Feature-aligned Stacked Autoencoder: A Novel Semi-supervised Deep Learning Model for Pattern Classification of Industrial Faults

Xinmin Zhang, *Member, IEEE*, Hongyi Zhang, and Zhihuan Song

***Abstract*—Autoencoder is a widely used deep learning method, which first extracts features from all data through unsupervised reconstruction, and then fine-tunes the network with labeled data. However, due to the limited number of labeled data samples, the network may lack sufficient generalization ability and is prone to overfitting. This paper proposes a new semi-supervised deep learning method called feature-aligned stacked autoencoder (FA-SAE). FA-SAE takes advantage of the unlabeled data during the fine-tuning process by aligning the feature of both labeled and unlabeled data. In FA-SAE, a new training loss function is designed by integrating the Sinkhorn distance measure of the difference between the features extracted from labeled and unlabeled data through the neural network into the cross-entropy classification loss. The effectiveness of the proposed FA-SAE is verified through its application to two industrial processes, and the application results demonstrated that the proposed FA-SAE has better generalization ability and higher fault classification accuracy as compared to the state-of-the-art methods.**

***Impact Statement*—Autoencoder is a popular data-driven modeling technology in deep learning. It can deal with the nonlinear relationships among process variables, and has a powerful feature extraction ability. Autoencoder has been widely utilized for fault detection and fault diagnosis in a wide variety of industrial processes. However, due to the limited number of labeled data samples in practical applications, the network of autoencoder may lack sufficient generalization ability and is prone to overfitting. The proposed feature-aligned stacked autoencoder in this paper can overcome these limitations. The application results on two industrial processes demonstrated that the proposed technology has better generalization ability and higher fault classification accuracy as compared to the state-of-the-art methods. The proposed technology offers a good solution to the semi-supervised deep learning field and is ready to support users in a wide variety of applications.**

***Index Terms*—Autoencoder, Deep learning, Fault classification, Feature alignment, Semi-supervised learning**

I. Introduction

In the modern industries, large amounts of data have been generated at each process of production. Data is a key enabler to promote the development of smart manufacturing. Data can be used for monitoring [1], [2], [3], [4], prediction [5], [6], [7], [8], optimization [9], control [10], etc. Data-driven process monitoring is an important application of process data analysis techniques in manufacturing industry with the goal of ensuring the safety of industrial process. Fault classification is a part of process monitoring that can help us identify the types of faults, based on which the process can be recovered by taking corrective actions.

Typically, from the viewpoint of machine learning, the task of fault classification can be seen as a multi-class classification problem, and thus many machine learning methods, such as k-nearest neighbor (kNN) [11], principal component analysis (PCA) [12], fisher discriminant analysis (FDA) [13], Bayesian network [14], random forests [15], and artificial neural networks (ANNs) [16], [17], can be employed for fault classification application. Deep learning is part of a broader family of machine learning methods based on artificial neural networks with representation learning [18]. Since the use of deep learning architectures, deep learning algorithms can deal with the nonlinear relationships among process variables, and has a powerful feature extraction ability. In recent years, a series of deep learning algorithms have been proposed and successfully applied to various tasks [19], [20], [21]. Autoencoder [22] is a popular deep learning method, which uses a neural network as encoder to transform the input variables into the latent vector, and then tries to reconstruct the inputs with latent vector using another neural network which is regarded as decoder. Through the encode-decode process, autoencoder has the ability of learning high-level feature representation from data and dealing with the process nonlinearity. Autoencoder has been widely utilized for soft-sensor and fault classification. For example, Yuan et. al. [23] proposed a variable-wise weighted stacked autoencoder for soft-sensor modeling. Wang and Yan [24] proposed a soft-sensor modeling method based on the weighted maximal information coefficients and stacked autoencoder. Yuan et. al. [25] proposed a hierarchical quality relevant autoencoder for soft-sensor modeling. Tao et. al. [26] proposed a bearing fault diagnosis method based on stacked autoencoder. Sun et. al. [27] proposed a sparse autoencoder for fault classification. Lu et. al. [28] proposed a stacked denoising autoencoder-based health state identification method for fault diagnosis. Shen et. al. [29] proposed a contractive autoencoder for fault diagnosis. Luo et. al. [30] proposed a semi-supervised discriminant autoencoder for fault diagnosis. Note that the above autoencoder-based fault classification models have a limitation that they only use labeled data in the fine-tuning phase. When the number of labeled data is extremely limited, this would increase the risk of overfitting the labeled data.

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62003301, 61833014) and in part by the State Key Laboratory of Industrial Control Technology, Zhejiang University, China (Grant No. ICT2021A14). (*Corresponding author: Zhihuan Song.*)

Xinmin Zhang, Hongyi Zhang, 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: xinminzhang@zju.edu.cn, hongyizhang@zju.edu.cn, songzhihuan@zju.edu.cn).

However, in practice, obtaining sufficient labeled data is a difficult task, since it not only requires expert experiences and prior knowledge of the process, but also is expensive and time-consuming.

To solve this problem, this work proposes a novel semi-supervised deep learning modeling framework called feature-aligned stacked autoencoder (FA-SAE). Unlike the conventional semi-supervised stacked autoencoder which only uses label data in the fine-tuning stage, FA-SAE uses both unlabeled data and labeled data in the fine-tuning stage. In FA-SAE, a new training loss function is designed by integrating the Sinkhorn distance measure of the difference between the features extracted from labeled and unlabeled data through the neural network into the cross-entropy classification loss. By aligning the features of labeled and unlabeled data during the fine-tuning phase, the generalization ability of the network can be significantly enhanced. The effectiveness of the proposed FA-SAE method was verified through its application to an industrial benchmark process and a real rolling bearing process. The main contributions of this work are summarized as follows:

(1) A new semi-supervised deep learning method, FA-SAE, is developed. FA-SAE takes advantage of the unlabeled data during the fine-tuning process by aligning the feature of both labeled and unlabeled data.
(2) In FA-SAE, a new training loss function is designed by integrating the Sinkhorn distance measure of the difference between the features extracted from labeled and unlabeled data through the neural network into the cross-entropy classification loss.
(3) The proposed FA-SAE method is evaluated through an industrial benchmark process and a real-world rolling bearing dataset. The application results demonstrated that the proposed technology has better generalization ability and higher fault classification accuracy as compared to the state-of-the-art methods.

The rest of this paper is organized as follows. In Section 2, the basic theories of autoencoder, stacked autoencoder, and semi-supervised stacked autoencoder are briefly reviewed. In Section 3, the proposed feature-aligned stacked autoencoder (FA-SAE) model is presented in detail, and applied for fault classification. The results of applying the proposed FA-SAE on two industrial case studies are provided in Section 4. Finally, conclusions are summarized in Section 5.

II. Preliminaries

A. Autoencoder (AE)

AE is a type of neural network used to learn efficient data representations from a set of data [22]. The learned representations are also called codings or codes, and have been proven effective for subsequent classification and regression tasks. The basic architecture of an AE is presented in Fig. 1. An AE consists of an encoder and a decoder. The encoder uses a one-layer neural network to map the input vector $\boldsymbol{x}$ to a hidden variable $\boldsymbol{z}$, while the decoder uses a one-layer neural network to reconstruct the input $\boldsymbol{x}$ with the hidden variable $\boldsymbol{z}$. The encoder and decoder processes are defined as

$$\boldsymbol{z} = \sigma(\boldsymbol{w}_e\boldsymbol{x} + \boldsymbol{b}_e) \tag{1}$$

$$\tilde{\boldsymbol{x}} = \sigma(\boldsymbol{w}_d\boldsymbol{z} + \boldsymbol{b}_d) \tag{2}$$

where $\boldsymbol{w}_e$ is the weight matrix of the encoder, $\boldsymbol{w}_d$ is the weight matrix of the decoder, $\boldsymbol{b}_e$ and $\boldsymbol{b}_d$ denote the biases, $\tilde{\boldsymbol{x}}$ is the reconstruction of the input $\boldsymbol{x}$, and σ is a non-linear activation function. Generally, the model parameters of AE can be obtained by minimizing the reconstruction errors (often referred to as the "loss") using the stochastic gradient descent algorithm [31]. The loss function is defined as

$$loss_{AE} = \frac{1}{N}\sum_{i=1}^{N}\|\boldsymbol{x}_i - \tilde{\boldsymbol{x}}_i\|_2^2 \tag{3}$$

where N is the number of the training samples.

