Engineering 7 (2021) 1262–1273

Contents lists available at ScienceDirect

Engineering

journal homepage: www.elsevier.com/locate/eng

Research
AI Energizes Process Manufacturing—Article

A Robust Transfer Dictionary Learning Algorithm for Industrial Process Monitoring

Chunhua Yang, Huiping Liang, Keke Huang *, Yonggang Li, Weihua Gui

School of Automation, Central South University, Changsha 410083, China

ARTICLE INFO

Article history:
Received 13 May 2020
Revised 27 June 2020
Accepted 25 August 2020
Available online 4 August 2021

Keywords:
Process monitoring
Multimode process
Dictionary learning
Transfer learning

ABSTRACT

Data-driven process-monitoring methods have been the mainstream for complex industrial systems due to their universality and the reduced need for reaction mechanisms and first-principles knowledge. However, most data-driven process-monitoring methods assume that historical training data and online testing data follow the same distribution. In fact, due to the harsh environment of industrial systems, the collected data from real industrial processes are always affected by many factors, such as the changeable operating environment, variation in the raw materials, and production indexes. These factors often cause the distributions of online monitoring data and historical training data to differ, which induces a model mismatch in the process-monitoring task. Thus, it is difficult to achieve accurate process monitoring when a model learned from training data is applied to actual online monitoring. In order to resolve the problem of the distribution divergence between historical training data and online testing data that is induced by changeable operation environments, a robust transfer dictionary learning (RTDL) algorithm is proposed in this paper for industrial process monitoring. The RTDL is a synergy of representative learning and domain adaptive transfer learning. The proposed method regards historical training data and online testing data as the source domain and the target domain, respectively, in the transfer learning problem. Maximum mean discrepancy regularization and linear discriminant analysis-like regularization are then incorporated into the dictionary learning framework, which can reduce the distribution divergence between the source domain and target domain. In this way, a robust dictionary can be learned even if the characteristics of the source domain and target domain are evidently different under the interference of a realistic and changeable operation environment. Such a dictionary can effectively improve the performance of process monitoring and mode classification. Extensive experiments including a numerical simulation and two industrial systems are conducted to verify the efficiency and superiority of the proposed method.

1. Introduction

Process monitoring is necessary and meaningful for industrial systems, and attracts a considerable amount of attention from both industry and academia [1–4]. In general, process-monitoring methods are divided into three categories: the model-based approach, the knowledge-based approach, and the data-driven approach [5–8]. The model-based approach uses a mathematical representation of the system and thus incorporates a physical understanding of the system into the monitoring scheme. The knowledge-based approach uses graphical models such as PetriNets, multi-signal flow graphs, and Bayesian networks (BNs) for system monitoring and troubleshooting. This approach is especially well-suited for the prognosis of coupled systems [9]. The data-driven approach for monitoring does not require reaction mechanisms or first-principles knowledge of the process. In recent years, by leveraging the rapid progress that has been made in smart sensors, data analytics, and deep learning technologies, the data-driven approach has been developed to enhance the effectiveness and performance of diagnoses [10]; advances in this area include Boltzmann machines, support vector machines (SVMs), convolutional neural networks (CNNs), and more [11–13]. Recently, data-driven methods have become the mainstream of complex industrial process monitoring.

* Corresponding author.
E-mail address: huangkeke@csu.edu.cn (K. Huang).

https://doi.org/10.1016/j.eng.2020.08.028

However, most data-driven methods currently assume that the historical training data and the online monitoring data follow the same distribution [14–16]. In fact, the collected data from real industrial processes are always affected by many factors, such as the changeable operating environment, variations in raw materials, and production indexes [17]. These factors often lead to problems such as model mismatching when the model that was learned based on training data is applied to actual online monitoring. Such problems make it difficult to achieve accurate process monitoring. In order to resolve the problem of historical training data and online testing data following different distributions, pioneering works have been proposed. Hou et al. [18] proposed an incremental principal component analysis (PCA) online model for time-varying process monitoring. In this model, when a new sample is obtained, the PCA model is updated by the original PCA model and the new sample, the squared prediction error (SPE), and the T^2 limits are updated as well. Jiang et al. [19] proposed a multi-model discriminant partial least-squares (DPLS) method based on the current data and historical data to diagnose the data. Zhang et al. [20] established a deep belief network (DBN), which had a strong generalization ability, to monitor welding status online. In order to adapt to the addition of new data samples, Zeng et al. [21] proposed an incremental local preservation projection (LPP) algorithm, which is updated by using the Laplacian matrix and the projection value of the original sample. However, these algorithms cannot cope with the large difference between two domains that occurs when, for example, the monitored industrial process is located in a completely different operating environment. Ge and Song [22] proposed the online batch independent component analysis–principal component analysis (ICA–PCA) method, in which new data is monitored using a fitted ICA–PCA model selected from the model storeroom. However, this method requires the construction of multiple models, and unbalances in data will reduce the monitoring performance.

Dictionary learning usually involves learning an over-complete dictionary; then, the raw data can be reconstructed by a dictionary and a sparse matrix. Real raw data usually possess the characteristic of structural redundancy. Through dictionary learning, the raw data are mapped to a lower dimensional space, which removes the structural redundancy of the raw data while simultaneously retaining the most concise information. Peng et al. [23] proposed a method that worked out a mapping dictionary through LPP in order to retain the geometric structure of the raw data. Chen et al. [24] proposed a method that utilized dictionary regularization to create a dictionary that learned from a small amount of target domain data; this dictionary was similar to a dictionary that learned from the source domain data. Zhang et al. [25] proposed a method in which a common dictionary, a source domain dictionary, and a target domain dictionary were learned in order to realize cross-domain classification. In the method proposed by Jie et al. [26], several subspace dictionaries between the source domain dictionary and target domain dictionary were learned. Long et al. [27] proposed a transfer sparse code (TSC) that introduced graph regularization and reduced the distribution distance in order to realize knowledge transfer. These dictionary learning methods have made great achievements in signal reconstruction, signal noise reduction, image recognition, image correction, and other aspects, which have attracted a great deal of attention in both academia and industry. Moreover, recent studies have shown that dictionary learning has an extraordinary advantage in process monitoring. Huang et al. [28] proposed a kernel dictionary learning method to achieve nonlinear process monitoring and a distributed dictionary learning method to achieve high-dimensional process monitoring.

Although the abovementioned methods exhibit superior process-monitoring performance in industrial systems, they do not take the distribution divergence of historical data and online monitoring data into account. Transfer learning, which is a general framework for knowledge transfer in different domains, has been extensively investigated in recent years, especially in the communities of artificial intelligence, image recognition, and computer vision. Inspired by the powerful representation ability of dictionary learning and the cross-domain knowledge-transfer ability of transfer learning, a robust transfer dictionary learning (RTDL) method is herein proposed to deal with the problem of the distribution divergence of realistic industrial process monitoring. The proposed method is a synergy of representative learning and domain adaptive transfer learning. To summarize, historical training data and online testing data are separately treated as the source domain and target domain in the transfer learning problem. For an industrial system, although the system always operates in different operating environments, which inevitably results in different data distribution, the underlying internal information, such as the mechanism, is often the same or similar. In other words, the high-dimensional observation data of an industrial system often have invariant subspaces in different domains. Therefore, it is feasible to map the source domain and the target domain to a common subspace, in which the distribution difference between the source domain and the target domain is eliminated. After that, process monitoring is carried out in the learned subspace. For practical use, a discriminative dictionary learning method is proposed to extract features from multimodal industrial data. Next, maximum mean discrepancy (MMD) regularization [29] is proposed as a non-parametric distance metric to express the distribution distance, which reduces the distribution divergence between the source domain data and the target domain data. In addition, in order to reduce the distance of the interior mode data, linear discriminant analysis (LDA) regularization is introduced. Accordingly, the proposed method can learn a robust common dictionary even if the source domain and target domain are seriously affected by a different operating environment; thus, this method can effectively improve the performance of process monitoring and mode classification.

The main contributions of this paper are summarized as follows. First, an RTDL method is proposed to reduce the negative effect of an industrial system's changeable operating environment. By reducing the inter-domain differences, the proposed model can reduce the performance degradation of process monitoring and mode classification caused by a changeable operating environment. Second, detailed optimization steps about the constrained nondifferentiable dictionary learning problem are given, which can efficiently solve the RTDL problem. Third, the proposed method is verified by extensive experiments, including a numerical experiment and real industrial experiments. The results demonstrate that the proposed method can outperform some state-of-the-art methods in accuracy; thus, this method is suitable for the task of process monitoring industrial systems.

The rest of this paper is organized as follows. Section 2 briefly introduces domain adaptive transfer learning, dictionary learning, and the motivation for this paper. In Section 3, the RTDL model is proposed and its effective optimization steps are given. In Section 4, extensive experiments including a numerical simulation, the continuous stirred tank heater (CSTH) benchmark case, and a wind turbine system case are conducted to verify the effectiveness of the proposed RTDL method. Finally, Section 5 provides a conclusion and summary remarks.

2. Preliminaries

2.1. Domain adaptive transfer

Suppose that there are a source domain with a large amount of data, $\boldsymbol{X}_s = [x_{s1}, x_{s2}, \ldots, x_{n_s}]$ (where n_s is the number of source domain data $\boldsymbol{X}_s$), and a target domain with a small amount of data,

$\boldsymbol{X}_t = [x_{t1}, x_{t2}, ..., x_{n_t}]$ (where n_t is the number of target domain data $\boldsymbol{X}_t$). Here, the source domain and target domain are related but not the same. Taking a wind turbine system as an example, set the process data of the wind turbine system in winter as the source domain and set the process data in summer as the target domain. These two domains are related, since the data of each domain are collected from the same wind turbine system under the same mechanism. However, due to the external operating environments being different, the observation data from the two seasons often have different distributions. Mathematically, the input feature space of the domains is the same, but the marginal distribution and conditional distribution of the domains are different; namely, $\boldsymbol{X}_s \in \chi$, $\boldsymbol{X}_t \in \chi$, $P_s(x) \neq P_t(x)$, and $P_s(y|x) \neq P_t(y|x)$, where P_s and P_t represent the probability distribution of source domain and target domain, respectively; χ represents data space; x represents the data sample; and y represents the label of the data x.

