

Received December 3, 2019, accepted December 15, 2019, date of publication December 24, 2019, date of current version January 15, 2020.

Digital Object Identifier 10.1109/ACCESS.2019.2962008

Fault Classification in Dynamic Processes Using Multiclass Relevance Vector Machine and Slow Feature Analysis

JIAN HUANG^{1,2}, XU YANG¹, YURI A. W. SHARDT³, AND XUEFENG YAN²

¹Key Laboratory of Knowledge Automation for Industrial Processes, Ministry of Education, School of Automation and Electrical Engineering, University of Science and Technology Beijing, Beijing 100083, China

²Key Laboratory of Advanced Control and Optimization for Chemical Processes, Ministry of Education, East China University of Science and Technology, Shanghai 200237, China

³Department of Automation Engineering, Technical University of Ilmenau, 98684 Ilmenau, Germany

Corresponding author: Xu Yang (yangxu@ustb.edu.cn)

This research was supported in part by the National Natural Science Foundation of China under Grant 61903026 and Grant 61673053, in part by the China Postdoctoral Science Foundation under Grant 2019M660462, in part by the Fundamental Research Funds for the Central Universities under Grant 222201917006, and in part by the National Key Research and Development Program of China under Grant 2017YFB0306403.

ABSTRACT This paper proposes a modified relevance vector machine with slow feature analysis fault classification for industrial processes. Traditional support vector machine classification does not work well when there are insufficient training samples. A relevance vector machine, which is a Bayesian learning-based probabilistic sparse model, is developed to determine the probabilistic prediction and sparse solutions for the fault category. This approach has the benefits of good generalization ability and robustness to small training samples. To maximize the dynamic separability between classes and reduce the computational complexity, slow feature analysis is used to extract the inner dynamic features and reduce the dimension. Experiments comparing the proposed method, relevance vector machine and support vector machine classification are performed using the Tennessee Eastman process. For all faults, relevance vector machine has a classification rate of 39%, while the proposed algorithm has an overall classification rate of 76.1%. This shows the efficiency and advantages of the proposed method.

INDEX TERMS Slow feature analysis, relevance vector machine, dynamic process, fault classification, process monitoring, statistical learning, support vector machine, feature extraction, process control.

I. INTRODUCTION

With the increase in industrial plant complexity, the ability to easily extract information has become more difficult, especially when considering product quality and process safety as critical parameters in the manufacturing process. Data-driven process monitoring techniques have been applied in modern industry to ensure the safety and stability of processes [1]–[8]. In recent years, many multivariate statistical process monitoring algorithms have been proposed for large-scale industrial processes. He *et al.* proposed a transition identification and process monitoring method for multimode processes using a dynamic mutual information similarity analysis [9]. A supervised non-Gaussian latent structure was introduced by He *et al.* to model the relationship between predictor and quality variables [10]. He and Zeng proposed double layer

distributed process monitoring based on hierarchical multi-block decomposition for large-scale distributed processes [11]. However, the multivariable statistical methods are not designed for fault classification. Fault detection is used to identify whether the process is under normal condition, but cannot provide any information about the fault type. The main idea of fault classification is to build a classifier on the basis of some known fault categories. Fault classification can provide the connection between the detected fault occurrence and some known faults. Recognizing the type of fault is an important step for process monitoring.

A. CLASSIFICATION METHODS

Numerous machine learning methods have been used for fault classification. Ge *et al.* proposed a kernel-driven semisupervised Fisher discriminant analysis (FDA) model for nonlinear fault classification, which incorporates the advantages of

The associate editor coordinating the review of this manuscript and approving it for publication was Gerard-Andre Capolino.

FDA for fault classification [12]. However, Chiang *et al.*, who investigated the advantages of FDA and support vector machine (SVM), showed that nonlinear SVM outperforms FDA [13]. Jing and Hou studied the multiclass classification problem of SVM and principal component analysis (PCA) [14], while Gao and Hou improved the multiclass SVM using a PCA approach and applied it to the Tennessee Eastman (TE) process [15]. An SVM classifier is the most commonly used classification technique. However, SVM has the following three disadvantages [16], [17]. First, the number of support vectors grows as the size of the training set increases, which also causes the computational complexity to rapidly increase. Secondly, the output from the SVM does not use a probabilistic approach for estimating the conditional distribution, which implies that the prediction cannot capture the uncertainty. This means that the accuracy of SVM is sensitive to the training set. Thus, the SVM training set should include a variety of training samples. Thirdly, the classification performance of SVM is sensitive to the parameters. Therefore, a probabilistic and sparse classification algorithm needs to be developed.

Relevance vector machine (RVM) [16], [17] is a classification method that does not have any of the above limitations. Given its probabilistic Bayesian framework, RVM can be applied in cases where there are limited training samples. However, no research exists on the application of RVMs to the data-driven fault classification of chemical processes. Considering the sensitivity of the RVM prediction to the input dimension of the training set in industrial data, a sound method should be used to transform the training set into a lower-dimensional feature space, which maximizes the separability between classes.

B. DYNAMIC DIMENSIONAL REDUCTION METHODS

PCA, a widely used multivariate statistical algorithm, is used to reduce the dimensions. However, PCA is a static dimension reduction method. Sometimes it is necessary to consider the presence of time delay in dynamic systems. In recent years, slow feature analysis (SFA) [18] has been a popular method for extracting dynamic features by focusing on learning the time features that have a slow frequency. Shang *et al.* pointed out that the time dynamics in the extracted slow features (SFs) is an indicator of process changes [19]. Modified SFA methods have also been proposed [20], [21]. SFA can extract the dynamic representations of the original data from different levels of dynamics. The above studies have shown that SFA can easily extract dynamic features. However, in previous studies, the SFA algorithm was only used for fault detection, which does not provide information about the fault type. In fact, the SFA algorithm has not been applied to fault classification.

C. MOTIVATION FOR THIS PAPER

In modern chemical processes, sufficient training samples for fault classifiers are usually not available, which worsens the performance of traditional classifiers. RVM has the advantages of sparsity and probabilistic prediction. It can be applied

when there are insufficient training samples. Furthermore, many chemical processes are large dimensional and have dynamic characteristics. Thus, such processes require that a dynamic dimensional reduction method be used to transform the training set into a lower-dimensional feature space. Since the SFA algorithm shows the inner dynamic characteristics of the faulty samples, it will indirectly increase the classification performance of RVM. Therefore, it seems plausible to combine SFA and RVM.

The contributions of this paper are the development of a robust, sparse RVM fault classification method that can be used when there are insufficient training samples and an SFA dimension reduction method that when combined with the RVM classifier can extract the distinct features, and hence, enhance the classification results. This paper proposes a modified multiclass RVM with SFA fault classification strategy for use in industrial processes. First, an SFA dynamic feature extraction model is built on the benchmark data. Then the important features are selected in order to reduce the dimension. After reducing the dimension, the high-dimensional training set is transformed into a low-dimensional dynamic feature space, which decreases the computational complexity of the RVM algorithm. Afterwards, some typical fault samples are collected and used for RVM training. RVM classifies the type of fault, providing a connection between the detected faulty samples and the known faults.