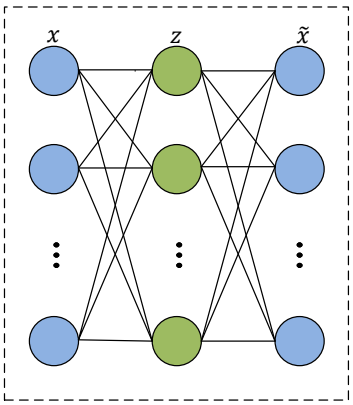

Fig. 1. Autoencoder.

B. Stacked autoencoder (SAE)

SAE is a deep version of the basic AE, which contains several layers of autoencoders where the output of each hidden layer is connected to the input of the successive hidden layer. The structure of SAE is depicted in Fig. 2, in which the decoder layers are drawn with dotted lines to illustrate that they do not actually exist. Commonly, a greedy layer-wise algorithm [32] is used to train SAE. For each iteration, an AE is trained using the output of the last encoder, and when the training process is finished, the encoder part of the trained AE is stacked to the last encoder. Through this stacking process, SAE can get a higher-level feature representation as compared to the basic AE. More specifically, the training process of the k-th AE in SAE can be calculated as

$$z_{k-1} = \sigma\Big(\boldsymbol{w}_e^{k-1}\big(\ldots\sigma(\boldsymbol{w}_e^2\sigma(\boldsymbol{w}_e^1\boldsymbol{x} + \boldsymbol{b}_e^1) + \boldsymbol{b}_e^2)\big) + \boldsymbol{b}_e^{k-1}\Big) \tag{4}$$

$$z_k = \sigma(\boldsymbol{w}_e^k z_{k-1} + \boldsymbol{b}_e^k) \tag{5}$$

$$\tilde{z}_{k-1} = \sigma(\boldsymbol{w}_d^k z_k + \boldsymbol{b}_d^k) \tag{6}$$

where z_{k-1} is the input of the k-th AE, $\{\boldsymbol{w}_e^k, \boldsymbol{b}_e^k\}$ and $\{\boldsymbol{w}_d^k, \boldsymbol{b}_d^k\}$ are the weight matrices and bias vectors of the encoder and decoder of the k-th AE, respectively. Similarly, the model

parameters of SAE can be obtained by minimizing the reconstruction errors using the stochastic gradient descent algorithm. The loss function of SAE is defined as

$$loss_{SAE} = \frac{1}{N}\sum_{i=1}^{N} \|\boldsymbol{z}_{k-1}^{i} - \tilde{\boldsymbol{z}}_{k-1}^{i}\|_2^2. \tag{7}$$

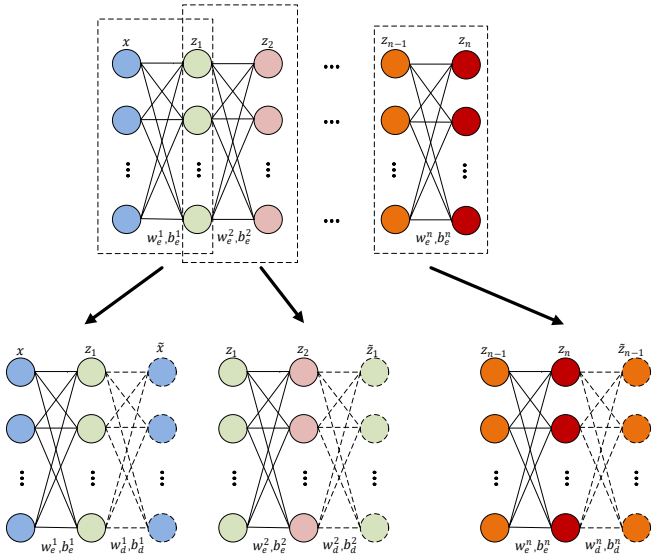

Fig. 2. Stacked autoencoder.

C. Semi-supervised stacked autoencoder (SS-SAE)

SAE is an unsupervised learning method, which can be easily extended to the semi-supervised learning methods. A semi-supervised form of the stacked autoencoder can be obtained by stacking a fully connected layer on the pre-trained SAE, as shown in Fig. 3. In SS-SAE, the labeled data are used to fine-tune the whole network. The feedforward process of SS-SAE is given by

$$\boldsymbol{z}_k = f_{sae}(\boldsymbol{x}) = \sigma\Big(\boldsymbol{w}_e^k\big(\ldots\sigma(\boldsymbol{w}_e^2\sigma(\boldsymbol{w}_e^1\boldsymbol{x} + \boldsymbol{b}_e^1) + \boldsymbol{b}_e^2)\big) + \boldsymbol{b}_e^k\Big) \tag{8}$$

$$\boldsymbol{p} = \text{softmax}(\boldsymbol{w}_c\boldsymbol{z}_k + \boldsymbol{b}_c) \tag{9}$$

where $\{\boldsymbol{w}_c, \boldsymbol{b}_c\}$ denote the weight matrix and bias vectors of the fully connected layer, $\boldsymbol{p} = \{p^f\}_{f=1}^{F}$ denotes the probabilities of the predicted sample belonging to each category, F denotes the number of categories, and softmax$(\cdot)$ denotes the softmax function. The purpose of using softmax function is to normalize the output of the network to a probability distribution over predicted output classes, and softmax$(\cdot)$ can be defined as

$$\text{softmax}(\boldsymbol{o})_f = \frac{\exp(o^f)}{\sum_{h=1}^{F}\exp(o^h)}. \tag{10}$$

By applying the standard exponential function to each element o^f of the input vector $\boldsymbol{o}$ and normalizing these values by dividing by the sum of all these exponentials, the softmax function can ensure that each component will be in the interval $[0, 1]$, and the components will add up to 1, so that they can be interpreted as probabilities. The model parameters of SS-SAE can be obtained by minimizing the cross-entropy loss using the stochastic gradient descent algorithm [22]. The cross-entropy loss is calculated using labeled data as follows

$$\begin{aligned} loss_{SS-SAE} &= \frac{1}{N_l}\sum_{i=1}^{N_l} \text{cross\_entropy}(\boldsymbol{y}_i, \boldsymbol{p}_i) \\ &= \frac{1}{N_l}\sum_{i=1}^{N_l}\sum_{f=1}^{F} -y_i^f \log p_i^f \end{aligned} \tag{11}$$

where N_l denotes the number of labeled training samples, p_i^f represents the probability of the i-th sample belonging to f-th category, and y_i^f represents the corresponding target value.

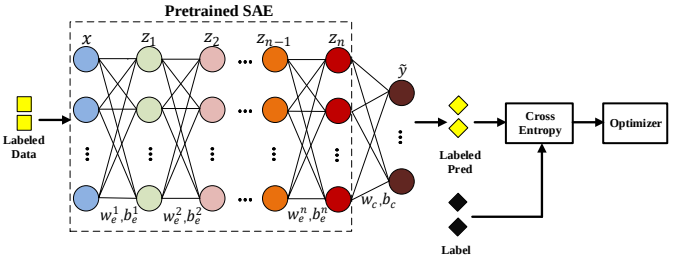

Fig. 3. Semi-supervised stacked autoencoder.