In order to achieve process monitoring in the target domain, it is necessary to eliminate the distribution divergence between the source domain and the target domain—namely, the conditional distribution difference and the marginal distribution difference. When the distribution divergence of the source domain and the target domain is eliminated, the source domain data can show the same process information as the target domain data, so the method can take advantage of abundant source domain data to aid the training model, and thus achieve a positive knowledge transfer effect.

2.2. Dictionary learning

The philosophy of dictionary learning is to minimize data reconstruction errors by learning a dictionary composed of a series of atoms and a sparse matrix. Let $\boldsymbol{X}_N = [x_1, x_2, ..., x_N] \in \boldsymbol{R}^{m\times N}$ be the set of raw samples, where x_N represents the Nth sample with a dimension of m, m is the number of data dimensions, $\boldsymbol{R}$ is the vector space, and N represents the amount of data. $\boldsymbol{D}_K = [d_1, d_2, ..., d_K] \in \boldsymbol{R}^{m\times K}$ is a dictionary composed of K atoms, where d_K is the Kth atom and K is the number of atoms. $\boldsymbol{S}_N \in \boldsymbol{R}^{K\times N}$ stands for the sparse matrix. The problem of dictionary learning can be expressed as follows:

$$(\boldsymbol{D}_K, \boldsymbol{S}_N) = \underset{\boldsymbol{D}_K, \boldsymbol{S}_N}{\operatorname{argmin}} \| \boldsymbol{X}_N - \boldsymbol{D}_K \boldsymbol{S}_N \|_F^2 + \alpha \| \boldsymbol{S}_N \|_0 \tag{1}$$

where α ($\alpha > 0$) is a parameter to control the sparsity of $\boldsymbol{S}_N$, $\| \cdot \|_F$ represents the F norm of the matrix $\boldsymbol{X}_N$, and $\| \cdot \|_0$ represents the L_0 norm of the matrix $\boldsymbol{S}_N$.

2.3. Motivation

As mentioned earlier, although an industrial system may operate under different operating environments (e.g., due to external interference such as different locations, time, weather, and manual operation), which inevitably results in a divergence in the data distribution, the underlying internal mechanisms are often the same or similar to each other. That is, the high-dimensional observation data of an industrial system under different domains often have invariant subspaces. Therefore, it is necessary to extract the invariant knowledge or subspace in order to eliminate the extrinsic interference and thereby further enhance the performance of industrial process monitoring.

In order to show the effect of the transferable feature between the source domain data and target domain data vividly, Fig. 1 shows a scatter diagram of data that are influenced only by different environmental factors that cause different distributions. The marginal distribution and conditional distribution of the source domain and target domain are obviously different. The traditional data-driven method has two common strategies: As shown in Fig. 1(a), the first strategy is to ignore the source domain data and only use the target domain data as the input data, so as to meet the assumption that the distribution of the training data is the same as the distribution of the testing data. However, because there is very little training data in the target domain, the final model is prone to overfitting. As shown in Fig. 1(b), the second strategy is to ignore the different characteristics of the source domain and the target domain. This strategy uses a large amount of historical data and a small amount of new data directly to conduct the model training task; thus, the final model confuses inter-domain difference information with abnormal information. Moreover, it is easy for the model to be dominated by the source domain, which has a large amount of samples. In contrast, Fig. 1(c) shows the RTDL model proposed in this paper. This method attempts to find a mapping relationship function $\Phi(\cdot)$. Through this mapping relationship, the raw data are mapped into a subspace. In this subspace, the marginal distribution and conditional distribution of the source domain are the same as those of the target domain; that is, $P_s(\Phi(x)) = P_t(\Phi(x))$ and $P_s(y|\Phi(x)) = P_t(y|\Phi(x))$ (where $\Phi(x)$ is the mapping with respect to x). We believe that if $\Phi(\cdot)$ can overcome the interference of the extrinsic environment and only keep the most concise internal mechanism information, it can transfer knowledge from the source domain to the target domain. That is, by incorporating MMD and LDA-like regularizations into the dictionary learning objective function, the proposed method can take advantage of the abundant

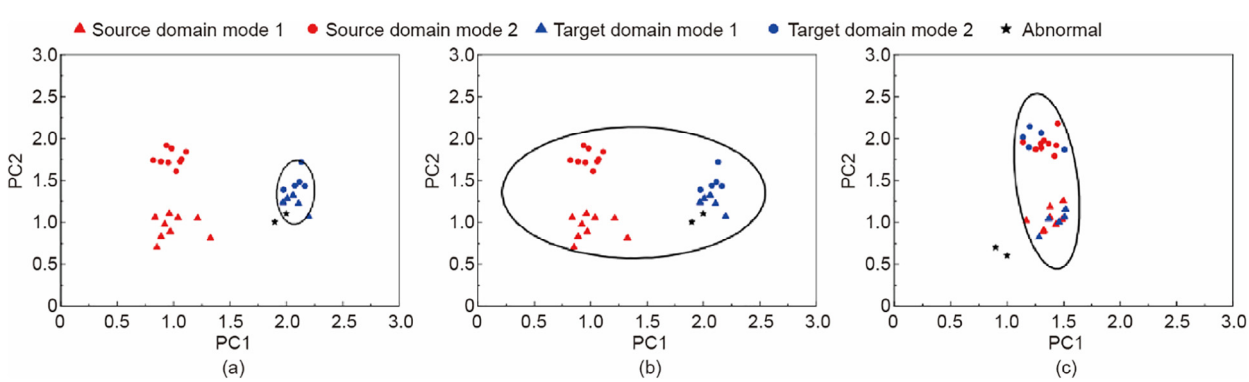

Fig. 1. This figure depicts a situation with a large amount of source domain data and a small amount of target domain data. (a) Traditional strategy 1 ignores the source domain and only utilizes the target domain data for model training. (b) Traditional strategy 2 ignores the different characteristics of the source domain and target domain and directly utilizes all data for model training. (c) The RTDL model, which regularizes the constraints on the data of the source domain and target domain, eliminates inter-domain differences; this model is the most reasonable of the three. PC1 and PC2 represent the two principal components of data.

source domain data to aid the training model and achieve the transfer effect, in order to improve industrial process monitoring.

3. Method

Before discussing the method in detail, an assumption is introduced here for the proposed method. This assumption is reasonable and is often satisfied by an industrial system.

Assumption: A complex industrial process usually runs in different modes to meet different realistic demands. The characteristics of the observed variables are different under different modes. In order to clearly describe different observations, the historical training data and online testing data are regarded as the source domain and target domain, respectively.

In general, there are two ways of conducting multimode data process monitoring. The first way is to treat the multimode data separately, and then fulfil the process-monitoring tasks individually. The second way is to treat the multimode data globally, and then fulfil the process-monitoring task using a single model. When the data collected in each individual mode is sufficient, the first way is a better choice. However, for a real industrial process-monitoring task, the data in the target domain is always seriously inadequate and the separate method is prone to overfitting; therefore, it is better to treat multimode data globally. Moreover, the observed variables are not only determined by the internal mechanism of the industrial process, but also influenced by the extrinsic environment (e.g., manual operation, uncertainties, discontinuities of parameter measurement, and noise). The extrinsic environment of online testing data is different from that of historical training data, so domain divergence occurs. In order to obtain an accurate process-monitoring result, a wise option is to eliminate the irrelevant extrinsic interference by using the domain adaptive transfer learning method.

3.1. Discriminative dictionary

Traditional dictionary learning has been extensively introduced for process monitoring. Moreover, recent studies have shown that learning a discriminative dictionary can enable the dictionary to possess the mode-recognition ability [30–33]. Therefore, a discriminative dictionary learning method is urgently needed for the process-monitoring task. Here, the discriminative dictionary is recorded as $\boldsymbol{D} = [\boldsymbol{D}_1, \boldsymbol{D}_2, ..., \boldsymbol{D}_C] \in \boldsymbol{R}^{m\times Ck}$, where C represents the number of modes, k represents the number of atoms of each mode, and $\boldsymbol{D}_C$ is a subdictionary of k atoms used to represent the characteristics of the Cth mode. The sparse matrix ($\boldsymbol{S}$) of raw samples matrix ($\boldsymbol{X}$) over $\boldsymbol{D}$ is $\boldsymbol{S} = [\boldsymbol{S}_1, \boldsymbol{S}_2, ..., \boldsymbol{S}_C] \in \boldsymbol{R}^{Ck\times(n_s+n_t)}$, where $\boldsymbol{S}_C = [\boldsymbol{S}_{s_C}, \boldsymbol{S}_{t_C}]$, $\boldsymbol{S}_{sC}$ represents the sparse coding of Cth mode source domain data and $\boldsymbol{S}_{tC}$ represents the sparse coding of Cth mode target domain data. For the sake of simplicity, we record the sparse coding of $\boldsymbol{X}_i$ over $\boldsymbol{D}$ as $\boldsymbol{S}_i = \left[\boldsymbol{S}_i^1; \boldsymbol{S}_i^2; ...; \boldsymbol{S}_i^C\right]$, where $\boldsymbol{S}_i^C$ is the coding coefficient of $\boldsymbol{X}_i$ over the sub-dictionary $\boldsymbol{D}_C$, $\boldsymbol{X}_i$ is a matrix composed of the ith mode samples ($i \in \{1, 2, ..., C\}$), which is a submatrix of $\boldsymbol{X}$.