D. NOMENCLATURE

I :	Identity matrix
$s(t)$:	Slow feature vector
S_k :	Selected slow feature matrix
s_{test} :	Slow features of the test sample
t_i :	The label for sample i
X :	Data matrix
x_{test} :	Testing sample
w :	Weight vector
α :	Classification parameters
M :	The number of classes
S :	Slow feature matrix
S_{train} :	Training set for RVM
\dot{s} :	First derivative of s
$x(t)$:	Input sample
X_{train} :	Training set
W :	Weight matrix
W_k :	Selected weight matrix
$\langle \cdot \rangle_t$:	Expectation

II. THEORETICAL BASIS

In this section, the SFA, SVM and RVM algorithms are reviewed. As well, the training results of the SVM and RVM algorithms are compared using a numerical example.

A. SLOW FEATURE ANALYSIS

SFA [22]–[24] extracts the slowly varying features. The input signal is expressed as $x(t) = [x_1(t), x_2(t), \dots, x_m(t)]^T$. The optimization objective of SFA is to find a transformation

function $\mathbf{g}(\mathbf{x})$ such that the feature $s(t) = \mathbf{g}(\mathbf{x}(t))$ varies as slowly as possible [22], [23]. The function $\mathbf{g}(\mathbf{x}) = [\mathbf{g}_1(\mathbf{x}), \mathbf{g}_2(\mathbf{x}), \dots, \mathbf{g}_m(\mathbf{x})]^T$ is the transformation function, while $\mathbf{s}(t) = [s_1(t), s_2(t), \dots, s_m(t)]^T$ is the slow feature. The optimization problem for SFA is

$$\min_{\mathbf{g}(\cdot)} \langle \dot{s}_i^2 \rangle_t \quad (1)$$

with the constraints

$$\langle s_i \rangle_t = 0 \quad (\text{zero mean}), \quad (2)$$

$$\langle s_i^2 \rangle_t = 1 \quad (\text{unit variance}), \quad (3)$$

$$\forall i \neq j: \langle s_i s_j \rangle_t = 0 \quad (\text{uncorrelated}), \quad (4)$$

The SFA algorithm is formalized as

$$\mathbf{s} = \mathbf{W}\mathbf{x} \quad (5)$$

where \mathbf{W} is the weight matrix which is expressed as $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \dots, \mathbf{w}_m]^T$. The whitening step is carried out by the singular value decomposition (SVD) of $\mathbf{R} = \langle \mathbf{x}(t)\mathbf{x}(t)^T \rangle_t$, which is given as

$$\mathbf{R} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \quad (6)$$

The whitening matrix is $\mathbf{Q} = \mathbf{\Lambda}^{-1/2}\mathbf{U}^T$. Then, the whitening step is written as

$$\mathbf{z} = \mathbf{\Lambda}^{-1/2}\mathbf{U}^T\mathbf{x} = \mathbf{Q}\mathbf{x} \quad (7)$$

Based on (5) and (7), the SF vector is given as

$$\mathbf{s} = \mathbf{W}\mathbf{x} = \mathbf{W}\mathbf{Q}^{-1}\mathbf{z} = \mathbf{P}\mathbf{z} \quad (8)$$

where $\mathbf{P} = \mathbf{W}\mathbf{Q}^{-1}$. Clearly, $\langle \mathbf{z}\mathbf{z}^T \rangle_t = \mathbf{Q}\langle \mathbf{x}\mathbf{x}^T \rangle_t\mathbf{Q}^T = \mathbf{I}$ and $\langle \mathbf{z} \rangle_t = 0$. Constraints (3) and (4) imply that

$$\langle \mathbf{s}\mathbf{s}^T \rangle_t = \mathbf{P}\langle \mathbf{z}\mathbf{z}^T \rangle_t\mathbf{P}^T = \mathbf{P}\mathbf{P}^T = \mathbf{I} \quad (9)$$

The optimization problem for SFA is to find an orthogonal matrix \mathbf{P} to minimize $\langle \dot{s}_i^2 \rangle_t$, which is given as

$$\langle \dot{s}_i^2 \rangle_t = \mathbf{p}_i^T \langle \dot{\mathbf{z}}\dot{\mathbf{z}}^T \rangle_t \mathbf{p}_i \quad (10)$$

The optimal solution is obtained by taking the SVD of the covariance matrix $\langle \dot{\mathbf{z}}\dot{\mathbf{z}}^T \rangle_t$, that is,

$$\langle \dot{\mathbf{z}}\dot{\mathbf{z}}^T \rangle_t = \mathbf{P}^T\mathbf{\Omega}\mathbf{P} \quad (11)$$

where \mathbf{P} is the eigenvector matrix and $\mathbf{\Omega} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_m)$ is the eigenvalue matrix, where the eigenvalues λ_i are placed in ascending order. Thus, $\langle \dot{s}_i^2 \rangle_t = \mathbf{p}_i^T \langle \dot{\mathbf{z}}\dot{\mathbf{z}}^T \rangle_t \mathbf{p}_i = \lambda_i$. The weight matrix \mathbf{W} is then

$$\mathbf{W} = \mathbf{P}\mathbf{Q} = \mathbf{P}\mathbf{\Lambda}^{-1/2}\mathbf{U}^T \quad (12)$$

B. SUPPORT VECTOR MACHINE

The main idea of the SVM classifier is to find support vectors that define the bounding hyperplanes, so as to maximize the margin between both planes. Let the training set be (\mathbf{x}_i, t_i) for $i = 1, 2, \dots, n$ and $t_i \in \{-1, 1\}$, where \mathbf{x}_i is the i th input sample and t_i is the label corresponding to sample \mathbf{x}_i . The SVM optimization problem requires solving the optimization problem

$$\begin{aligned} \min & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C \sum_{i=1}^n \xi_i \\ & t_i(\mathbf{w}^T\mathbf{z}_i + b) \geq 1 - \xi_i \quad \text{with } \xi_i \geq 0 \text{ for } i = 1, 2, \dots, n \end{aligned} \quad (13)$$

where the slack variable ξ_i represents the violation for training sample \mathbf{x}_i . A penalty parameter C is introduced to control the total violation while maximizing the margin. This introduces a trade-off between maximizing the margin and minimizing the violation. As well, a nonlinear transformation $\mathbf{z} = \Phi(\mathbf{x})$ is used to project the training data onto a high-dimensional linear space. Based on functional theory, a kernel function should satisfy the Mercer condition.