III. Feature-aligned stacked autoencoder (FA-SAE)

Note that SS-SAE has a limitation that it only uses labeled data for training in the fine-tuning stage. When the number of labeled data is extremely limited, this would increase the risk of overfitting the labeled data. To handle this issue, this work proposes a novel SS-SAE modeling framework called feature-aligned stacked autoencoder (FA-SAE), and applies it to fault classification. Similar to the traditional SS-SAE, the training process of the proposed FA-SAE can also be divided into unsupervised pre-training and supervised fine-tuning stages. However, unlike the conventional SS-SAE which only uses labeled data in the fine-tuning stage, FA-SAE uses both unlabeled data and labeled data in the fine-tuning stage in order to improve the network's generalization ability. The basic principle of FA-SAE is as follows. If the labeled and unlabeled data belong to the same class, the features extracted from the labeled and unlabeled data through the network model should also belong to the same distribution, when the network model has strong generalization ability. To consider this assumption, FA-SAE designed a new training loss function that integrates the Sinkhorn distance measure of the difference of the extracted feature distributions from labeled and unlabeled data into the cross-entropy classification loss. By aligning the features of labeled and unlabeled data during the fine-tuning phase, the generalization ability of the network can be significantly enhanced. The framework of FA-SAE is shown in Fig. 4. By comparing Fig. 3 and Fig. 4, it can be clearly seen that the main difference between SS-SAE and FA-SAE is that the proposed FA-SAE designs a new training loss function, which can make full use of unlabeled data information in the fine-tuning stage.

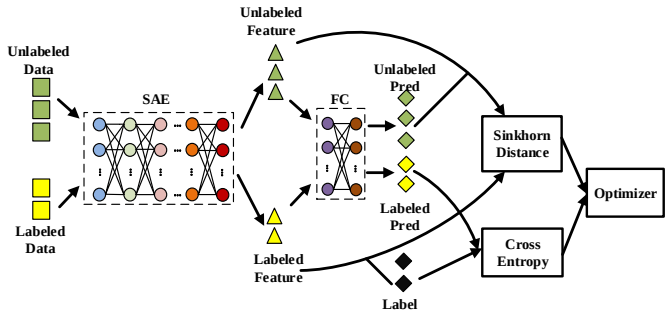

Fig. 4. Framework of FA-SAE.

The training procedures of the proposed FA-SAE consist of an unsupervised pre-training stage and a supervised fine-tuning stage. The unsupervised pre-training method is similar to the pre-training method mentioned in Section II-B. Firstly, a SAE is trained as the initialization weights of the network. After the input passes the trained SAE, it is mapped to the deep hidden variables, which can be calculated as

$$z_k = f_{sae}(\boldsymbol{x}) \tag{12}$$

where f_{sae} denotes the trained SAE. Then, a fully connected layer is stacked on the pre-training SAE, and the output can be calculated as

$$\tilde{y} = \text{argmax}(\boldsymbol{w}_c f_{sae}(\boldsymbol{x}) + \boldsymbol{b}_c). \tag{13}$$

More specifically, assuming that the training data includes labeled samples $\boldsymbol{S}_l = \{\boldsymbol{x}_l^i, y_i\}_{i=1}^m$ and unlabeled samples $\boldsymbol{S}_u = \{\boldsymbol{x}_u^j\}_{j=1}^n$. The high-level hidden variables $\boldsymbol{Z}_l = \{\boldsymbol{z}_l^i\}_{i=1}^m$ and predicted outputs $\tilde{\boldsymbol{Y}}_l = \{\tilde{y}_l^i\}_{i=1}^m$ of the labeled data can be calculated as

$$\boldsymbol{z}_l^i = f_{sae}(\boldsymbol{x}_l^i) \tag{14}$$

$$\tilde{y}_l^i = \text{argmax}(\boldsymbol{w}_c \boldsymbol{z}_l^i + \boldsymbol{b}_c). \tag{15}$$

Similarly, the high-level hidden variables $\boldsymbol{Z}_u = \{\boldsymbol{z}_u^j\}_{j=1}^n$ and predicted outputs $\tilde{\boldsymbol{Y}}_u = \{\tilde{y}_u^i\}_{j=1}^n$ of the unlabeled data can be calculated as

$$\boldsymbol{z}_u^j = f_{sae}(\boldsymbol{x}_u^j) \tag{16}$$

$$\tilde{y}_u^j = \text{argmax}(\boldsymbol{w}_c \boldsymbol{z}_u^j + \boldsymbol{b}_c). \tag{17}$$

For each class $f \in F$, the features extracted from the unlabeled data and labeled data are denoted as $\boldsymbol{Z}_u^f = \{z_u^f, where\ \tilde{y}_u^j = f\}$ and $\boldsymbol{Z}_l^f = \{z_l^i, where\ y_i = f\}$, respectively. Assuming that the features of the labeled data and unlabeled data that belong to the same class after passing through the neural network should also belong to the same distribution. Based on this assumption, we can measure the distance between the distributions of those two features. In this work, Sinkhorn distance [33], an accelerated version of optimal transport distance, is used to measure the difference between two distributions. The optimal transport distance [34], also known as earth mover distance or Wasserstein distance, is a typical method of measuring the difference between two distributions. Compared with KL divergence or Hellinger distance that requires a probability density function to calculate the distribution distance, the optimal transport distance can be calculated directly in a metric space. More specifically, the optimal transport distance is generally defined as follows: given two distributions $\boldsymbol{R} : \{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_m\}$ and $\boldsymbol{C} : \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_n\}$, a cost matrix $\boldsymbol{M}$ between distribution $\boldsymbol{R}$ and distribution $\boldsymbol{C}$ can be calculated as

$$\boldsymbol{M} = \begin{pmatrix} d_{11} & d_{12} & \ldots & d_{1n} \\ d_{21} & d_{22} & \ldots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & \ldots & d_{mn} \end{pmatrix} \tag{18}$$

where d_{ij} denotes the distance of $\boldsymbol{r}_i$ to $\boldsymbol{c}_j$. Generally, the Euclidean distance metric space is selected, and the distance function d_{ij} can be calculated as

$$d_{ij} = \|\boldsymbol{r}_i - \boldsymbol{c}_j\|_2. \tag{19}$$

Given a transition matrix or joint distribution matrix $\boldsymbol{P}$

$$\boldsymbol{P} = \begin{pmatrix} p_{11} & p_{12} & \ldots & p_{1n} \\ p_{21} & p_{22} & \ldots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{m1} & p_{m2} & \ldots & p_{mn} \end{pmatrix} \tag{20}$$

where $p_{ij} = \boldsymbol{P}\{\boldsymbol{R} = \boldsymbol{r}_i, \boldsymbol{C} = \boldsymbol{c}_j\}$. Then, the distance of mapping the distribution $\boldsymbol{R}$ to $\boldsymbol{C}$ is the dot product of $\boldsymbol{P}$ and $\boldsymbol{M}$, which can be written as $\langle \boldsymbol{P}, \boldsymbol{M} \rangle$. The optimal transport distance tries to find an optimal joint distribution matrix $\boldsymbol{P}$ such that $\langle \boldsymbol{P}, \boldsymbol{M} \rangle$ is the smallest. This problem is called an *optimal transportation* problem, which can be formulated as

$$d_M(\boldsymbol{R}, \boldsymbol{C}) = \min_{\boldsymbol{P} \in \boldsymbol{U}(\boldsymbol{R}, \boldsymbol{C})} \langle \boldsymbol{P}, \boldsymbol{M} \rangle. \tag{21}$$