In order to improve the representation ability of the multimode data, prior constraints should be incorporated into the dictionary learning. First, the data should be well reconstructed by a dictionary and the corresponding sparse matrix; that is, $\boldsymbol{X} \approx \boldsymbol{DS}$. Second, the data should be well represented by its own sub-dictionary $\boldsymbol{D}_i$ and sub-sparse matrix $\boldsymbol{S}_i^i$; that is, $\boldsymbol{X}_i \approx \boldsymbol{D}_i\boldsymbol{S}_i^i\ \forall i \in \{1, 2, ..., C\}$. Third, since the data can be well represented by its own sub-dictionary and sub-sparse matrix, the item $\boldsymbol{S}_i^{i'}\ (i' \neq i)$ should be as close to zero as possible, so the Eq. (1) can be transformed into the following formation [31]:

$$\begin{aligned}(\boldsymbol{D}, \boldsymbol{S}) = &\underset{\boldsymbol{D},\boldsymbol{S}}{\operatorname{argmin}} \parallel \boldsymbol{X} - \boldsymbol{DS} \parallel_F^2 + \parallel \boldsymbol{X} - \boldsymbol{DS}_{\text{in}} \parallel_F^2 \\ &+ \parallel \boldsymbol{DS}_{\text{out}} \parallel_F^2 + \alpha \parallel \boldsymbol{S} \parallel_0 \\ \text{s.t. } &\forall \parallel d_b \parallel_2 \leq 1\end{aligned} \tag{2}$$

where d_b represents bth atom of the dictionary $\boldsymbol{D}$. $\boldsymbol{S}_{\text{in}}$ and $\boldsymbol{S}_{\text{out}}$ are the expressions about $\boldsymbol{S}$, which are given as follows:

$$\boldsymbol{S}_{\text{in}}[a, b] = \begin{cases} \boldsymbol{S}[a, b] & x_a, d_b \in \text{same mode} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$\boldsymbol{S}_{\text{out}}[a, b] = \begin{cases} \boldsymbol{S}[a, b] & x_a, d_b \notin \text{same mode} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where x_a is the ath sample of $\boldsymbol{X}$.

In particular, $\parallel \boldsymbol{X} - \boldsymbol{DS}_{\text{in}} \parallel_F^2 = \sum_{i=1}^{C} \parallel \boldsymbol{X}_i - \boldsymbol{D}_i\boldsymbol{S}_i^i \parallel_F^2$ means that the data X_i should be well represented by its own sub-dictionary $\boldsymbol{D}_i$ and sub-sparse matrix $\boldsymbol{S}_i^i$. That is, $\parallel \boldsymbol{DS}_{\text{out}} \parallel_F^2 = \sum_{i=1}^{C} \sum_{i'\neq i} \parallel \boldsymbol{D}_i\boldsymbol{S}_i^{i'} \parallel_F^2$. Thus, the item $\boldsymbol{S}_i^{i'}\ (i' \neq i)$ should be as close to zero as possible.

3.2. Regularizations

Since the source domain and target domain of the industrial system are affected by different environmental factors, the data distributions are different. To ensure that the learned dictionary captures the latent common mechanism information rather than irrelevant extrinsic interference in the source domain and target domain, a straightforward way is to reduce the distribution difference by minimizing some predefined metrics. MMD regularization is considered to be a nonparametric metric to express the distribution difference between domains [27], which can make the centers of the sparse matrices $S_{s,i}$ and $S_{t,i}$ close. Mathematically, the MMD regularization is expressed as follows:

$$c_1 = \sum_{i=1}^{C} \left\| \frac{1}{n_{si}} \sum_{s_j \in S_{si}} s_j - \frac{1}{n_{ti}} \sum_{s_j \in S_{ti}} s_j \right\|_F^2 = \sum_{i=1}^{C} \operatorname{tr}\left(\boldsymbol{S}_i\boldsymbol{M}^i\boldsymbol{S}_i^{\mathrm{T}}\right) = \operatorname{tr}\left(\boldsymbol{SMS}^{\mathrm{T}}\right) \tag{5}$$

where c_1 represents the MMD regularization between the source domain and the target domain, n_{si} and n_{ti} represent the numbers of samples of ith mode in source domain and target domain, respectively, s_j is a sparse code in sparse matrix $\boldsymbol{S}$, $\boldsymbol{M}^i$ represents the ith mode part of MMD matrix $\boldsymbol{M}$, which can be calculated as follows:

$$\boldsymbol{M}_{\bar{a}\underline{a}}^i = \begin{cases} 1/n_{si}^2 & x_{\bar{a}}, x_{\underline{a}} \in \text{source domain} \\ 1/n_{ti}^2 & x_{\bar{a}}, x_{\underline{a}} \in \text{target domain} \\ -1/(n_{si}n_{ti}) & \text{otherwise} \end{cases} \tag{6}$$

$$\boldsymbol{M} = \operatorname{diag}\left(\boldsymbol{M}^1, \boldsymbol{M}^2, \ldots, \boldsymbol{M}^C\right)$$

where $M_{\bar{a}\underline{a}}^i$ represents the element of M^i in the $\bar{a}$th row and $\underline{a}$th column, $x_{\bar{a}}$ is the $\bar{a}$th sample of X , and $x_{\underline{a}}$ is the $\underline{a}$th sample of X.

It is well known that the distribution of observed data in the same mode should be the same. However, due to the uncertainty of the operating environment, the data in the same mode also exhibit certain differences. In order to eliminate the interference from the uncertain environment, it is preferable to make the sparse code of the same mode data become closer to each other no matter which domain the data comes from. That is, each column vector in $\boldsymbol{S}_i = [\boldsymbol{S}_{si}, \boldsymbol{S}_{ti}]\ \forall i \in \{1, 2, ..., C\}$ should be close to each other. Thus, the value of $\sum_{s_j \in S_i} \left(s_j - \bar{s}_i\right)^{\mathrm{T}} \left(s_j - \bar{s}_i\right)$ should be as small as possible,

where $\bar{s}_i$ is the center of S_i. Accordingly, the following constraint should be introduced.

$$c_2 = \sum_{i=1}^{C} \sum_{s_j \in S_i} \left(s_j - \bar{s}\right)_i^{\mathrm{T}} \left(s_j - \bar{s}\right)_i = \mathrm{tr}\left(\boldsymbol{S}(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{S}^{\mathrm{T}}\right) \tag{7}$$

where c_2 refers the intra-mode distances of all modes, and $\boldsymbol{I}$ is the identity matrix. The LDA-like matrix $\boldsymbol{H}$ can be obtained as follows:

$$\boldsymbol{H} = \mathrm{diag}(\boldsymbol{H}_1, \boldsymbol{H}_2, ..., \boldsymbol{H}_C), \boldsymbol{H}_i = \frac{1}{n_i} \boldsymbol{1}_{n_i} \boldsymbol{1}_{n_i}^{\mathrm{T}} \tag{8}$$

where $\boldsymbol{1}_{n_i}$ is a vector with length n_i, with all elements equal to 1, n_i is the total number of data for ith mode in the source domain and target domain. The regularization of Eq. (7) appears to be similar to the LDA regularization [34], so it will be called "LDA-like regularization" from this point on.

In summary, the MMD and LDA-like regularizations can be thought of as a progressive relationship. The MMD regularization makes the source domain center of each mode close to the target domain center of each mode. In a complementary way, the LDA-like regularization makes the data of the same mode closer to each other, in order to reduce the intra-mode distance. Based on the synergy effect of these two regularizations, the proposed method was formulated by joining Eqs. (2), (3), and (7) together; the objective function of the proposed method can be rewritten as follows:

$$\begin{aligned} (\boldsymbol{D}, \boldsymbol{S}) = \underset{\boldsymbol{D},\boldsymbol{S}}{\mathrm{argmin}} &\| \boldsymbol{X} - \boldsymbol{D}\boldsymbol{S} \|_F^2 + \| \boldsymbol{X} - \boldsymbol{D}\boldsymbol{S}_{\mathrm{in}} \|_F^2 \\ &+ \| \boldsymbol{D}\boldsymbol{S}_{\mathrm{out}} \|_F^2 + \alpha \| \boldsymbol{S} \|_0 + \beta_1 \mathrm{tr}\left(\boldsymbol{S}\boldsymbol{M}\boldsymbol{S}^{\mathrm{T}}\right) \\ &+ \beta_2 \mathrm{tr}\left(\boldsymbol{S}(\boldsymbol{I} - \boldsymbol{H})\boldsymbol{S}^{\mathrm{T}}\right) \\ \text{s.t. } &\forall d_b \le 1 \end{aligned} \tag{9}$$

where β_1 and β_2 $(\beta_1, \beta_2 > 0)$ are hyper-parameters of the MMD regularization and the LDA-like regularization, respectively; and these hyper-parameters can balance the weight between the reconstruction error and the regularization constraints. Since the proposed method reduces the distribution divergence between the source domain and the target domain, it can achieve the effect of eliminating extrinsic environmental interference.

3.3. Optimization

The optimization variables of the above objective function are $\boldsymbol{D}$ and $\boldsymbol{S}$. Since the optimization problem is not a joint convex problem for both variables, but is separately convex to $\boldsymbol{D}$ (while holding $\boldsymbol{S}$ fixed) and convex to $\boldsymbol{S}$ (while holding $\boldsymbol{D}$ fixed), the alternative optimization method is introduced to calculated the optimal values of $\boldsymbol{D}$ and $\boldsymbol{S}$ iteratively [35].

