Research Article

Study on Support Vector Machine-Based Fault Detection in Tennessee Eastman Process

Shen Yin,[1] Xin Gao,[1] Hamid Reza Karimi,[2] and Xiangping Zhu[1]

[1] *College of Engineering, Bohai University, Liaoning 121013, China*
[2] *Department of Engineering, Faculty of Engineering and Science, University of Agder, 4898 Grimstad, Norway*

Correspondence should be addressed to Xin Gao; gaoxin.jessica@gmail.com

This paper investigates the proficiency of support vector machine (SVM) using datasets generated by Tennessee Eastman process simulation for fault detection. Due to its excellent performance in generalization, the classification performance of SVM is satisfactory. SVM algorithm combined with kernel function has the nonlinear attribute and can better handle the case where samples and attributes are massive. In addition, with forehand optimizing the parameters using the cross-validation technique, SVM can produce high accuracy in fault detection. Therefore, there is no need to deal with original data or refer to other algorithms, making the classification problem simple to handle. In order to further illustrate the efficiency, an industrial benchmark of Tennessee Eastman (TE) process is utilized with the SVM algorithm and PLS algorithm, respectively. By comparing the indices of detection performance, the SVM technique shows superior fault detection ability to the PLS algorithm.

1. Introduction

Fault detection in manufacture process aims at timely nosing out abnormal process behavior. Early fault detection of process makes the process safer, more efficient and more economical and guarantees the quality of products. Therefore, several statistical methods appeared for fault diagnosis and detection, such as artificial neural networks (ANN), fuzzy logic systems, genetic algorithms, principal component analysis (PCA), and more recently support vector machine (SVM) [1]. SVM has been extensively studied and has been used to construct data-driven modeling, classifying, and fault detection due to its better generalization ability and nonlinear classification ability [2–4]. On the other hand, since massive amounts of data produced in the industrial manufacturing process can be recorded and collected, the study of data-driven technique has become an active research field. It can benefit diverse communities including process engineering [5, 6]. Over the past few years, linear supervised classification technique, for example, K-nearest neighbor [7], PCA [8, 9], Fisher discriminant analysis [8], discriminant partial least squares (DPLS) [8–10], and nonlinear classification technique, for example, ANN [11] and SVM [12],

have been proposed and greatly improved [13, 14]. Moreover, several machine learning algorithms have been applied in real processes and simulation processes [15–17]; for example, the PCA algorithm is used to analyse product quality for a pilot plant [18], PLS and PCA are applied in chemical industry for process fault monitoring [19], and the key performance indicator prediction scheme is applied into an industrial hot strip mill in [20]. The SVM algorithm can improve the detection accuracy and it started to be used for fault detection. Now it has been extensively used to solve classification problems in many domains, for example, face, object and text detection and categorization, information and image retrieval, and so forth.

The algorithm of support vector machine (SVM) is studied detailedly in this paper. SVM is a representative nonlinear technique and it is a potentially effective technique for classifying all kinds of datasets [21]. Fault detection can be considered as a special classification problem involved in model-based method [22] and data-based method, with the purpose to timely recognise faulty condition. With the help of cross-validation algorithm to optimise parameters, the performance of classification is greatly enhanced [23–25]. Then, to test the classification performance of SVM algorithm,

a simulation model, the Tennessee Eastman process, is used for detecting fault, which has 52 variables representing the dynamics of the process [26]. In this simulation, original dataset is handled using SVM algorithm and it obtains satisfactory fault detection result. In the process of model building and test data classifying, no other theory is added, relatively decreasing the calculation time and reducing the computational burden. Compared with PLS algorithm, classifier based on SVM performs higher accuracy. Finally, using the SVM-based classifier with optimal parameters, faulty station of the process is detected.

The paper is arranged as follows. The SVM classification algorithm, PLS algorithm, and cross-validation algorithm are introduced in the next section. Sections 3 and 4 present an application to Tennessee Eastman process simulator using SVM and PLS algorithms, respectively, and SVM-based fault detection outperforms that of PLS algorithm. Section 5 summarizes a conclusion.