Note that the calculation of (21) is time-consuming. Suppose that each of $\boldsymbol{R}$ and $\boldsymbol{C}$ contains d samples, the complexity of calculating (21) is $O(d^3 \log d)$. To speed up the calculation of the optimal transport distance, Cuturi [33] proposed an approximate calculation method called Sinkhorn distance, which adds an entropy regularization term to the traditional optimal transport distance, and then uses Sinkhorn's matrix scaling algorithm [35] to calculate the distance. For $\lambda > 0$, Sinkhorn distance is defined as

$$d_M^\lambda(\boldsymbol{R}, \boldsymbol{C}) = \langle \boldsymbol{P}^\lambda, \boldsymbol{M} \rangle \tag{22}$$

where

$$\boldsymbol{P}^\lambda = \arg \min_{\boldsymbol{P} \in \boldsymbol{U}(\boldsymbol{R}, \boldsymbol{C})} \langle \boldsymbol{P}, \boldsymbol{M} \rangle - \frac{1}{\lambda} h(\boldsymbol{P}). \tag{23}$$

$h(\boldsymbol{P}) = \sum_{ij} p_{ij} \log p_{ij}$ denotes an entropy regularization term. After adding the entropy regularization term into the optimal transport distance, the joint distribution matrix $\boldsymbol{P}$ can be approximated by a linear iterative algorithm, and thus the *optimal transportation* problem can be optimized directly by the gradient backpropagation method. The detailed calculation process of d_M is shown in Algorithm 1.

Based on (22), the Sinkhorn distance measure of the difference between the features extracted from labeled and unlabeled data through the stacked autoencoder can be calculated

as

$$\text{sinkhorn\_distance}(\boldsymbol{Z}_l^f, \boldsymbol{Z}_u^f) = \frac{1}{F}\sum_{f=1}^{F} d_M^{\lambda}(\boldsymbol{Z}_l^f, \boldsymbol{Z}_u^f). \tag{24}$$

By integrating the Sinkhorn distance measure of the feature distributions of the labeled and unlabeled data into the cross-entropy classification loss, a new training loss function in the fine-tuning process can be obtained as follows:

$$\begin{aligned} Loss_{FA-SAE} = \frac{1}{m}\sum_{i=1}^{m} \text{cross\_entropy}(\boldsymbol{y}_i, \boldsymbol{p}_l^i) \\ +\alpha \text{sinkhorn\_distance}(\boldsymbol{Z}_l^f, \boldsymbol{Z}_u^f) + \beta\|\Theta\|_2^2 \end{aligned} \tag{25}$$

with

$$\boldsymbol{p}_l^i = \text{softmax}(\boldsymbol{w}_c f_{sae}(\boldsymbol{x}_l^i) + \boldsymbol{b}_c) \tag{26}$$

where α is the ratio of the Sinkhorn distance, Θ denotes the network parameters, $\|\Theta\|_2^2$ denotes the L_2-norm regularization term that is used to prevent overfitting of the network, and β denotes the ratio factor of the L_2-norm. In this work, the ratio of cross-entropy loss is set to 1 by default. Similarly, the stochastic gradient descent algorithm is used to minimize the loss function of FA-SAE. The procedures of the proposed FA-SAE for fault classification are summarized in Algorithm 2.

Algorithm 1 The calculation of Sinkhorn distance.

Input: $\boldsymbol{M}, \lambda, \boldsymbol{R}, \boldsymbol{C}$
$\boldsymbol{I} = (\boldsymbol{R} > 0); \boldsymbol{R} = \boldsymbol{R}(I); \boldsymbol{M} = \boldsymbol{M}(I, :); \boldsymbol{K} = e^{-\lambda \boldsymbol{M}}$
$\boldsymbol{u} = ones(length(\boldsymbol{R}), n)/length(\boldsymbol{R})$
$\tilde{\boldsymbol{K}} = diag(1./\boldsymbol{R})\boldsymbol{K}$
while $\boldsymbol{u}$ changes or other relevant stopping criterion **do**
  $\boldsymbol{u} = 1./(\tilde{\boldsymbol{K}}(\boldsymbol{C}./(\boldsymbol{K}'\boldsymbol{u})))$
end while
$\boldsymbol{v} = \boldsymbol{C}./(\boldsymbol{K}'\boldsymbol{u})$
$d_M = sum(\boldsymbol{u}.*(\boldsymbol{K}.*\boldsymbol{M})\boldsymbol{v})$
Output: d_M

Algorithm 2 FA-SAE for fault classification.

1: Given the labeled dataset $\boldsymbol{S}_l = \{\boldsymbol{x}_l^i, y_i\}_{i=1}^m$ and unlabeled dataset $\boldsymbol{S}_u = \{\boldsymbol{x}_u^j\}_{j=1}^n$.
2: Divide the data into training set and test set, and normalize the data.
3: Train a SAE model on the training set as the initialization parameters of the network.
4: Add a fully connected layer on the trained SAE model and train FA-SAE.
5: Predict the class of the query sample using the trained FA-SAE model.

IV. Case studies

In this section, the effectiveness of the proposed FA-SAE method was verified through its application to an industrial benchmark process and a real rolling bearing process, and its application results are compared with support vector machine (SVM) [36], multi-layer perceptron (MLP) [37], semi-supervised stacked autoencoder (SS-SAE), semi-supervised sparse SAE (SS-SSAE) [38], and semi-supervised denoised SAE (SS-DSAE) [39]. SVM and MLP are supervised methods. SS-SAE, SS-SSAE, SS-DSAE, and the proposed FA-SAE are semi-supervised methods.

A. TE process

In this section, an industrial benchmark of Tennessee Eastman (TE) process is first used to evaluate the proposed FA-SAE method. The TE simulation process was first proposed by Downs and Vogel [40], and created by the Eastman Chemical Company to provide a realistic industrial process for evaluating process control techniques. The TE process has been widely used by the process control community as a source of data for testing various process monitoring and fault diagnosis methods [41], [42]. The flowsheet of the TE process is shown in Fig. 5. The TE process consists of 5 operation units including a two-phase reactor, a product condenser, a flash separator, a recycle compressor, and a product stripper. The entire process includes 42 measured variables and 12 manipulated variables.

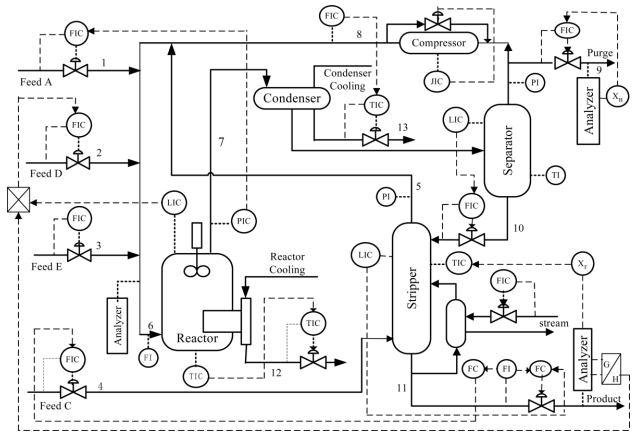

Fig. 5. Flowsheet of TE process.