3.3.1. Updating D

When updating the dictionary by fixing $\boldsymbol{S}$, the objective function can be simplified to

$$\begin{aligned} \boldsymbol{D} = \underset{\boldsymbol{D}}{\mathrm{argmin}} &\| \boldsymbol{X} - \boldsymbol{D}\boldsymbol{S} \|_F^2 + \| \boldsymbol{X} - \boldsymbol{D}\boldsymbol{S}_{\mathrm{in}} \|_F^2 + \| \boldsymbol{D}\boldsymbol{S}_{\mathrm{out}} \|_F^2 \\ \text{s.t. } &\forall \| d_b \|_2 \le 1 \end{aligned} \tag{10}$$

After constructing a new data matrix and a new sparse matrix $\boldsymbol{X}_{\mathrm{new}} = [\boldsymbol{X}, \boldsymbol{X}, \boldsymbol{O}]$, $\boldsymbol{S}_{\mathrm{new}} = [\boldsymbol{S}, \boldsymbol{S}_{\mathrm{in}}, \boldsymbol{S}_{\mathrm{out}}]$, where $\boldsymbol{O}$ is the zero matrix of the same dimension as $\boldsymbol{X}$, the objective function (Eq. (10)) becomes

$$\boldsymbol{D} = \underset{\boldsymbol{D}}{\mathrm{argmin}} \| \boldsymbol{X}_{\mathrm{new}} - \boldsymbol{D}\boldsymbol{S}_{\mathrm{new}} \|_F^2, \ \forall \| d_b \|_2 \le 1 \tag{11}$$

The objective function can be solved effectively by using the Lagrange dual method [35]. First, consider the Lagrangian:

$$\begin{aligned} L\left(\boldsymbol{D}, \vec{\lambda}\right) = &\mathrm{trace}\left((\boldsymbol{X}_{\mathrm{new}} - \boldsymbol{D}\boldsymbol{S}_{\mathrm{new}})^{\mathrm{T}} (\boldsymbol{X}_{\mathrm{new}} - \boldsymbol{D}\boldsymbol{S}_{\mathrm{new}})\right) \\ &+ \sum_{j=1}^{Ck} \lambda_j \left(\sum_{i=1}^{m} \boldsymbol{D}[i,j]^2 - 1\right), \forall \lambda_j \ge 0 \end{aligned} \tag{12}$$

where L is Lagrange function, $\vec{\lambda} = [\lambda_1, \lambda_2, ..., \lambda_{Ck}]$ is the introduced nonnegative parameter.

Minimizing the Lagrangian (Eq. (12)) by $\boldsymbol{D}$ yields the Lagrangian dual problem, as follows:

$$\begin{aligned} B\left(\vec{\lambda}\right) &= \min_{\boldsymbol{D}} L\left(\boldsymbol{D}, \vec{\lambda}\right) \\ &= \mathrm{trace}\left(\boldsymbol{X}_{\mathrm{new}}^{\mathrm{T}} \boldsymbol{X}_{\mathrm{new}} - \boldsymbol{\Lambda} - \boldsymbol{X}_{\mathrm{new}} \boldsymbol{S}_{\mathrm{new}}^{\mathrm{T}} \left(\boldsymbol{S}_{\mathrm{new}} \boldsymbol{S}_{\mathrm{new}}^{\mathrm{T}} + \boldsymbol{\Lambda}\right)^{-1} \left(\boldsymbol{X}_{\mathrm{new}} \boldsymbol{S}_{\mathrm{new}}^{\mathrm{T}}\right)^{\mathrm{T}}\right) \end{aligned} \tag{13}$$

where B is the Lagrangian dual formula; $\boldsymbol{\Lambda}$ is a diagonal matrix consisting of $\vec{\lambda}$, $\boldsymbol{\Lambda} = \mathrm{diag}(\vec{\lambda})$. The Lagrangian dual problem (Eq. (13)) can be optimized by the Newton method or by a conjugate gradient. After maximizing $B\left(\vec{\lambda}\right)$, we obtain the optimal bases $\boldsymbol{D}$ as follows:

$$\boldsymbol{D}^{\mathrm{T}} = \left(\boldsymbol{S}_{\mathrm{new}} \boldsymbol{S}_{\mathrm{new}}^{\mathrm{T}} + \boldsymbol{\Lambda}\right)^{-1} \left(\boldsymbol{X}_{\mathrm{new}} \boldsymbol{S}_{\mathrm{new}}^{\mathrm{T}}\right)^{\mathrm{T}} \tag{14}$$

3.3.2. Updating S

The matrix $\boldsymbol{S}$ will be updated column by column. When s_j is being updated, the objective function can be expressed as follows:

$$\begin{aligned} s_j &= \underset{s_j}{\mathrm{argmin}} f(s_j) \\ &= \underset{s_j}{\mathrm{argmin}} \| x_j - \boldsymbol{D}s_j \|_F^2 + \| x_j - \boldsymbol{D}\boldsymbol{Q}s_j \|_F^2 + \| \boldsymbol{D}\boldsymbol{P}s_j \|_F^2 \\ &\quad + \alpha \sum_{p=1}^{Ck} \left|s_j^{(p)}\right| + \left(\beta_1 \boldsymbol{M}_{jj} + \beta_2 (\boldsymbol{I} - \boldsymbol{H})_{jj}\right) s_j^{\mathrm{T}} s_j + s_j^{\mathrm{T}} h_j \end{aligned} \tag{15}$$

where $\boldsymbol{Q}$, $\boldsymbol{P}$, and h_j are intermediate variables in the simplification process of the objective functions. $\boldsymbol{Q} = \mathrm{diag}(0, 0, ..., 1, 1, 1, ..., 0, 0, 0) \in \boldsymbol{R}^{Ck \times Ck}$, $\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{Q}$, and $h_j = 2\sum_{j' \ne j} \left(\beta_1 \boldsymbol{M}_{jj'} + \beta_2 (\boldsymbol{I} - \boldsymbol{H})_{jj'}\right) s_{j'}$. $\boldsymbol{Q}$ is a diagonal matrix, and 1 is present only on the corresponding k positions of the mode of s_j. x_j is the jth sample in $\boldsymbol{X}$. $s_j^{(p)}$ is the pth element in s_j. We optimize Eq. (15) through a feature-sign search algorithm [27,35]. Define $g(s_j)$ as follows:

$$\begin{aligned} g(s_j) = &\| x_j - \boldsymbol{D}s_j \|_F^2 + \| x_j - \boldsymbol{D}\boldsymbol{Q}s_j \|_F^2 + \| \boldsymbol{D}\boldsymbol{P}s_j \|_F^2 \\ &+ \left(\beta_1 \boldsymbol{M}_{jj} + \beta_2 (\boldsymbol{I} - \boldsymbol{H})_{jj}\right) s_j^{\mathrm{T}} s_j + s_j^{\mathrm{T}} h_j \end{aligned} \tag{16}$$

where $g(s_j)$ is the differentiable part of the Eq. (15). In order to address the feature-sign search algorithm for the problem in Eq. (15), a lemma should be introduced.

Lemma 1: Define a continuous function over x as $F(x) = G(x) + \lambda \| x \|_1$; the optimal necessary conditions of $x = \underset{x}{\mathrm{argmin}} F(x)$ are

$$\begin{cases} \nabla^p G(x) + \lambda \mathrm{sign}(x^p) = 0, & \text{if } |x^p| \ne 0 \\ \left|\nabla^p G(x)\right| \le \lambda, & \text{if } |x^p| = 0 \end{cases} \forall x^p \tag{17}$$

where $G(x)$ is a continuous differentiable function over vector $\boldsymbol{x}$ [36,35], x^p is the pth element in vector $\boldsymbol{x}$, and $\nabla^p G(x)$ is the partial derivative of $G(x)$ over x^p.

Algorithm 1. Sparse matrix optimizing algorithm.

Input: Data matrix $\boldsymbol{X}$, MMD matrix $\boldsymbol{M}$, LDA-like matrix $\boldsymbol{H}$, parameters α, β_1, and β_2
Output: Optimal sparse matrix $\boldsymbol{S}$
Begin optimizing algorithm
For each s_j in $\boldsymbol{S}$ do
Step initialization:
$s_j = \mathbf{0}$, $\boldsymbol{\theta} = \mathbf{0}$, active set $\boldsymbol{A} = \{\}$, where $\boldsymbol{\theta}$ is a vector that represents a symbol of s_j, $\theta_p \in \{-1, 0, 1\}$ denotes sign $(s_j^{(p)})$
Step activate:
From zero coefficients of s_j, select $p := \underset{p}{\operatorname{argmax}} |\nabla^{(p)} g(s_j)|$. Activate $s_j^{(p)}$ (add p to $\boldsymbol{A}$) only if it locally improves Eq. (15); that is, if $\nabla^{(p)} g(s_j) > \alpha$, then set $\theta_p = -1$, $\boldsymbol{A} = \{p\} \cup \boldsymbol{A}$; otherwise, if $\nabla^{(p)} g(s_j) < -\alpha$, then set $\theta_p = 1$, $\boldsymbol{A} = \{p\} \cup \boldsymbol{A}$
Step feature-sign search:
Let $\hat{\boldsymbol{D}}$, $\widehat{\boldsymbol{DQ}}$, and $\widehat{\boldsymbol{DP}}$ be a submatrix of $\boldsymbol{D}$, $\boldsymbol{DQ}$, and $\boldsymbol{DP}$ that contains only the columns in $\boldsymbol{A}$; let $\hat{\boldsymbol{s}}_j$, $\hat{\boldsymbol{h}}_j$, and $\hat{\theta}_j$ be sub-vectors of s_j, h_j, and $\boldsymbol{\theta}$ in $\boldsymbol{A}$, respectively
Compute the solution to the resulting unconstrained quadratic optimization problem (QP):
$\min \hat{f}\left(\hat{\boldsymbol{s}}_j\right) = ||x_j - \hat{\boldsymbol{D}}\hat{\boldsymbol{s}}_j||_F^2 + ||x_j - \widehat{\boldsymbol{DQ}}\hat{\boldsymbol{s}}_j||_F^2 + ||\widehat{\boldsymbol{DP}}\hat{\boldsymbol{s}}_j||_F^2 + \left(\beta_1 \boldsymbol{M}_{jj} + \beta_2 (\boldsymbol{I} - \boldsymbol{H})_{jj}\right)\hat{\boldsymbol{s}}_j^{\mathrm{T}}\hat{\boldsymbol{s}}_j + \hat{\boldsymbol{s}}_j^{\mathrm{T}}\hat{\boldsymbol{h}}_j + \alpha\hat{\theta}_j\hat{\boldsymbol{s}}_j$
If we let $\partial\hat{f}(\hat{\boldsymbol{s}}_j)/\partial\hat{\boldsymbol{s}}_j = 0$, we can obtain the optimal value of s_j under the current $\boldsymbol{A}$:
$\hat{\boldsymbol{s}}_j^{\mathrm{new}} = \left\{\hat{\boldsymbol{D}}^{\mathrm{T}}\hat{D} + \widehat{\boldsymbol{DQ}}^{\mathrm{T}}\widehat{\boldsymbol{DQ}} + \widehat{\boldsymbol{DP}}^{\mathrm{T}}\widehat{\boldsymbol{DP}} + \left(\beta_1 \boldsymbol{M}_{jj} + \beta_2 (I - \boldsymbol{H})_{jj}\right) I\right\}^{-1} \left(\hat{\boldsymbol{D}}^{\mathrm{T}} x_j + \widehat{\boldsymbol{DP}}^{\mathrm{T}} x_j - \left(\alpha\hat{\boldsymbol{\theta}}_j + \hat{\boldsymbol{h}}_j\right)/2\right)$
Perform a discrete line search on the closed line segment from $\hat{\boldsymbol{s}}_j$ to $\hat{\boldsymbol{s}}_j^{\mathrm{new}}$
Check the objective value at $\hat{\boldsymbol{s}}_j^{\mathrm{new}}$ and all points where any coefficient changes sign
Update $\hat{\boldsymbol{s}}_j$ (and the corresponding entries in s) to the point with the lowest objective value
Remove zero coefficients of $\hat{\boldsymbol{s}}_j$ from the active set $\boldsymbol{A}$ and update $\boldsymbol{\theta} := \operatorname{sign}(s_j)$
Step check optimality conditions:
(i) Optimality condition for nonzero coefficients: $\nabla^{(p)} g(s_j) + \alpha \operatorname{sign}\left(s_j^{(p)}\right) = 0, \forall s_j^{(p)} \neq 0$
If condition (i) is not satisfied, go to the step "feature-sign search" (without any new activation); otherwise, check condition (ii)
(ii) Optimality condition for zero coefficients: $\left|\nabla^{(p)} g(s_j)\right| \leq \alpha, \forall s_j^{(p)} = 0$
If condition (ii) is not satisfied, go to the step "activate"; otherwise, return s_j as the optimal solution
End for
End optimal algorithm