2. Method Algorithms

2.1. Support Vector Machines Theory. Support vector machine (SVM) is a relatively new multivariate statistical approach and has become popular due to its preferable effect of classification and regression. SVM-based classifier has better generalization property because it is based on the structural risk minimization principle [27]. SVM algorithm has the nonlinear attribute; thus, it can deal with large feature spaces [28]. Due to the aforementioned two factors, SVM algorithm begins to be used in machine fault detection. The fundamental principle of SVM is separating dataset into two classes according to the hyperplane (a decision boundary) which should have maximum distance between support vectors in each class. Support vectors are representative data points and their increasing number may increase the complexity of problem [28, 29].

This thesis uses a binary classifier with dataset and the corresponding labels. Training dataset containing two classes is given in matrix X with the form of $m \times n$ [6], in which m represents the number of observe samples, while n stands for the quantity of the observed variables. x_i is denoted as a column vector to stand for the ith row of X. Each sample is assumed to be in a positive class or in a negative class. Besides, a column vector Y serves as the class label, containing two entries -1 and 1. Denote that $y_i = 1$ is associated with one class and $y_i = -1$ with the other class. If the training dataset is linearly separable, the SVM will try to separate it by a linear hyperplane:

$$f(x) = \langle w, x \rangle + b = 0, \tag{1}$$

where w is an m-dimensional vector and b is a scalar.

The parameters w and b decide the separating hyperplane's position and orientation. A separating hyperplane is considered to be optimal if it creates maximum distance between the closest vectors and the hyperplane. The closest points in each class are denoted as support vectors. If other points in the training set are removed, the calculated decision boundary remains the same one. That is to say the support

vectors contain all information in the dataset to define the hyperplane. The distance d from a data point x_i to the separating hyperplane is

$$d = \frac{|\langle w, x \rangle + b|}{\|w\|}. \tag{2}$$

Vapink in 1995 put forward a canonical hyperplane [30], where w and b should satisfy

$$\min_i |\langle w, x_i \rangle + b| = 1. \tag{3}$$

That is to say if the nearest point is taken to the hyperplane function, the result is constrained to be 1. This restriction on the parameters is to simplify the formation of problem. In a way, as for a training data x_i, y_i, the separating hyperplane of the above-mentioned form will become

$$y_i f(x_i) = y_i (\langle w, x_i \rangle + b) \geq 1, \quad i = 1, \ldots, m. \tag{4}$$

The best separating hyperplane is the one that makes maximum distance from support vectors to decision boundary. The maximum distance is denoted as ρ.

Consider

$$\rho = \frac{1}{\|w\|}. \tag{5}$$

Hence, as for linear separable data, the optimal separating hyperplane satisfies the following function:

$$\min \phi(w) = \frac{1}{2} \|w\|^2. \tag{6}$$

To solve the optimal problem equation (6) under the constrain of (4), define the Lagrangian to be

$$\ell(w, b, \alpha) = \frac{1}{2} \|w\|^2 - \sum_{i=1}^{m} \alpha_i \left[y_i (\langle w, x_i \rangle + b) - 1 \right], \tag{7}$$

where α_i is called Lagrangian multiplier. The Lagrangian should be maximised by choosing appropriate α_i and should be minimised by w, b.

Taking the noise in the data and the misclassification of hyperplane into consideration, the above function describing the separate hyperplane equation (4) is not accurate enough. To make the optimal separating boundary to be generalised, we reformulate the described function of the separate hyperplane:

$$y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i, \quad i = i \ldots, m. \tag{8}$$

where the variable ξ_i represents a measure of distance from hyperplane to misclassified points and $\xi_i \geq 0$. To find the optimal generalised separating hyperplane, the following optimal problem should be solved:

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} \|w\|^2 + C \sum_{k=1}^{m} \xi_i \\ \text{s.t.} \quad & y_i (\langle w, x_i \rangle + b) \geq 1 - \xi_i, \\ & \xi_i \geq 0, \quad i = i \ldots, m, \end{aligned} \tag{9}$$

where the parameter C, a given value, is called error penalty. As for the above-mentioned data inseparable case, in order to simplify the optimal problem, define the Lagrangian to be

$$\ell\left(w, b, \xi, \alpha, \gamma\right) = \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i$$
$$- \sum_{i=1}^{m}\alpha_i\left[y_i\left(\langle w, x_i\rangle + b\right) - 1 + \xi_i\right] - \sum_{i=1}^{m}\gamma_i\xi_i, \tag{10}$$

where α, γ are the Lagrangian multipliers. We consider the minimization problem as original, primal problem.

