IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 23 October 2023, accepted 13 November 2023, date of publication 16 November 2023,
date of current version 21 November 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3334012

RESEARCH ARTICLE

Ensemble Monitoring Model Based on Multi-Subspace Partition of Deep Features

ZHICHAO LI[1,2], LI TIAN[1], AND XUEFENG YAN[3], (Member, IEEE)
[1]Department of Electrical Engineering and Automation, Shaoxing University, Shaoxing 312000, China
[2]Institute of Artificial Intelligence, Shaoxing University, Shaoxing 312000, China
[3]Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China

Corresponding author: Li Tian (tianli_712@163.com)

This work was supported in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LQ22F030019, in part by the National Key Research and Development Program of China under Grant 2020YFA0908303, and in part by the National Natural Science Foundation of China under Grant 21878081.

ABSTRACT Traditional deep neural network (DNN) based process monitoring methods only use the deep features of the last layer and residuals to achieve fault detection. However, the features in different hidden layers are different representations of the input data, which may be beneficial to process monitoring. Only using the deepest features for process monitoring will cause the problems of information loss and low monitoring performance. To obtain more useful information for fault detection, this paper considers the features in all hidden layers and proposed an ensemble monitoring model based on multi-subspace partition of deep features. Firstly, a DNN model is established based on the collected faultless data to obtain the features in all hidden layers and residuals. Secondly, a new feature matrix is constructed based on the retained deep features and residuals. Then, the multi-subspace partition of the new feature matrix is realized by combining correlation analysis and cluster analysis. Finally, the monitoring statistics that are established based on the features in each subspace are fused to realize process monitoring. The proposed method can not only reduce information loss but also enrich the fault-related information. The monitoring performance is verified through two benchmark processes and one actual industrial process.

INDEX TERMS Deep neural network, process monitoring, fault detection, cluster analysis.

I. INTRODUCTION

As the increasing application of information technology in industrial processes, modern industrial processes are becoming larger and more complex [1], [2]. This trend is conducive to improving product quality and reducing production costs, but also makes the coupling between industrial process devices complicated [3], [4]. Any minor deviation may affect the normal production status, resulting in decline of product quality or even serious accidents. As a key link to improve the economic benefits and core competitiveness of enterprises, process monitoring can ensure the product quality and the production safety [5], [6]. Moreover, due to the rise of artificial intelligence and big data, data-driven process monitoring methods have attracted extensive attention [7], [8], [9], [10]. Multivariate statistical process monitoring (MSPM) methods project raw data into a low dimensional subspace and a residual space through multivariate statistical analysis methods. Then, two monitoring statistics are established to realize process monitoring [11], [12]. Different methods can extract different types of features and are suitable for different industrial processes. Principal component analysis (PCA) realizes dimensionality reduction according to the variance information of raw data, which assumes that the process variables are subject to Gaussian distribution [13], [14], [15]. Local preserving projection fully consider the local proximity of raw data during feature extraction, which can maintain the local structure of input data in low dimensional space [16]. Independent component analysis is used to extract mutually independent non-Gaussian components and is mainly used for non-Gaussian process monitoring [17], [18].

The associate editor coordinating the review of this manuscript and approving it for publication was Mehrdad Saif.

However, the traditional MSPM methods can only express the linear relationship between input data. Directly applying these methods to modern industrial processes with complex nonlinear characteristics may neglect the nonlinear relationship between input data. To enhance the nonlinear process monitoring performance, several improved methods are proposed. The first method is based on kernel function [19], where kernel function is adopted to project input data into high-dimensional space. Then, traditional MSPM methods are executed in high-dimensional space [20], [21]. The second method is based on multiple models. This method divides the input data into several subspaces through process decomposition, and then establishes a linear model in each subspace [22]. The nonlinear characteristics are approximately characterized by fusing the information of multiple linear models [23], [24]. The third method is based on machine learning and shallow neural network, which can directly establish a nonlinear monitoring model based on the input data [25], [26]. However, these methods face different problems. Firstly, the monitoring performance is affected by the determination of kernel function and parameters [27]. In addition, in the online monitoring phase, the real-time monitoring will be limited by the size of training samples [28]. Secondly, multiple models based methods emphasize the local information of industrial processes and ignore the global information [3]. Third, machine learning and shallow neural networks based methods can only extract the shallow representation of input data, which have poor learning ability for complex nonlinear processes [29].

In recent years, deep neural networks (DNNs) with multiple hidden layers have been widely used [30], [31], [32]. More abstract representation of the input data can be obtained by DNNs through extracting features layer by layer. Therefore, DNNs have more powerful feature extraction capabilities and function fitting capabilities. As unsupervised learning frameworks, deep belief network (DBN) and stacked autoencoder (SAE) are widely used in fault detection for industrial processes. However, SAE and DBN are mainly used to extract higher-order abstract features of process data. Then, higher-order abstract features are further processed to achieve various monitoring tasks. For example, Wu et al. used bidirectional long short-term memory as the layers of variational autoencoder to handle the dynamic nonlinear characteristics in process data. Then, the deepest features are used to construct a statistic for the trend prediction of production processes [33]. Zheng and Zhao used the extracted features as input to the visualization algorithm and then combined with visual clustering technology to realize fault diagnosis [34]. Zhang et al. established monitoring statistics for fault detection by using the extracted deepest features as inputs to k-nearest neighbor [35]. Lyu et al. used multiple DBNs to extract local features of industrial images and fuse them to realize a picture-based industrial process monitoring method. This method can effectively reduce the complexity and computational burden of the model [36]. Recently, utilizing convolutional neural network (CNN) and generative adversarial network (GAN) to enhance the feature extraction of DNNs have gradually gained popularity in the field of fault detection. For example, Liu et al. proposed the residual attention convolutional autoencoder for feature enhancement and selection [37]. Deng et al. introduced 1-D convolution operation and residual module into GANomaly, enhancing its reconstruction and feature extraction performance for industrial processes [38].

Although these methods have improved process monitoring performance, only the deepest features extracted by DNN are utilized. Considering that the layer-by-layer transformation and extraction of process information may result in the loss of useful information, some scholars further improved the training process of DNNs and strengthened the extraction of process related features. Wang et al. added the input information in the layer-by-layer pre-training to reduce information loss [39]. In addition, DNNs based on the supervised learning framework have also been studied for the quality prediction and fault diagnosis. Liu et al. used DBN to extract features of quality spectrum and realize production quality status discrimination [40]. By adding the correlation constraint between features and output variables, Yuan et al. proposed several models for soft sensor [41], [42]. Although these methods can enhance the extraction of process related features, they still only utilize the deepest features to represent process information and establish predictive models, ignoring the feature information of other hidden layers.

At present, there are some researches have demonstrated the effectiveness of intermediate layer features in DNNs. For example, Matthew et al. proposed a visualization method to understand the working principle and feature extraction process of CNNs by visualizing intermediate layer features [43]. Yosinski et al. found that features of different levels have different sensitivities to different attributes of the image [44]. Yu and Yan also illustrated that the intermediate layer features of DNNs can provide favorable information for fault detection [45]. Therefore, only using the information in the last layer for process monitoring can cause information loss and lower model monitoring performance.

To avoid this problem, this paper will fully consider the deep features of each hidden layer to enrich the useful information for process monitoring. However, directly using the deep features in all hidden layers to establish monitoring statistics may also submerge the fault information, which is not conducive to fault detection. The effectiveness of subspace partitioning based on correlation analysis and clustering analysis in solving this problem has been demonstrated in distributed process monitoring [46]. Considering this situation, on the basis of fully considering the deep features in all hidden layers, this paper proposes an ensemble monitoring model based on multi-subspace partition of deep features (SAE-MSPM). Firstly, a DNN model is established with collected faultless data, and the deep features in each hidden layer are extracted. Secondly, considering that the residual

of the DNN also contains fault-relevant information, a new feature matrix is obtained by combining the deep features with the residual information. Then, the new feature matrix is divided into multiple subspaces by using cluster analysis technology. The partition principle is that the features in the same subspace have high correlation, while the features in different subspaces have high independence. Based on the features in each subspace, the monitoring statistics and control limits are calculated. Finally, through a decision fusion strategy, monitoring results of the subspaces are integrated to obtain the final monitoring result. Because this paper fully considers the features in all hidden layers and the residual information of DNN, it can effectively avoid information loss. Moreover, the proposed multi-subspace partition principle can make each subspace have different sensitivity to faults. Integrating multiple subspaces with different fault sensitivity can help improve the model's generalization ability and enhance the fault detection performance. The main contributions of this paper are as follows:

(1) The deep features in all hidden layers are considered for fault detection. This not only avoids information loss caused by layer-by-layer information extraction, but also enhances the stability of the monitoring model;

(2) A deep feature multi-subspace partition with diverse information representation has been implemented based on correlation and clustering analysis. This can enhance the expression of fault information and reduce the impact of irrelevant variables on fault detection;

(3) The generalization performance of the ensemble model has been improved by integrating the detection results of multiple subspaces.