The Lagrange dual problem is

$$\begin{aligned} \min_{\boldsymbol{\alpha}} & \frac{1}{2}\boldsymbol{\alpha}^T(\tilde{\mathbf{d}} \cdot \tilde{\mathbf{d}}^T)\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha} \\ \text{s.t.} & : \boldsymbol{\alpha}^T\mathbf{t} = 0, \quad 0 \leq \boldsymbol{\alpha} \leq C \end{aligned} \quad (14)$$

where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \dots, \alpha_n]^T$ is the Lagrange multiplier vector with $\alpha_i \geq 0$, $\mathbf{t} = [t_1, t_2, \dots, t_n]^T$, $\tilde{\mathbf{d}} = [t_1\mathbf{z}_1, t_2\mathbf{z}_2, \dots, t_n\mathbf{z}_n]^T$, and \mathbf{e} is a unit column vector, that is, a column vector where each entry is 1. The input sample \mathbf{x}_i for which $\alpha_i \neq 0$ is called a support vector (SV).

The discriminant function $f(\mathbf{x})$ for a test sample \mathbf{x} can be obtained from

$$f(\mathbf{x}) = \text{sign} \left(\sum_{\mathbf{x}_i \in \text{SVs}} \alpha_i t_i \Phi(\mathbf{x} \cdot \mathbf{x}_i) + b \right) \quad (15)$$

C. RELEVANCE VECTOR MACHINE

RVM seeks to maximize the possible sparsity, by discarding a large number of extremely small weights and thus, training samples that have no effect on the classification function. Originally, RVM [16], [17], [25], [26] was derived for binary classification. RVM is designed to predict the posterior probability of the membership of the input \mathbf{x} . The training set for RVM is defined as (\mathbf{x}_i, t_i) for $i = 1, 2, \dots, n$. The learning function is

$$y(\mathbf{x}; \mathbf{w}) = \sum_{i=1}^n \omega_i K(\mathbf{x}, \mathbf{x}_i) + \omega_0 \quad (16)$$

where $K(\mathbf{x}, \mathbf{x}_i)$ is a kernel transformation and ω_i is the weight. Generally, a Gaussian kernel function is used. Bayes' rule gives the posterior probability of \mathbf{w} , that is,

$$p(\mathbf{w}|t, \boldsymbol{\alpha}) = \frac{p(t|\mathbf{w}, \boldsymbol{\alpha})p(\mathbf{w}|\boldsymbol{\alpha})}{p(t|\boldsymbol{\alpha})} \quad (17)$$

where $p(t|w, \alpha)$ is the posterior probability, $p(w|a)$ is the conditional prior likelihood for the hyperparameters $\alpha = [\alpha_1, \dots, \alpha_n]^T$, and $p(t|a)$ is the probability of the evidence. The logistic sigmoid function $\sigma(y) = 1/(1 + e^{-y})$ and Bernoulli distribution $p(t|x)$ are used to generalize the linear model. Thus, (17) can be rewritten as

$$p(t|w, \alpha) = \prod_{i=1}^n \sigma\{y(x_i; w)\}^{t_i} [1 - \sigma\{y(x_i; w)\}]^{1-t_i} \quad (18)$$

where $t = [t_1, t_2, \dots, t_n]^T$, $t_i \in \{0, 1\}$, and $w = [\omega_0, \omega_1, \dots, \omega_n]^T$. In (17), $p(w|a)$ is the Gaussian kernel function, where the weights w_i are obtained from the parameter α_i using

$$p(w|\alpha) = \prod_{i=1}^n \frac{\sqrt{\alpha_i}}{\sqrt{2\pi}} \exp\left(-\frac{\alpha_i w_i^2}{2}\right) \quad (19)$$

The approximation procedure is based on Laplace's method [27], which consists of the following three steps:

1. Given a fixed value of α , find the maximum posterior weights w_{MP} to provide the location of the mode of the posterior distribution. To solve the problem, the penalized logistic log-likelihood function is introduced. Since $p(w|t, \alpha) \propto P(t|w)p(w|\alpha)$, the weights w_{MP} can be obtained from

$$\begin{aligned} & \log\{p(t|w)p(w|\alpha)\} \\ &= \sum_{i=1}^n [t_i \log y_i + (1 - t_i) \log(1 - y_i)] - \frac{1}{2} w^T A w \quad (20) \end{aligned}$$

where $y_i = \sigma\{y(x_i; w)\}$ and $A = \text{diag}(\alpha_0, \alpha_1, \dots, \alpha_n)$.

2. Laplace's method is used to find a quadratic approximation of the log-posterior based on its mode. Equation (20) is solved using the second-order Newton optimization method. The Hessian is

$$\nabla_w \nabla_w \log p(w|t, \alpha)|_{w_{MP}} = -(\Phi^T B \Phi + A) \quad (21)$$

where $\Phi = [\phi(x_1), \phi(x_2), \dots, \phi(x_n)]^T$ is a structure matrix, in which $\phi(x_i) = [1, K(x_i, x_1), K(x_i, x_2), \dots, K(x_i, x_n)]^T$; and $B = \text{diag}(\beta_1, \beta_2, \dots, \beta_n)$ is a diagonal matrix with $\beta_i = \sigma\{y(x_i)\} [1 - \sigma\{y(x_i)\}]$. The Hessian matrix is converse and inverted to give $\Sigma = (\Phi^T B \Phi + A)^{-1}$ for the Gaussian approximation to the posterior function at w_{MP} . Given that $\nabla_w \log p(w|t, \alpha)|_{w_{MP}} = 0$, w_{MP} can be written as $w_{MP} = \Sigma \Phi^T B t$.

3. The hyperparameters α are updated using Σ and w_{MP}

$$\alpha_i^{new} = \frac{\gamma_i}{\mu_i^2} \quad (22)$$

where $\gamma_i = 1 - \alpha_i \Sigma_{ii}$, Σ_{ii} is the (i, i) diagonal entry of the matrix Σ , and $\mu = [\mu_1, \dots, \mu_n]^T$ is the posterior mean weight vector.

This optimization procedure forces most of the α_i to infinity, which implies that, based on (19), the corresponding weights w_i are discarded. The remaining vectors, which are called the relevance vectors (RVs), give a sparse solution.

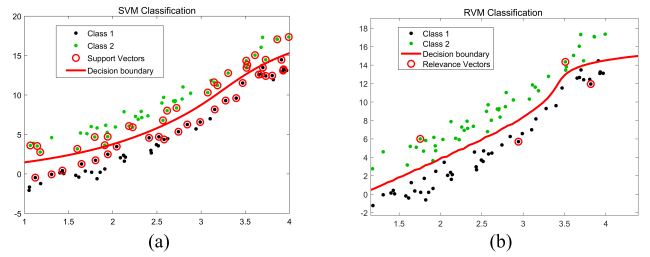


FIGURE 1. Training results for the numerical example: (a) SVM classifier and (b) RVM classifier.

The discriminant function $f(x)$ can be calculated as

$$f(x) = \sum_{i=1}^r w_i K(x, x_i) + w_0 \quad (23)$$

where r is the number of RVs, x_i is the RV, and w_i represents the remaining weights.