Consider

$$\min_{w,b,\xi}\theta_p\left(w\right) = \min_{w,b,\xi}\max_{\alpha,\gamma}\ell\left(w, b, \xi, \alpha, \gamma\right). \tag{11}$$

When satisfying the Kuhn-Tucker condition, then the primal problem is transformed to its dual problem, which is

$$\max_{\alpha,\gamma}\theta_d\left(\alpha, \gamma\right) = \max_{\alpha,\gamma}\min_{w,b,\xi}\ell\left(w, b, \xi, \alpha, \gamma\right). \tag{12}$$

Then, the task is minimizing ℓ in (10) by adjusting the value of w, b, ξ. At the optimal point, derivatives of ℓ should be zero. The saddle-point equation is as follows:

$$\frac{\partial\ell}{\partial w} = 0, \quad \Longrightarrow \sum_{i=1}^{m}\alpha_i y_i = 0 \tag{13}$$

$$\frac{\partial\ell}{\partial b} = 0, \quad \Longrightarrow w = \sum_{i=1}^{m}\alpha_i y_i x_i \tag{14}$$

$$\frac{\partial\ell}{\partial\xi} = 0, \quad \Longrightarrow \alpha_i + \gamma_i = C. \tag{15}$$

If we take (13), (14), and (15) back into (10), we can obtain the dual quadratic optimization problem [30, 31]:

$$\max_{\alpha}W\left(\alpha\right) = \max_{\alpha}\left\{-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j\langle x_i, x_j\rangle + \sum_{k=1}^{m}\alpha_k\right\}. \tag{16}$$

Satisfying the constrains:

$$0 \le \alpha_i \le C, \quad i = 1\ldots, m,$$
$$\sum_{i=1}^{m}\alpha_i y_i = 0. \tag{17}$$

When solved the dual quadratic optimization problem shown in (16), the α_i will be obtained. Then take a look back at (14) and describe the optimal w using α_i and the form of hyperplane can be changed to

$$\langle w, x_i\rangle + b = \sum_{i,j=1}^{m}\alpha_i y_i\langle x_i, x_j\rangle + b. \tag{18}$$

TABLE 1: Formulation of Kernel functions.

Kernel	$K(x_i, x_j)$
Linear	$x_i^T \cdot x_j$
Polynomial	$(\gamma x_i^T \cdot x_j + \text{constant})^d$
Gaussian RBF functions	$e^{-\gamma\|x_i - x_j\|^2}$

γ, constant and d are kernel parameters.

The classifier implementing the optimal separating hyperplane comes out in the following form:

$$f\left(x\right) = \text{sgn}\left(\sum_{i,j=1}^{m}\alpha_i y_i\langle x_i, x_j\rangle + b\right). \tag{19}$$

However, in some cases linear classifier is not suitable; for example, data is overlapped or cannot be linearly separated. Therefore, the input vectors should be projected into a higher dimensional feature space and there the data may be linearly classified more efficiently with the use of SVM algorithm. However, it may cause computational problem due to the large vectors and high dimensionality. The idea of using Kernel function enables the calculation performed in the original space instead of in the projected high dimensioned future space, avoiding the curse of dimensionality [27, 31]. Given a feature mapping ϕ, we define the corresponding Kernel function to be

$$K\left(x_i, x_j\right) = \langle\phi\left(x_i\right), \phi\left(x_j\right)\rangle. \tag{20}$$

Thus, the linear decision hyperplane in the high feature space is

$$f\left(x\right) = \text{sgn}\left(\sum_{i,j=1}^{m}\alpha_i y_i\langle K\left(x_i, x_j\right)\rangle + b\right). \tag{21}$$

$K(x_i, x_j)$ is inexpensive to calculate. Kernel function returns an inner production of vectors in the high feature space which is evaluated in the original space not in high feature space. The commonly used four Kernel functions are as shown in Table 1.

