A Novel Sequence Discriminative Feature Extraction Network and Its Application in Offline Industrial Fault Pattern Clustering

Jinchuan Qian, Chihang Wei, *Member, IEEE*, and Zhihuan Song

***Abstract*—A valid fault pattern clustering method for stored fault data can effectively help offline fault diagnosis, and the clustering results can provide solid data support for the training of the fault diagnosis model. To achieve it, a novel sequence discriminative feature extraction network (SDFEN) is developed for extracting the discriminative features underlying the industrial time series. The proposed SDFEN composes a prediction network and an extraction network, which are trained successively. The prediction network is designed for preliminary extraction of the discriminative features, and a set of contribution rate sequences are calculated for the supervised training. Besides, to extract local information and contract the features from the same class, a local information extraction network is further connected, trained by reconstruction and neighbor prediction. Dynamic time warping helps to select neighbors among time series, while a parallel strategy is designed to reduce computing load. The final clustering result is given by the Gaussian mixture model. The feasibility and effectiveness of the proposed method are verified by experiments on the Tennessee Eastman benchmark process and the multiphase flow facility.**

***Index Terms*—Deep learning, discriminative feature extraction, fault clustering, neighbor searching.**

I. Introduction

INFORMATION technology and artificial intelligence have achieved great developments in recent years, which makes it possible to extract complex information from the industrial process without sufficient expert knowledge [1], [2]. The collected fault samples usually contain fault information, which can be used to develop fault diagnosis methods and expand the fault knowledge base. However, in the industrial process without online fault diagnosis systems, these fault samples cannot be diagnosed and labeled in time. Offline fault pattern clustering is designed to preprocess the collected unlabeled fault samples by separating them into several blocks, and the aims of it mainly include the following two parts. 1) Facilitate the subsequent offline fault diagnosis and fault labeling. 2) Provide valid data support for fault diagnosis model training.

Data-driven methods can be performed without sufficient process knowledge, of which clustering methods, such as Gaussian mixture model (GMM) [3] and Kmeans [4], can be directly used to achieve the fault pattern clustering. These methods are less constrained by physical knowledge, so to achieve better clustering results, practical industrial characteristics should be analyzed and considered during the method development. Combining some dimension reduction methods [5], [6] is an effective approach to improve the performance, and because the dynamic correlation wildly exists in practical industrial processes, some works further transform original data into time series to improve the clustering accuracy. Singhal et al. [7] proposed a principal components analysis (PCA)-based similarity factors to improve the performance of Kmeans in industrial time series clustering tasks. Barragan et al. [8] designed a time series clustering method according to the characteristics of the industrial process, which effectively combines wavelet features, PCA-based similarity metric, and fuzzy clustering.

According to the manifestation of fault data, this article divides fault data into two types, stable deviation fault and dynamic deviation fault. Stable deviation fault refers to the fault that the fault information can be stably reflected by the deviation on the variables of a single sample, such as step fault and slope fault. Dynamic deviation fault denotes the fault type with dynamically changing fault information, which could be caused by random noise, stickiness, and sensor degradation. The stable deviation fault can be easily identified once the faulty variables are detected, while the dynamic deviation fault is difficult to deal with because the deviation amplitude may be within the normal range and different fault types may have overlapping areas. The collected offline fault samples usually contain both

Manuscript received 5 February 2023; revised 17 April 2023 and 12 June 2023; accepted 23 July 2023. Date of publication 10 August 2023; date of current version 19 January 2024. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 61833014 and Grant 62103364. Paper no. TII-23-0375. *(Corresponding authors: Chihang Wei; Zhihuan Song.)*

Jinchuan Qian and Zhihuan Song are with the State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: qianjinchuan@zju.edu.cn; songzhihuan@zju.edu.cn).

Chihang Wei is with the School of Information Science and Technology, Hangzhou Normal University, Hangzhou 311121, China (e-mail: chhwei@zju.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TII.2023.3301045.

Digital Object Identifier 10.1109/TII.2023.3301045

aforementioned fault types, to manage the clustering task of the mixed dataset, serializing the original data and transforming it to a time series clustering task can be a good solution, since faults usually lead to changes in the dynamic correlation between data. Moreover, discriminative features underlying the times series should be well extracted to reveal the difference between faults, which can effectively help to improve the clustering accuracy, however, the traditional linear methods, such as PCA, are difficult to achieve it.

Deep learning includes a variety of neural network-based models, which has strong feature extracting ability [9], [10]. Because of it, the deep learning model is wildly used in various industrial applications [11], [12], [13], [14]. In the procedure of deep learning-based feature extraction, labeled data are not always needed, thus, it can help to develop semisupervised and unsupervised methods [15], [16]. A commonly used strategy for feature extraction is reconstructing the input, and hidden outputs are utilized as features for specific tasks, such as fault classification and clustering [17], [18], [19]. The reconstruction mainly focuses on extracting the complex correlation between variables, and the feature might not be discriminative enough to distinguish the patterns, so this strategy is more suitable for model pretraining before supervised fine-tuning. Another way to achieve it is to combine feature extraction process with the pseudolabel obtained by clustering methods and introduce the label information during the feature extraction [20], [21]. The feature obtained by this strategy can be directly used for fault clustering, however, the quality of features can be affected by the quality of pseudolabels, which can easily accumulate errors and lead to confusing results. In fault pattern clustering tasks, it is necessary to increase the discrimination of the feature, and combining discriminative indexes is a feasible strategy. In addition, extra information within the fault pattern needs to be added to enhance the quality of the feature and improve robustness to address errors on the constructed discriminative index.

In this article, a novel sequence discriminative feature extraction network (SDFEN) is proposed for fault pattern clustering. Discriminative features aim to increase the difference between different categories and make the decision boundary between categories easier to describe. To achieve effective discriminative feature extraction, SDFEN is developed based on the deep neural network, which can deeply explore the process information underlying the data. Moreover, for the lack of fault labels, a special-designed prediction index and a feature refined strategy are further developed in SDFEN. The structure mainly composes of the following two parts: a contribution rate sequence prediction network and a local information extraction network (LIEN). The prediction network is designed to extract the discriminative features in time series preliminarily without fault labels, where contribution rate sequences are used as supervision. Additionally, to further extract local information of the data, the feature output of the prediction network is further fed to the LIEN. The parameters in the LIEN are trained by reconstruction and neighbor prediction, motivated by [22], [23]. Because a set of time series is considered here, the dynamic time warping (DTW) is employed as the index to select the neighbors [24]. However, the calculation of DTW is time consuming, thus, a parallel strategy with Euclidean distance-based preselecting process is designed to accelerate the neighbor selection process. The GMM trained by the obtained features will provide the final clustering result. The whole method is composed of several parts and has fully considered the characteristics of practical application so that the performance can be improved.