Fault classification usually involves more than 2 classes, whereas RVM and SVM are intrinsically binary classifiers. To convert binary classifiers into multiclass classifiers, the general strategy is to combine an ensemble of binary classifiers based on some decision rules. In this study, the one-versus-one strategy is used [17]. In the one-versus-one approach, an ensemble of models, whose training set is made of any two classes is designed. Therefore, the total number of classifiers is $M(M - 1)/2$. When testing an unknown sample, all the discriminant functions need to be calculated. A voting mechanism is used to count the score of each category. The category that obtains the most votes determines the category of the test sample.

D. COMPARISON OF SVM AND RVM USING A NUMERICAL EXAMPLE

Consider the following numerical example with the measurements $x = [x_1, x_2]$

$$x_2 = x_1^2 + ya + v \quad (24)$$

where x_1 is uniformly distributed over the interval $[1, 4]$, v is a random Gaussian noise with zero mean and variance of 0.25, a is an unmeasurable variable with a uniform distribution over the interval $[1, 3]$, and $y \in \{-1, 1\}$ is the label for sample x . The training set consists of 100 samples of which the first 50 samples are labelled 1 and the last 50 samples are labelled -1.

Figure 1 shows the results of training the SVM and RVM classifiers. The subfigure on the left shows the results for the SVM classifier, while the subfigure on the right shows the same for the RVM classifier. The circled samples represent the support (respectively, relevant) vectors. Comparing the two subfigures, it can be seen that SVM requires 40 support vectors, while RVM only requires 4 relevance vectors. Thus, RVM shows higher sparsity.

III. SLOW FEATURE ANALYSIS AND MULTICLASS RELEVANCE VECTOR MACHINE-BASED FAULT CLASSIFICATION METHOD

Since the number of extracted SFs using the SFA algorithm is always equal to the number of original variables, dimension reduction is required to select an appropriate number of SFs. The L_2 -norm method can be used for SF selection. The main idea of the L_2 -norm method is that a large L_2 norm of a row in \mathbf{W} is assumed to capture more process information. The squared coefficient can be computed using

$$L_i = \sum_{j=1}^m W_{ij}^2 \quad (25)$$

where W_{ij} is the (i, j) -entry of the weight matrix \mathbf{W} .

As the training set is chosen randomly from the industrial data set, the correlation between samples may be broken. To extract the inner dynamic features, benchmark data are collected. The normalized benchmark dataset is written as \mathbf{X} . The dynamic feature extraction from the dataset \mathbf{X} can be expressed as $\mathbf{S}_k = \mathbf{W}_k \mathbf{X}$, where \mathbf{S}_k is the selected slow feature matrix and weight matrix is given as \mathbf{W}_k .

The normalization for both the training set and testing set are based on the normalization of the benchmark data. Assume that the training set is expressed as \mathbf{X}_{train} . Dynamic features are extracted to avoid the computational complexity during RVM training. The dimension reduction operation is carried out before RVM training, so that,

$$\mathbf{S}_{train} = \mathbf{W}_k \mathbf{X}_{train} \quad (26)$$

Similarly, the testing sample \mathbf{x}_{test} can be written as $\mathbf{s}_{test} = \mathbf{W}_k \mathbf{x}_{test}$.

The training set for each binary RVM model can be expressed as $\mathbf{S}_{train}^{(i)}$ and $\mathbf{S}_{train}^{(j)}$, where $\mathbf{S}_{train}^{(i)}$ are the faults with label i and $\mathbf{S}_{train}^{(j)}$ are the faults with label j ($i \neq j$) chosen from the training set \mathbf{S}_{train} . The discriminant function $f_{ij}(x)$ can be obtained from the RVM training based on (23). Then, the multiclass RVM models are made up of an ensemble of binary RVM classifiers according to the one-versus-one approach.

The multiclass RVM procedure is determined as follows. First, all possible pairs of classifications are built. The total number of binary RVMs is $M(M-1)/2$. The discriminant function $f_{ij}(x)$ is used to define the assignment of each sample. The assignments to classes are expressed as $P(x, \mathbf{S}_{train}^{(i)})$ and $P(x, \mathbf{S}_{train}^{(j)})$. A vote is then taken using the vote function $S_i(x)$

$$S_i(x) = \sum_{\substack{j=1 \\ j \neq i}}^M \text{sgn}\{f_{ij}(x)\} \quad (27)$$

The assignment of the sample to a class is based on the total votes obtained by the class based on the following maximization

$$C(x) = \arg \max_{i=1,2,\dots,M} \{S_i(x)\} \quad (28)$$

The multiclass RVM approach is summarized by Algorithm 1.

Algorithm 1 The Multiclass RVM approach

Step 0: Initialization

Define the class number M . Set the training class set

$$C = \{\mathbf{S}_{train}^{(1)}, \mathbf{S}_{train}^{(2)}, \dots, \mathbf{S}_{train}^{(M)}\}.$$

Step 1: Training

For $i = 1, 2, \dots, M$

For $j = i + 1, i + 2, \dots, M$

Choose the training set $\mathbf{S}_{train}^{(i)}$ and $\mathbf{S}_{train}^{(j)}$. Obtain the discriminant function $f_{ij}(x)$

End

End

Step 2: Testing

For $i = 1, 2, \dots, M$

For $j = i + 1, i + 2, \dots, M$

Test sample \mathbf{x} using $f_{ij}(x)$ to define $P(x, \mathbf{S}_{train}^{(i)})$ and $P(x, \mathbf{S}_{train}^{(j)})$. If $P(x, \mathbf{S}_{train}^{(i)}) > P(x, \mathbf{S}_{train}^{(j)})$, set $S_i(x) = S_i(x) + 1$; otherwise, set $S_j(x) = S_j(x) + 1$.

End

End

Step 3: Decision Making

Define the class assignment for \mathbf{x} using (28).

Stop.

For the testing vector \mathbf{s}_{test} , the discriminant function set $f_{ij}(x)$ is used to determine the fault category for each binary RVM classifier. Then, a voting mechanism is used to count the score of each fault category. The test vector belongs to the fault category that obtains the most votes.

Figure 2 shows the flowchart of the SFA-RVM algorithm.

IV. SIMULATIONS AND EXPERIMENTS USING THE TE PROCESS

A. BACKGROUND ABOUT THE TE PROCESS

The TE process was introduced by Downs and Vogel [28]. The control system for the TE process is shown in Figure 3 [29]. There are 22 measured variables and 12 manipulated variables for the TE process. There are 500 samples in the normal dataset. To further enhance the TE process, 21 different faults are simulated. For the training dataset, there are 480 faulty samples for each fault. For the testing dataset, the first 160 samples in the faulty dataset are collected under normal condition, while the remaining 800 samples are faulty data. The measurements contain random noise which is carried out using a random seed. The sampling frequency for TE process is 3 minutes. The simulation dataset of the TE is provided at <http://web.mit.edu/braatzgroup/links.html>. In the fault classification procedure, the training set consists of 100 randomly chosen samples from the training faulty samples for each class.