2.2. SVM Model Selection. SVM algorithm is a very effective data classifying technique, and building a model to detect fault based on SVM algorithm is not so complex. Training dataset and testing sets are usually involved when a classification is done. Besides, every instance in the training set contains two parts: one is target value (i.e., the class label) and the other is several attributes (i.e., observed variables). The basic process of using SVM algorithm to classify data is as follows: at first build a classifier based on the training set and then use it to predict the target value of the data in testing set where only the attributes are known.

To construct a classifier model and figure out faulty data, the following procedure is used.

(i) Transform data collected from real process to the format that SVM classifier can use.

(ii) Try a few kinds of Kernels and find out the best one; then search the optimal parameters for it. This thesis uses Gaussian RBF Kernel function and the optimal parameters are obtained by using the cross-validation algorithm.

(iii) Use the optimal parameters and appropriate Kernel function to build a classifier.

(iv) Take testing data into the constructed classifier and do a test. As a result the faulty data will be figured out and in this way the faulty station can be detected.

2.2.1. RBF Kernel. Which kind of Kernel functions to use is significant, because it decides the feature space where training data is classified. Theoretically, the Kernel function is considered to be better if it can provide higher upper bound on the VC dimension. However, it requires evaluating the hypersphere enclosing all data in the nonlinear feature space, which is difficult to accomplish. Among the commonly used Kernel functions for the SVM algorithm, the RBF Kernel is preferable according to practical experience.

Kernel function can nonlinearly project original input data to a higher dimensional space; thus, SVM equipped with Kernel function is able to deal with the case where the input data cannot be linearly separated. Generally speaking, the RBF Kernel is the first choice for the following two reasons. Firstly, linear Kernel is a special RBF Kernel—when the parameter C is adjusted to a certain value, linear Kernel behaves similarly to RBF Kernel and so it is the sigmoid Kernel at some parameters. Secondly, the RBF Kernel brings fewer computational costs and fewer hyperparameters [23].

2.2.2. Cross-Validation. When using an RBF Kernel function, it needs appropriate parameters to make sure that the classifier accurately predicts unknown data. It is not known beforehand what the best parameter values are; nevertheless, the optimal parameter searching can be accomplished using cross-validation algorithm [23].

The cross-validation algorithm is a model validation method which is used to evaluate the accuracy of a predictive model. The goal of cross-validation is to give an insight on how the model generalizes to an independent dataset (i.e., an unknown dataset) by defining a dataset in the training phase to test the model.

In the process of k-fold cross-validation, original training set is randomly divided into k equal size parts. Sequentially, one subset is used as the testing dataset to test the predictive model and the rest of $k - 1$ subsets are combined as the training dataset. The aforementioned validation process should be repeated k times in all, with every subset performed as testing data once. Using the k results, the predictive result of the model is produced and it is the overall misclassification rate across all testing sets.

This thesis uses 5-fold cross-validation algorithm to find out the overall misclassification rate across all testing sets. The cross-validation process should be performed many times to pick out the parameters which make the overall misclassification minimise. In this way, the optimal parameters are found and the classifier can obtain the best accuracy rate.

2.3. PLS for Fault Detection. The PLS algorithm is introduced by Dayal and Macgregor [32] due to its simplicity and the lesser computational effort when dealing with the process monitoring with large data. In this technique, we also denote training data in the form of $m \times n$ matrix X, in which m represents the number of observed samples, while n is the quantity of the attributes and $Y \in \mathbf{R}^{m \times 1}$. The PLS algorithm projects the matrix X into a low dimensional space with latent variables, and Y can be constructed by these latent variables. The construct model is as follows:

$$X = TP^T + E = \sum_{i=1}^{A} t_i p_i^T + E,$$
$$Y = TQ^T + F = \sum_{i=1}^{A} t_i q_i^T + F, \tag{22}$$

where $T = [t_1, \dots t_A]$ is the score matrix of X. $P = [p_1, \dots p_A]$ and $Q = [q_1, \dots q_A]$ are the loading matrix of X and Y, respectively. The latent variables $t_i \in T$ $(i = 1, \dots, A)$ can be directly calculated from X by $t_i = X r_i$; $R = [r_1, \dots r_A]$ and r_i are calculated by the following:

$$r_1 = w_1,$$
$$r_i = w_i - p_1^T w_i r_1 - p_{i-1}^T w_i r_{i-1}, \quad i = 2, \dots, A, \tag{23}$$

where the $w_i (i = 1, \dots, A)$ is the weight vector of the ith deflated X. PLS decomposes X into two parts:

$$x = \hat{x} + \tilde{x},$$
$$\hat{x} = PR^T x \in S_{\hat{x}}, \tag{24}$$
$$\tilde{x} = \left(I - PR^T\right) x \in S_{\tilde{x}}.$$