This paper is arranged as follows. The structure and principle of SAE is introduced in Section II. Section III describes the proposed monitoring method in detail. Three case studies are performed to compare the performance of algorithms in Section IV. The last section draws conclusion and deficiencies.

II. BACKGROUND

A. SAE-BASED FAULT DETECTION

Autoencoder (AE) is firstly reviewed before introducing SAE. As shown in Figure 1, AE is a three-layer network: input layer ($\boldsymbol{x}$), hidden layer ($\boldsymbol{h}$) and output layer ($\hat{\boldsymbol{x}}$). The process of mapping input data from input layer to hidden layer is called encoder; the process of mapping the features of the hidden layer to the output layer is called decoder.

The mapping functions of encoder and decoder are as follows,

$$\boldsymbol{h} = f\left(\boldsymbol{x}; \boldsymbol{W}_{en}, \boldsymbol{b}_{en}\right) = f\left(\boldsymbol{W}_{en}\boldsymbol{x} + \boldsymbol{b}_{en}\right) \tag{1}$$

$$\hat{\boldsymbol{x}} = g\left(\boldsymbol{h}; \boldsymbol{W}_{de}, \boldsymbol{b}_{de}\right) = g\left(\boldsymbol{W}_{de}\boldsymbol{h} + b_{de}\right) \tag{2}$$

where $\boldsymbol{W}_{en}$ and $\boldsymbol{b}_{en}$ are the weights and biases in encoder, $\boldsymbol{W}_{de}$ and $\boldsymbol{b}_{de}$ are the weights and biases in decoder, $f(\cdot)$ and $g(\cdot)$ are the activation functions in hidden layer and output layer.

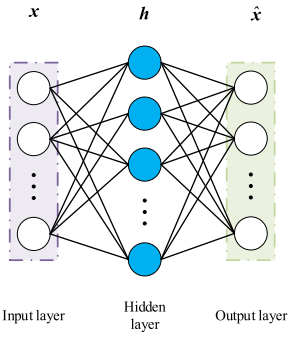

FIGURE 1. Structure of AE.

As an unsupervised learning framework, AE usually realizes feature extraction by reconstructing input data. Therefore, the objective function of AE is defined as follows,

$$\min \ \ l\left(\boldsymbol{W}_{en}, \boldsymbol{b}_{en}, \boldsymbol{W}_{de}, \boldsymbol{b}_{de}\right) = \sum_{i=1}^{n} L\left(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_i\right) \tag{3}$$

where $L\left(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_i\right)$ is the square loss function,

$$L\left(\boldsymbol{x}_i, \hat{\boldsymbol{x}}_i\right) = \sum_{j=1}^{r} \left(x_{i,j} - \hat{x}_{i,j}\right)^2 \tag{4}$$

where n and r represent the number of samples and variables, respectively.

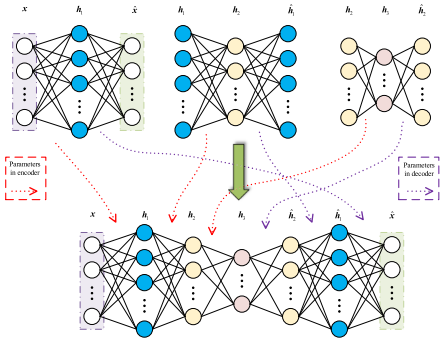

FIGURE 2. Structure of SAE.

SAE is formed by cascading multiple AEs. The features extracted from the ith AE is used as the input of the $(i+1)$th AE to realize the layer-by-layer feature extraction of the input data, which can obtain more effective features than a single AE. Figure 2 shows a SAE consisting of three AEs, where $\boldsymbol{h}_i$ represents the features extracted by the ith hidden layer.

SAE adopts a two-step training method of layer-by-layer pre-training and final fine-tuning. Layer-by-layer pre-training refers to completing the training of a single AE in sequence. First, the first AE is trained with the input data, then the parameters in the encoder and decoder $(\boldsymbol{W}_{en}^1, \boldsymbol{b}_{en}^1, \boldsymbol{W}_{de}^1, \boldsymbol{b}_{de}^1)$ and the features in the first hidden layer $(\boldsymbol{h}_1)$ are obtained. Secondly, the second AE is trained

with $\boldsymbol{h}_1$ as input, then the parameters in the encoder and decoder ($\boldsymbol{W}_{en}^2, \boldsymbol{b}_{en}^2, \boldsymbol{W}_{de}^2, \boldsymbol{b}_{de}^2$) and the features in the second hidden layer ($\boldsymbol{h}_2$) are obtained. The same training strategy is adopted for the subsequent AEs: the $(i+1)$th AE is trained with $\boldsymbol{h}_i$ as input, and the corresponding mapping parameters ($\boldsymbol{W}_{en}^{i+1}, \boldsymbol{b}_{en}^{i+1}, \boldsymbol{W}_{de}^{i+1}, \boldsymbol{b}_{de}^{i+1}$) and deep features $\boldsymbol{h}_{i+1}$ are obtained.

After the pre-training is completed, the trained multiple AEs are cascaded in sequence. The parameters obtained during pre-training are used as the initial parameters in the SAE model. Then, the SAE model is fine-tuned with the goal of minimizing the reconstruction error.

Suppose that the encoder and decoder of the trained SAE are as follows,

$$\boldsymbol{h}_f = F(\boldsymbol{x}) \tag{5}$$

$$\hat{\boldsymbol{x}} = G(\boldsymbol{h}_f) \tag{6}$$

where $\boldsymbol{h}_f$ is the deepest features, $F(\cdot)$ is the mapping function from the input layer to the deepest hidden layer in SAE, and $G(\cdot)$ is the mapping function from the deepest hidden layer to the output layer in SAE.

Then, two monitoring statistics are obtained for fault detection,

$$T^2 = \left(\boldsymbol{h}_f - \bar{\boldsymbol{h}}_{f,tr}\right)^T \sum\nolimits_{\boldsymbol{H}_{f,tr}}^{-1} \left(\boldsymbol{h}_f - \bar{\boldsymbol{h}}_{f,tr}\right) \tag{7}$$

$$SPE = \left(\boldsymbol{x} - \hat{\boldsymbol{x}}\right)^T \left(\boldsymbol{x} - \hat{\boldsymbol{x}}\right) \tag{8}$$

where $\boldsymbol{h}_f$ and $\sum \boldsymbol{H}_{f,tr}$ are the mean and covariance of the deepest features based on the training data, respectively. In addition, to test whether the monitoring statistics are abnormal, control limits are calculated with kernel density estimation (KDE).

B. PROBLEM DESCRIPTION

SAE extracts the features from input data by minimizing the reconstruction error, which is an unsupervised feature extraction method. At present, process monitoring methods based on DNNs usually use DNN to extract low-dimensional features from high-dimensional input data. Moreover, only the features in the last layer are used for fault detection. With the increase of network depth, the extracted features become more and more abstract. In fact, the shallow features in DNNs also contain information that is conducive to process monitoring. Therefore, the traditional DNN may suffer from the loss of information. Therefore, how to further integrate deep features in different hidden layers to achieve better monitoring performance is worthy studying. This study fully considers the features in all hidden layers, and the correlation among all deep features is analyzed. Then, the features are divided into multiple subspaces. The features in the same subspace have similar responses to faults, while the features in different subspaces have different responses to faults. This study can enrich the fault-related information through the similar representation of features in the same subspace, and avoid information loss.

