0.00)	1.21 (±0.00)	1.17 (±0.25)	0.79 (±0.08)	0.80 (±0.10)	0.73 (±0.06)	0.69 (±0.08)	**0.63** (±0.02)
Pbt	MAE	0.22 (±0.00)	0.46 (±0.00)	0.35 (±0.06)	0.24 (±0.03)	0.20 (±0.01)	**0.10** (±0.01)	0.37 (±0.02)	0.20 (±0.01)
	RMSE	0.31 (±0.00)	0.64 (±0.00)	0.46 (±0.08)	0.34 (±0.06)	0.32 (±0.02)	**0.15** (±0.02)	0.55 (±0.03)	0.32 (±0.02)
	MAPE	0.47 (±0.00)	1.02 (±0.00)	0.76 (±0.13)	0.51 (±0.07)	0.43 (±0.03)	**0.22** (±0.03)	0.80 (±0.06)	0.43 (±0.02)
Rbt	MAE	0.98 (±0.00)	0.92 (±0.00)	0.86 (±0.15)	0.57 (±0.03)	0.51 (±0.08)	0.74 (±0.15)	0.62 (±0.03)	**0.43** (±0.03)
	RMSE	1.10 (±0.00)	1.19 (±0.00)	1.04 (±0.13)	0.76 (±0.04)	0.74 (±0.13)	0.91 (±0.10)	0.97 (±0.05)	**0.57** (±0.05)
	MAPE	2.27 (±0.00)	2.20 (±0.00)	2.02 (±0.34)	1.38 (±0.06)	1.23 (±0.19)	1.74 (±0.33)	1.50 (±0.07)	**1.03** (±0.07)
Ott	MAE	0.35 (±0.00)	0.37 (±0.00)	0.41 (±0.06)	0.26 (±0.04)	0.28 (±0.04)	0.25 (±0.02)	0.27 (±0.01)	**0.23** (±0.03)
	RMSE	0.40 (±0.00)	0.44 (±0.00)	0.49 (±0.05)	**0.34** (±0.05)	0.45 (±0.07)	0.35 (±0.03)	0.40 (±0.02)	0.37 (±0.04)
	MAPE	1.05 (±0.00)	1.13 (±0.00)	1.24 (±0.17)	0.79 (±0.13)	0.87 (±0.13)	0.78 (±0.05)	0.82 (±0.04)	**0.73** (±0.08)

* The experimental results are the format of *a* (±*s*), where *a* and *s* denote the mean values and standard deviations after five executions. Significant values are boldfaced.

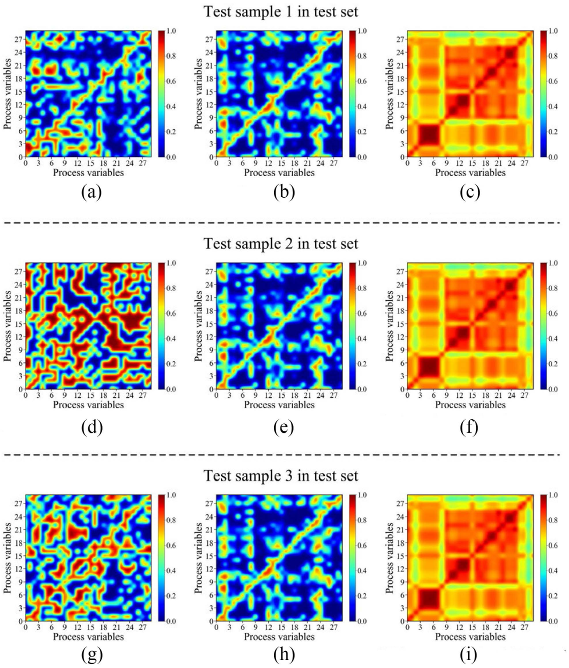

Fig. 5. Visualization of different graphs in different models. (a) Dynamic graph in DGDL. (b) Static graph in w/o DG-2. (c) Static graph in w/o DG-1 and GCN (Defferrard et al., 2016). (d) Dynamic graph in DGDL. (e) Static graph in w/o DG-2. (f) Static graph in w/o DG-1 and GCN (Defferrard et al., 2016). (g) Dynamic graph in DGDL. (h) Static graph in w/o DG-2. (i) Static graph in w/o DG-1 and GCN (Defferrard et al., 2016).

2) w/o DG-2: DGDL without the dynamic kernel. The dynamic kernel is removed so that the dynamic graph is converted to the static graph proposed by Wu et al. [29].