Usually, T^2 statistic is used to detect abnormalities and the calculation method is

$$T^2 = x^T R \left(\frac{T^T T}{n-1} \right)^{-1} R^T x. \tag{25}$$

With a given confidence level α, the threshold for T^2 will be calculated by the following:

$$J_{\text{th},T^2} = \frac{A\left(n^2 - 1\right)}{n\left(n - A\right)} F_{l,n-l,\alpha}, \tag{26}$$

where $F_{l,n-l,\alpha}$ represents F-distribution with l and $N - 1$ degrees of freedom and its confidence level is α. If T^2 are all less than their corresponding thresholds, the process is out of fault [32].

3. Simulation

3.1. Tennessee Eastman Process. The Tennessee Eastman process (TE process) is a simulation model of a practical industrial process. The simulation model is very suitable for researching on process control technology and also applicable
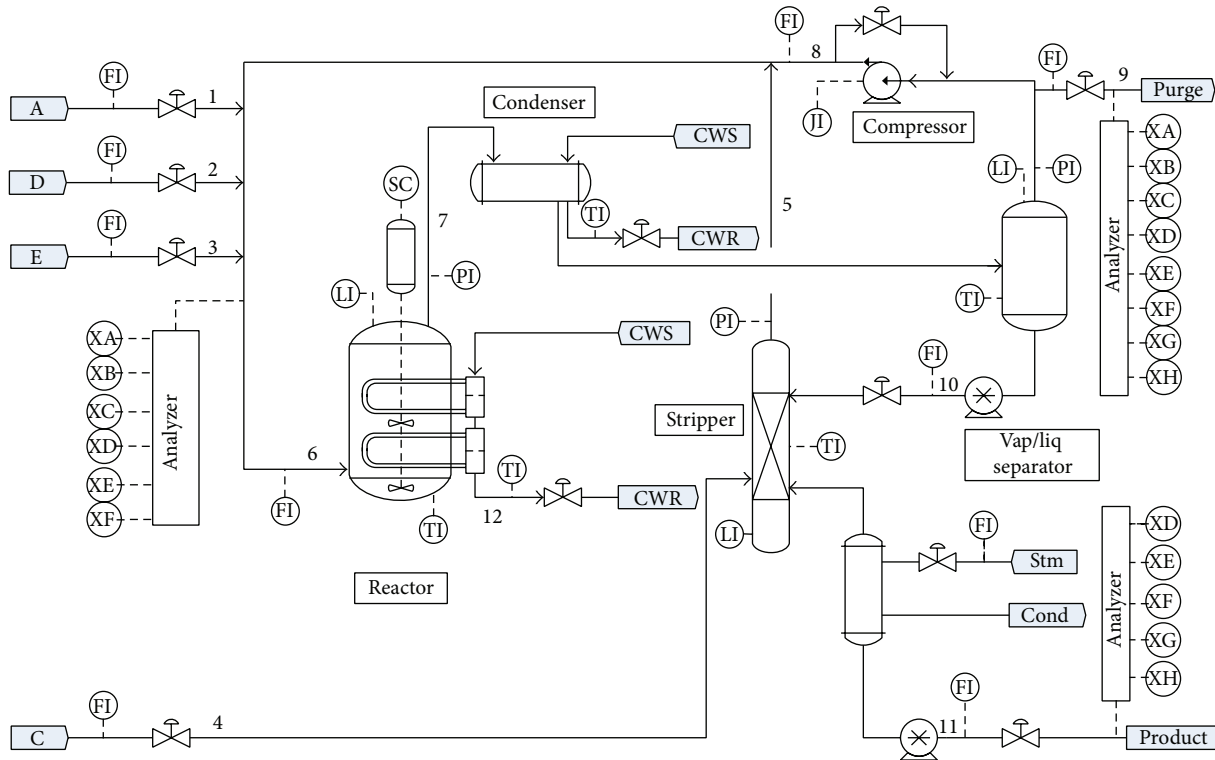
FIGURE 1: The flowsheet of TE process.

for other kinds of control issues. Figure 1 shows the flowsheet of TE process. The TE process consists of five major parts: the reactor, the product condenser, a vapor-liquid separator, a recycle compressor, and a product stripper to help accomplish reactor, separator, and recycle arrangement. More details can be found in [33, 34].