B. FEATURE EXTRACTION AND DIMENSION REDUCTION OF THE TE PROCESS DATA

The goal of feature extraction is to retain only the most significant features. The number of SFs retained for dimension

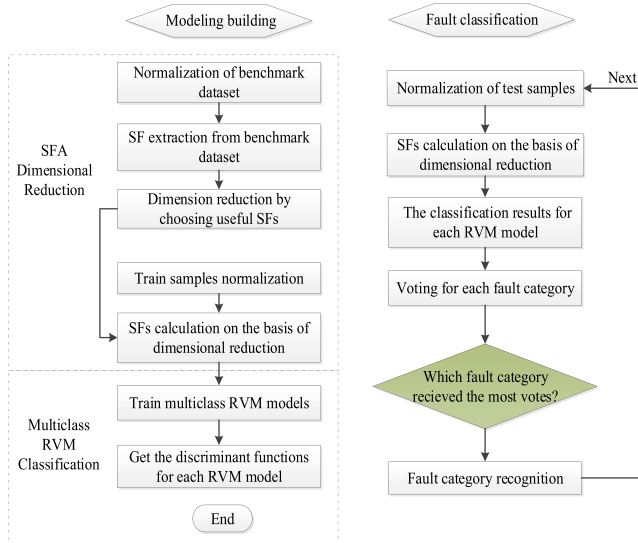


FIGURE 2. The flowchart for the proposed SFA-RVM fault classification algorithm.

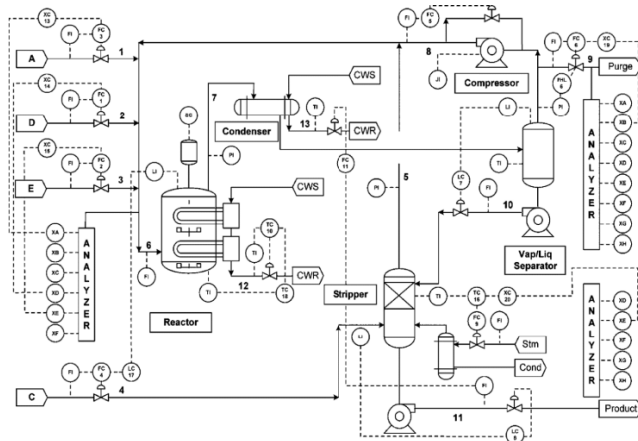


FIGURE 3. Control strategy for the Tennessee Eastman process.

reduction is based on the L_2 -norm method. The weight vectors are re-ordered in descending order based on their L_2 -norm. The number of slow features k is determined by

$$\sum_{i=1}^k L_i / \sum_{i=1}^m L_i > \theta \quad (29)$$

where θ is the selection threshold.

The number of SFs depends on the data for the current application. After testing a number of threshold values, the most important SFs are retained when the θ equals to 0.9 with the data used. In the TE process, the number of slow features is 21 according to (29). Figure 4 shows the visualization of the first, second, and third dimensions of the training set. The different categories of fault samples are separated using different colors. Since faults 3, 9, and 15 are commonly considered to be hard to detect, the visualization of the training set does not include faults 3, 9, and 15. The training samples are gathered in a small area and are hardly visually separated. The TE process contains 33 variables

TABLE 1. Fault classification results for the TE process.

CFCR/ Precision	All faults			Without faults 3, 9, and 15		
	RVM	SVM	SFA-RVM	RVM	SVM	SFA-RVM
1	0.98/0.64	0.96/1.00	0.98/0.98	0.98/0.68	0.99/1.00	0.98/0.98
2	0.94/0.98	0.96/0.99	0.93/0.99	0.94/0.98	0.98/0.96	0.94/1.00
3	0.10/0.10	0.18/0.17	0.47/0.25	—	—	—
4	0.13/0.08	0.93/0.87	0.87/0.92	0.23/0.13	0.95/0.88	0.96/0.91
5	0.13/0.09	0.39/0.15	0.97/0.93	0.18/0.14	0.54/0.47	0.98/0.97
6	0.57/1.00	0.11/0.85	1.00/0.95	0.58/1.00	0.11/0.99	1.00/0.97
7	0.89/0.90	0.92/0.99	0.99/0.99	0.91/0.96	0.98/1.00	0.99/0.98
8	0.40/0.64	0.74/0.83	0.75/0.70	0.42/0.64	0.81/0.90	0.75/0.67
9	0.10/0.09	0.26/0.25	0.25/0.21	—	—	—
10	0.10/0.10	0.40/0.41	0.68/0.67	0.14/0.13	0.49/0.54	0.83/0.65
11	0.11/0.12	0.51/0.52	0.66/0.81	0.13/0.12	0.57/0.69	0.74/0.79
12	0.38/0.61	0.51/0.79	0.77/0.81	0.40/0.54	0.64/0.89	0.76/0.84
13	0.68/0.69	0.70/0.68	0.75/0.92	0.62/0.70	0.81/0.87	0.78/0.82
14	0.50/0.77	0.97/0.96	0.96/0.97	0.50/0.71	0.98/0.97	0.97/0.98
15	0.21/0.09	0.27/0.18	0.11/0.26	—	—	—
16	0.21/0.10	0.43/0.35	0.77/0.84	0.12/0.12	0.52/0.55	0.77/0.86
17	0.47/0.59	0.82/0.85	0.87/0.91	0.48/0.54	0.82/0.86	0.88/0.93
18	0.50/1.00	0.10/0.48	0.88/0.83	0.52/1.00	0.10/1.00	0.86/0.88
19	0.25/0.31	0.27/0.41	0.82/0.84	0.29/0.29	0.44/0.53	0.77/0.95
20	0.16/0.14	0.55/0.39	0.76/0.84	0.24/0.17	0.69/0.58	0.82/0.83
21	0.40/0.30	0.74/0.33	0.61/0.61	0.45/0.37	0.82/0.24	0.75/0.69

that are combinations of latent factors. The characteristic of high dimensionality makes the classification model complex. Figures 5 and 6 show the visualization for the first 6 dimensions of the SFA dimensionally reduced dataset. Compared with Figure 4, part of the faulty samples in the SFA dimensionally reduced dataset are visually separated from the other fault categories. The extracted dynamic features maximize the separation between classes, which highlights the inner dynamic characteristics of faulty samples and indirectly increases the classification performance of RVM.

C. SIMULATION RESULTS

To test the online fault classification performance of the proposed method, regular RVM and SVM are used for comparison. The optimal parameters of the SVM are obtained using cross-validation and the grid search method. Table 1 gives the correct fault classification rate (CFCR) and precision for each fault for RVM, SVM, and SFA-RVM, while Table 2 shows the fusion matrix of the classification results using the RVM, SVM, and SFA-RVM methods for each fault type. Each row gives the fault classification results for a particular fault. The bold numbers on the diagonal show the number of samples that are properly classified. The CFCR is calculated using

$$CFCR = \frac{\text{Number of samples correctly classified}}{\text{Total number of testing samples}} \quad (30)$$

The precision for each fault is calculated by

$$\text{Precision} = \frac{\text{Number of samples correctly classified}}{\text{Total number of samples classified as this fault}} \quad (31)$$

In Tables 1 and 2, it can be seen that all three methods have very high correct fault classification rates for Faults 1, 2, and 7. The fault magnitudes for Faults 1, 2, and 7 are large,

TABLE 2. The fusion matrix for fault classification (without faults 3, 9, and 15).