We input some test samples from the test set into different GCN-based models and visualize the corresponding graphs of these test samples, as shown in Fig. 5. The horizontal coordinate and vertical coordinate represent the corresponding indices of process variables. The redder area and bluer area denote the stronger and weaker spatial coupling relations between pairwise process variables, respectively. In this experiment, we regard the prediction process of test samples as the online soft sensor process. Due to the mutable characteristics of spatial coupling relations, spatial coupling relations among process variables change in different test samples. However, it can be seen that all the static graphs (w/o DG-2, w/o DG-1, GCN [5]) use fixed spatial coupling relations in different test samples, which is inconsistent with the real situation that spatial coupling relations change with time and may negatively influence the final prediction results. In contrast, dynamic graph generates different graphs for different test samples because the dynamic graph is capable of automatically inferring mutable spatial coupling relations in different test samples based on different test samples, thus helping to improve the performance of the online industrial soft sensor.

V. CONCLUSION

In this article, a novel deep learning model DGDL was proposed to improve the online soft sensor performance in complex industrial processes. It was based on the fact that underlying spatial coupling relations among process variables had mutable characteristics (i.e., it changes with time). For the first time, the dynamic graph was proposed to adaptively learn and automatically infer mutable spatial coupling relations, which was different from the static graph in GCN-based soft sensor studies. Meanwhile, the proposed MAGCN can leverage the dynamic graph to effectively capture fine-grained spatial dependence features and the proposed MIT can further improve the prediction performance. Comprehensive experiments were conducted on a real coal mill rig to demonstrate the superior performance of DGDL and the effectiveness of dynamic graph in DGDL. Based on the experimental results, it can be concluded that DGDL had lower prediction errors than the other baseline models. In addition, DGDL was more suitable for the online industrial soft sensor because the dynamic graph can leverage current input process variables to automatically infer mutable spatial coupling relations among process variables. Future research directions can focus on the development for the soft sensor of industrial processes with missing data or nonstationary characteristics.

References

[1] R. Chiplunkar and B. Huang, "Siamese neural network-based supervised slow feature extraction for soft sensor application," *IEEE Trans. Ind. Electron.*, vol. 68, no. 9, pp. 8953–8962, Sep. 2021.

[2] Z. Chai, C. Zhao, and B. Huang, "Variational progressive-transfer network for soft sensing of multirate industrial processes," *IEEE Trans. Cybern.*, 2021, to be published, doi: 10.1109/TCYB.2021.3090996.

[3] Y. Chen and X. Chen, "A novel reinforced dynamic graph convolutional network model with data imputation for network-wide traffic flow prediction," *Transp. Res. C, Emerg. Technol.*, vol. 143, 2022, Art. no. 103820.

[4] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 933–941.

[5] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. 30th Annu. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.

[6] J. L. Deng, "Introduction to grey system theory," *J. Grey Syst.*, vol. 1, pp. 1–24, 1989.

[7] S. Duan, C. Zhao, and M. Wu, "Multiscale partial symbolic transfer entropy for time-delay root cause diagnosis in nonstationary industrial processes," *IEEE Trans. Ind. Electron.*, vol. 70, no. 2, pp. 2015–2025, Feb. 2023, doi: 10.1109/TIE.2022.3161761.

[8] P. Facco, F. Doplicher, F. Bezzo, and M. Barolo, "Moving average PLS soft sensor for online product quality estimation in an industrial batch polymerization process," *J. Process Control*, vol. 19, no. 3, pp. 520–529, 2009.

[9] S. D. Grantham and L. H. Ungar, "A first principles approach to automated troubleshooting of chemical plants," *Comput. Chem. Eng.*, vol. 14, no. 7, pp. 783–798, 1990.

[10] M. Han and C. Liu, "Endpoint prediction model for basic oxygen furnace steel-making based on membrane algorithm evolving extreme learning machine," *Appl. Soft Comput.*, vol. 19, pp. 430–437, 2014.

[11] D. Hendrycks and K. Gimpel, "Gaussian error linear units (gelus)," 2016, *arXiv:1606.08415*.

[12] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, no. 1/3, pp. 489–501, 2006.

[13] Y. Huang et al., "Grassnet: Graph soft sensing neural networks," in *Proc. IEEE Int. Conf. Big Data*, 2021, pp. 746–756.

[14] M. Jia, Y. Dai, D. Xu, T. Yang, Y. Yao, and Y. Liu, "Deep graph network for process soft sensor development," in *Proc. IEEE 8th Int. Conf. Inf., Cybern., Comput. Social Syst.*, 2021, pp. 44–49.

[15] W. Ke, D. Huang, F. Yang, and Y. Jiang, "Soft sensor development and applications based on LSTM in deep neural networks," in *Proc. IEEE Symp. Ser. Comput. Intell.*, 2017, pp. 1–6.

[16] S. Khatibisepehr and B. Huang, "Dealing with irregular data in soft sensors: Bayesian method and comparative study," *Ind. Eng. Chem. Res.*, vol. 47, no. 22, pp. 8713–8723, 2008.

[17] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, 2015.