3.2. Simulated Data Intro. The TE process is a plant-wide closed-loop control structure. The simulated process can produce normal operating condition as well as 21 faulty conditions and generate simulated data at a sampling interval of 3 min. For each case (no matter normal or faulty condition), two sets of data are produced, training datasets and testing datasets. The training sets are used to construct statistical predictive model and the testing datasets are used to estimate the accuracy of the built classifier. In training sets, the normal dataset contains 500 observation samples, while each faulty dataset contains 480 observation samples. As for testing data, each dataset (both normal and faulty conditions) consists of 960 observations. In the faulty condition, faulty information emerges 8h later since the TE process is turned on. That is to say, in each faulty condition, the former 160 samples are shown normally, while the remaining 800 instances are shown really faulty and should be detected. Every observation sample contains 52 variables which consists of 22 process measure variables, 19 component measure variables, and 12 control variables. All the datasets used in this paper can be found in [1].

In the following section, the fault detection result of SVM-based classifier will be compared with the one based on the PLS algorithm using the dataset generated by TE process simulator. For SVM classifier, the RBF Kernel is used and the related parameters will be set ahead according to the result of cross-validation to make sure of the classifier's high accuracy. In the comparison, the performance is measured by fault detection rates.

4. Result and Discussion

TE process simulator can generate 22 types of conditions, containing the normal condition and 21 kinds of programmed faults caused by various known disturbances in the process. Once fault is added, all variables will be affected and some changes will emerge. According to Chiang et al. [8] and Zhang [18] detection for faults 3, 9, 15, and 21 is very difficult for there are not any observable changes in the means, variance, or the peak time. Therefore, these four faults always cannot be detected by any statistics technique; thus, the four faults are not analysed in this paper. The information of all the faults is presented in Table 2.

In order to profoundly test the detection effect of the classifier based on theory of SVM, the SVM algorithm and PLS algorithm are applied to the TE process, respectively. As for SVM classifier, all faults data are by turns combined with the normal condition data as the dataset for binary classifier. After models are built by training data, we use testing data to evaluate the prediction result through the following common indices: accuracy (Acc) and fault detection rate (FDR) [35]. Then the detection result for each fault is shown in Table 3. Table 4 represents the detection result using PLS technique.

Table 2: Descriptions of 21 faults in TE process.

Number	Fault description
IDV0	Normal
IDV1	A/C feed ratio, B composition constant
IDV2	B composition, A/C ratio constant
IDV4	Reactor cooling water inlet temperature
IDV5	Condenser cooling water inlet temperature
IDV6	A feed loss (stream 1)
IDV7	C header pressure loss-reduced availability (stream 4)
IDV8	A, B, and C feed composition (stream 4)
IDV10	C feed temperature (stream 4)
IDV11	Reactor cooling water inlet temperature
IDV12	Condenser cooling water inlet temperature
IDV13	Reaction kinetics
IDV14	Reactor cooling water valve
IDV16	Unknown
IDV17	Unknown
IDV18	Unknown
IDV19	Unknown
IDV20	Unknown

Table 3: Results of faults detection with one against one classifier.

Fault	Acc (%)	FDR (%)
IDV1	98.44	99.5
IDV2	98.12	98.12
IDV4	99.9	99.88
IDV5	91.98	90.75
IDV6	66.77	60.13
IDV7	99.58	98.91
IDV8	96.35	96
IDV10	77.71	81
IDV11	73.02	80.25
IDV12	97.4	97.75
IDV13	93.33	92.5
IDV14	89.06	91
IDV16	80.73	89.38
IDV17	83.85	81.63
IDV18	90.1	89.5
IDV19	74.48	85.88
IDV20	77.5	80.5

Consider

(i) $$\text{Acc} = \frac{\text{the quantity of data correctly predicated}}{\text{the quantity of all data}} \times 100\%$$

(ii) $$\text{FDR} = \frac{\text{number of faulty data correctly predicated}}{\text{number of all faulty data}} \times 100\%.$$

$$(27)$$

Table 4: Results of faults detection using PLS algorithm.