III. PROPOSED METHOD

A. DATA PROCESSING AND LAYER-BY-LAYER FEATURE EXTRATION BASED ON DNN

For a large amount of historical normal operating data collected, maximum and minimum normalization method is firstly used to make each variable change in the [0, 1]. Subsequently, the dataset is randomly divided into two parts, 80% of which is used for model establishing, and the remaining 20% is used for model evaluation. In addition, early stopping strategy is used in the training process to avoid the overfitting problem. After determining the structure of the DNN model, the model is trained. Then, the mapping functions from the input layer to each hidden layer (assuming that there are K hidden layers in the DNN model) can be obtained,

$$\boldsymbol{h}_1 = F_1(\boldsymbol{x}) \in \boldsymbol{R}^{l_1} \tag{9}$$

$$\boldsymbol{h}_2 = F_2(\boldsymbol{x}) \in \boldsymbol{R}^{l_2} \tag{10}$$

$$\vdots$$

$$\boldsymbol{h}_K = F_K(\boldsymbol{x}) \in \boldsymbol{R}^{l_K} \tag{11}$$

where $\boldsymbol{h}_k$ is the deep features extracted by the kth hidden layer, $F_k(\cdot)$ represents the mapping function from the input layer to the kth hidden layer, l_j is the number of nodes in the jth hidden layer ($j = 1,2,\ldots,K$). The mapping function from the input layer to the output layer is as follows,

$$\hat{\boldsymbol{x}} = \Psi(\boldsymbol{x}) \tag{12}$$

As mentioned in Sections I and II, most DNN based process monitoring methods only utilize the output information of the last layer ($\boldsymbol{h}_K$). Although $\boldsymbol{h}_K$ is the most abstract and highest-order feature, the information obtained by each layer of the network is different. The information carried by each layer of neurons may be beneficial to the monitoring process. Process monitoring methods based on the features in the last layer may cause information loss and low monitoring performance. In addition, the residual information obtained by the DNN are also beneficial to monitoring operating status of the processes [47]. Therefore, comprehensive consideration of the features in all hidden layers and the reconstruction error of the DNN model can further enhance the monitoring performance for industrial processes.

However, different deep features are different representations of the input data. Directly using the features in all hidden layers and residual information to establish monitoring statistics may also submerge the fault information, which is not conducive to fault detection. To solve this problem, a multi-subspace partition method based on correlation analysis and cluster analysis is proposed. Deep features and residual information are combined into a new feature matrix, and then the correlation analysis technology is used to measure the correlation between the features in the new feature matrix. Finally, multi-subspace partition based on correlation can be realized by combining cluster analysis. This method divides the highly correlated deep features and residual variables into the same subspace, which enables the features in the same subspace to

monitor the process variables similarly. Moreover, different subspaces represent different process information and have different monitoring behaviors. Comprehensively considering the fault detection results of multiple subspaces with different monitoring characteristics is beneficial to strengthen the fault information expression and further improve the generalization ability of the monitoring method.

B. MULTI-SUBSPACE PARTITION METHOD BASED ON DIVERSITY OF DEEP FEATIRES

Assume that the training data obtained after normalization is $\boldsymbol{X} \in \boldsymbol{R}^{n\times r}$, the features in each hidden layer extracted by DNN with K hidden layers are $\boldsymbol{H} = [\boldsymbol{H}_1, \boldsymbol{H}_2, \cdots, \boldsymbol{H}_K] \in \boldsymbol{R}^{n\times(l_1+l_2+\cdots+l_K)}$, and residual matrix is $\boldsymbol{E} = \boldsymbol{X} - \hat{\boldsymbol{X}} \in \boldsymbol{R}^{n\times r}$. Then, a new feature matrix, $\boldsymbol{M} = [\boldsymbol{H}, \boldsymbol{E}] = \left[\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_{l_1+l_2+\cdots+l_K+r}\right] \in \boldsymbol{R}^{n\times(l_1+l_2+\cdots+l_K+r)}$, is obtained by combining $\boldsymbol{H}$ and $\boldsymbol{E}$. Using $\boldsymbol{M}$, the multi-subspace partition method based on diversity of deep features can be realized by combining correlation analysis and cluster analysis.

In this paper, Pearson correlation coefficient is used to calculate the distance between features (the distance between clusters), and then hierarchical clustering algorithm is used to realize the multi-subspace partition. The correlation coefficient between any two features ($\boldsymbol{m}_i$ and $\boldsymbol{m}_j$) in $\boldsymbol{M}$ is calculated as follows,

$$\rho_{\boldsymbol{m}_i,\boldsymbol{m}_j} = \frac{cov\left(\boldsymbol{m}_i, \boldsymbol{m}_j\right)}{\sigma_{\boldsymbol{m}_i}\sigma_{\boldsymbol{m}_j}} = \frac{E\left(\boldsymbol{m}_i - \bar{\boldsymbol{m}}_i\right) E\left(\boldsymbol{m}_j - \bar{\boldsymbol{m}}_j\right)}{\sigma_{\boldsymbol{m}_i}\sigma_{\boldsymbol{m}_j}} \tag{13}$$

where $\bar{\boldsymbol{m}}_i$ and $\bar{\boldsymbol{m}}_j$ are the mean values of $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$ respectively, and $\sigma_{\boldsymbol{m}_i}$ and $\sigma_{\boldsymbol{m}_j}$ are the standard deviations of $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$, respectively. $\rho_{\boldsymbol{m}_i,\boldsymbol{m}_j}$ varies between -1 and 1, "+" and "−" respectively indicate that they have positive and negative correlation. The larger the value of $\left|\rho_{\boldsymbol{m}_i,\boldsymbol{m}_j}\right|$ is, the greater the correlation between $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$ is. The smaller the value of $\left|\rho_{\boldsymbol{m}_i,\boldsymbol{m}_j}\right|$ is, the smaller the correlation between $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$ is. Since this paper only considers the correlation between features, and usually a smaller distance value represents a larger similarity, the following indicator is used to represent the distance between $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$,

$$dis\left(\boldsymbol{m}_i, \boldsymbol{m}_j\right) = 1 - \left|\rho_{\boldsymbol{m}_i,\boldsymbol{m}_j}\right| \tag{14}$$

It can be concluded from Eq. (14) that when there is a complete positive or negative correlation between $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$, the distance between the two features is 0; when there is no linear correlation between $\boldsymbol{m}_i$ and $\boldsymbol{m}_j$, the distance between the two features is 1.

Hierarchical clustering analysis usually has two ways, one is bottom-up aggregation, and the other is top-down division. In this paper, the first way is used for multi-subspace partition. At the beginning of clustering, all features are regarded as a single cluster, and then the clustering is completed by continuously merging the two nearest clusters. The distance between clusters ($\boldsymbol{C}_i$, and $\boldsymbol{C}_j$) can be calculated as,

$$dis\left(\boldsymbol{C}_i, \boldsymbol{C}_j\right) = \frac{1}{\left|\boldsymbol{C}_i\right|\left|\boldsymbol{C}_j\right|} \sum_{\boldsymbol{m}_i\in\boldsymbol{C}} \sum_{\boldsymbol{m}_j\in\boldsymbol{C}_j} dis\left(\boldsymbol{m}_i, \boldsymbol{m}_j\right) \tag{15}$$

In addition, determining the number of clusters is also a problem worth studying. This paper adopts silhouette coefficient to determine the final number of subspaces. Silhouette coefficient considers not only the distance between individuals in the cluster, but also the distance between clusters. The silhouette coefficient of $\boldsymbol{m}_i$ is calculated as follows,

$$SC\left(\boldsymbol{m}_i\right) = \frac{DIS_{out}\left(\boldsymbol{m}_i\right) - DIS_{in}\left(\boldsymbol{m}_i\right)}{\max\left(DIS_{out}\left(\boldsymbol{m}_i\right), DIS_{in}\left(\boldsymbol{m}_i\right)\right)} \tag{16}$$

where $DIS_{in}(\boldsymbol{m}_i)$ represents the average distance between $\boldsymbol{m}_i$ and other features in the same cluster, and $DIS_{out}(\boldsymbol{m}_i)$ represents the average distance between $\boldsymbol{m}_i$ and features in other clusters. Equation (16) indicates that $SC(\boldsymbol{m}_i)$ varies between -1 and 1. A larger $SC(\boldsymbol{m}_i)$ indicates the smaller intra-cluster distance and the larger inter-cluster distance, which further indicates that the clustering performance is better. The average of silhouette coefficients of all features ($\overline{SC}$) can be used to represent the performance of current clustering. A larger $\overline{SC}$ represents a better clustering performance. The proposed multi-subspace partition method can be summarized as Algorithm 1.