Proof: We provide a brief proof through a reduction to absurdity. Assume that there is an element x^p in the optimal solution x that does not meet the condition. First, for $|x^p| \neq 0$, $\nabla^p G(x) + \lambda \operatorname{sign}(x^p) \neq 0$, it is obvious that $\nabla^p F(x) = \nabla^p G(x) + \lambda \operatorname{sign}(x^p) \neq 0$. Therefore, we can find another value x^{*p} to take the place of x^p to make $F(x^*)$ be smaller. This is contradictory to the assumption. Second, for $|x^p| = 0$, $\nabla^p G(x) > \lambda$, since $G(x)$ is a continuous differentiable function, we can find an $x^{*p} < 0$ to take the place of x^p and $G(x^*) - G(x) < \lambda x^{*p}$ is met. Therefore, $F(x^*) = G(x^*) + \lambda \| x^* \|_1 = G(x^*) + \lambda \| x \|_1 - \lambda x^{*p} < G(x) + \lambda \| x \|_1 = F(x)$, which is also contradictory to the assumption. For $|x^p| = 0$, $\nabla^p G(x) < -\lambda$, the same way can be used to show that the assumption is not true.

According to **Lemma 1**, the necessary condition in Eq. (15) can be described as follows:

$$\begin{cases} \nabla^{(p)} g(s_j) + \alpha \operatorname{sign}\left(s_j^{(p)}\right) = 0, & \text{if } \left|s_j^{(p)}\right| \neq 0 \\ \left|\nabla^{(p)} g(s_j)\right| \leq \alpha, & \text{if } \left|s_j^{(p)}\right| = 0 \end{cases} \tag{18}$$

When the first condition is violated, The objective function $f(s_j)$ in Eq. (15) is differentiable over $s_j^{(p)}$ because the sign of $s_j^{(p)}$ is known, and it becomes an unconstrained optimization problem (QP). When the second condition is violated, assume that $\nabla^{(p)} g(s_j) > \alpha$. Since $\nabla^{(p)} f(s_j)$ must be greater than zero, in order to minimize the local value of $f(s_j)$, $s_j^{(p)}$ must decrease. Since $s_j^{(p)}$ starts at zero, any infinitesimal adjustment to $s_j^{(p)}$ will make the sign of $s_j^{(p)}$ negative. Thus, we directly let the sign of $s_j^{(p)}$ be -1. Then, $f(s_j)$ is similarly differentiable over $s_j^{(p)}$, and the problem can be solved simply. If $\nabla^{(p)} g(s_j) < -\alpha$, $s_j^{(p)}$ can be updated in the same way.

Accordingly, the complete process of the sparse matrix optimization algorithm is summarized as shown in **Algorithm 1**.

3.4. Online process monitoring

After the common dictionary $\boldsymbol{D}$ is obtained, the process monitoring and mode classification task will be carried out.

3.4.1. Process monitoring

Dictionary $\boldsymbol{D}$ and the orthogonal matching pursuit (OMP) algorithm [37] are used to calculate the sparse code s_j of the target domain data x_j in the training set; the reconstructed residual (RES) of the target domain data in the training set can be obtained according to Eq. (19).

$$r = \| x - \boldsymbol{D}s \|_2 \tag{19}$$

where $\| \cdot \|_2$ represents the L_2 norm of the vector, and r is the reconstruction error of sample x.

Next, the kernel density estimation (KDE) [38] is used to calculate the residual distribution interval of the data in the target domain, which can be used to detect whether the new testing data is normal or abnormal. When the new testing data x comes from the target domain, the sparse code s is obtained by using the same OMP algorithm; then RES r of the testing data can be obtained using Eq. (19). When the RES r belongs to the above distribution interval, it is normal data. Otherwise, it is faulty data.

3.4.2. Mode classification

After testing data is detected as normal data, mode classification is carried out, and the data x is identified by Eq. (20).

$$\text{mode} = \underset{i}{\operatorname{argmin}} \| x - \boldsymbol{D}_i s^i \|_F^2 \tag{20}$$

where s^i is sub-sparse code of s.

In summary, the complete model flow of the proposed method is summarized in **Algorithm 2**.

Algorithm 2. RTDL.

Input: Source domain data matrix $\boldsymbol{X}_s = [\boldsymbol{x}_{s1}, \boldsymbol{x}_{s2}, ..., \boldsymbol{x}_{n_s}]$, target domain data matrix $\boldsymbol{X}_t = [\boldsymbol{x}_{t1}, \boldsymbol{x}_{t2}, ..., \boldsymbol{x}_{n_t}]$, parameters α, β_1, and β_2

Begin RTDL algorithm

Step 1. Initialization

Structure data matrix $\boldsymbol{X} = [\boldsymbol{X}_{s1}, \boldsymbol{X}_{t1}, ..., \boldsymbol{X}_{sc}, \boldsymbol{X}_{tc}, ..., \boldsymbol{X}_{sC}, \boldsymbol{X}_{tC}]$, MMD matrix $\boldsymbol{M}$ by Eq. (4) and LDA-like matrix $\boldsymbol{H}$ by Eq. (8). $\boldsymbol{D}$ and $\boldsymbol{S}$ are initialized as random values, where $\boldsymbol{X}_{sC}$ and $\boldsymbol{X}_{tC}$ represent the datasets of Cth mode of the source domain and target domain, respectively

Step 2. Off-line training

Obtain $\boldsymbol{D}$ and $\boldsymbol{S}$ iteratively

To fix $\boldsymbol{S}$, update $\boldsymbol{D}$ using Eq. (14)

To fix $\boldsymbol{D}$, update $\boldsymbol{S}$ using **Algorithm 1**

Step 3. Online testing

(i) Process monitoring

Get the RES limit by $\boldsymbol{D}$ and the OMP and KDE algorithms

Conduct process monitoring by comparing the RES statistic and RES limit

(ii) Mode classification

After the testing data is detected as normal data, conduct mode classification using Eq. (20)

End RTDL algorithm

4. Illustrative experiments

In this section, extensive experiments are carried out on a numerical simulation, the CSTH benchmark, and a wind turbine system to demonstrate the effectiveness and superiority of the proposed RTDL method. For the sake of performance visualization and parameter sensitivity analysis, the MMD regularization and LDA-like regularization are introduced separately in the numerical simulation in order to make it possible to visually observe the distribution of the samples. For the CSTH benchmark and the wind turbine system, the proposed method is compared with some state-of-the-art methods for a performance comparison.

4.1. Performance visualization and parameter sensitivity analysis

4.1.1. Datasets

Experiments on a numerical simulation are performed to verify the motivation and intuitively evaluate the performance. The data generation model of the numerical simulation is as follows:

$$\bar{\boldsymbol{x}} = \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \\ \bar{x}_4 \\ \bar{x}_5 \end{bmatrix} = \begin{bmatrix} 0.5768 & 0.3766 \\ 0.7382 & 0.0566 \\ 0.8291 & 0.4009 \\ 0.6519 & 0.2070 \\ 0.3972 & 0.8045 \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \end{bmatrix} \tag{21}$$

where $\bar{\boldsymbol{x}} = \left[\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4, \bar{x}_5\right]^{\mathrm{T}}$ is the observed variable of process monitoring, t_1 and t_2 are two independent input variables of the data simulation model, and $e_1, e_2, ..., e_5$ are the noise of the observed variable due to the current environment interference. For the sake of brevity, we assume that there are two modes in the source domain and target domain. The mode state is determined by the independent input variables t_1 and t_2, while the environment interference $(e_1, e_2, ..., e_5)$ results in domains difference. A detailed description of the data is shown in Table 1, where $N(u, \sigma^2)$ represents the Gaussian distribution whose mean is u and whose variance is σ^2. $U\left(\underline{\omega}, \bar{\omega}\right)$ denotes a uniform distribution that ranges from $\underline{\omega}$ to $\bar{\omega}$. $E(\lambda)$ denotes an exponential distribution with the parameter λ. $\bar{F}(\gamma, \rho)$ represents an F distribution with the degrees of freedom as γ and ρ. Abnormal data are collected from mode 2 in the target domain, with the bias faults occurring in x_2.

For the training data, we collect 100 pieces of normal data for each mode in the source domain and 10 pieces of normal data for each mode in target domain. For the testing data, we collect 50 pieces of normal data for each mode and 300 pieces of abnormal data in the target domain.