Fault	FDR (%)	
	$T_{\bar{x}}^2$	$T_{\bar{x}}^2$
IDV1	99.37	99.62
IDV2	98.75	98.12
IDV4	0.63	97.25
IDV5	16.65	30.04
IDV6	99	100
IDV7	29.16	100
IDV8	89.86	96.87
IDV10	18.15	43.55
IDV11	2.63	68.84
IDV12	76.85	98.37
IDV13	92.62	94.99
IDV14	2.25	100
IDV16	9.39	29.41
IDV17	4.88	91.74
IDV18	87.48	89.86
IDV19	0.38	15.14
IDV20	18.4	50.44

Figure 2 shows the predicted label y using three classifiers with the training and testing data respectively come from normal condition and fault 1, normal condition and fault 2, and normal condition and fault 4. The first 160th test, the process shows normally and y should be -1. The fault information appears at the moment of 161st sample-taken and the label y should be 1 since then. From Figure 2, we can see that the SVM classifier's prediction result for most of the time is right. In addition, Table 3 presents detailed detection indices of these 21 SVM classifiers. It is worth mentioning that the hyperparameters of the classifiers are all optimized beforehand. Without the optimization of classifier parameter, the predicted effect will not be so good. Moreover, the index accuracy will be 16.77% when using classifier's default parameters to detect fault 1. Table 4 shows the detailed indices using PLS technique to detect faults. It can be seen that the SVM classifiers with optimal hyperparameters are mostly able to detect the faulty data, and the accuracy is higher than that given by using PLS algorithm. Therefore, the SVM algorithm has a better performance of fault detection.

To further test the predictive ability of the SVM classifier, that is, the detective ability of fault in the TE process, we use normal condition data combined with three faulty condition datasets (fault 1, fault 2, fault 4) as the training data to construct a classification model and then used corresponding test data as testing data to observe the classification result. As shown in Table 5, the detection indices are also good though the computing time is a little longer with the same compute facility. Thus the SVM algorithm can perform satisfactorily on the original dataset containing 52 attributes without any transformation. In this way, in real process, with advanced compute facility, we can train normal data and all faults data to build a classifier. Once the fault label is figured out, the fault in process is detected.
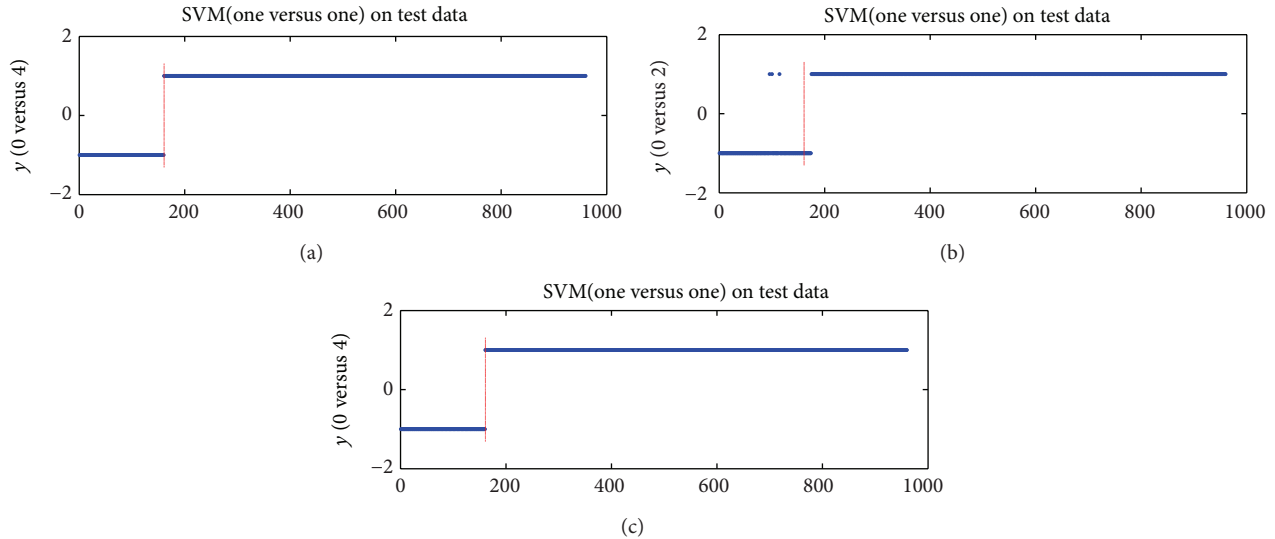
(a)