Algorithm 1 Multi-Subspace Partition Method Based on Diversity of Deep Features

Input: New feature $\boldsymbol{M} = [\boldsymbol{H}, \boldsymbol{E}] = \left[\boldsymbol{m}_1, \boldsymbol{m}_2, \cdots, \boldsymbol{m}_{l_1+l_2+\cdots+l_K+r}\right] \in \boldsymbol{R}^{n\times(l_1+l_2+\cdots+l_K+r)}$

Procedure:

1. Set the current cluster number: $q = l_1 + l_2 + \cdots + l_K + r$
2. Initialize clustering result $\boldsymbol{C}$: Treat each column in $\boldsymbol{M}$ as an independent cluster: $\boldsymbol{C} = \left\{\boldsymbol{C}_1, \boldsymbol{C}_2, \cdots \boldsymbol{C}_q\right\}$ where $\boldsymbol{C}_1 = [\boldsymbol{m}_1], \boldsymbol{C}_2 = [\boldsymbol{m}_2], \cdots, \boldsymbol{C}_q = \left[\boldsymbol{m}_q\right]$
3. Calculate the silhouette coefficient of the current clustering result
4. while $q > 1$:
5. Calculate the distance matrix or similarity matrix between clusters (Eq. (15))
6. Find cluster pairs with minimum distance or maximum similarity ($\boldsymbol{C}_i, C_j$)
7. Merge clusters $\boldsymbol{C}_i$ and $\boldsymbol{C}_j$ to form a new cluster $\boldsymbol{C}$
8. Update clustering result $\boldsymbol{C}$
9. Calculate the silhouette coefficient of the current clustering result
10. Select the clustering result with maximum silhouette coefficient as the final clustering result

Output: Clustering result $\boldsymbol{C}$

C. FAULT DETECTION BASED ON MULTI-SUBSPACE FUSION

For on-line fault detection, the features in each hidden layer ($\boldsymbol{h}^T = \left[\boldsymbol{h}_1^T, \boldsymbol{h}_2^T, \cdots, \boldsymbol{h}_K^T\right]$) and the residual ($\boldsymbol{e} = \boldsymbol{x} - \hat{\boldsymbol{x}} \in \boldsymbol{R}^r$) are firstly extracted and calculated based on the trained model. Then, the new feature matrix is constructed as follows, $\boldsymbol{m} = \left[\boldsymbol{h}^T, \boldsymbol{e}^T\right]^T \in \boldsymbol{R}^{l_1+l_2+\cdots+l_K+r}$.

Suppose that $\boldsymbol{m} = \left[\boldsymbol{h}^T, \boldsymbol{e}^T\right]^T$ is divided into T subspaces through Section III-B, $\boldsymbol{c}_1 \in \boldsymbol{R}^{b_1}, \boldsymbol{c}_2 \in \boldsymbol{R}^{b_2}, \cdots, \boldsymbol{c}_T \in \boldsymbol{R}^{b_T}$, where b_i represents the number of features that are divided into the ith subspace. According to the features in each subspace, T^2 statistics are established as follows (taking the ith subspace as an example),

$$T_i^2 = \left(\boldsymbol{c}_i - \bar{\boldsymbol{c}}_{i,tr}\right)^T \sum\nolimits_{C_{i,tr}}^{-1} \left(\boldsymbol{c}_i - \bar{\boldsymbol{c}}_{i,tr}\right) \tag{17}$$

where the meaning of each parameter is similar to that in Eq. (7). Due to the complexity and abstraction of feature extraction by DNN, it is impossible to predict the distribution knowledge of deep features. Therefore, the nonparametric estimation method, KDE, is adopted to solve the control limits of T^2 statistics.

To show the monitoring results more intuitively, the monitoring results are integrated by Bayesian inference. For the online sample $\boldsymbol{x}_{new}$, the probability that the ith subspace identifies it as a fault is as follows,

$$p_{T_i^2}\left(F|\boldsymbol{x}_{new}\right) = \frac{p_{T_i^2}\left(\boldsymbol{x}_{new}|F\right) p_{T_i^2}\left(F\right)}{p_{T_i^2}\left(\boldsymbol{x}_{new}\right)} \tag{18}$$

where $p_{T_i^2}\left(\boldsymbol{x}_{new}\right)$ can be obtained from the total probability formula,

$$p_{T_i^2}\left(\boldsymbol{x}_{new}\right) = p_{T_i^2}\left(\boldsymbol{x}_{new}|F\right) p_{T_i^2}\left(F\right) + p_{T_i^2}\left(\boldsymbol{x}_{new}|N\right) p_{T_i^2}\left(N\right) \tag{19}$$

$p_{T_i^2}\left(F\right)$ and $p_{T_i^2}\left(N\right)$ represent the prior probabilities of industrial processes in fault and normal states, respectively. The conditional probability in Eq. (19) is defined as follows,

$$P_{T_i^2}\left(\boldsymbol{x}_{new}|N\right) = \exp\left(-\frac{T_i^2}{T_{i,\lim}^2}\right) \tag{20}$$

$$P_{T_i^2}\left(\boldsymbol{x}_{new}|F\right) = \exp\left(-\frac{T_{i,\lim}^2}{T_i^2}\right) \tag{21}$$

After the fault probability of each subspace with respect to $\boldsymbol{x}_{new}$ is obtained, the comprehensive statistic is calculated by weighted average,

$$BIC = \sum_{i=1}^{T} \left\{ \frac{P_{T_i^2}\left(\boldsymbol{x}_{new}|F\right) P_{T_i^2}\left(F|\boldsymbol{x}_{new}\right)}{\sum_{i=1}^{T} P_{T_i^2}\left(\boldsymbol{x}_{new}|F\right)} \right\} \tag{22}$$

If the value of BIC is greater than $p_{T_i^2}\left(F\right)$, a fault has occurred. Otherwise, the industrial process is in normal status. Figure 3 shows the flow chart of the proposed SAE-MSPM.

Offline modeling:

1) Collect faultless data and standardize it to [0, 1];

2) Set the structure of SAE and train the SAE model;

3) Calculate the features in each hidden layer and residual, and form the new feature matrix;

4) Use correlation coefficients as distance metrics to achieve multi-subspace partitioning of the new feature matrix;

5) Construct T^2 statistics based on the features in each subspace and use KDE to calculate control limits;

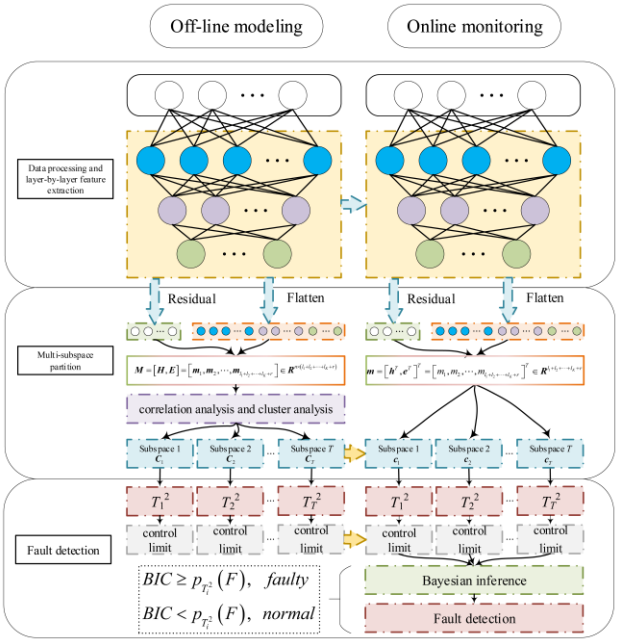

FIGURE 3. Flow chart of SAE-MSPM.

Online monitoring:

6) Standardize test data based on statistical information from training data;

7) Similar to 3), the new feature space is formed for the test data, and it is divided into different subspaces based on 4);

8) Calculate the monitoring statistics of each subspace;

9) Fault detection results are integrated by Bayesian inference to achieve fault detection.

IV. CASE STUDY

In this paper, fault detection rate (FDR) and false alarm rate (FAR) are adopted to evaluate the performance of fault detection model. FDR and FAR are defined as follows,

$$FDR = \frac{n\left(faulty|faulty\right)}{n\left(faulty\right)} \tag{23}$$

$$FAR = \frac{n\left(faulty|normal\right)}{n\left(normal\right)} \tag{24}$$

where $n\left(faulty\right)$ and $n\left(normal\right)$ are the numbers of faulty and normal samples used for testing, respectively. $n\left(faulty|faulty\right)$ and $n\left(faulty|normal\right)$ are the number of fault samples detected and the number of normal samples falsely reported as faults, respectively. Under the condition that FAR can meet industrial requirements, a larger FDR indicates better performance of the fault detection model.

A. TE BENCHMARK PROCESS

TE benchmark process is a simulation model based on an actual industrial process [48]. This process can collect

TABLE 1. FDRs of comparison algorithms and SAE-MSPM.