The TE process simulator recommended in [43] was implemented to generate the process data for each fault. In this experiment, the collected data include one normal class and seven fault classes. The detailed fault descriptions and modeling variables are summarized in Table I and Table II, respectively. The sampling rate is 1000 points per hour, and the fault occurs at the 3001th point. In each class, 1100 data samples were collected to construct the model. That is, for each fault class, the 3001th to 4100th faulty samples are used. Similarly, for the normal class, the 3001th to 4100th normal samples are used. Thus, a total of 8800 samples were collected. To build the model, the collected data were divided into the training dataset and the testing dataset. In each class, 1100 data samples were randomly divided into 1000 unlabeled samples and 100 labeled samples. Among them, 1050 samples (1000 unlabeled samples+50 labeled samples) in each class

are divided into the training dataset, and the remaining 50 labeled samples are divided into the testing dataset. Thus, the training dataset totally consists of 8000 unlabeled samples ($1000 \times 8 = 8000$) and 400 labeled samples ($50 \times 8 = 400$), and the testing dataset totally consists of 400 labeled samples ($50 \times 8 = 400$). Table III shows the average classification accuracy and standard deviation of each method with the increasing of unlabeled training samples. In Table III, N_u denotes the number of unlabeled training samples. To make the results more convincing, we repeated each experiment 20 times. The network architecture and parameter settings used in FA-SAE are summarized in Table IV. From Table III, it can be seen that compared with supervised models (SVM and MLP), semi-supervised models (SS-SAE, SS-DSAE, SS-SSAE, and FA-SAE) have an advantage in classification accuracy due to the introduction of unlabeled data for training. Compared with the traditional autoencoder-based semi-supervised algorithms (SS-SAE, SS-DSAE, and SS-SSAE), FA-SAE achieved higher classification accuracy due to the introduction of feature-aligned strategy in the modeling process. For comparison, Fig. 6 shows the detailed classification results on the testing dataset by SS-SAE and FA-SAE, where the samples located in the ranges of 0-50, 50-100, 100-150, 150-200, 200-250, 250-300, 300-350, 350-400 should belong to normal condition and faults 1-7, respectively. From Fig. 6, it can be seen that FA-SAE achieved better classification accuracy than SS-SAE. Regarding the stability of the proposed method, it can be found that the standard deviation of the proposed FA-SAE method is at the same level as other methods, which confirms that the stability of the FA-SAE method is similar to the traditional semi-supervised methods.

TABLE I
FAULT DESCRIPTIONS IN TE PROCESS

Fault No.	Process variable	Type
Normal	—	—
Fault 1	A/C feed ratio, B composition constant (stream 4)	Step
Fault 2	B composition, A/C ratio constant (stream 4)	Step
Fault 3	Condenser cooling water inlet temperature	Step
Fault 4	C header pressure loss-reduced availability	Step
Fault 5	A,B,C feed composition (steam 4)	Random variation
Fault 6	Condenser cooling water inlet temperature	Random variation
Fault 7	Reactor cooling water value	Sticking

TABLE II
TE PROCESS VARIABLES.

No.	Measured variables	No.	Measured variables
1	A feed	9	Product separator temperature
2	D feed	10	Product separator pressure
3	E feed	11	Product separator underflow
4	A and C Feed	12	Stripper pressure
5	Recycle flow	13	Stripper temperature
6	Reactor feed rate	14	Stripper steam flow
7	Reactor temperature	15	Reactor cooling water outlet temperature
8	Purge rate	16	Separator cooling water outlet temperature

To further explore the effectiveness of the proposed FA-SAE, the quality of the features learned by FA-SAE is compared with the features learned by SS-SAE. Firstly, the

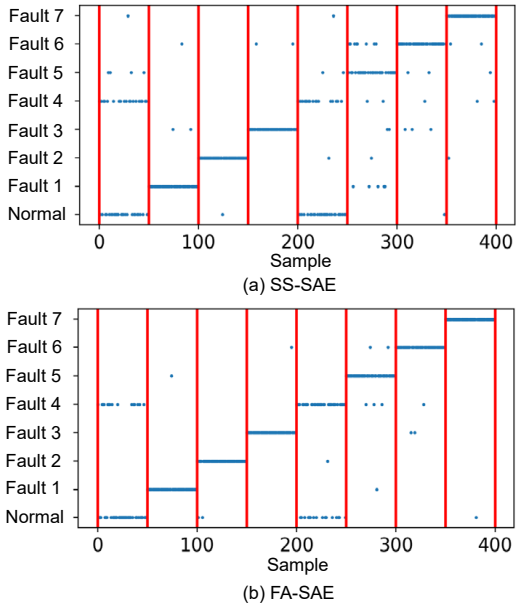

Fig. 6. Classification results of SS-SAE and FA-SAE.

features learned by SS-SAE and FA-SAE on the labeled data are shown in Fig. 7, in which different colors stand for different types of faults. In Fig. 7, a dimensionality reduction technique called PCA [44] is used to produce the visualization results. From Fig. 7, it can be found that the quality of the features learned by FA-SAE is much better than SS-SAE, that is, the features of different categories learned by FA-SAE are more separable and the features of the same category are more concentrated. This is further demonstrated why the proposed FA-SAE model can get better classification performance. The superiority of the proposed FA-SAE in learning good features can also be illustrated from Fig. 8, in which the green curve represents the feature distribution of unlabeled data, and the red curve represents the feature distribution of labeled data. For convenience, Fig. 8 only shows the first three dimensions of the feature distribution of SS-SAE and FA-SAE on labeled and unlabeled data. From Fig. 8, it can be seen that compared with SS-SAE, the feature distribution of FA-SAE for labeled and unlabeled data is much closer. This is due to the fact that the SS-SAE model only uses labeled data for fine-tuning and ignores the informative information of unlabeled data, making the feature distribution of labeled and unlabeled data more different than that of FA-SAE.

To show the superiority of the proposed FA-SAE in alleviating the over-fitting problem, the cross-entropy losses of SS-SAE and FA-SAE on the training dataset and testing dataset are shown in Fig. 9. As can be seen from Fig. 9, SS-SAE has a serious over-fitting problem, although the L_2-norm regularization term was added to the training loss function to prevent over-fitting. In contrast, the over-fitting problem was significantly alleviated by FA-SAE. The better generalization ability of the proposed FA-SAE can be attributed to the fact that both labeled and unlabeled data were used simultaneously in the fine-tuning process.

TABLE III
CLASSIFICATION ACCURACY OF SIX ALGORITHMS WITH DIFFERENT NUMBER OF UNLABELED TRAINING SAMPLES IN TE PROCESS.

N_u	SVM	MLP	SS-SAE	SS-DSAE	SS-SSAE	FA-SAE
0	0.7850±0	0.7794±0.0083				
2000			0.7906±0.0120	0.7941±0.01054	0.8085±0.0095	0.8518±0.0087
4000			0.7933±0.0135	0.7996±0.01184	0.8114±0.0099	0.8611±0.0084
8000			0.7985±0.0134	0.8128±0.01031	0.8093±0.0090	0.8634±0.0092

TABLE IV
NETWORK ARCHITECTURE AND PARAMETER SETTINGS OF FA-SAE IN TE PROCESS.

Input layer size	Three hidden layers size	Output layer size	Activation function	α	β
16	$70 \times 100 \times 70$	8	Tanh	0.05	0.0001

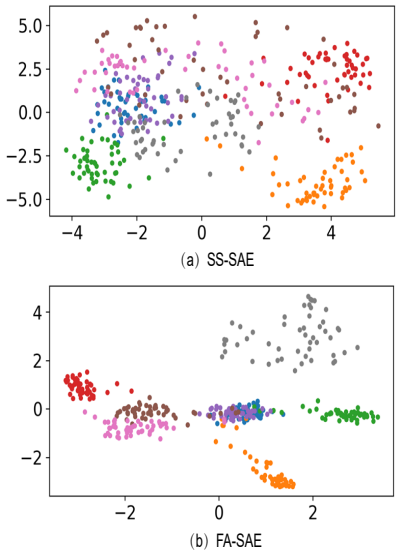

Fig. 7. Extracted features of SS-SAE and FA-SAE on labeled TE data. Figure legends: eight colored dots ('blue','green','orange','saddlebrown','darkorchid','violet','crimson','grey') correspond to the fault No. in Table I.