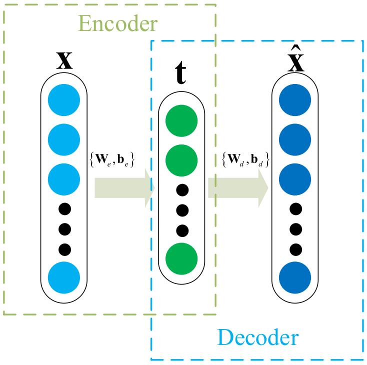

Encoder
$\mathbf{x}$
$\{\mathbf{W}_e, \mathbf{b}_e\}$
$\mathbf{t}$
$\{\mathbf{W}_d, \mathbf{b}_d\}$
$\hat{\mathbf{x}}$
Decoder

Fig. 1. Architecture of AE.

Compared with traditional clustering methods, SDFEN focuses on discriminative feature extraction, so that when the traditional clustering method is directly performed on these features, an accurate clustering result can be stably obtained without further features processing or complex parameters tuning tricks. The main contributions of this article are highlighted in detail.

1) To extract discriminative features with no need of fault labels, a contribution rate sequence prediction network is proposed, which utilizes the calculated contribution rate as the training label.
2) To further extract local information of the time series and contract features of the same category, an LIEN is designed, and this part is trained by the neighbors searched by DTW.
3) To accelerate the DTW-based neighbor selection, this article proposes a parallel strategy with Euclidean distance-based preselecting process.

The rest of this article is organized as follows. Section II presents a brief review of autoencoder (AE), recurrent neural network (RNN), and DTW. Then, a detailed introduction about the proposed SDFEN is given in Section III. Section IV provides simulation results based on the Tennessee Eastman (TE) benchmark process and the multiphase flow facility. Finally, Section VI concludes this article.

II. Preliminaries

In this section, the neural network models involved in the article, and the DTW are briefly reviewed.

A. Autoencoder

AE is a basic unsupervised model for nonlinear feature extraction [9]. As shown in Fig. 1, the structure of AE contains the following two parts: an encoder and a decoder.

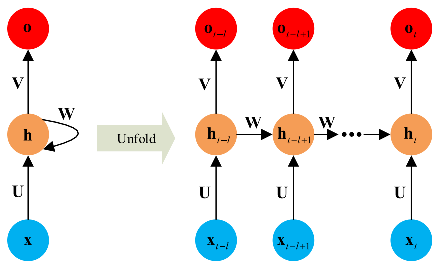


Fig. 2. Architecture of RNN.

Assume the input sample is denoted as $\mathbf{x} \in \mathbb{R}^m$, and $\mathbf{t} \in \mathbb{R}^d$ is the mapped feature, where m and d are the dimensions of input and feature, respectively. Then, the encoder function and the decoder function can be described as follows:

$$\mathbf{t} = \sigma(\mathbf{W}_e\mathbf{x} + \mathbf{b}_e) \tag{1}$$

$$\hat{\mathbf{x}} = \mathbf{W}_d\mathbf{t} + \mathbf{b}_d \tag{2}$$

where $\sigma(*)$ in the encoder function denotes the activation function. The parameters $\{\mathbf{W}_e, \mathbf{W}_d, \mathbf{b}_e, \mathbf{b}_d\}$ of AE can be obtained by minimizing the reconstruction error. In addition, several variants, such as denoise AE (DAE) and sparse AE, are also widely used in feature extraction tasks.

B. Recurrent Neural Network

RNN is proposed for dealing with time series [25]. The structure of RNN is shown in Fig. 2, and can be unfolded as shown on the right-hand side of Fig. 2.

The hidden layer function and the output function at the tth time step are defined as following equations:

$$\mathbf{h}_t = f\left(\mathbf{U}\mathbf{x}_t + \mathbf{W}\mathbf{h}_{t-1}\right) \tag{3}$$

$$\mathbf{o}_t = g\left(\mathbf{V}\mathbf{h}_t\right) \tag{4}$$

where $\mathbf{x}_t$, $\mathbf{h}_t$, and $\mathbf{o}_t$ denote the input, the feature, and the output of RNN at the tth time step, respectively; $\boldsymbol{W}$, $\boldsymbol{U}$, $\boldsymbol{V}$ are the weights, shared by different time steps, and $f(*)$, $g(*)$ denote the activation functions.

To solve the problems of long-term dependencies, gradience vanishing, and gradient exploding, several modified modules, such as long short-term memory and gate recurrent unit are developed to improve the performance.

C. Dynamic Time Warping

Traditional similarity measurement methods will show a large deviation when dealing with unsynchronized sequences, DTW is designed to solve this problem and provide accurate similarities between sequences [24].

Suppose there are two sequences $\mathbf{A}$ and $\mathbf{B}$ with lengths of L_a and L_b to construct a distance matrix $\mathbf{Dist} \in \mathbb{R}^{L_a \times L_b}$ in the first step. The element $\mathbf{Dist}(i,j)$ of the distance matrix denotes the Euclidean distance between the ith sample in $\mathbf{A}$ and the jth sample in $\mathbf{B}$, respectively, denoted as $\mathbf{a}_i$ and $\mathbf{b}_j$. The next step is to find a route in the distance matrix with the least sum of the elements from $\mathbf{Dist}(1,1)$ to $\mathbf{Dist}(L_a, L_b)$, and only three kinds of steps are allowed: 1) $\mathbf{Dist}(i,j)$ to $\mathbf{Dist}(i{+}1,j)$, 2) $\mathbf{Dist}(i,j)$ to $\mathbf{Dist}(i,j{+}1)$, and 3) $\mathbf{Dist}(i,j)$ to $\mathbf{Dist}(i{+}1, j{+}1)$. The distance of the route from $\mathbf{Dist}(1,1)$ to $\mathbf{Dist}(i,j)$ is denoted as $\mathbf{D}(i,j)$. For any i and j, we have

$$\begin{aligned}\mathbf{D}(i,j) &= \mathbf{Dist}(i,j)\\ &\quad + \min\{\mathbf{D}(i-1,j), \mathbf{D}(i,j-1), \mathbf{D}(i-1,j-1)\}.\end{aligned} \tag{5}$$

Dynamic programming can be used to solve this problem, and finally, $\mathbf{D}(L_a, L_b)$ is the value to measure the similarity.