	PCA		KPCA		SAE		SDAE		DBN		SAE-MSPM
	T^2	*SPE*	T^2	*SPE*	T^2	*SPE*	T^2	*SPE*	T^2	SPE	BIC
FAR	0.030	0.039	0.015	0.013	0.010	0.007	0.007	0.003	0.029	0.073	0.048
F1	0.992	0.997	0.993	0.999	0.991	0.998	0.991	0.998	0.128	0.998	**1.000**
F2	0.980	0.986	0.986	**0.989**	0.981	0.984	0.980	0.983	0.101	0.985	0.983
F3	0.002	0.009	0.029	0.144	0.010	0.018	0.008	0.011	0.133	**0.273**	0.128
F4	0.544	**1.000**	0.121	**1.000**	0.044	**1.000**	0.019	0.995	0.135	0.935	**1.000**
F5	0.225	0.254	0.268	0.529	0.240	0.248	0.235	0.235	0.164	0.425	**1.000**
F6	0.992	**1.000**	0.991	**1.000**	0.993	**1.000**	0.995	**1.000**	0.954	**1.000**	**1.000**
F7	**1.000**	**1.000**	**1.000**	**1.000**	0.381	**1.000**	0.368	**1.000**	0.226	**1.000**	**1.000**
F8	0.975	0.975	0.974	0.974	0.963	0.971	0.955	0.973	0.469	**0.993**	0.979
F9	0.006	0.019	0.031	0.038	0.025	0.015	0.013	0.013	0.129	**0.211**	0.133
F10	0.334	0.341	0.364	0.571	0.306	0.355	0.314	0.309	0.259	0.673	**0.919**
F11	0.570	0.845	0.199	0.770	0.224	0.731	0.129	0.705	0.100	0.769	**0.953**
F12	0.971	0.975	0.993	0.970	0.969	0.970	0.950	0.975	0.579	0.996	**0.999**
F13	0.940	0.955	0.950	0.958	0.929	0.953	0.928	0.949	0.515	0.956	**0.960**
F14	**1.000**	0.989	0.999	0.999	0.800	**1.000**	0.738	**1.000**	0.076	**1.000**	**1.000**
F15	0.012	0.027	0.041	0.219	0.026	0.046	0.014	0.020	0.108	**0.263**	0.165
F16	0.157	0.245	0.176	0.586	0.161	0.276	0.126	0.235	0.209	0.663	**0.963**
F17	0.741	0.892	0.778	0.955	0.733	0.933	0.679	0.926	0.515	0.930	**0.980**
F18	0.887	0.899	0.889	0.900	0.885	0.899	0.884	0.898	0.854	**0.918**	0.910
F19	0.139	0.280	0.199	0.575	0.050	0.114	0.003	0.266	0.115	0.206	**0.998**
F20	0.315	0.602	0.361	0.695	0.360	0.481	0.314	0.454	0.065	0.676	**0.916**
F21	0.264	0.430	0.440	0.509	0.315	0.498	0.299	0.466	0.001	0.534	**0.650**
Avg.	0.574	0.653	0.561	0.732	0.495	0.642	0.473	0.639	0.278	0.734	0.840

52 variables, including 33 measured variables and 19 component variables. The system simulates 21 process faults, which have been widely used to verify the performance of monitoring algorithms. Each fault data set contains 960 test samples, of which the 161st sample is the time of fault occurrence. In this paper, 33 measured variables are collected for modeling and testing. The structure of SAE is set to 33-27-20-14-8. According to the silhouette coefficient, the features in the new feature matrix composed of all hidden layer features and residuals are divided into 15 subspaces, as shown in Figure 4.

The comparison results of the proposed SAE-MSPM and several other algorithms (PCA, Kernel PCA (KPCA), SAE, stacked denoising autoencoder (SDAE), DBN) are shown in Table 1. The bold indicates the maximum FDR achieved on the corresponding fault. It can be concluded from Table 1 that the proposed SAE-MSPM has achieved 15 maximum FDRs among the 21 faults, which is much higher than other comparison algorithms. In addition, Table 1 shows that although the comparison algorithms achieved the highest FDRs on some faults (such as faults 1, 2, 4, 6, 7, 8, 12, 13, 14, 17 and18), the performance difference from SAE-MSPM is small. This is because these faults have a great impact on the process. Almost all process monitoring methods can effectively detect these faults. Faults 3, 9 and 15 have little interference to the process. At present, few algorithms can achieve high FDRs for these faults. For the remaining faults, SAE-MSPM greatly improves the detection results of these faults.

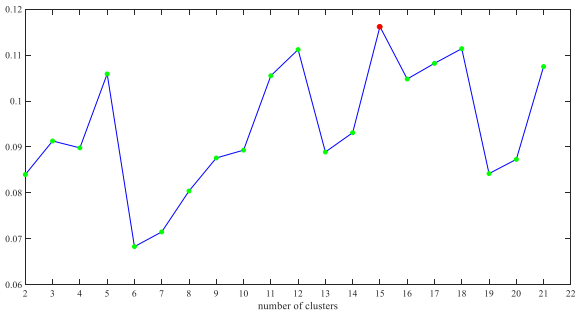

FIGURE 4. Silhouette coefficient values obtained from different cluster numbers.

For faults 10, 11, 16, 19, 20 and 21, Figure 5 shows the FDRs obtained by the comparison algorithms (blue histogram) and the percentage increase of FDRs by SAE-MSPM relative to the comparison algorithms (orange histogram). Figure 5 indicates that the monitoring performance of SAE-MSPM on these faults is significantly improved compared to the comparison algorithms. For example, the FDRs of PCA-based T^2 and *SPE* statistics for fault 19 are 0.139 and 0.280, respectively, while the FDR of SAE-MSPM for fault 19 reaches 0.998, achieving an improvement of 618% and 256%. The FDRs of SAE-based T^2 and *SPE* statistics for fault 20 is 0.360 and 0.481, respectively, while the FDR of SAE-MSPM for fault 20 reaches 0.916, achieving an improvement of 154% and 90%.

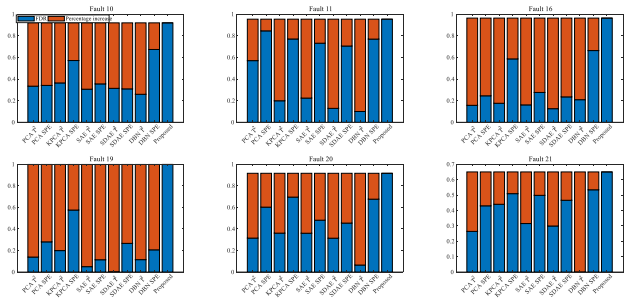

FIGURE 5. FDRs of algorithms (blue histogram) and percentage increase of FDRs by SAE-MSPM (orange histogram).

To further verify the monitoring performance of SAE-MSPM for 21 faults, Friedman's test was adopted. The results are shown in Figure 6. Friedman's test can comprehensively consider the detection results for 21 faults by different algorithms, and obtain the ranking of algorithm's performance. The smaller the ranking value is, the better the ranking of the algorithm and the higher the performance is. Friedman's test results show that SAE-MSPM ranks the highest in terms of the detection results for 21 faults, and has achieved the best process monitoring performance.

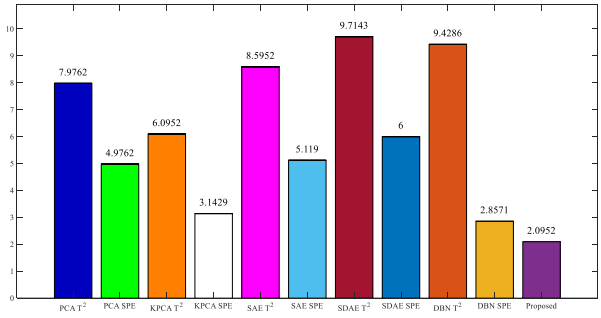

FIGURE 6. Ranking values of algorithms.

Take fault 5 and fault 10 as examples for further analysis. Fault 5 is a change of the condenser cooling water inlet temperature. Figure 7 (a), (b), (c) shows the monitoring charts of SAE, SDAE and SAE-MSPM on this fault, respectively, which indicate that SAE, SDAE and SAE-MSPM can quickly and accurately identify the occurrence of the fault in the initial stage of the fault. However, after about the 350th samples, the monitoring performance of SAE and SDAE for this fault decreased sharply, and almost all abnormal samples have failed to report. In fact, with the action of the control loop, most of the process variables will eventually return to the steady-state value, and only a small number of manipulated variables have changed due to compensation. SAE and SDAE cannot effectively detect such abnormal changes. Figure 7 (c) shows that SAE-MSPM can detect all 800 fault samples, further improving the fault detection performance.