[18] W. Li, C. Yang, and S. Jabari, "Nonlinear traffic prediction as a matrix completion problem with ensemble learning," *Transp. Sci.*, vol. 56, no. 1, pp. 52–78, 2022.

[19] W. Li and Y. Wang, "A robust supervised subspace learning approach for output-relevant prediction and detection against outliers," *J. Process Control*, vol. 106, pp. 184–194, 2021.

[20] M. Lin, Q. Chen, and S. C. Yan, "Network in network," in *Proc. 34th Int. Conf. Mach. Learn.*, Banff, Canada, Apr. 6–11, 2013, pp. 1–10.

[21] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn.*, 2010, pp. 807–814.

[22] A. Sadeghian, O. Wu, and B. Huang, "Robust probabilistic principal component analysis based process modeling: Dealing with simultaneous contamination of both input and output data," *J. Process Control*, vol. 67, pp. 94–111, 2018.

[23] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[24] P. Song and C. Zhao, "Slow down to go better: A survey on slow feature analysis," *IEEE Trans. Neural Netw. Learn. Syst.*, 2022, to be published, doi: 10.1109/TNNLS.2022.3201621.

[25] P. Song, C. Zhao, and B. Huang, "SFNet: A slow feature extraction network for parallel linear and nonlinear dynamic process monitoring," *Neurocomputing*, vol. 488, pp. 359–380, 2022.

[26] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.

[27] G. Wang, Q.-S. Jia, M. Zhou, J. Bi, and J. Qiao, "Soft-sensing of wastewater treatment process via deep belief network with event-triggered learning," *Neurocomputing*, vol. 436, pp. 103–113, 2021.

[28] Y. Wang, P. Yan, and M. Gai, "Dynamic soft sensor for anaerobic digestion of kitchen waste based on SGSTGAT," *IEEE Sensors J.*, vol. 21, no. 17, pp. 19198–19208, Sep. 2021.

[29] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, "Connecting the dots: Multivariate time series forecasting with graph neural networks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 753–763.

[30] Y. Xu et al., "Prediction intervals based soft sensor development using fuzzy information granulation and an improved recurrent ELM," *Chemometrics Intell. Lab. Syst.*, vol. 195, 2019, Art. no. 103877.

[31] W. Yu, M. Wu, and C. Lu, "Meticulous process monitoring with multiscale convolutional feature extraction," *J. Process Control*, vol. 106, pp. 20–28, 2021.

[32] W. Yu, C. Zhao, and B. Huang, "MoniNet with concurrent analytics of temporal and spatial information for fault detection in industrial processes," *IEEE Trans. Cybern.*, vol. 52, no. 8, pp. 8340–8351, Aug. 2022, doi: 10.1109/TCYB.2021.3050398.

[33] X. Yuan, S. Qi, Y. A. W. Shardt, Y. Wang, C. Yang, and W. Gui, "Soft sensor model for dynamic processes based on multichannel convolutional neural network," *Chemometrics Intell. Lab. Syst.*, vol. 203, 2020, Art. no. 104050.

[34] C. Zhao, "Perspectives on nonstationary process monitoring in the era of industrial artificial intelligence," *J. Process Control*, vol. 116, pp. 255–272, 2022.

[35] C. Zhao, J. Chen, and H. Jing, "Condition-driven data analytics and monitoring for wide-range nonstationary and transient continuous processes," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 4, pp. 1563–1574, Oct. 2021.

[36] C. Zhao and H. Sun, "Dynamic distributed monitoring strategy for large-scale nonstationary processes subject to frequently varying conditions under closed-loop control," *IEEE Trans. Ind. Electron.*, vol. 66, no. 6, pp. 4749–4758, Jun. 2019.

[37] K. Zhu, S. Zhang, J. Li, D. Zhou, H. Dai, and Z. Hu, "Spatiotemporal multi-graph convolutional networks with synthetic data for traffic volume forecasting," *Expert Syst. Appl.*, vol. 187, 2022, Art. no. 115992.

Kun Zhu received the M.S. degree in management science and engineering from the Zhejiang University of Finance and Economics, Hangzhou, China, in 2022. He is currently working toward the Ph.D. degree in electronic information with the College of Control Science and Engineering, Zhejiang University, Hangzhou.

His research interests mainly include industrial soft sensor and deep learning.

Chunhui Zhao (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from Northeastern University, Shenyang, China, in 2009.

From 2009 to 2012, she was a Postdoctoral Fellow with the Hong Kong University of Science and Technology and the University of California, Santa Barbara, Los Angeles, CA, USA. Since January 2012, she has been a Professor with the College of Control Science and Engineering, Zhejiang University, Hangzhou, China. Her research interests include statistical machine learning and data mining for industrial application. She has authored or coauthored more than 170 papers in peer-reviewed international journals. She has served as a Senior Editor for the *Journal of Process Control*, AEs for two International Journals, including *Control Engineering Practice* and *Neurocomputing*.