III. Model Structure and Training Strategy

In order to extract sequence discriminative features in industrial time series and achieve an accurate clustering result, SDFEN is proposed, which is mainly achieved by combining two aspects of information: the discriminative information extracted by the supervision of contribution rate sequence and the local information learned from neighbor samples. In the first part, contribution rate sequence construction is the preparation, while in the second part, neighbor searching is the key step.

A. Contribution Rate Sequence Construction

Contribution rate sequence is an index to indicate the difference between different types of faults. It is motivated by the contribution plot, which is a traditional fault identification method [26]. In order to extract nonlinear features, DAE is used instead of PCA here. Assume the reconstruction mapping of the trained DAE is denoted as $R(*)$, the dataset is denoted as $\mathbf{X} \in \mathbb{R}^{N \times m}$. The contribution index $\mathbf{cp}_i \in \mathbb{R}^m$ of sample $\mathbf{x}_i \in \mathbb{R}^m$ can be calculated as follows:

$$\hat{\mathbf{x}}_i = R\left(\mathbf{x}_i\right) \tag{6}$$

$$\mathrm{cp}_{i,j} = \left(\hat{x}_{i,j} - x_{i,j}\right)^2 \tag{7}$$

where $\hat{x}_{i,j}$, $x_{i,j}$, and $cp_{i,j}$ represent the jth element of $\hat{\mathbf{x}}_i$, $\mathbf{x}_i$, and $\mathbf{cp}_i$, respectively.

The index of contribution plot is usually different in different faults, thus, it can be used as a label to supervise network to extract discriminative features. Besides, the contribution index constructed here mainly identifies the fault type by the changing pattern of fault location, thus, to reduce the influence caused by the amplitude of $\mathbf{cp}_i$, the contribution index is normalized and transformed to the contribution rate $\mathbf{cr}_i \in \mathbb{R}^m$ by the following equation:

$$\mathrm{cr}_{i,j} = \frac{\mathrm{cp}_{i,j}}{\sum_{j=1}^{m} \mathrm{cp}_{i,j}} \tag{8}$$

where $\mathrm{cr}_{i,j}$ represents the jth element of $\mathbf{cr}_i$ and the value is in the range of [0,1].

In addition, directly using contribution rate of a single sample would lead to a confusing result, especially when dealing with dynamic deviation type fault, where the value of contribution rate changes frequently. To better reveal the fault information, the calculated contribution rate is extended to contribution rate sequence (CRS) by combining with the contribution rates calculated by previous samples. For simplicity, the tth time series is

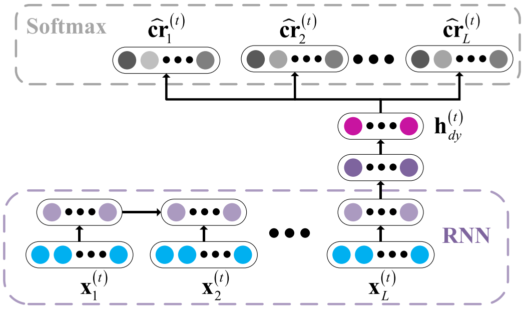

Softmax
$\widehat{\mathbf{cr}}_1^{(t)}$
$\widehat{\mathbf{cr}}_2^{(t)}$
$\widehat{\mathbf{cr}}_L^{(t)}$
$\mathbf{h}_{dy}^{(t)}$
RNN
$\mathbf{x}_1^{(t)}$
$\mathbf{x}_2^{(t)}$
$\mathbf{x}_L^{(t)}$

Fig. 3. Architecture of contribution rate sequence prediction network.

denoted as $\mathbf{X}_t = [\mathbf{x}_1^{(t)} \quad \mathbf{x}_2^{(t)} \quad \cdots \quad \mathbf{x}_L^{(t)}] \in \mathbb{R}^{m\times L}$, and the tth CRS is denoted as $\mathbf{crs}_t = [\mathbf{cr}_1^{(t)} \quad \mathbf{cr}_2^{(t)} \quad \cdots \quad \mathbf{cr}_L^{(t)}] \in \mathbb{R}^{m\times L}$, where L is length of the time series, and $\mathbf{cr}_l^{(t)}$ is the contribution rate calculated by sample $\mathbf{x}_l^{(t)}$.

Note that the DAE model used here is trained by a preselected normal state dataset. For the practical industrial process, it is easy and achievable to label a set of data from a steady normal state.

B. Contribution Rate Sequence Prediction Network

When dealing with unlabeled data, directly using features obtained by traditional unsupervised learning methods cannot give a convincing clustering result without label information aided fine tuning, because the extracted features are not discriminative enough to distinguish the types. Contribution rate sequence prediction network is designed to solve this problem, and Fig. 3 shows its structure.

In the designed prediction network, the feature extraction is realized by a prediction task, and the built CRS is used as the supervision. To extract dynamic features, an RNN structure is used as the input part, note that to reduce the parameters, only the basic RNN module is used here. Then, several fully connected (FC) layers are connected to the feature output of the last time step for further feature extraction and dimension reduction. Finally, the output layer is designed to predict the CRS. To match the range of CRS, the output layer is composed of L softmax. The mean square prediction error is used as the loss function, shown as follows:

$$L_{dy} = \frac{1}{N_s}\sum_{i=1}^{N_s}\sum_{l=1}^{L}\left\|\mathbf{cr}_l^{(i)} - \widehat{\mathbf{cr}}_l^{(i)}\right\|^2 + \lambda L_2\left(\boldsymbol{\theta}_{dy}\right) \qquad (9)$$

where N_s is the number of time series, $\widehat{\mathbf{cr}}_l^{(i)}$ is the prediction value of $\mathbf{cr}_l^{(i)}$, θ_{dy} denotes the parameters in contribution rate sequence prediction network, and $L_2(*)$ is the regularization of parameters, which is used for preventing overfitting. As mentioned above, CRS can be used as a discriminative index, thus, the prediction training process can effectively utilize the index for discriminative feature extraction.

The parameters θ_{dy} can be trained by minimizing the loss function shown above. When the training procedure is completed, the features in this part, denoted as $\mathbf{h}_{dy}$, can be obtained by mapping the input time series to the deepest hidden layer.