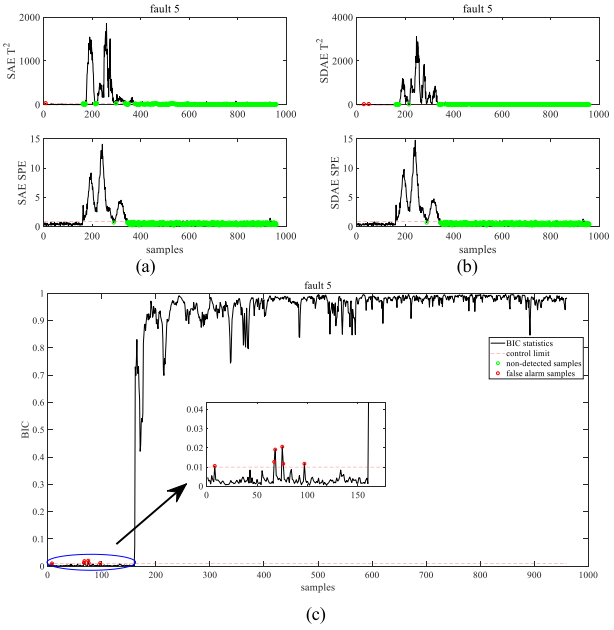

FIGURE 7. Monitoring charts for fault 5 in TE process: (a) SAE; (b) SDAE; (c) SAE-MSPM.

Fault 10 is the random disturbance in C feed temperature. Figure 8 shows the monitoring charts of SAE, SDAE and SAE-MSPM on this fault, respectively. Figure 8 (a) and (b) show that neither SAE nor SDAE can continuously and effectively alarm the fault, and only about 30% of the fault samples can be detected. Figure 8 (c) indicates that SAE-MSPM has greatly improved the FDR for this fault, and only a small number of fault samples have not been detected.

To further illustrate the necessity of retaining deep features of all hidden layers to establish a monitoring model, this paper adopts visualization technology (t-SNE) to show the information carried by the features in different hidden layers. Figure 9 shows the visualization results of features information extracted from different hidden layer. The red dot represents the extracted information of normal operating status, the green dot represents the extracted information of fault 1, the blue dot represents the extracted information of fault 2, and the black dot represents the extracted information of fault 4. Figure 9 indicates that the features in the first three hidden layers also contain information that is conducive to

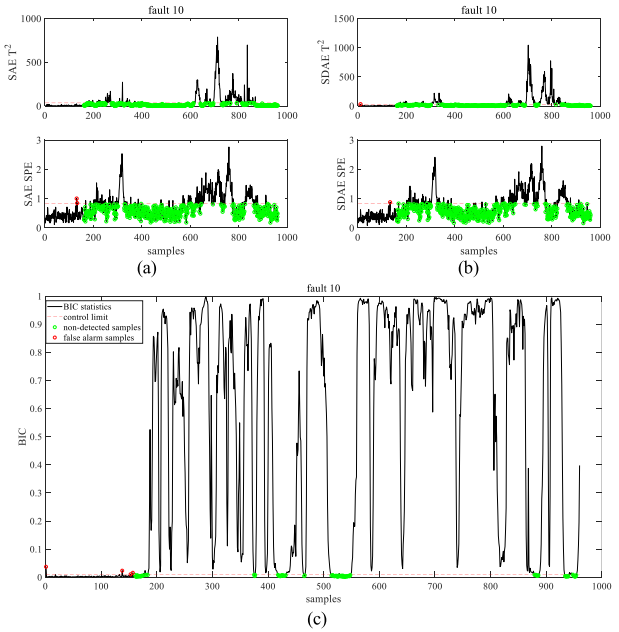

FIGURE 8. Monitoring charts for fault 10 in TE process: (a) SAE; (b) SDAE; (c) SAE-MSPM.

fault detection/identification of normal status and fault status, even better than that of the deepest features. By comparing Figure 9 (a) to Figure 9 (d), it can be further concluded that process data information may be lost with the layer-by-layer feature extraction and information transmission, resulting in poor performance of monitoring model constructed by the deep features. Therefore, comprehensive consideration of the features in all hidden layers can achieve the enrichment of fault related information.

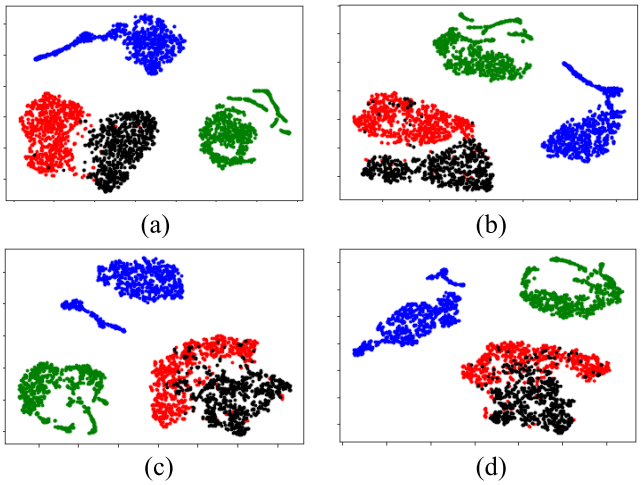

FIGURE 9. Visualization results of features information extracted from different hidden layer under the network structure of 33-27-20-14-8: (a) the 1st hidden layer; (b) the 2nd hidden layer; (c) the 3rd hidden layer; (d) the 4th hidden layer.

B. WASTEWATER TREATMENT PROCESS

Wastewater treatment process is a typical multivariable and nonlinear system. Monitoring the process can ensure the drainage quality and process safety. In this paper, BSM1 proposed by the International Water Association is used to verify the monitoring performance. The detailed introduction of this system can be referred to the reference [49]. This paper uses the eight variables suggested in reference [50] for modeling and testing. In addition, three kinds of faults are simulated to evaluate the monitoring performance. Fault 1: the kinetic parameter μ_A in the bioreactor increased from 0.5 to 0.8; Fault 2: the external carbon flow rate in the fifth reactor was randomly disturbed; Fault 3: a fixed deviation of S_{NO2} sensor. The structure of SAE is set to 8-10-8-6-4.

The FDRs of KPCA, SAE, SDAE, DBN and SAE-MSPM on these three faults are shown in Table 2. Compared with the three comparison algorithms, SAE-MSPM has greatly improved the detection ability for these three faults. For fault 1, the three comparison algorithms can only detect 49% of the fault samples at most, while SAE-MSPM can identify all the fault samples. For faults 2 and 3, KPCA, SDAE and DBN can hardly detect the occurrence of these two faults. The FDRs of the three methods for these two faults are less than 20%, and there are a large number of undetected samples. The monitoring performance of SAE based *SPE* statistics is better than that of KPCA, SDAE and DBN, but there is still a large gap from satisfactory performance. The FDRs of SAE-MSPM for fault 2 and fault 3 are 96% and 82%, respectively, which is much higher than the other comparison algorithms. This further verifies the advantages of SAE-MSPM in dealing with systems with complex nonlinear characteristics.

Furthermore, the performance of DNNs is often unstable due to the effects of random initialization. This paper also analyzes and compares the monitoring performance stability of SAE, SDAE and SAE-MSPM. Each algorithm is independently trained and tested for 10 times. Figure 10 shows the average FDRs and standard deviations of the three algorithms for the three faults, which indicates that SAE-MSPM not only obtains the highest FDR (maximum average value) but also has the most stable performance (minimum standard deviation). Actually, in the fault detection methods based on SAE and SDAE, only the deepest features and residuals are used to establish monitoring statistics. Different initialization will lead to certain differences in the features extracted by the model, leading to the instability of monitoring performance. SAE-MSPM considers the features in all hidden layers and preserves the information of the input data in each layer. It can not only avoid the information loss during the layer-by-layer feature extraction but also strengthen the stability of the information retention of the input data. This further explains why SAE-MSPM has high monitoring performance stability.

C. REAL WIND TURBINE POWER GENERATION PROECSS

Wind power generation has been widely used in the world, where wind turbines face the risk of blade icing. Therefore, timely and effective monitoring of the working status of blades is conducive to ensuring power generation efficiency and equipment safety. Twenty-six consecutive numerical

TABLE 2. FDRs of KPCA, SAE, SDAE and SAE-MSPM.