NO.	Predicted class																				
	1	2	4	5	6	7	8	10	11	12	13	14	16	17	18	19	20	21			
RVM	1	783	0	2	0	0	0	4	3	2	2	0	0	2	0	0	2	0	0		
	2	0	752	5	2	0	0	10	11	4	0	0	2	5	2	0	1	3	3		
	4	0	1	180	155	0	0	7	58	72	3	8	8	55	18	0	83	80	72		
	5	0	1	208	145	0	0	6	68	68	15	18	10	66	19	0	54	63	59		
	6	275	0	6	6	461	0	26	5	2	1	1	2	7	1	1	1	1	2	3	
	7	0	0	6	9	0	725	6	0	8	19	9	0	3	0	0	6	5	4		
	8	6	2	41	78	0	3	338	51	26	46	53	2	42	16	0	20	51	25		
	10	0	2	140	110	0	0	9	108	66	7	3	7	76	24	0	75	107	66		
	11	0	1	126	85	0	0	6	73	102	3	11	35	77	31	0	84	93	73		
	12	1	0	42	69	0	11	58	61	38	319	58	7	41	18	0	18	45	14		
	13	1	0	20	35	0	5	21	25	14	80	499	5	17	13	0	18	14	33		
	14	0	1	71	36	0	0	5	23	89	0	1	403	22	64	0	35	34	16		
	16	0	3	141	70	0	0	6	93	86	2	13	3	94	30	0	74	106	79		
	17	0	1	61	47	0	0	2	27	55	3	3	65	34	388	0	47	44	23		
	18	120	0	14	29	0	8	6	19	9	90	23	0	7	9	418	11	15	22		
	19	0	0	96	65	0	0	8	71	61	1	1	8	54	22	0	269	95	49		
	20	2	2	93	87	0	0	6	76	65	2	9	9	98	41	0	79	168	63		
	21	1	0	90	49	0	0	5	42	58	0	6	7	59	24	0	53	54	352		
SVM	1	795	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	3		
	2	0	782	0	1	1	0	1	2	0	0	0	0	0	0	2	0	1	5	5	
	4	0	0	760	0	0	0	0	0	40	0	0	0	0	0	0	0	0	0	0	
	5	0	2	1	435	0	0	7	42	26	11	18	0	40	6	0	66	57	89		
	6	0	0	0	2	88	0	0	0	0	0	0	0	0	0	0	0	1	709		
	7	0	0	0	0	0	786	2	0	1	2	0	0	0	0	0	0	0	0	9	
	8	0	6	0	10	0	1	646	2	1	18	26	0	10	0	0	15	7	58		
	10	0	1	0	69	0	0	1	398	16	8	3	0	123	11	0	41	57	72		
	11	0	2	105	54	0	0	3	30	457	0	5	0	18	7	0	53	37	29		
	12	2	1	0	26	0	0	30	55	6	510	8	0	26	2	0	8	7	119		
	13	0	0	0	7	0	0	5	7	2	11	647	0	5	3	0	7	2	104		
	14	0	0	0	0	0	0	0	0	1	0	0	781	0	17	0	1	0	0		
	16	0	5	0	80	0	0	7	82	22	0	9	0	417	6	0	51	64	57		
	17	0	1	0	21	0	0	5	3	27	0	1	28	16	653	0	11	23	11		
	18	0	0	0	32	0	0	0	8	1	10	8	0	10	1	79	10	11	630		
	19	0	3	0	85	0	0	8	53	43	0	11	0	38	22	0	353	99	85		
	20	0	0	0	58	0	0	5	38	14	0	5	0	35	13	0	40	552	40		
	21	0	8	0	37	0	0	0	22	7	0	2	0	18	17	0	10	24	655		
SFA-RVM	1	780	0	0	0	0	0	8	0	0	6	3	0	0	0	0	3	0	0		
	2	0	749	0	0	0	0	36	7	2	0	0	0	0	1	2	1	1	1		
	4	0	0	768	0	0	0	2	0	29	0	0	0	0	0	0	0	1	0	0	
	5	0	0	0	786	0	0	2	1	0	9	1	1	0	0	0	0	0	0	0	
	6	0	0	0	0	800	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	7	0	0	0	0	0	792	1	0	0	3	2	0	0	0	0	2	0	0	0	
	8	8	2	0	0	0	7	635	13	9	17	41	0	18	3	0	5	28	14		
	10	0	0	0	0	0	0	20	663	11	6	10	0	8	1	5	5	21	50		
	11	0	0	74	0	0	0	18	40	591	2	9	6	0	8	3	2	11	36		
	12	6	1	0	19	2	11	44	18	5	611	28	0	18	1	23	1	5	7		
	13	5	0	0	0	18	1	13	24	3	53	625	0	7	5	22	0	8	16		
	14	0	0	0	0	0	0	0	0	2	3	0	772	0	21	2	0	0	0		
	16	0	0	0	0	1	0	13	84	13	0	13	0	618	1	3	6	5	43		
	17	0	0	1	0	0	0	13	10	23	6	3	8	5	706	8	0	10	7		
	18	0	0	0	6	0	0	18	24	9	7	3	0	3	0	686	1	3	40		
	19	0	0	0	0	0	0	60	27	19	4	3	0	21	2	3	619	13	29		
	20	0	0	0	0	1	0	31	27	15	1	5	0	7	2	21	3	658	29		
	21	0	0	0	0	0	0	30	74	21	0	13	0	17	6	5	6	27	601		

and their distinct fault features make it easy to classify the fault. However, for all methods, the correct fault classification rates for Faults 3, 9, and 15 are very poor. This can be attributed to the fact that these 3 faults have little extractable fault information, which then confuses the fault detection algorithms. The fault classification results for faults without Faults 3, 9, and 15 are also given in Table 1. Furthermore, RVM has a low correct fault classification rate (< 30%) for Faults 4, 5, 10, 11, 16, 19, and 20. These faults are conflated with each other, that is, RVM cannot distinguish these faults. Compared with RVM, SVM has better fault classification

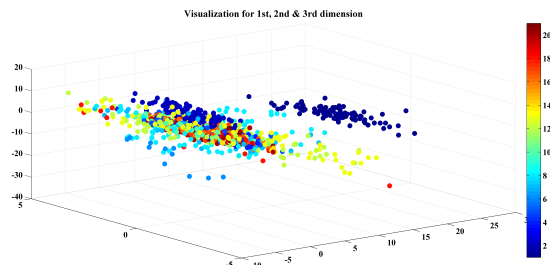


FIGURE 4. Visualization of the first, second, and third dimensions of the scaled training dataset.