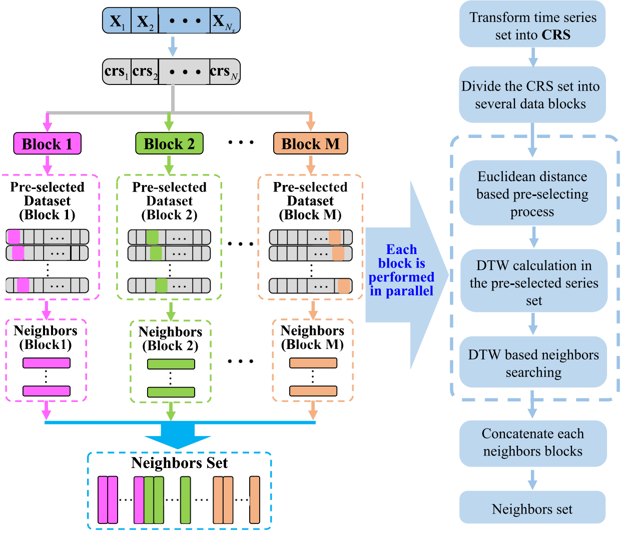

$\mathbf{X}_1$ $\mathbf{X}_2$ • • • $\mathbf{X}_{N_s}$
$\mathbf{crs}_1$ $\mathbf{crs}_2$ • • • $\mathbf{crs}_{N_s}$
Block 1
Block 2
Block M
Pre-selected Dataset (Block 1)
Pre-selected Dataset (Block 2)
Pre-selected Dataset (Block M)
Each block is performed in parallel
Neighbors (Block1)
Neighbors (Block 2)
Neighbors (Block M)
Neighbors Set
Transform time series set into CRS
Divide the CRS set into several data blocks
Euclidean distance based pre-selecting process
DTW calculation in the pre-selected series set
DTW based neighbors searching
Concatenate each neighbors blocks
Neighbors set

Fig. 4. Flowchart of the proposed neighbor searching process.

C. Local Information Extraction Network

Local information can assist the network to explore the low-dimension manifold, which is usually achieved by using neighbor samples and the similarity between samples [27]. Moreover, when dealing with a mixed dataset, the information from the neighbor samples can effectively contract the features of the same class, and make the features more discriminative. LIEN is performed on the feature $\mathbf{h}_{dy}$, and the structure is a single layer AE.

Different from the commonly used local information extraction methods, the data used here are composed of a set of time series. In order to select neighbors more effectively, DTW is performed on CRS to measure the similarity between time series, because CRS can better reveal the fault type. Besides, to accelerate the DTW-based neighbor searching process, a parallel strategy with Euclidean distance-based preselection is proposed.

1) Divide the whole dataset into several blocks.
2) In a single block, for each time series in it, calculate the Euclidean distances between the time series and all-time series in the dataset, then remove the time series with large distances.
3) Calculate the DTW between the time series and the kept time series in the dataset, and select K most similar ones.

Fig. 4 illustrates the flowchart of the proposed neighbor searching method. Preselection is set to pick out of the sequences with the lowest similarities to reduce the burden of DTW calculation, so the accuracy of the similarity index is not highly required, and Euclidean distance can be used in this process for its simple and fast calculation. Moreover, the searching process in each block can be performed independently, therefore, parallel strategy can be used to further speed up the process of neighbor selection. Note that there is no specific principle for dataset division, and the dataset only needs to be divided on average according to the size of the dataset.

The feature extracted in this part is required to be able to reconstruct the input and predict the neighbors. Therefore, the

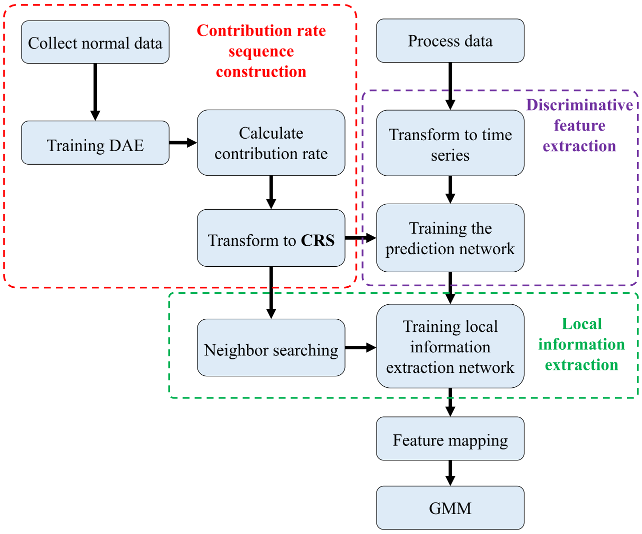

Collect normal data
Contribution rate sequence construction
Training DAE
Calculate contribution rate
Transform to CRS
Process data
Discriminative feature extraction
Transform to time series
Training the prediction network
Neighbor searching
Training local information extraction network
Local information extraction
Feature mapping
GMM

Fig. 5. Flowchart of the proposed method.

loss function, as shown in (10), includes a reconstruction error and the prediction error of the neighbors

$$L_{li}=\sum_{i=1}^{N_s}\left(\left\|\mathbf{h}_{dy}^{(i)}-\hat{\mathbf{h}}_{dy}^{(i)}\right\|+\lambda_l\sum_{k=1}^{K}\left\|\hat{\mathbf{h}}_{dy}^{(i)}-\mathbf{h}_{ne_i}^{(k)}\right\|\right)+\lambda L_2\left(\theta_{li}\right) \tag{10}$$

where $\mathbf{h}_{dy}^{(i)}$ is the $\mathbf{h}_{dy}$ of the ith time series, $\hat{\mathbf{h}}_{dy}^{(i)}$ is the reconstruction value, and $\mathbf{h}_{ne_i}^{(k)}$ denotes the kth neighbor of $\mathbf{h}_{dy}^{(i)}$; λ_l is an adjustable hyperparameter; K is the number of the selected neighbors. θ_{li} denotes the parameters in LIEN.

The $\sum_{k=1}^{K}\|\hat{\mathbf{h}}_{dy}^{(i)}-\mathbf{h}_{ne_i}^{(k)}\|$ part in the loss function enables the model to extract information from neighbor samples. Generally speaking, neighbor samples usually contain the information within the same fault pattern, thus, predicting neighbor samples enables the feature to maintain similarity between original samples. The enhanced feature can be effectively constrained and is more suitable for subsequent fault pattern clustering.

IV. SDFEN-Based Clustering Application

SDFEN uses two parts of network structure to realize the deep mining of discriminative features. Like the layer-wise training strategy in Stack AE, these two parts are trained successively. Moreover, in practical applications, the DTW-based neighbor searching process can be performed once the CRS set has been built, and it is independent of the training of the prediction network part. Therefore, these two processes can be executed in parallel. Fig. 5 shows the schematic diagram of the whole process.

Once the network is well-trained, the deepest feature, i.e., the hidden layer output of the LIEN, can be calculated by mapping the time series in these two parts successively. The calculated features are transferred to train the clustering model and the final clustering result can be obtained afterward. The extracted feature is pursued to distinguish the types, and there is no need to introduce a specially designed and complex clustering method. Therefore, GMM is employed to calculate the final clustering result here, because it can be easily achieved and well adapt to the high noise in the industrial processes.