	KPCA		SAE		SDAE		DBN		SAE-MSPM
	T^2	*SPE*	T^2	*SPE*	T^2	*SPE*	T^2	*SPE*	BIC
F1	0.06	0.36	0.27	0.49	0.11	0.29	0.00	0.17	**1.00**
F2	0.02	0.11	0.14	0.61	0.04	0.11	0.11	0.01	**0.96**
F3	0.04	0.15	0.14	0.38	0.04	0.16	0.00	0.07	**0.82**
Avg.	0.04	0.21	0.18	0.49	0.06	0.19	0.04	0.08	0.93

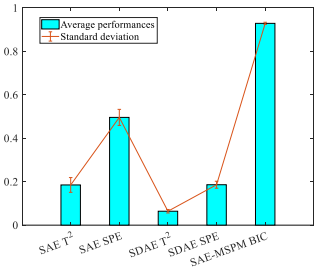

FIGURE 10. Average FDRs and standard deviation of SAE, SDAE and SAE-MSPM.

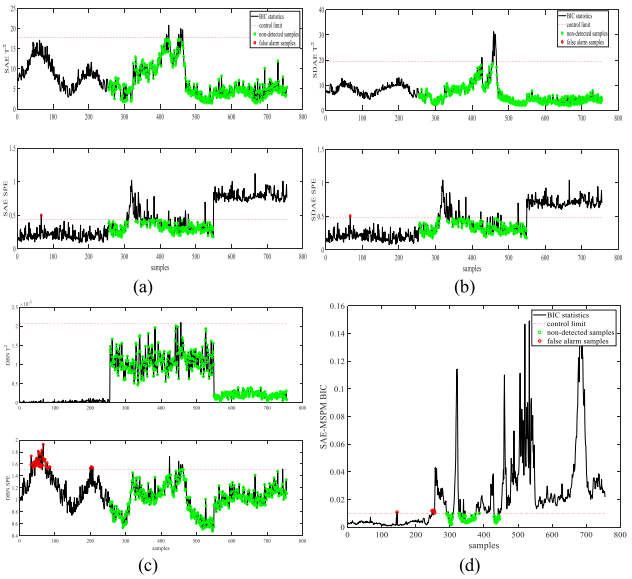

FIGURE 11. Monitoring charts for wind blade icing fault: (a) SAE; (b) SDAE; (c) DBN; (d) SAE-MSPM.

variables collected by SCADA system of a wind power plant is used for modeling and testing, which can be available from http://www.industrial-bigdata.com/datasets. The model is trained with 1000 normal samples, and the model is tested with 255 normal samples and 500 fault samples. The structure of SAE is set to 26-20-15-11-8.

The monitoring charts of SAE, SDAE, DBN and SAE-MSPM on the test data are shown in Figure 11. For 255 normal data, SAE and SDAE falsely reported only one normal sample as a fault state, and SAE-MSAE falsely reported 4 normal samples as a fault state. Although the FAR of SAE-MSAE is slightly higher than that of SAE and SDAE, the FAR is only 1.57%, which fully meets the industrial requirement. The FAR of *SPE* statistic based on DBN reached 17.25%, which cannot meet the requirements. Furthermore, T^2 statistics based on SAE and SDAE and monitoring statistics based on DBN can hardly detect fault samples, while *SPE* statistics based on SAE and SDAE can only detect about half of fault samples. This will seriously affect the power generation efficiency of the wind turbine and reduce the operating life of the wind turbine. Figure 11 (d) shows that SAE-MSPM can detect the vast majority of fault samples, and the FDR can reach 82.64%. Therefore, compared with SAE SDAE and DBN, the proposed SAE-MSPM can provide more accurate detection results for wind turbine blade icing status.

V. CONCLUSION

Regarding the problem that DNN based process monitoring methods usually only consider the features in the last layer while ignoring the features in other hidden layers, an ensemble monitoring model based on multi-subspace partition of deep features is proposed. The features in all hidden layers are fully considered to enrich useful information for process monitoring. Then, combined with the residuals, multi-subspace partition is achieved based on the rich diversity of deep features. The detection results of multiple subspaces are integrated through the Bayesian inference to realize process monitoring. Since the features in all hidden layers are fully considered, the problem of information loss can be effectively avoided. In addition, the features in the same subspace have a high similarity, and the features among different subspaces have a large diversity. Therefore, the features in the same subspace have similar monitoring behaviors for process variables, and the features in different subspaces have different monitoring behaviors. Integrating such multiple subspaces is beneficial to improve the generalization ability of the monitoring method. Case studies on TE process, WWTPs and an actual wind turbine power generation process demonstrated the effectiveness of SAE-MSPM.

Although this paper uses correlation analysis and cluster analysis to achieve the partition of multiple subspaces, it is not necessarily the best partition method. How to combine the known fault information to achieve appropriate partition of multiple subspaces and reduce the monitoring redundancy between subspaces is worth further study. In addition, how to establish interpretable models of subspaces and process variables and achieve fault tracing after detecting faults is also a future research work.

DATA AVAILABILITY

The code and data that support this study are available on request from the corresponding author.

REFERENCES

[1] K. Severson, P. Chaiwatanodom, and R. D. Braatz, "Perspectives on process monitoring of industrial systems," *Annu. Rev. Control*, vol. 42, pp. 190–200, Jan. 2016.

[2] K. Huang, K. Wei, Y. Li, and C. Yang, "Distributed dictionary learning for industrial process monitoring with big data," *Appl. Intell.*, vol. 51, no. 11, pp. 7718–7734, Nov. 2021.

[3] Q. Jiang, S. Yan, H. Cheng, and X. Yan, "Local–global modeling and distributed computing framework for nonlinear plant-wide process monitoring with industrial big data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 8, pp. 3355–3365, Aug. 2021.

[4] T. J. Rato and M. S. Reis, "An integrated multiresolution framework for quality prediction and process monitoring in batch processes," *J. Manuf. Syst.*, vol. 57, pp. 198–216, Oct. 2020.

[5] L. Yao and Z. Ge, "Industrial big data modeling and monitoring framework for plant-wide processes," *IEEE Trans. Ind. Informat.*, vol. 17, no. 9, pp. 6399–6408, Sep. 2021.

[6] Y. Zhou, Z. Cao, J. Lu, C. Zhao, D. Li, and F. Gao, "Objectives, challenges, and prospects of batch processes: Arising from injection molding applications," *Korean J. Chem. Eng.*, vol. 39, no. 12, pp. 3179–3189, Dec. 2022.

[7] M. Onel, C. A. Kieslich, Y. A. Guzman, C. A. Floudas, and E. N. Pistikopoulos, "Big data approach to batch process monitoring: Simultaneous fault detection and diagnosis using nonlinear support vector machine-based feature selection," *Comput. Chem. Eng.*, vol. 116, pp. 503–520, Jul. 2018.

[8] 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.

[9] C. Shang and F. You, "Data analytics and machine learning for smart process manufacturing: Recent advances and perspectives in the big data era," *Engineering*, vol. 5, no. 6, pp. 1010–1016, Dec. 2019.

[10] M. Fernandes, J. M. Corchado, and G. Marreiros, "Machine learning techniques applied to mechanical fault diagnosis and fault prognosis in the context of real industrial manufacturing use-cases: A systematic literature review," *Appl. Intell.*, vol. 52, no. 12, pp. 14246–14280, Sep. 2022.

[11] S. J. Qin, Y. Dong, Q. Zhu, J. Wang, and Q. Liu, "Bridging systems theory and data science: A unifying review of dynamic latent variable analytics and process monitoring," *Annu. Rev. Control*, vol. 50, pp. 29–48, Jan. 2020.

[12] A. Bakdi and A. Kouadri, "A new adaptive PCA based thresholding scheme for fault detection in complex systems," *Chemometric Intell. Lab. Syst.*, vol. 162, pp. 83–93, Mar. 2017.

[13] T. Rato, M. Reis, E. Schmitt, M. Hubert, and B. De Ketelaere, "A systematic comparison of PCA-based statistical process monitoring methods for high-dimensional, time-dependent processes," *AIChE J.*, vol. 62, no. 5, pp. 1478–1493, May 2016.

[14] Q. Jiang, X. Yan, and B. Huang, "Performance-driven distributed PCA process monitoring based on fault-relevant variable selection and Bayesian inference," *IEEE Trans. Ind. Electron.*, vol. 63, no. 1, pp. 377–386, Jan. 2016.

[15] S. Gajjar, M. Kulahci, and A. Palazoglu, "Real-time fault detection and diagnosis using sparse principal component analysis," *J. Process Control*, vol. 67, pp. 112–128, Jul. 2018.

[16] Y.-L. He, X. Yan, and Q.-X. Zhu, "Novel pattern recognition using bootstrap-based discriminant locality-preserving projection and its application to fault diagnosis," *Ind. Eng. Chem. Res.*, vol. 58, no. 38, pp. 17906–17917, Sep. 2019.