4.1.2. Performance visualization experiments

As mentioned above, MMD regularization and LDA-like regularization can eliminate distribution divergence from different perspectives. In order to verify that both have the ability to eliminate distribution divergence, we separately introduce one of them to observe the distribution of the samples visually by setting the other regularization parameter as zero. The training data are utilized to learn a common dictionary. Next, the sparse code of the source domain data, target domain data, and abnormal data are obtained through the common dictionary and OMP algorithm. We visualize the sparse code by means of PCA. The visualization is shown in Fig. 2.

Fig. 2(a) shows the raw data distribution. The domain distribution divergence that occurs is due to environmental interference. The abnormal information is obscure. Figs. 2(b) and (c) show that, with one of the regularizations used on its own, the distribution divergence is partly eliminated and the abnormal information emerges by means of dictionary learning. Fig. 2(d) shows that the source data and target data are completely mixed and the abnormal information can easily be identified by means of the proposed method using both regularizations. Accordingly, the experimental results agree with the motivation of this paper (Section 2.3).

Table 1
Detailed description of data generation.

Mode	Variable		
	t_1	t_2	$(e_1, e_2, ..., e_5)$
Mode 1 in source domain	$N(5, 0.8)$	$N(15, 1.3)$	$3E(0.5)$
Mode 2 in source domain	$U(2, 3)$	$N(15, 1.7)$	$3E(0.5)$
Mode 1 in target domain	$N(5, 0.8)$	$N(15, 1.3)$	$0.1\,\bar{F}(5, 10)$
Mode 2 in target domain	$U(2, 3)$	$N(15, 1.7)$	$0.1\,\bar{F}(5, 10)$

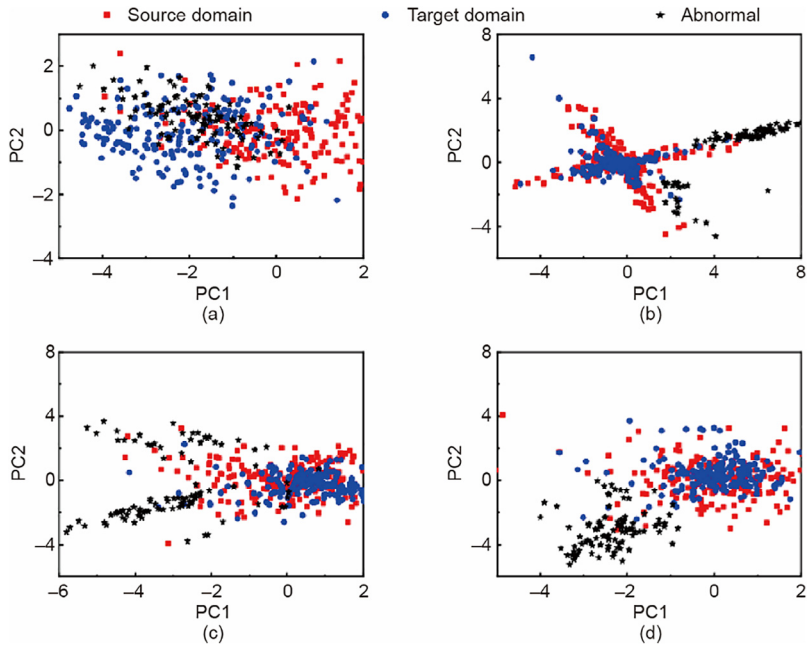

Fig. 2. Distribution of samples. (a) Raw data; (b) sparse code learned by dictionary learning with LDA-like regularization; (c) sparse code learned by dictionary learning with MMD regularization; (d) sparse code learned by dictionary learning with both regularizations.

4.1.3. Parameter sensitivity analysis experiments

The parameter α is an important adjusting parameter in dictionary learning that controls the sparsity of $\boldsymbol{S}$. In general, α can be selected by observing the sparsity of $\boldsymbol{S}$, which represents the percentage of nonzero elements in matrix $\boldsymbol{S}$. As shown in Fig. 3, a satisfactory process-monitoring result can be obtained when the sparsity rate (SPR) ranges from 20% to 50% approximately; thus, the parameter sensitivity analysis for SPR verified the robustness of the proposed method.

Next, the process-monitoring performance is shown for the modified β_1 and β_2 parameters. Let β_1 change from 10^1 to 10^5, while β_2 changes from 10 to 70. The false alarm rate (FAR) and fault detection rate (FDR) are considered to be two evaluation indexes for process monitoring. The results are shown in Figs. 4 and 5. It can be seen that the FAR is always lower than 3% and the FDR is always greater than 80% when β_1 and β_2 are changed. This performance of the process monitoring is commendable, even though β_1 and β_2 are changed in a wide range. When the values of β_1 and β_2 are too small, the learned dictionary can reconstruct the training data well; however, the irrelevant environmental interference of the training data is also learned by the dictionary. When the values of β_1 and β_2 are too big, the objective function is only to reduce the distribution difference between the domains; the learned dictionary cannot reconstruct the process data well, and underlying semantic information such as the mechanism information from the data will be lost. Both of these cases reduce the ability of the dictionary to capture the common latent information in the process data, resulting in a poor performance in process monitoring. In order to choose a suitable value for these hyper-parameters, an optimization method such as the grid search can be selected.

4.2. Performance comparison experiments

4.2.1. Datasets

Since the performance of the proposed method is commendable in the numerical simulation, we will now consider its performance in realistic industrial scenarios. Two datasets were prepared for these experiments.

(1) **CSTH**: The CSTH process is a nonlinear real platform that has been widely used as a benchmark for evaluating different

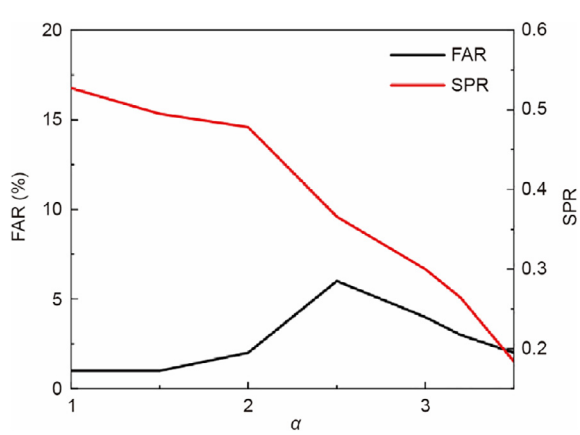

Fig. 3. Sensitivity analysis of the RTDL for FAR and SPR with parameter α. FAR: false alarm rate.

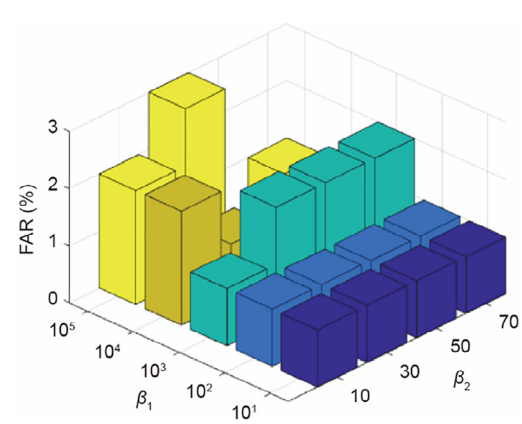

Fig. 4. Sensitivity analysis of the RTDL for FAR with parameters β_1 and β_2.

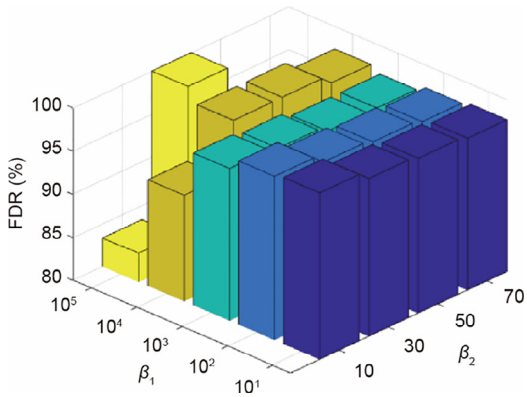

Fig. 5. Sensitivity analysis of the RTDL for FDR with parameters β_1 and β_2.

process-monitoring methods [39]. The principle of CSTH is shown in Fig. 6. In the CSTH process, there are two physical balances: a mass balance and a thermal balance. Cold water and hot water simultaneously flow into the sink, are stirred, and are heated by steam [40]. Suppose that this process has two modes, where the mode state is determined by the liquid level setting, temperature setting, and hot water valve position setting. The details are given in Table 2. To simulate environmental interference, we add an exponential distribution variable $E(0.5)$ to all observed variables in order to form the source domain and an F distribution variable $0.1\bar{F}(5,10)$ to all observed variables in order to form the target domain. An additivity fault is imposed onto the observed flow variable to generate abnormal data. We collect 100 data for each mode in the source domain and 30 normal data for each mode in the target domain in order to form training data. The testing data consist of 50 data for each mode and 200 abnormal data in the target domain.

(2) **Wind turbine system**: The data on the wind turbine system come from a wind power company in Beijing. The data were sampled every minute from 1 January 2011 to 11 November 2011 by eight wind turbines, and the 15-dimensional data shown in Table 3 were used for process monitoring and mode classification. There are different manifold structures among the data of each wind turbine, which can be considered to be the operating modes of the wind turbine system. The temperature and wind in summer are different from those in winter, resulting in different operating temperatures and operating powers for the wind turbine, and leading

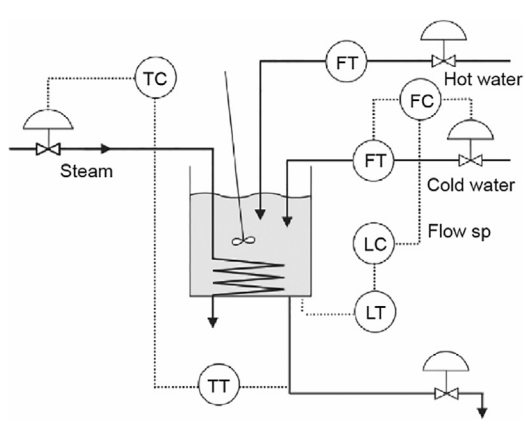

Fig. 6. CSTH schematic diagram. TC: temperature controller; FC: flow controller; LC: liquid level controller; TT: temperature sensor; FT: flow sensor; LT: liquid level sensor; sp: set point. Reproduced from Ref. [40] with permission of Elsevier, ©2008.