V. Case Studies

The case studies on the TE process and multiphase flow facility are presented to verify the superiority and feasibility of the proposed method. Basic methods of spectral clustering [28], DTW-based Kmeans and Kshape [29] are used as comparisons. Besides, to achieve effective ablation experiments, the sequential feature extracting method Seq2Seq [30] and the contribution rate sequence prediction part of SDFEN, denoted as CRSPN, are also employed. Both CRS and original data are used in the experiments, and the performance of TSNE is also compared here. All the networks are trained by Adam optimizer and Rectified linear unit (ReLU) is chosen as the activation function. Because the application background is mainly performed offline, all the parameters are tuned based on the whole collected dataset. The RNN hidden layer dimension in SDFEN is determined by the training loss of CRSPN, and the classic criterion Bayesian information criterion (BIC) is used for determining the cluster number [31]. Detailed determination processes are illustrated in the supplementary file.

To evaluate the clustering performance, normalized mutual information (NMI) and adjusted rand index (ARI) are used to indicate the accuracy from both global and fine-grained perspectives. The three basic clustering methods are tested 100 times, and to measure the discrimination of the extracted features, GMM models trained on the extracted features are also tested 100 times with different initialization. The collected results are shown by boxplots. Moreover, in practical applications, the data with significant fault information are more likely to be detected and collected in the offline dataset, and the sample numbers of different faults in the collected offline dataset are usually different. Therefore, similar situations are simulated in the experiments. All the experiments are performed in the hardware environment of Intel Core i7-6700 CPU with deep learning framework Pytorch 1.8.1.

A. TE Process

TE process is an experimental platform designed, which can simulate various typical process characteristics in practical application [32].

Among all the variables, 33 variables including 22 process measurements and 11 manipulated variables are chosen for the experiment. The data used here include six operation states, which have included typical fault types. The data are selected from the standard 960-sample dataset, according to the fault detection results [33], [34], and they are further transformed to the time series with the length of 32. The detailed descriptions of the data including the sample location of original TE dataset are given in Table I.

The hidden layer dimension of RNN used in both SDFEN and Seq2Seq is 64, and is further reduced to 20 by FC; the dimension of the hidden layer in the LIEN part is set to 16, the number of the neighbors is set to 8, and λ_l is set to 0.1, detailed network structure parameters are tabulated in Table II. In addition, the

TABLE I
DESCRIPTIONS OF THE CHOSEN TE PROCESS TIME SERIES

Index	Condition	Type	Sample location
1–200	Fault 1	Step	361-560
201–600	Fault 4	Step	361-760
601–800	Fault 10	Random variation	621-820
801–1300	Fault 11	Random variation	361-860
1301–1600	Fault 14	Sticking	361-660
1601–2000	Fault 17	Unknown	361-760

TABLE II
NETWORK STRUCTURE IN TE PROCESS EXPERIMENT

Model	Structure
Seq2Seq	RNN hidden dimension: 64, FC: 64-32-20
CRSPN	RNN hidden dimension: 64, FC: 64-32-20
SDFEN	RNN hidden dimension: 64, FC: 64-32-20, LIEN: 20-16-20

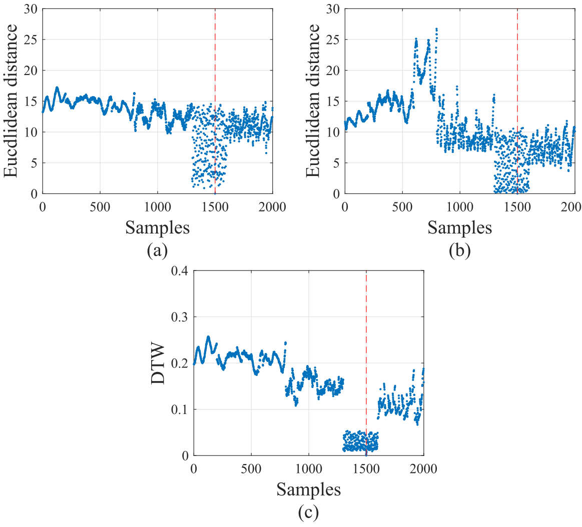


Fig. 6. Similarity calculated by Euclidean distance and DTW. (a) Euclidean distance between CRS. (b) Euclidean distance between $\mathbf{h}_{dy}$. (c) DTW between CRS.

TABLE III
AVERAGE TIME COST OF DIFFERENT METHODS

Methods	Time (s)
DTW	2.410
DTW with preselection	0.613
Proposed parallel method	0.233

TABLE IV
MEDIAN NMI AND ARI IN TE PROCESS

Methods	Median NMI	Median ARI
Spectral	0.6748	03634
Spectral-CRS	0.7096	0.4665
Kmeans	0.3827	0.1743
Kmeans-CRS	0.5937	0.4388
Kshape	0.1809	0.0999
Kshape-CRS	0.7237	0.6017
Seq2Seq	0.7854	0.6186
Seq2Seq-TSNE	0.7224	0.5936
CRS	0.7928	0.6663
CRS-TSNE	0.9310	0.8752
CRSPN	0.9213	0.8665
SDFEN	**0.9685**	**0.9612**

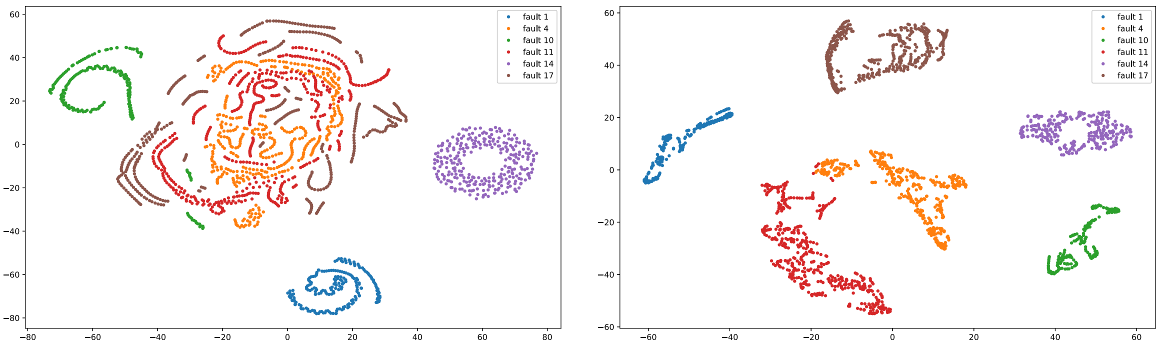
DAE is trained by 500 normal state samples, and the noise for training follows $N(0, 0.5^2)$.