[17] J. E. Garcia-Bracamonte, J. M. Ramirez-Cortes, J. de Jesus Rangel-Magdaleno, P. Gomez-Gil, H. Peregrina-Barreto, and V. Alarcon-Aquino, "An approach on MCSA-based fault detection using independent component analysis and neural networks," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 5, pp. 1353–1361, May 2019.

[18] J. Huang, X. Yang, and X. Yan, "Slow feature analysis-independent component analysis based integrated monitoring approach for industrial processes incorporating dynamic and static characteristics," *Control Eng. Pract.*, vol. 102, Sep. 2020, Art. no. 104558.

[19] A. Apsemidis, S. Psarakis, and J. M. Moguerza, "A review of machine learning kernel methods in statistical process monitoring," *Comput. Ind. Eng.*, vol. 142, Apr. 2020, Art. no. 106376.

[20] J. M. B. de Lázaro, A. P. Moreno, O. L. Santiago, and A. J. da Silva Neto, "Optimizing kernel methods to reduce dimensionality in fault diagnosis of industrial systems," *Comput. Ind. Eng.*, vol. 87, pp. 140–149, Sep. 2015.

[21] N. M. Nor, M. A. Hussain, and C. R. C. Hassan, "Fault diagnosis and classification framework using multi-scale classification based on kernel Fisher discriminant analysis for chemical process system," *Appl. Soft Comput.*, vol. 61, pp. 959–972, Dec. 2017.

[22] Z. Ge, M. Zhang, and Z. Song, "Nonlinear process monitoring based on linear subspace and Bayesian inference," *J. Process Control*, vol. 20, no. 5, pp. 676–688, Jun. 2010.

[23] L. X. You and J. Chen, "A variable relevant multi-local PCA modeling scheme to monitor a nonlinear chemical process," *Chem. Eng. Sci.*, vol. 246, Dec. 2021, Art. no. 116851.

[24] Z. Li and X. Yan, "Fault-relevant optimal ensemble ICA model for non-Gaussian process monitoring," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 6, pp. 2581–2590, Nov. 2020.

[25] P. Kazemi, C. Bengoa, J.-P. Steyer, and J. Giralt, "Data-driven techniques for fault detection in anaerobic digestion process," *Process Saf. Environ. Protection*, vol. 146, pp. 905–915, Feb. 2021.

[26] Y. Xu, S.-Q. Shen, Y.-L. He, and Q.-X. Zhu, "A novel hybrid method integrating ICA-PCA with relevant vector machine for multivariate process monitoring," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 4, pp. 1780–1787, Jul. 2019.

[27] N. Li and Y. Yang, "Ensemble kernel principal component analysis for improved nonlinear process monitoring," *Ind. Eng. Chem. Res.*, vol. 54, no. 1, pp. 318–329, Jan. 2015.

[28] R. Fezai, M. Mansouri, O. Taouali, M. F. Harkat, and N. Bouguila, "Online reduced kernel principal component analysis for process monitoring," *J. Process Control*, vol. 61, pp. 1–11, Jan. 2018.

[29] K. Su, J. Liu, and H. Xiong, "Hierarchical diagnosis of bearing faults using branch convolutional neural network considering noise interference and variable working conditions," *Knowl.-Based Syst.*, vol. 230, Oct. 2021, Art. no. 107386.

[30] Q. Li, L. Li, W. Wang, Q. Li, and J. Zhong, "A comprehensive exploration of semantic relation extraction via pre-trained CNNs," *Knowl.-Based Syst.*, vol. 194, Apr. 2020, Art. no. 105488.

[31] J. Liu, X. Wang, S. Shen, G. Yue, S. Yu, and M. Li, "A Bayesian Q-learning game for dependable task offloading against DDoS attacks in sensor edge cloud," *IEEE Internet Things J.*, vol. 8, no. 9, pp. 7546–7561, May 2021.

[32] R. Arunthavanathan, F. Khan, S. Ahmed, and S. Imtiaz, "A deep learning model for process fault prognosis," *Process Saf. Environ. Protection*, vol. 154, pp. 467–479, Oct. 2021.

[33] H. Wu and J. Zhao, "Self-adaptive deep learning for multimode process monitoring," *Comput. Chem. Eng.*, vol. 141, Oct. 2020, Art. no. 107024.

[34] 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, Apr. 2020, Art. no. 106755.

[35] Z. Zhang, T. Jiang, S. Li, and Y. Yang, "Automated feature learning for nonlinear process monitoring—An approach using stacked denoising autoencoder and K-nearest neighbor rule," *J. Process Control*, vol. 64, pp. 49–61, Apr. 2018.

[36] Y. Lyu, J. Chen, and Z. Song, "Image-based process monitoring using deep learning framework," *Chemometric Intell. Lab. Syst.*, vol. 189, pp. 8–17, Jun. 2019.

[37] X. Liu, J. Yu, and L. Ye, "Residual attention convolutional autoencoder for feature learning and fault detection in nonlinear industrial processes," *Neural Comput. Appl.*, vol. 33, no. 19, pp. 12737–12753, Oct. 2021.

[38] X. Deng, L. Xiao, X. Liu, and X. Zhang, "One-dimensional residual GANomaly network-based deep feature extraction model for complex industrial system fault detection," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.

[39] Y. Wang, Z. Pan, X. Yuan, C. Yang, and W. Gui, "A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network," *ISA Trans.*, vol. 96, pp. 457–467, Jan. 2020.

[40] Y. Liu, H. Zhou, F. Tsung, and S. Zhang, "Real-time quality monitoring and diagnosis for manufacturing process profiles based on deep belief networks," *Comput. Ind. Eng.*, vol. 136, pp. 494–503, Oct. 2019.

[41] X. Yuan, B. Huang, Y. Wang, C. Yang, and W. Gui, "Deep learning-based feature representation and its application for soft sensor modeling with variable-wise weighted SAE," *IEEE Trans. Ind. Informat.*, vol. 14, no. 7, pp. 3235–3243, Jul. 2018.

[42] X. Yuan, C. Ou, Y. Wang, C. Yang, and W. Gui, "Deep quality-related feature extraction for soft sensing modeling: A deep learning approach with hybrid VW-SAE," *Neurocomputing*, vol. 396, pp. 375–382, Jul. 2020.

[43] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision—ECCV 2014*. Cham, Switzerland: Springer, 2014, pp. 818–833.

[44] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," 2015, *arXiv:1506.06579*.

[45] J. Yu and X. Yan, "Whole process monitoring based on unstable neuron output information in hidden layers of deep belief network," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3998–4007, Sep. 2020.
[46] Q. Jiang, X. Yan, and B. Huang, "Review and perspectives of data-driven distributed monitoring for industrial plant-wide processes," *Ind. Eng. Chem. Res.*, vol. 58, no. 29, pp. 12899–12912, Jul. 2019.
[47] H. Yi and Q. Jiang, "Discriminative feature learning for blade icing fault detection of wind turbine," *Meas. Sci. Technol.*, vol. 31, no. 11, Nov. 2020, Art. no. 115102.
[48] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Comput. Chem. Eng.*, vol. 17, no. 3, pp. 245–255, Mar. 1993.
[49] J. Alex, "Benchmark simulation model no. 1 (BSM1)," Lund Univ., Lund, Sweden, Tech. Rep., 2008.
[50] Z. Li and X. Yan, "Adaptive selective ensemble-independent component analysis models for process monitoring," *Ind. Eng. Chem. Res.*, vol. 57, no. 24, pp. 8240–8252, Jun. 2018.

ZHICHAO LI received the B.S. degree in automation and the Ph.D. degree in control theory and engineering from the East China University of Science and Technology, Shanghai, China, in 2016 and 2021, respectively.

He is currently a Lecturer with Shaoxing University, China. His research interests include multivariate statistical process monitoring, data-driven fault diagnosis, data mining, differential evolution algorithm, and machine learning.

LI TIAN received the B.S. degree in automation and the Ph.D. degree in control theory and engineering from the East China University of Science and Technology, Shanghai, China, in 2014 and 2021, respectively.

She is currently a Lecturer with Shaoxing University, China. Her current research interests include differential evolution algorithm, multi-objective optimization, data-driven process modeling, and their real-world applications.

XUEFENG YAN (Member, IEEE) received the B.S. degree in biochemical engineering and the Ph.D. degree in control theory engineering from Zhejiang University, Hangzhou, China, in 1995 and 2002, respectively.

He is currently a Professor with the East China University of Science and Technology, Shanghai, China. His research interests include complex chemical process modeling, optimizing and controlling, process monitoring, fault diagnosis, and intelligent information processing.