B. Rolling bearing process

In this section, the effectiveness of the proposed FA-SAE was further demonstrated by the CWRU motor bearing dataset, which is provided by the Western Reserve University (CWRU) Bearing Data Center [45]. The CWRU data were collected from normal and faulty bearings. The bearing system consists of a dynamometer, a torque transducer, and a 2 hp motor. Different kinds of faults were introduced to the test bearings using electro-discharge machining with fault diameters of 0.007 inches, 0.014 inches, and 0.021 inches. The fault positions are located in the inner race, the ball, and the outer race. The detailed fault descriptions of the CWRU dataset are shown in Table V. The measured variables in CWRU data include fan end and drive end vibration signals as well as motor rotational speed. To construct the classification models, since the original data is the raw vibration signals, a data preprocessing technique of wavelet packet analysis [46] is first used to process the original data. After preprocessing, the data has eight variables. By using 10 labeled and 40 unlabeled samples of each class, a total of 170 labeled data and 680 unlabeled data are applied for model training, and another dataset which contains 850 labeled samples is used for model testing.

TABLE V
FAULT DESCRIPTIONS IN BEARING PROCESS.

Fault No.	Fault Diameter	Motor Load	Fault position
Normal	0.007	0	
IR007_0	0.007	0	Inner Race
IR007_1	0.007	1	Inner Race
IR014_0	0.014	0	Inner Race
IR014_1	0.014	1	Inner Race
IR021_0	0.021	0	Inner Race
IR021_1	0.021	1	Inner Race
IR007_0	0.007	0	Ball
IR007_1	0.007	1	Ball
IR014_0	0.014	0	Ball
IR014_1	0.014	1	Ball
IR021_0	0.021	0	Ball
IR021_1	0.021	1	Ball
OR007@6_0	0.007	0	Outer Race centered
OR007@6_1	0.007	1	Outer Race centered
OR014@6_0	0.014	0	Outer Race centered
OR021@6_0	0.021	0	Outer Race centered

Table VI provides the 20 times average classification accuracy and standard deviation of each method with the increasing of unlabeled training samples. In this experiment, the network architecture and parameter settings used in FA-SAE are summarized in Table VII. As shown in Table VI, the semi-supervised methods (SS-SAE, SS-DSAE, SS-SSAE, and FA-SAE) provided higher classification accuracy than the supervised algorithms (SVM and MLP). As a result, FA-SAE obtained the best classification performance. Furthermore, as shown in Table VI, the standard deviation of the proposed FA-SAE method is at the same level as other methods, indicating that the stability of the proposed FA-SAE method is similar to the traditional semi-supervised methods.

Similar to the case study 1, Fig. 10 shows the learned features of SS-SAE and FA-SAE based on the PCA visualization technique on labeled CWRU data, and Fig. 11 shows the feature distributions of SS-SAE and FA-SAE on labeled and unlabeled CWRU data. From Fig. 10, it can be seen that the features of different types of faults have been well separated in FA-SAE, and the features within the same category are more concentrated. Also, the feature distributions of FA-SAE for labeled and unlabeled data are closer than that of SS-SAE, as shown in Fig. 11.

The results of two case studies have demonstrated that the proposed FA-SAE is significantly superior to the conventional methods. In FA-SAE, the Sinkhorn distance plays an important role. To discuss the impact of Sinkhorn distance ratio α on the model performance, the classification accuracy of the FA-SAE model with 170 unlabeled samples and different values of α were evaluated in the CWRU bearing example, as shown in Fig. 12. Figure 12 indicates that the value of the ratio

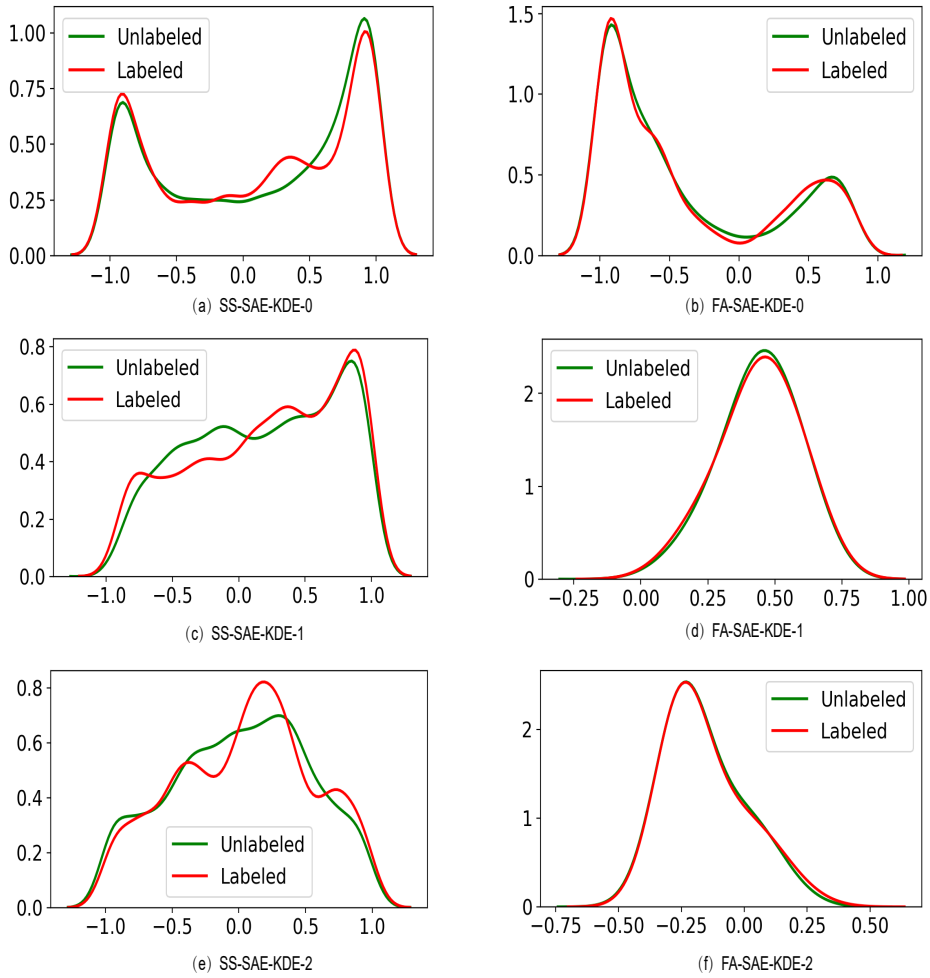

Fig. 8. Feature distribution of SS-SAE and FA-SAE on labeled and unlabeled TE data.

TABLE VI
CLASSIFICATION ACCURACY OF SIX ALGORITHMS WITH DIFFERENT NUMBER OF UNLABELED SAMPLES IN BEARING PROCESS.

N_u	SVM	MLP	SS-SAE	SS-DSAE	SS-SSAE	FA-SAE
0	0.8705±0	0.8751±0.0061				
170			0.8762±0.0063	0.8803±0.0065	0.8780±0.0053	0.8879±0.0059
340			0.8769±0.0059	0.8811±0.0077	0.8818±0.0071	0.8887±0.0070
680			0.8778±0.0066	0.8819±0.0071	0.8865±0.0068	0.8927±0.0066

TABLE VII
NETWORK ARCHITECTURE AND PARAMETER SETTINGS OF FA-SAE IN BEARING PROCESS.

Input layer size	Three hidden layers size	Output layer size	Activation function	α	β
8	$16 \times 80 \times 40$	17	Tanh	0.01	0.0001

parameterthe α should not be set too large, generally less than 0.1.

V. CONCLUSIONS

In this paper, a novel semi-supervised deep fault classification method called FA-SAE was proposed. FA-SAE takes advantage of the unlabeled data during the fine-tuning process by aligning the feature of both labeled and unlabeled data. Through the feature alignment strategy, the potential of unlabeled data samples has been intensively explored, and the model generalization ability has been improved. The usefulness and advantages of the proposed FA-SAE were validated through an industrial benchmark process and a real-world rolling bearing dataset. The application results demonstrated that FA-SAE has higher generalization ability and better classification performance compared with the conventional SVM, MLP, SS-SAE, SS-DSAE, and SS-SSAE. In future work, it is interesting to explore some improved sinkhorn diatances to further improve the fault classification performance of the proposed FA-SAE model.