Fig. 6 illustrates the results of the similarities between the 1500th time series and the whole time series set, where three methods are involved: the Euclidean distance between CRSs, the Euclidean distance between the features of CRSPN, $\mathbf{h}_{dy}$, and the DTW between CRSs. It can be seen that the result given by DTW is more accurate than that of Euclidean distance because all the time series from the same class have shorter distances, which means greater similarities, while the results of Euclidean distance are confusing. As mentioned above, the similarity is used to choose the neighbors, therefore, DTW can give a more accurate search space, which is helpful for local information extraction.

The average time costs of the neighbor searching process for a single time series are tabulated in Table III, where DTW, DTW with preselection, and DTW with proposed parallel strategy are compared. A quarter of the time series in the dataset are retained by the Euclidean distance-based preselection, and the parallel strategy is realized by separating the task to the 4 cores in the CPU and performing multiprocessing. It can be seen that with the preselecting process, the neighbor searching process is accelerated obviously, and the proposed parallel strategy further reduces the time cost to one-tenth of the original time cost. Compared with DTW, the calculation of Euclidean distance is much faster, and the neighbors are searched in the preselected dataset, which means less DTW calculations. Moreover, the proposed strategy allows several time series to search neighbors in parallel, which further accelerates the whole process.

Through TSNE, two-dimension visualizations of original times series and the features extracted by SDFEN are illustrated in Fig. 7. It can be seen that after SDFEN-based feature extraction, the obtained features show obvious discrimination, and the difference between fault types is significantly increased compared with the original time series.

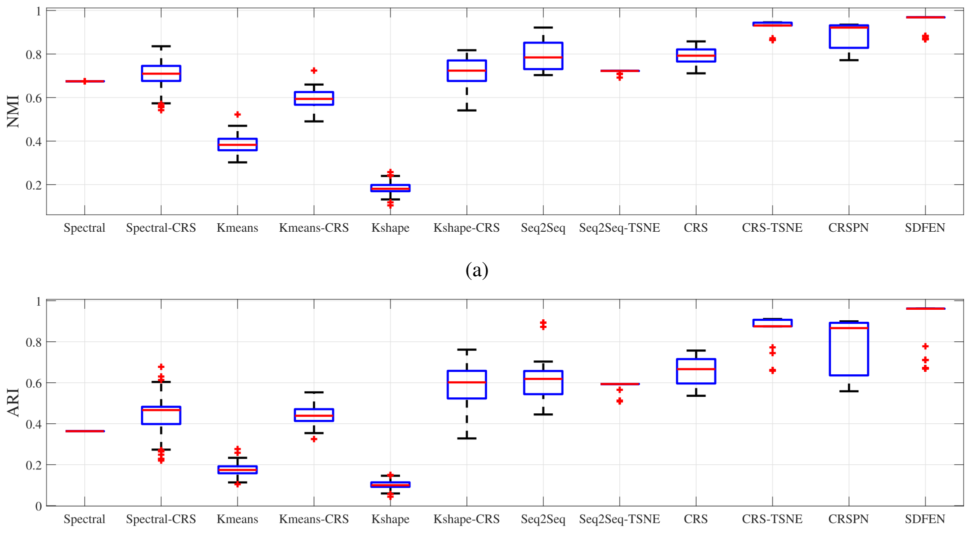
Fig. 8 illustrates the boxplots of NMI and ARI given by each method, and the median index values of different methods are recorded in Table IV, where the numbers in bold represent the best performance. It can be seen that SDFEN gives the most accurate result, while CRS-TSNE achieves the second best. Traditional clustering methods, such as spectral clustering, Kmeans, and Kshape, cannot give convincing results, but after combining CRS, the performances are effectively improved. Moreover, Seq2Seq is trained in an unsupervised way, and it gives a confusing clustering result, however, after combining CRS prediction, the clustering accuracy is improved, as shown in the performance of CRSPN. Therefore, CRS is effective supervision for discriminative feature extraction. In addition, TSNE is a dimension reduction method, which can increase the distance between different clusters, but it shows that the performance can be affected by the original data distribution. CRS is a calculated index, which can be used to indicate the difference between faults, and is discriminative itself, thus, with the help of TSNE, the performance of CRS is improved. However, the performance of Seq2Seq-TSNE is not as good as that of Seq2Seq, it might be because the feature given by



(a)

(b)

Fig. 7. Two-dimension visualization by TSNE. (a) Original time series. (b) Features extracted by SDFEN.



(a)

(b)

Fig. 8. Boxplots of NMI and ARI in TE process experiment. (a) NMI. (b) ARI.

Seq2Seq in this experiment is not discriminative enough, which makes the results of TSNE more chaotic. SDFEN is completely achieved by neural networks, and on the basis of CRSPN, local information is further extracted, therefore, SDFEN shows a more accurate and stable performance, and it can be seen from the boxplots that the results given by SDFEN are robust to GMM initialization because the obtained results are concentrated at the median, except for several outliers.

Note that to demonstrate the effectiveness of the proposed method more comprehensively, the clustering results using all the data of the chosen faults are given in the supplementary file, and the model parameters are the same as listed in Table II. In addition, to provide a set of performance references for future research, an additional experiment including all the TE faults is also presented in the supplementary file.

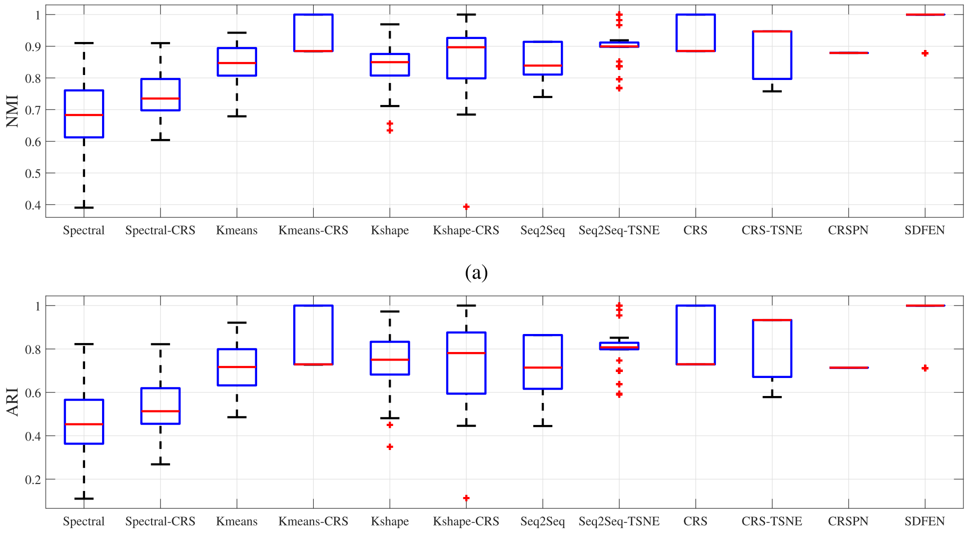
B. Multiphase Flow Facility

The multiphase flow facility is an experimental facility developed by Cranfield University. The data are collected under the real industrial scene, and can be used to evaluate the performances of fault detection and diagnosis methods. A more detailed description can be referred to [35].