Table 2
The sensor current signal setting parameters of CSTH.

Mode	Liquid level sensor cuttent (mA)	Temperature sensor current (mA)	Water valve position sensor current (mA)
Mode 1	12	10.5	5.5
Mode 2	12	10.5	5.0

to data distribution divergence. The wind turbine system is affected by different environmental interferences in different seasons. We assume the winter season to be the source domain while the summer season is the target domain. A detailed description is shown in Table 4. The training data is composed of 350 normal pieces of data for each mode in the source domain and 50 normal pieces of data for each mode in the target domain. A number of 50 normal pieces of data for each mode and 300 abnormal pieces of data in the target domain are taken to constitute the testing data. Due to the huge difference in the dimensions of the wind turbine system data, all the data were normalized before the experiment.

4.2.2. Comparison experiments for process monitoring

In order to evaluate the proposed method quantitatively, two other novel dictionary learning methods and an adaptive monitoring method are used for comparison. In addition, two data-processing strategies mentioned in Section 2.3 are used. Comparison methods include: label consistent K-singular value decomposition (LC-KSVD)(S+T), LC-KSVD(T), Fisher discrimination dictionary learning (FDDL)(S+T), FDDL(T), moving window PCA (MWPCA), and RTDL.

The LC-KSVD [41], FDDL [31], and MWPCA [42] methods are three state-of-the-art methods for process monitoring. LC-KSVD and FDDL possess the ability of mode classification, while MWPCA is another adaptive method for process monitoring. Here, LC-KSVD(S+T) and FDDL(S+T) refer to the LC-KSVD method and FDDL method, respectively, using all the source domain training data and target domain training data directly as the input data points without considering the different characteristics between the domains. LC-KSVD(T) and FDDL(T) refer to the LC-KSVD method and FDDL method, respectively, only using the target domain training data as the input data points. The MWPCA method uses all the source domain training data and the target domain training data directly as input data points without considering the different characteristics between domains. RTDL uses the source domain training data and the target domain training data discriminately as the input data points. In order to make the comparison fair, the size of the dictionary and other parameters are set to be the same. In order to compare the performance of the models, we refer to two indexes: FAR and FDR.

The results are shown in Figs. 7 and 8. As shown in these figures, the proposed method outperforms the baselines in terms of accuracy in both of the realistic datasets. There are also some interesting results. In CSTH, the FDR of LC-KSVD(S+T) is close to the FDR of LC-KSVD(T), but the FAR of LC-KSVD(S+T) is clearly greater than the FAR of LC-KSVD(T). This result agrees with the observation that if the distribution divergence of the domains is ignored, the model may easily confuse inter-domain difference information with abnormal information, leading to a poorer process-monitoring result.

4.2.3. Comparison experiments for mode classification

The proposed method can deal with multimode data. In addition, when data is detected to be normal data, mode classification can be carried out; thus, in this section, the effectiveness of the mode classification is evaluated by a comparison with the

Table 3
Features used in the wind turbine system experiment.

Serial number	Feature	Serial number	Feature	Serial number	Feature
001	Active power	006	First phase current	007	Second phase current
008	Third phase current	012	Motor speed	026	Rotor speed
035	First phase voltage	036	Second phase voltage	038	Third phase voltage
038	Average wind speed	061	Gear shaft temperature	065	Gear box temperature
068	Engine room temperature	069	External ambient temperature	098	The stator temperature

Table 4
Detailed description of the wind turbine system data.

Domain	Model		Abnormal data
	Mode 1	Mode 2	
Source domain	Normal data in January 2011 in wind turbine 20	Normal data in January 2011 in wind turbine 70	No
Target domain	Normal data in July and August 2011 in wind turbine 20	Normal data in July and August 2011 in wind turbine 70	Abnormal data in July and August 2011 in wind turbine 70

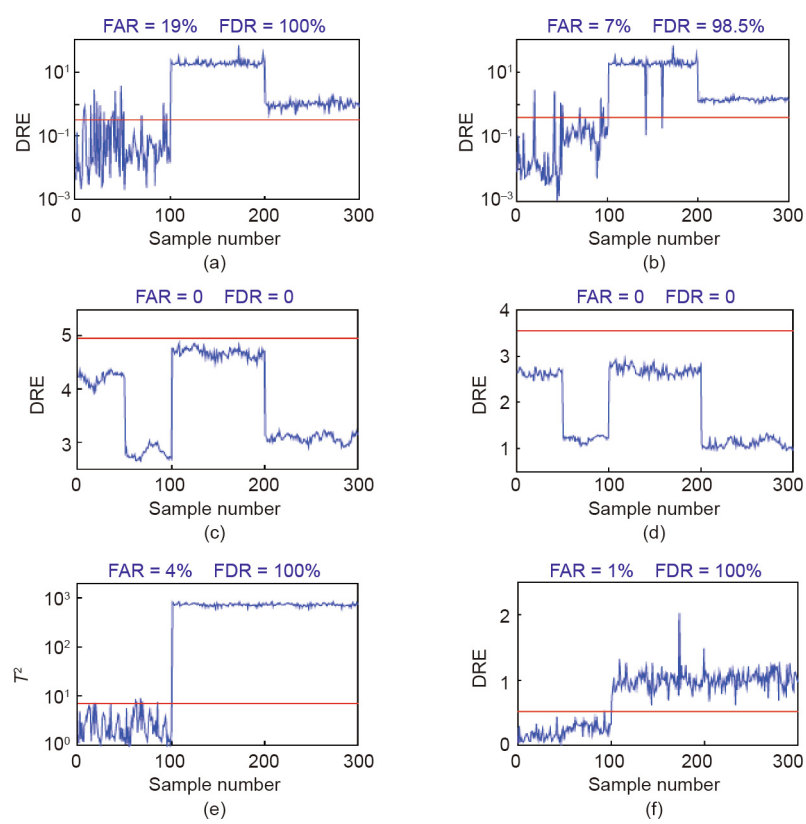

Fig. 7. Process-monitoring results of baselines and the proposed method on CSTH. (a) The DRE statistic of LC-KSVD(S+T); (b) the DRE statistic of LC-KSVD(T); (c) the DRE statistic of FDDL(S+T); (d) the DRE statistic of FDDL(T); (e) the T^2 statistic of MWPCA; and (f) the DRE statistic of RTDL. DRE: dictionary reconstruction error.

baselines. Like the parameters setting, baselines can be used as the comparison experiment for process monitoring. Note that the mode classification task cannot be carried out by the MWPCA method, so that method is not utilized for the mode classification experiment. In order to compare the performance of the models, we refer to two indexes: mode 1 accuracy and mode 2 accuracy. The result is shown in Table 5. It can be seen that the performance of the proposed method in mode classification is better than that of the baselines, which further verifies the effectiveness of the proposed method.

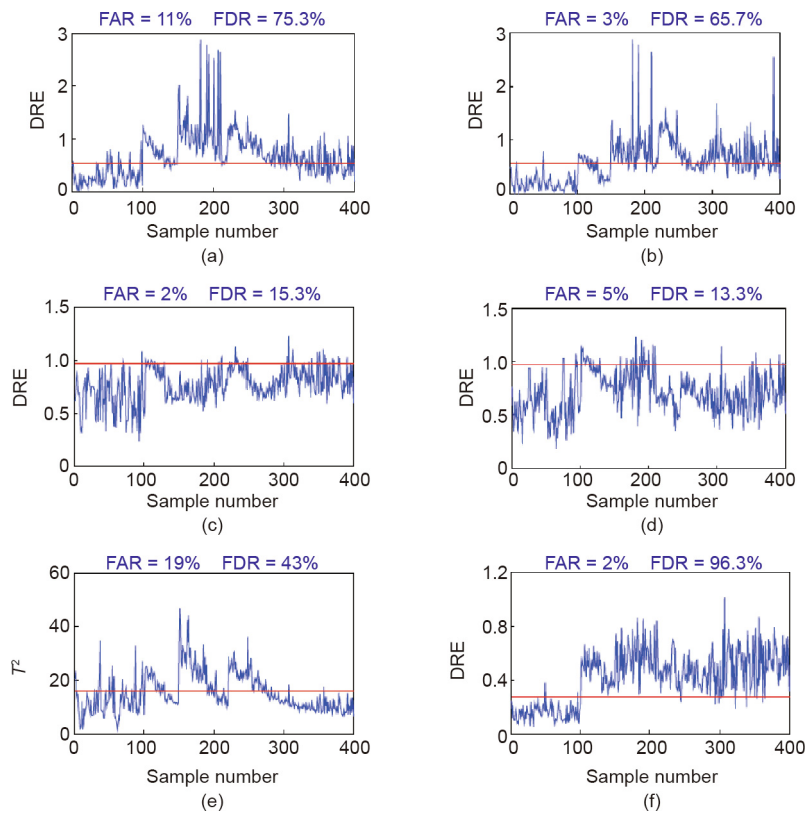

Fig. 8. Process monitoring results of the baselines and the proposed method on the wind turbine system. (a) The DRE statistic of LC-KSVD(S+T); (b) the DRE statistic of LC-KSVD(T); (c) the DRE statistic of FDDL(S+T); (d) the DRE statistic of FDDL(T); (e) the T^2 statistic of MWPCA; and (f) the DRE statistic of RTDL.

Table 5
Mode classification result.