REFERENCES

[1] 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, 2014.

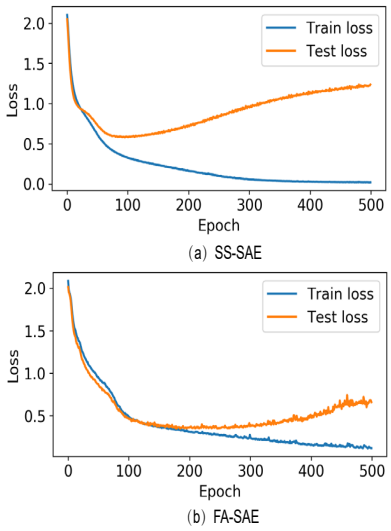

Fig. 9. Cross-entropy losses of SS-SAE and FA-SAE on the training data and testing data.

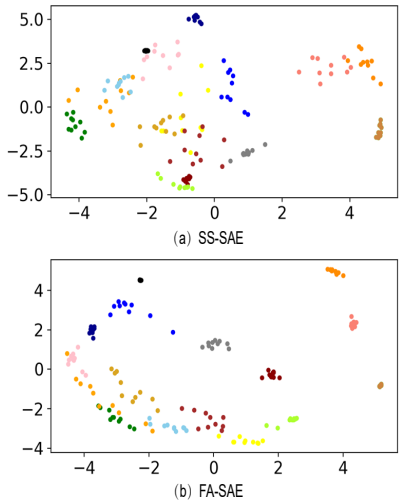

Fig. 10. Extracted features of SS-SAE and FA-SAE on labeled CRWU data. Figure legends: seventeen colored dots ('black','green','blue','orange', 'yellow', 'brown', 'darkblue', 'darkred', 'gray', 'greenyellow', 'pink', 'skyblue', 'goldenrod', 'lime', 'peru', 'salmon', 'darkorange') correspond to the fault No. in Table V.

[2] Z. Ge, "Review on data-driven modeling and monitoring for plant-wide industrial processes," *Chemom. Intell. Lab. Syst.*, vol. 171, pp. 16–25, Dec. 2017.

[3] 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. 12 899–12 912, Jul. 2019.

[4] 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.*, 2020.

[5] X. Yuan, Y. Wang, C. Yang, Z. Ge, Z. Song, and W. Gui, "Weighted linear dynamic system for feature representation and soft sensor application in nonlinear dynamic industrial processes," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1508–1517, Jul. 2017.

[6] X. Zhang, M. Kano, M. Tani, J. Mori, J. Ise, and K. Harada, "Prediction and causal analysis of defects in steel products: Handling nonnegative and highly overdispersed count data," *Control Eng. Pract.*, vol. 95, p. 104258, Feb. 2020.

[7] X. Zhang, M. Kano, and S. Matsuzaki, "A comparative study of deep and shallow predictive techniques for hot metal temperature prediction in blast furnace ironmaking," *Comput. Chem. Eng.*, vol. 130, p. 106575, Nov. 2019.

[8] Z. Yang and Z. Ge, "Industrial virtual sensing for big process data based on parallelized nonlinear variational bayesian factor regression," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 10, pp. 8128–8136, May 2020.

[9] J. H. Lee, J. Shin, and M. J. Realff, "Machine learning: Overview of the recent progresses and implications for the process systems engineering field," *Comput. Chem. Eng.*, vol. 114, pp. 111–121, Jun. 2018.

[10] J. Shin, T. A. Badgwell, K.-H. Liu, and J. H. Lee, "Reinforcement learning–overview of recent progress and implications for process control," *Comput. Chem. Eng.*, vol. 127, pp. 282–294, Aug. 2019.

[11] J. M. Johnson and A. Yadav, "Fault detection and classification technique for hvdc transmission lines using knn," in *Information and Communication Technology for Sustainable Development*. Springer, 2018, pp. 245–253.

[12] C. Jing and J. Hou, "Svm and pca based fault classification approaches for complicated industrial process," *Neurocomputing*, vol. 167, pp. 636–642, Nov. 2015.

[13] L. H. Chiang, M. E. Kotanchek, and A. K. Kordon, "Fault diagnosis based on fisher discriminant analysis and support vector machines," *Comput. Chem. Eng.*, vol. 28, no. 8, pp. 1389–1401, Jul. 2004.

[14] M. A. Atoui, A. Cohen, S. Verron, and A. Kobi, "A single bayesian network classifier for monitoring with unknown classes," *Eng. Appl. Artif. Intell.*, vol. 85, pp. 681–690, Oct. 2019.

[15] Y. Liu and Z. Ge, "Weighted random forests for fault classification in industrial processes with hierarchical clustering model selection," *J. Process Control*, vol. 64, pp. 62–70, Apr. 2018.

[16] Y. Zhang, X. Ding, Y. Liu, and P. Griffin, "An artificial neural network approach to transformer fault diagnosis," *IEEE Trans. Power Appar. Syst.*, vol. 11, no. 4, pp. 1836–1841, Oct. 1996.

[17] J. Yin and W. Zhao, "Fault diagnosis network design for vehicle on-board equipments of high-speed railway: A deep learning approach," *Eng. Appl. Artif. Intell.*, vol. 56, pp. 250–259, Nov. 2016.

[18] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015.

[19] I. Banerjee, Y. Ling, M. C. Chen, S. A. Hasan, C. P. Langlotz, N. Moradzadeh, B. Chapman, T. Amrhein, D. Mong, D. L. Rubin *et al.*, "Comparative effectiveness of convolutional neural network (cnn) and recurrent neural network (rnn) architectures for radiology text report classification," *Artif. Intell. Med.*, vol. 97, pp. 79–88, 2019.

[20] S. Wu, Y. Jiang, H. Luo, and S. Yin, "Remaining useful life prediction for ion etching machine cooling system using deep recurrent neural network-based approaches," *Control Eng. Pract.*, vol. 109, p. 104748, 2021.

[21] Q. Jiang, X. Fu, S. Yan, R. Li, W. Du, Z. Cao, F. Qian, and R. Grima, "Neural network aided approximation and parameter inference of non-markovian models of gene expression," *Nat. Commun.*, vol. 12, no. 1, pp. 1–12, 2021.

[22] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[23] 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. Inform*, vol. 14, no. 7, pp. 3235–3243, Feb. 2018.

[24] Y. Wang and X. Yan, "Soft sensor modeling method by maximizing output-related variable characteristics based on a stacked autoencoder and maximal information coefficients," *Int. J. Comput. Intell. Syst.*, vol. 12, no. 2, pp. 1062–1074, Sep. 2019.

[25] X. Yuan, J. Zhou, B. Huang, Y. Wang, C. Yang, and W. Gui, "Hierarchical quality-relevant feature representation for soft sensor modeling: a novel deep learning strategy," *IEEE Trans. Ind. Inform*, vol. 16, no. 6, pp. 3721–3730, Sep. 2019.

[26] S. Tao, T. Zhang, J. Yang, X. Wang, and W. Lu, "Bearing fault diagnosis method based on stacked autoencoder and softmax regression," in *2015 34th Chinese Control Conference (CCC)*. IEEE, Sep. 2015, pp. 6331–6335.

[27] W. Sun, S. Shao, R. Zhao, R. Yan, X. Zhang, and X. Chen, "A sparse auto-encoder-based deep neural network approach for induction motor faults classification," *Measurement*, vol. 89, pp. 171–178, Jul. 2016.

[28] C. Lu, Z.-Y. Wang, W.-L. Qin, and J. Ma, "Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification," *Signal Process.*, vol. 130, pp. 377–388, Jan. 2017.