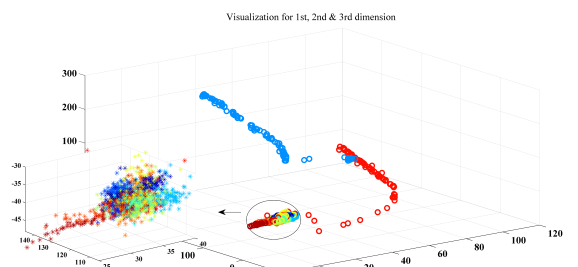


FIGURE 5. Visualization of the first, second, and third dimensions of the SFA dimensionally reduced training dataset.

results for these faults. However, SVM has poor correct fault classification rates for Faults 6 and 18, where the accuracy is only 10%. The fusion matrix in Table 2 shows that most test samples for Faults 6 and 18 are incorrectly classified as Fault 21. The proposed SFA-RVM algorithm has improved the correct fault classification rates. SFA-RVM achieves the best classification rates for Faults 5, 6, 10, 11, 12, 16, 19, and 20. Of note, the classification rates for SFA-RVM are over 70% for faults excluding Faults 3, 9, and 15 (the last 3 columns in Table 1). Furthermore, the classification rates for Faults 1, 2, 5, 6, and 7 are over 90%. Thus, it has been shown that the proposed SFA-RVM is more robust than RVM and SVM.

Fault 10 is a random variation in the C feed temperature. RVM can hardly recognize fault 10 with low fault classification rates (approximately 10%). SVM shows better fault classification results. However, the rates are lower than 50%. In the previous fault detection studies, the fault detection rates of some modified process monitoring method for Fault 10 are approximately 0.85 [7], [30], [31]. Some traditional methods such as PCA and DPCA can only detect 50% of the fault samples. The reason why Fault 10 is hard to recognize is that Fault 10 is a random fault, in which the fault amplitude varies considerably. Amplitudes of some fault samples are extremely low, which makes the fault samples similar to the normal samples. Fault detection is a binary classification problem, which is much easier than fault classification, especially for the TE process. The fault classification rate of SFA-RVM (excluding Faults 3, 9, and 15) reaches 0.83, which is close to the best fault detection rate.

Fault 11 is a random change in the reactor cooling inlet temperature. Fault 11 has the same problem as Fault 10 in

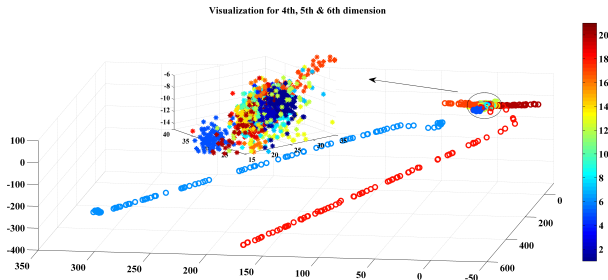


FIGURE 6. Visualization of the fourth, fifth, and sixth dimensions of the SFA dimensionally reduced training dataset.

TABLE 3. The total classification rates for the TE process.

Accuracy	RVM	SVM	Decision tree	SFA-decision tree	SFA-RVM
100 samples (all faults)	0.390	0.558	0.556	0.569	0.761
100 samples (excluding Faults 3, 9, and 15)	0.452	0.680	0.638	0.685	0.865
200 samples (excluding Faults 3, 9, and 15)	0.468	0.709	0.714	0.742	0.869

that it also has low fault amplitudes. The fault classification rates of RVM are 0.11 and 0.13. SVM has better results with a classification rate over 50%. In the previous studies, the fault detection rates of some advanced methods are approximately 0.85 [7], [30], [31]. The fault classification rate of SFA-RVM (excluding Faults 3, 9, and 15) is 0.74. It should be noted that Faults 4 and 11 occurred in the reactor cooling inlet temperature. The only difference between Faults 4 and 11 is that Fault 4 is a step fault whereas Fault 11 is random. Some of the fault samples from Fault 11 are quite similar to samples from Fault 4, which will confuse the fault classification model. It can be seen in Table 2 that there are 74 test samples belonging to Fault 11 that are classified as belonging to Fault 4. In industrial processes, different faults may result in similar evidence because of the presence of the control system. Therefore, the definition of fault categories before model training is also important.

Table 3 shows the overall classification rates for RVM, SVM, decision tree, SFA-based decision tree (SFA-decision tree), and SFA-RVM, as well as the classification rates for different number of training samples (100 and 200 training samples). For all faults (100 training samples), RVM can only recognize 39% of the test samples, while the accuracy of SVM reaches 55.8%. Furthermore, the SFA-RVM algorithm has an overall classification rate of 76.1%, which is the best fault recognition algorithm. The overall classification rates excluding Faults 3, 9, and 15 are higher than those for all faults. The RVM approach leads to a high dimensional solution that causes the accuracy to be lower than 50%. Since SVM is not a probabilistic algorithm, a complete training set is needed to contain enough samples. If the training set is

limited, the accuracy of SVM is sensitive to the size of the training set. For example, some of the TE process faults are random faults, which means that the amplitude of the faults varies. If the randomly chosen training sets differ greatly at different times, the training models of SVM may have many differences. The proposed SFA-RVM algorithm extracts the inner dynamic features to maximize the separability between classes. Dimension reduction avoids the computational complexity of the RVM training. The extracted features are good dynamic representations of the TE process. RVM has probabilistic prediction and sparsity properties, which achieve good generalization ability and robustness for small scaled training samples. The proposed method takes advantages of both SFA in extracting dynamic features and RVM in classifying.

V. CONCLUSION

In this paper, a modified RVM with the SFA algorithm is proposed for fault classification in dynamic industrial processes. SFA extracts the lower-dimensional dynamic features, which maximizes separability and reduces the computational complexity of RVM. Simultaneously, a one-versus-one multiclass RVM model is built to recognize the fault category when there are insufficient training samples. RVM classifies the type of faults, providing the connection between the detected faulty samples and known faults. RVM has the advantages of probabilistic prediction and sparsity properties for small training sets. Detailed comparative studies between the SFA-RVM method and traditional RVM and SVM were carried out using the TE benchmark process. The simulations show that the proposed SFA-RVM method has much better overall fault classification rates than either the RVM or the SVM method alone. For all faults, RVM can only recognize 39%. The SFA-RVM algorithm has an overall classification rate of 76.1%, which is a remarkable improvement.

However, the proposed method was not tested in a real process. Collecting sufficient data with faults can be an issue for real processes. Furthermore, the knowledge of fault type may be not clear, that is, the training samples for fault classifiers are not available as fast as required. Limited by the diversity and quantity of the training samples for fault classification, the dataset of a realistic experimental application is hard to obtain. Finally, even if faulty data can be obtained, it may not be possible to obtain meaningful fault classifications due to the fact that some of the faults may have a very weak signal. Thus, future work will focus on the application of this method to fault classification on a real process and improving its ability to better handle weak faults.