TABLE V
DESCRIPTIONS OF THE CHOSEN MULTIPHASE FLOW FACILITY TIME SERIES

Index	Condition	Description
1–300	Fault 1	Air line blockage
301–800	Fault 2	Water line blockage
801–1400	Fault 3	Top separator input blockage
1401–1600	Fault 4	Open direct bypass
1601–2100	Fault 5	Slugging condition
2101–2500	Fault 6	Pressurization of the 2” line

There are 24 variables in the process measured by the installed sensors, and all of them are chosen for the evaluation in this section. The dataset used here consists of six different faults and is further transformed to the time series with the length 32. The detailed descriptions of the data are given in Table V. Moreover, a set of stable data containing 1100 samples are selected from the normal dataset for training the DAE, and the noise for training follows $N(0, 0.5^2)$. Detailed network structure parameters are tabulated in Table VI. In LIEN part, the number of the neighbors is set to 12, and λ_l is set to 0.8.



(a)

(b)

Fig. 9. Boxplots of NMI and ARI in multiphase flow facility experiment. (a) NMI. (b) ARI.

TABLE VI
NETWORK STRUCTURE IN MULTIPHASE FLOW FACILITY EXPERIMENT

Model	Structure
Seq2Seq	RNN hidden dimension: 64, FC: 64-32-20
CRSPN	RNN hidden dimension: 64, FC: 64-32-20
SDFEN	RNN hidden dimension: 64, FC: 64-32-20, LIEN: 20-16-20

TABLE VII
MEDIAN NMI AND ARI IN MULTIPHASE FLOW FACILITY

Methods	Median NMI	Median ARI
Spectral	0.6830	0.4531
Spectral-CRS	0.7350	0.5131
Kmeans	0.8470	0.7168
Kmeans-CRS	0.8849	0.7291
Kshape	0.8498	0.7504
Kshape-CRS	0.8968	0.7811
Seq2Seq	0.8392	0.7142
Seq2Seq-TSNE	0.8996	0.8076
CRS	0.8861	0.7296
CRS-TSNE	0.9467	0.9333
CRSPN	0.8795	0.7139
SDFEN	**1.0000**	**1.0000**

Fig. 9 shows the boxplots of the NMI and ARI for each method, and the median NMIs and ARIs are tabulated in Table VI, where the numbers in bold represent the best performance.. It can be seen from Table VII and boxplots that the difference between fault types is greater in this experiment, so most of the methods show improvement comparing with TE process, especially Kmeans-based methods. Moreover, TSNE shows its superiority here. Compared with Seq2Seq and CRS, Seq2Seq-TSNE and CRS-TSNE have higher medians, although CRS has a possibility to give higher accuracy as shown in the boxplots in Fig. 9, its performances are unstable and sensitive to the initialization of GMM. In addition, the fault information is more obvious in this experiment, which makes CRS more discriminative to distinguish the fault types. Therefore, extracting discriminative features underlying the time series by CRSPN only makes the results more stable, but the median is similar. However, after local information is extracted, the proposed SDFEN shows a significant performance improvement compared with CRSPN. In addition, the result given by SDFEN is stable, and almost all the indexes are concentrated around 1, except for several outliers, validating the effectiveness of the local information extraction.

VI. CONCLUSION

The motivation of this article aims to solve the fault pattern clustering task in industrial processes. To achieve this, the SDFEN is novelly designed to extract sequence discriminative features underlying the time series. The highlights of this work include:

1) CRS is used as supervision to guide the prediction network part to extract discriminative features;
2) local information is further extracted by neighbors prediction;
3) a parallel strategy with Euclidean distance-based preselection is designed to accelerate the neighbor searching.

The features obtained by SDFEN is discriminative and more suitable for data clustering task, therefore, the traditional clustering method GMM can easily give an accurate clustering result using these features. The experiments on TE process and multiphase flow facility show the effectiveness and superiority of the proposed method. Our future work focuses on developing analysis methods for segmented data.

REFERENCES

[1] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annu. Rev. Control*, vol. 36, no. 2, pp. 220–234, 2012.

[2] X. Kong, X. Jiang, B. Zhang, J. Yuan, and Z. Ge, "Latent variable models in the era of industrial big data: Extension and beyond," *Annu. Rev. Control*, vol. 54, pp. 167–199, 2022.

[3] L. Yao and Z. Ge, "Scalable semisupervised GMM for big data quality prediction in multimode processes," *IEEE Trans. Ind. Electron.*, vol. 66, no. 5, pp. 3681–3692, May 2019.

[4] G. Chen, Y. Liu, and Z. Ge, "K-means Bayes algorithm for imbalanced fault classification and big data application," *J. Process Control*, vol. 81, pp. 54–64, 2019.

[5] M. C. Thomas, W. Zhu, and J. A. Romagnoli, "Data mining and clustering in chemical process databases for monitoring and knowledge discovery," *J. Process Control*, vol. 67, pp. 160–175, 2017.

[6] A. J. Torabi, J. E. Meng, L. Xiang, B. S. Lim, and O. P. Gan, "Application of clustering methods for online tool condition monitoring and fault diagnosis in high-speed milling processes," *IEEE Syst. J.*, vol. 10, no. 2, pp. 721–732, Jun. 2016.

[7] A. Singhal and D. E. Seborg, "Clustering multivariate time-series data," *J. Chemometrics*, vol. 19, no. 8, pp. 427–438, 2010.

[8] J. F. Barragan, C. H. Fontes, and M. Embirucu, "A wavelet-based clustering of multivariate time series using a multiscale SPCA approach," *Comput. Ind. Eng.*, vol. 95, pp. 144–155, 2016.

[9] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[10] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, 2015, Art. no. 436.

[11] K. Wang, X. Yuan, J. Chen, and Y. Wang, "Supervised and semi-supervised probabilistic learning with deep neural networks for concurrent process-quality monitoring," *Neural Netw.*, vol. 136, pp. 54–62, 2021.