[29] C. Shen, Y. Qi, J. Wang, G. Cai, and Z. Zhu, "An automatic and robust features learning method for rotating machinery fault diagnosis based on

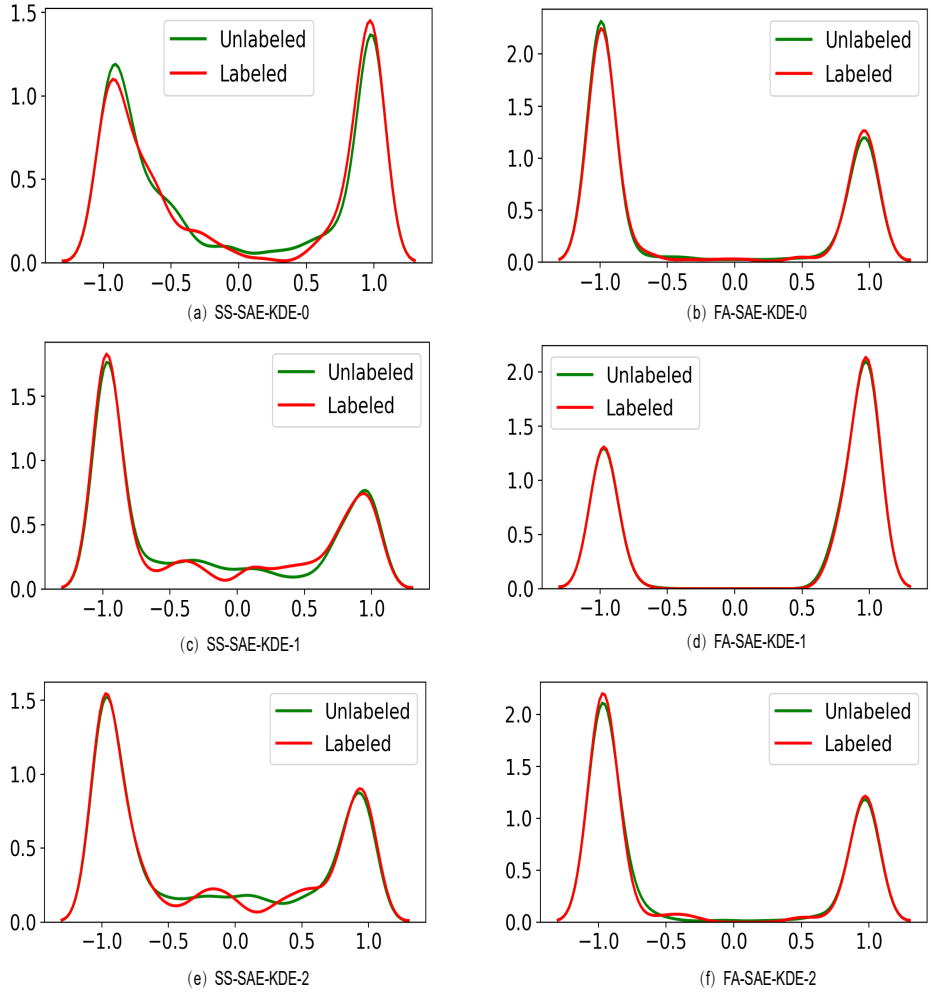

Fig. 11. Feature distribution of SS-SAE and FA-SAE on labeled and unlabeled CWRU data.

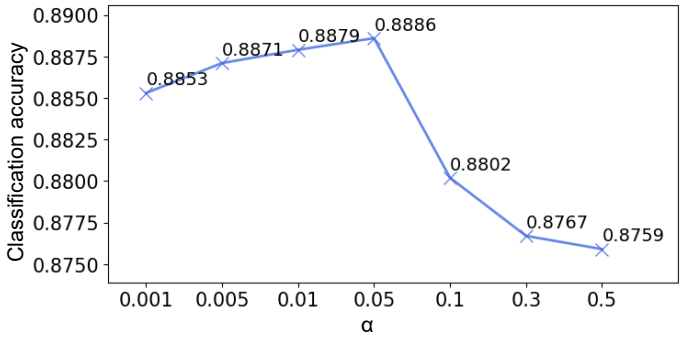

Fig. 12. The influence of Sinkhorn distances ratio α on classification accuracy.

contractive autoencoder," *Eng. Appl. Artif. Intell.*, vol. 76, pp. 170–184, Nov. 2018.

[30] X. Luo, X. Li, Z. Wang, and J. Liang, "Discriminant autoencoder for feature extraction in fault diagnosis," *Chemom. Intell. Lab. Syst.*, vol. 192, p. 103814, Sep. 2019.

[31] H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Stat.*, pp. 400–407, Sep. 1951.

[32] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," *Adv. Neural Inf. Process. Syst.*, vol. 19, p. 153, 2007.

[33] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Adv. Neural Inf. Process. Syst.*, vol. 26, pp. 2292–2300, Jun. 2013.

[34] C. Villani, *Optimal transport: old and new*. Springer Science & Business Media, 2008, vol. 338.

[35] P. A. Knight, "The sinkhorn–knopp algorithm: convergence and applications," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 1, pp. 261–275, Mar. 2008.

[36] K. S. Durgesh and B. Lekha, "Data classification using support vector machine," *J. Theor. Appl. Inf. Technol.*, vol. 12, no. 1, pp. 1–7, 2010.

[37] Y.-P. Lin, C.-H. Wang, T.-L. Wu, S.-K. Jeng, and J.-H. Chen, "Multilayer perceptron for eeg signal classification during listening to emotional music," in *TENCON 2007-2007 IEEE region 10 conference*. IEEE, 2007, pp. 1–3.

[38] A. Ng, "Sparse autoencoder," *CS294A Lecture notes*, vol. 72, no. 2011, pp. 1–19, Sep. 2011.

[39] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *J. Mach. Learn. Res*, vol. 11, no. 12, Dec. 2010.

[40] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Comput. Chem. Eng.*, vol. 17, no. 3, pp. 245–255, 1993.

[41] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process," *J. Process Control*, vol. 22, no. 9, pp. 1567–1581, Oct. 2012.

[42] S. Heo and J. H. Lee, "Parallel neural networks for improved nonlinear principal component analysis," *Comput. Chem. Eng.*, vol. 127, pp. 1–10, Aug. 2019.
[43] N. L. Ricker, "Decentralized control of the tennessee eastman challenge process," *J Process Control.*, vol. 6, no. 4, pp. 205–221, 1996.
[44] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 2, no. 4, pp. 433–459, Jul. 2010.
[45] K. Loparo. (2014) Case western reserve university bearing data center. [Online]. Available: https://csegroups.case.edu/bearingdatacenter/home
[46] Z. Peng and F. Chu, "Application of the wavelet transform in machine condition monitoring and fault diagnostics: a review with bibliography," *Mech. Syst. Signal Process.*, vol. 18, no. 2, pp. 199–221, Mar. 2004.

Xinmin Zhang (M'21) received the Ph.D. degree in System Science from Kyoto University, Japan, in 2019. From April 2019 to December 2019, he was a Postdoctoral Research Fellow in the Department of Systems Science, Kyoto University, Japan. Currently, he is an Associate Professor with the College of Control Science and Engineering, Zhejiang University, China.

His research interests include process control, process data analysis, fault diagnosis, soft-sensor, industrial big data, and machine learning and deep learning with application to industrial processes.

Hongyi Zhang received the B.Eng. degree in Measurement and Control Technology from Wuhan University, Wuhan, China, in 2018 and the M.Eng. degree in Control Science and Engineering from Zhejiang University, Zhejiang, China, in 2021. His research interests include deep learning and the modeling and fault diagnosis of industrial process.

Zhihuan Song received the B.Eng. and M.Eng. degrees in industrial automation from Hefei University of Technology, Anhui, 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 in the College 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 published 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.