REFERENCES

- [1] Z. Ge, Z. Song, and F. Gao, "Review of recent research on data-based process monitoring," *Ind. Eng. Chem. Res.*, vol. 52, no. 10, pp. 3543–3562, 2013.
- [2] S. J. Qin, "Survey on data-driven industrial process monitoring and diagnosis," *Annu. Rev. Control*, vol. 36, no. 2, pp. 220–234, Dec. 2012.
- [3] S. Yin, X. Li, H. Gao, and O. Kaynak, "Data-based techniques focused on modern industry: An overview," *IEEE Trans. Ind. Electron.*, vol. 62, no. 1, pp. 657–667, Jan. 2015.

- [4] V. Venkatasubramanian, R. Rengaswamy, S. N. Kavuri, and K. Yin, "A review of process fault detection and diagnosis Part III: Process history based methods," *Comput. Chem. Eng.*, vol. 27, no. 3, pp. 327–346, Mar. 2003.
- [5] S. I. Alabi, A. J. Morris, and E. B. Martin, "On-line dynamic process monitoring using wavelet-based generic dissimilarity measure," *Chem. Eng. Res. Des.*, vol. 83, no. A6, pp. 698–705, Jun. 2005.
- [6] Q. Jiang and X. Yan, "Parallel PCA-KPCA for nonlinear process monitoring," *Control Eng. Pract.*, vol. 80, pp. 17–25, Nov. 2018.
- [7] Q. C. Jiang, X. Yan, and B. Huang, "Performance-driven distributed PCA process monitoring based on fault-relevant variable selection and Bayesian inference," *IEEE Trans. Ind. Electron.*, vol. 63, no. 1, pp. 377–386, Jan. 2016.
- [8] Q. C. Jiang, F. R. Gao, H. Yi, and X. F. Yan, "Multivariate statistical monitoring of key operation units of batch processes based on time-slice CCA," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 3, pp. 1368–1375, May 2019.
- [9] Y. C. He, L. Zhou, Z. Q. Ge, and Z. H. Song, "Dynamic mutual information similarity based transient process identification and fault detection," *Can. J. Chem. Eng.*, vol. 96, no. 7, pp. 1541–1558, Jul. 2018.
- [10] Y. He, B. Zhu, C. Liu, and J. Zeng, "Quality-related locally weighted non-Gaussian regression based soft sensing for multimode processes," *Ind. Eng. Chem. Res.*, vol. 57, no. 51, pp. 17452–17461, 2018.
- [11] Y. C. He and J. S. Zeng, "Double layer distributed process monitoring based on hierarchical multi-block decomposition," *IEEE Access*, vol. 7, pp. 17337–17346, 2019.
- [12] Z. Ge, S. Zhong, and Y. Zhang, "Semisupervised kernel learning for FDA model and its application for fault classification in industrial processes," *IEEE Trans. Ind. Informat.*, vol. 12, no. 4, pp. 1403–1411, Aug. 2016.
- [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] C. Jing and J. Hou, "SVM and PCA based fault classification approaches for complicated industrial process," *Neurocomputing*, vol. 167, pp. 636–642, Nov. 2015.
- [15] X. Gao and J. Hou, "An improved SVM integrated GS-PCA fault diagnosis approach of Tennessee Eastman process," *Neurocomputing*, vol. 174, pp. 906–911, Jan. 2016.
- [16] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, Sep. 2001.
- [17] F. A. Mianji and Y. Zhang, "Robust hyperspectral classification using relevance vector machine," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 6, pp. 2100–2112, Jun. 2011.
- [18] C. Shang, F. Yang, X. Gao, X. Huang, J. A. K. Suykens, and D. Huang, "Concurrent monitoring of operating condition deviations and process dynamics anomalies with slow feature analysis," *AIChE J.*, vol. 61, no. 11, pp. 3666–3682, May 2015.
- [19] C. Shang, B. Huang, F. Yang, and D. Huang, "Slow feature analysis for monitoring and diagnosis of control performance," *J. Process Control*, vol. 39, pp. 21–34, Mar. 2016.
- [20] C. Shang, F. Yang, B. Huang, and D. Huang, "Recursive slow feature analysis for adaptive monitoring of industrial processes," *IEEE Trans. Ind. Electron.*, vol. 65, no. 11, pp. 8895–8905, Nov. 2018.
- [21] F. H. Guo, C. Shang, B. Huang, K. C. Wang, F. Yang, and D. X. Huang, "Monitoring of operating point and process dynamics via probabilistic slow feature analysis," *Chemometr. Intell. Lab.*, vol. 151, pp. 115–125, Feb. 2016.
- [22] L. Wiskott and T. J. Sejnowski, "Slow feature analysis: Unsupervised learning of invariances," *Neural Comput.*, vol. 14, no. 4, pp. 715–770, Apr. 2002.
- [23] L. Wiskott, "Slow feature analysis: A theoretical analysis of optimal free responses," *Neural Comput.*, vol. 15, no. 9, pp. 2147–2177, Sep. 2003.
- [24] T. Blaschke, P. Berkes, and L. Wiskott, "What is the relation between slow feature analysis and independent component analysis?" *Neural Comput.*, vol. 18, no. 10, pp. 2495–2508, Oct. 2006.
- [25] A. Widodo, E. Y. Kim, J.-D. Son, B.-S. Yang, A. C. C. Tan, D.-S. Gu, B.-K. Choi, J. Mathew, "Fault diagnosis of low speed bearing based on relevance vector machine and support vector machine," *Expert Syst. Appl.*, vol. 36, pp. 7252–7261, Apr. 2009.
- [26] G. M. Foody, "RVM-based multi-class classification of remotely sensed data," *Int. J. Remote Sens.*, vol. 29, no. 6, pp. 1817–1823, 2008.
- [27] D. J. C. MacKay, "The evidence framework applied to classification networks," *Neural Comput.*, vol. 4, no. 5, pp. 720–736, Sep. 1992.
- [28] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Comput. Chem. Eng.*, vol. 17, no. 3, pp. 245–255, Mar. 1993.
- [29] P. R. Lyman and C. Georgakis, "Plant-wide control of the Tennessee Eastman problem," *Comput. Chem. Eng.*, vol. 19, no. 3, pp. 321–331, Mar. 1995.
- [30] J. Huang and X. Yan, "Double-step block division plant-wide fault detection and diagnosis based on variable distributions and relevant features," *J. Chemometrics*, vol. 29, no. 11, pp. 587–605, 2015.
- [31] Z. Ge and Z. Song, "Process monitoring based on independent component analysis-principal component analysis (ICA-PCA) and similarity factors," *Ind. Eng. Chem. Res.*, vol. 46, no. 7, pp. 2054–2063, 2007.

• • •