Method	CSTH		Wind turbine system	
	Mode 1 accuracy	Mode 2 accuracy	Mode 1 accuracy	Mode 2 accuracy
LC-KSVD(S+T)	96	98	92	74
LC-KSVD(T)	0	64	76	84
FDDL(S+T)	0	**100**	18	**88**
FDDL(T)	0	**100**	48	42
RTDL	**98**	**100**	**94**	**88**

5. Conclusions

Since industrial processes are often affected by a changeable operating environment, online monitoring data and historical training data do not always follow the same distribution. As a result, learned process-monitoring models based on historical training data cannot carry out the task of monitoring the online streaming data accurately. In this paper, an RTDL method was proposed. The proposed method is a synergy framework of representative learning and domain adaptive transfer learning. That is, the dictionary learning method, which projects the raw data into a subspace, is first used to learn a common dictionary to represent both the source domain data and the target domain data. After reducing the inter-domain distribution distance and the intra-mode distance in the subspace, the distribution divergence caused by environmental interference is eliminated, which improves the ability of the learned dictionary to represent internal semantic information, such as mechanism information. Through extensive experiments including a numerical simulation, the CSTH benchmark platform, and a real wind turbine system, the superiority of the proposed method for the domain transfer problem was demonstrated. Thus, it can be concluded that the proposed method can transfer knowledge from a single source domain to a single target domain. Since industrial processes usually encounter several operating environments, future works will focus on realizing knowledge transfer from multiple source domains to multiple target domains.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (61988101) and in part by the National Key R&D Program of China (2018YFB1701100).

Compliance with ethics guidelines

Chunhua Yang, Huiping Liang, Keke Huang, Yonggang Li, and Weihua Gui declare that they have no conflict of interest or financial conflicts to disclose.

References

[1] Qian F, Zhong W, Du W. Fundamental theories and key technologies for smart and optimal manufacturing in the process industry. Engineering 2017;3(2):154–60.
[2] Mao S, Wang B, Tang Y, Qian F. Opportunities and challenges of artificial intelligence for green manufacturing in the process industry. Engineering 2019;5(6):995–1002.
[3] Zhao C, Sun Y. Multispace total projection to latent structures and its application to online process monitoring. IEEE Trans Control Syst Technol 2014;22(3):868–83.
[4] Liu Q, Qin SJ, Chai T. Multiblock concurrent PLS for decentralized monitoring of continuous annealing processes. IEEE Trans Ind Electron 2014;61(11):6429–37.
[5] Jiang Q, Yan X, Huang B. Review and perspectives of data-driven distributed monitoring for industrial plant-wide processes. Ind Eng Chem Res 2019;58(29):12899–912.
[6] Shang C, You F. Data analytics and machine learning for smart process manufacturing: recent advances and perspectives in the big data era. Engineering 2019;5(6):1010–6.
[7] Zhang W, Yang D, Wang H. Data-driven methods for predictive maintenance of industrial equipment: a survey. IEEE Syst J 2019;13(3):2213–27.
[8] Zhou P, Zhang R, Xie J, Liu J, Wang H, Chai T. Data-driven monitoring and diagnosing of abnormal furnace conditions in blast furnace ironmaking: an integrated PCA–ICA method. IEEE Trans Ind Electron 2021;68(1):622–31.
[9] Sankavaram C, Pattipati BR, Kodali A, Pattipati K, Azam M, Kumar S, et al. Model-based and data-driven prognosis of automotive and electronic systems. In: Proceedings of 2009 IEEE International Conference on Automation Science and Engineering; 2009 Aug 22–25; Bangalore, India. New York City: Curran Associates; 2009. p. 22–5.
[10] Feng L, Zhao C. Fault description based attribute transfer for zero-sample industrial fault diagnosis. IEEE Trans Ind Inform 2021;17(3):1852–62.
[11] Liang YC, Wang S, Li WD, Lu X. Data-driven anomaly diagnosis for machining processes. Engineering 2019;5(4):646–52.
[12] Jiang Q, Yan X, Huang B. Deep discriminative representation learning for nonlinear process fault detection. IEEE Trans Autom Sci Eng 2020;17(3):1410–9.
[13] Jiang Q, Yan S, Yan X, Yi H, Gao F. Data-driven two-dimensional deep correlated representation learning for nonlinear batch process monitoring. IEEE Trans Ind Inform 2020;16(4):2839–48.
[14] Pan SJ, Yang Q. A survey on transfer learning. IEEE Trans Knowl Data Eng 2010;22(10):1345–59.
[15] Chai Z, Zhao C. A fine-grained adversarial network method for cross-domain industrial fault diagnosis. IEEE Trans Autom Sci Eng 2020;17(3):1432–42.
[16] Huang K, Wen H, Zhou C, Yang C, Gui W. Transfer dictionary learning method for cross-domain multimode process monitoring and fault isolation. IEEE Trans Instrum Meas 2020;69(11):8713–24.
[17] Wang J, Zhao C. Mode-cloud data analytics based transfer learning for soft sensor of manufacturing industry with incremental learning ability. Control Eng Pract 2020;98:104392.
[18] Hou R, Wang H, Xiao Y, Xu W. Incremental PCA based online model updating for multivariate process monitoring. In: Proceedings of the 10th World Congress on Intelligent Control and Automation; 2012 Jul 6–8; Beijing, China. New York City: Curran Associates; 2012.
[19] Jiang LY, Xie L, Wang SQ, Wang N. Monitoring batch processes using multi-model discriminant partial least squares. In: Proceedings of 2005 International Conference on Machine Learning and Cybernetics; 2005 Aug 18–21; Guangzhou, China. New York City: Curran Associates; 2005.
[20] Zhang Y, You D, Gao X, Katayama S. Online monitoring of welding status based on a DBN model during laser welding. Engineering 2019;5(4):671–8.
[21] Zeng J, Gao C, Luo S, Li Q. Online Process Monitoring Based on Incremental LPP. In: Process Monitoring Based on Incremental LPP; 2011 Jul 22–24; Yantai, China. New York City: Curran Associates; 2011.
[22] Ge Z, Song Z. Online batch process monitoring based on multi-model ICA–PCA method. In: Proceedings of the 2008 7th World Congress on Intelligent Control and Automation; 2008 Jun 25–27; Chongqing, China. New York City: Curran Associates; 2008.
[23] Peng X, Tang Y, Du W, Qian F. Multimode process monitoring and fault detection: a sparse modeling and dictionary learning method. IEEE Trans Ind Electron 2017;64(6):4866–75.
[24] Chen G, Xiong C, Corso JJ. Dictionary transfer for image denoising via domain adaptation. In: Proceedings of the 19th IEEE International Conference on Image Processing; 2012 Sep 30–Oct 3; Orlando, FL, USA. New York City: Curran Associates; 2013.
[25] Zhang K, Yuan M, Xiong Y, Qu L. Common dictionary and domain-specific dictionary based cross-domain image classification. In: Proceedings of 2017 Chinese Automation Congress (CAC); 2017 Oct 20–22; Jinan, China. New York City: Curran Associates; 2017. p. 2824–9.
[26] Jie N, Qiang Q, Chellappa R. Subspace interpolation via dictionary learning for unsupervised domain adaptation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2013 Jun 23–28; Portland, OR, USA. New York City: Curran Associates; 2013. p. 692–9.
[27] Long M, Ding G, Wang J, Sun J, Guo Y, Yu PS. Transfer sparse coding for robust image representation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2013 Jun 23–28; Portland, OR, USA. New York City: Curran Associates; 2013. p. 407–14.
[28] Huang K, Wen H, Ji H, Cen L, Chen X, Yang C. Nonlinear process monitoring using kernel dictionary learning with application to aluminum electrolysis process. Control Eng Pract 2019;89:94–102.
[29] Gretton A, Borgwardt K, Rasch MJ, Scholkopf B, Smola AJ. A kernel method for the two-sample problem. 2008. arXiv:0805.2368.
[30] Ramirez I, Sprechmann P, Sapiro G. Classification and clustering via dictionary learning with structured incoherence and shared features. In: Proceedings of 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2010 Jun 13–18; San Francisco, CA, USA. New York City: Curran Associates; 2010.
[31] Meng Y, Lei Z, Feng X, Zhang D. Fisher discrimination dictionary learning for sparse representation. In: Proceedings of 2011 International Conference on Computer Vision; 2011 Nov 6–13; Barcelona, Spain. New York City: Curran Associates; 2011.
[32] Shekhar S, Patel VM, Nguyen HV, Chellappa R. Generalized domain-adaptive dictionaries. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition; 2013 Jun 23–28; Partland, OR, USA. New York City: Curran Associates; 2013. p. 361–8.
[33] Huang K, Wu Y, Wen H, Liu Y, Yang C, Gui W. Distributed dictionary learning for high-dimensional process monitoring. Control Eng Pract 2020;98:104386.
[34] Kokiopoulou E, Chen J, Saad Y. Trace optimization and eigenproblems in dimension reduction methods. Numer Linear Algebra Appl 2011;18(3):565–602.
[35] Schölkopf B, Platt J, Hofmann T. Efficient sparse coding algorithms. Stanford: MIT Press; 2007.
[36] Fletcher R. Practical methods of optimization. 2nd ed. New York City: Wiley; 1987.
[37] Rebollo-Neira L, Lowe D. Optimized orthogonal matching pursuit approach. IEEE Signal Process Lett 2002;9(4):137–40.
[38] Chen Q, Wynne RJ, Goulding P, Sandoz D. The application of principal component analysis and kernel density estimation to enhance process monitoring. Control Eng Pract 2000;8(5):531–43.
[39] Huang K, Wu Y, Yang C, Peng G, Shen W. Structure dictionary learning-based multimode process monitoring and its application to aluminum electrolysis process. IEEE Trans Autom Sci Eng 2020;17(4):1989–2003.
[40] Thornhill NF, Patwardhan SC, Shah SL. A continuous stirred tank heater simulation model with applications. J Process Control 2008;18(3–4):347–60.
[41] Jiang Z, Lin Z, Davis LS. Label consistent K-SVD: learning a discriminative dictionary for recognition. IEEE Trans Pattern Anal Mach Intell 2013;35(11):2651–64.
[42] Jeng JC. Adaptive process monitoring using efficient recursive PCA and moving window PCA algorithms. J Taiwan Inst Chem Eng 2010;41(4):475–81.