[12] X. Jiang and Z. Ge, "Data augmentation classifier for imbalanced fault classification," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 3, pp. 1206–1217, Jul. 2021.

[13] X. Yuan, J. Rao, Y. Gu, L. Ye, K. Wang, and Y. Wang, "Online adaptive modeling framework for deep belief network-based quality prediction in industrial processes," *Ind. Eng. Chem. Res.*, vol. 60, no. 42, pp. 15208–15218, 2021.

[14] H. Zhang, G. Qiao, S. Lu, L. Yao, and X. Chen, "Attention-based feature fusion generative adversarial network for yarn-dyed fabric defect detection," *Textile Res. J.*, vol. 93, no. 5/6, pp. 1178–1195, 2023.

[15] L. Yao and Z. Ge, "Dynamic features incorporated locally weighted deep learning model for soft sensor development," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–11, 2021.

[16] S. Zheng and J. Zhao, "A new unsupervised data mining method based on the stacked autoencoder for chemical process fault diagnosis," *Comput. Chem. Eng.*, vol. 135, 2020, Art. no. 106755.

[17] X. Li, X. Li, and H. Ma, "Deep representation clustering-based fault diagnosis method with unsupervised data applied to rotating machinery," *Mech. Syst. Signal Process.*, vol. 143, 2020, Art. no. 106825.

[18] M. Xia, T. Li, L. Liu, L. Xu, and C. W. de Silva, "Intelligent fault diagnosis approach with unsupervised feature learning by stacked denoising autoencoder," *IET Sci., Meas. Technol.*, vol. 11, no. 6, pp. 687–695, 2017.

[19] G. Wang, J. Huang, and F. Zhang, "Ensemble clustering-based fault diagnosis method incorporating traditional and deep representation features," *Meas. Sci. Technol.*, vol. 32, no. 9, 2021, Art. no. 095110.

[20] H. Liu, J. Zhou, Y. Xu, Y. Zheng, X. Peng, and W. Jiang, "Unsupervised fault diagnosis of rolling bearings using a deep neural network based on generative adversarial networks," *Neurocomputing*, vol. 315, pp. 412–424, 2018.

[21] X. Hu, Y. Li, L. Jia, and M. Qiu, "A novel two-stage unsupervised fault recognition framework combining feature extraction and fuzzy clustering for collaborative AIoT," *IEEE Trans. Ind. Inform.*, vol. 18, no. 2, pp. 1291–1300, Feb. 2022.

[22] X. Zhao, M. Jia, and M. Lin, "Deep Laplacian auto-encoder and its application into imbalanced fault diagnosis of rotating machinery," *Measurement*, vol. 152, 2019, Art. no. 107320.

[23] C. Liu, K. Wang, L. Ye, Y. Wang, and X. Yuan, "Deep learning with neighborhood preserving embedding regularization and its application for soft sensor in an industrial hydrocracking process," *Inf. Sci.*, vol. 567, pp. 42–57, 2021.

[24] M. Müller, *Dynamic Time Warping*. Berlin, Germany: Springer, 2007.

[25] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. 11th Annu. Conf. Int. Speech Commun. Assoc.*, vol. 2, no. 3, 2010, pp. 1045–1048.

[26] J. A. Westerhuis, S. P. Gurden, and A. K. Smilde, "Generalized contribution plots in multivariate statistical process monitoring," *Chemometrics Intell. Lab. Syst.*, vol. 51, no. 1, pp. 95–114, 2000.

[27] C. Wei, W. Shao, and Z. Song, "Virtual sensor development for multioutput nonlinear processes based on bilinear neighborhood preserving regression model with localized construction," *IEEE Trans. Ind. Inform.*, vol. 17, no. 4, pp. 2500–2510, Apr. 2021.

[28] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. 28th Conf. Neural Inf. Process. Syst.*, vol. 14, 2001.

[29] J. Paparrizos and L. Gravano, "k-shape: Efficient and accurate clustering of time series," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2015, pp. 1855–1870.

[30] X. Zhang and Z. Ge, "Automatic deep extraction of robust dynamic features for industrial big data modeling and soft sensor application," *IEEE Trans. Ind. Inform.*, vol. 16, no. 7, pp. 4456–4467, Jul. 2020.

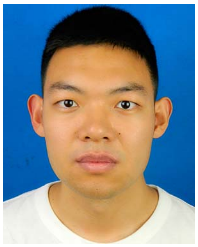
[31] C. Fraley and A. E. Raftery, "How many clusters? Which clustering method? Answers via model-based cluster analysis," *Comput. J.*, vol. 41, pp. 578–588, 1998.

[32] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault Detection and Diagnosis in Industrial Systems*. Berlin, Germany: Springer, 2000.

[33] Y. Cong, L. Zhou, Z. Song, and Z. Ge, "Multirate dynamic process monitoring based on multirate linear Gaussian state-space model," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1708–1719, Oct. 2019.

[34] S. Yin, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Trans. Ind. Electron.*, vol. 61, no. 11, pp. 6418–6428, Nov. 2014.

[35] C. Ruiz-Cárcel, Y. Cao, D. Mba, L. Lao, and R. Samuel, "Statistical process monitoring of a multiphase flow facility," *Control Eng. Pract.*, vol. 42, pp. 74–88, 2015.



Jinchuan Qian received the B.Eng. degree in automation from the Hefei University of Technology, Anhui, China, in 2017 and the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2022.

He is currently a Postdoctoral Research Fellow with the State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou, China. His research interests include fault detection, fault diagnosis, and industrial Big Data modeling.

Chihang Wei (Member, IEEE) received the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2018.

From 2019 to 2021, he was a Postdoctoral Research Fellow with the State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University. He is currently an Associate Professor with the School of Information Science and Technology, Hangzhou Normal University, Hangzhou, China. His research interests include system modeling, process data analysis, fault diagnosis, industrial big data, and soft sensor applications.

Zhihuan Song received the B.Eng. and M.Eng. degrees in industrial automation from the Hefei University of Technology, Hefei, China, in 1983 and 1986, respectively, and the Ph.D. degree in industrial automation from Zhejiang University, Hangzhou, China, in 1997.

Since 1997, he has been with the Department of Control Science and Engineering, Zhejiang University, where he was first a Postdoctoral Research Fellow, then an Associate Professor, and is currently a Professor. He has authored or coauthored more than 200 papers in journals and conference proceedings. His research interests include the modeling and fault diagnosis of industrial processes, analytics and applications of industrial big data, and advanced process control technologies.