(b)

(c)

FIGURE 2: Classification results for the testing dataset by the SVM-based classifier with optimal parameters. The value $y = -1$ represents at the sample-taken moment that the process is in normal condition, while $y = 1$ stands for faulty condition. The top plot is the classification result between normal condition and fault 1, the middle one is the result between normal condition and fault 2, and the last plot is result between normal condition and fault 4. In the simulation of fault condition, the fault information appears from the moment of 161th sample-taken and it is pointed out by red line.

TABLE 5: Results of faults detection with one against all classifier.

Test description	Test data	Acc (%)	FDR (%)
0 versus 1, 2	IDV1_test data	98.02	99.62
0 versus 1, 2	IDV2_test data	95.83	98.5
0 versus 1, 2	IDV0_test data	83.44	
0 versus 1, 2, 4	IDV1_test data	96.46	99.75
0 versus 1, 2, 4	IDV2_test data	95	98.88
0 versus 1, 2, 4	IDV4_test data	96.98	100
0 versus 1, 2, 4	IDV0_test data	79.62	

From the above-shown result of fault detection on TE process, we can conclude that the classifier based on SVM algorithm is of good predictive ability. In addition, there are two facts that should be mentioned. First, before detecting normal condition or faulty condition, we have used the technique of cross-validation to optimize classifier's hyperparameters. Therefore, the performance of classifier could be the best. Second, the classification on TE process based on SVM algorithm performs satisfactorily without any other process, for example, foregoing data dealing process and attributes selection. And this feature makes the SVM classifier easy to be built. Besides, the calculation and calculate time is relatively small since the algorithms used are fewer.

5. Conclusion

TE process, a benchmark chemical engineering model, is used in this paper for fault detection. It can be found that the fault detection ability of the classifier based on SVM algorithm using the TE process's original data is satisfactory and this indicates the advantage of using nonlinear classification

when the number of samples or attributes is very large. By comparing detection performance with classifier based on the PLS algorithm, the classifier based on SVM algorithm with Kernel function shows superior accuracy rate. In addition, parameter optimization beforehand plays a great role in improving the effectiveness of classification. It also simplifies the problem and by using no other technique relatively decreases the computational load to reach a satisfactory classification result as well.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] S. Yin, *Data-driven design of fault diagnosis systems [Ph.D. dissertation]*, University of Duisburg-Essen, 2012.

[2] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, New York, NY, USA, 1995.

[3] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "Training algorithm for optimal margin classifiers," in *Proceedings of the 5h Annual ACM Workshop on Computational Learning Theory*, pp. 144–152, New York, NY, USA, July 1992.

[4] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[5] A. Kulkarni, V. K. Jayaraman, and B. D. Kulkarni, "Support vector classification with parameter tuning assisted by agent-based technique," *Computers and Chemical Engineering*, vol. 28, no. 3, pp. 311–318, 2004.

[6] L. H. Chiang, M. E. Kotanchek, and A. K. Kordon, "Fault diagnosis based on Fisher discriminant analysis and support

vector machines," *Computers and Chemical Engineering*, vol. 28, no. 8, pp. 1389–1401, 2003.

[7] K. R. Beebe, R. J. Pell, and M. B. Seasholtz, *Chemometrics: A Practical Guide*, Wiley, New York, NY, USA, 1998.

[8] L. H. Chiang, E. L. Russell, and R. D. Braatz, *Fault Detection and Diagnosis in Industrial Systems*, Springer, New York, NY, USA, 2001.

[9] A. Raich and A. Çinar, "Statistical process monitoring and disturbance diagnosis in multivariable continuous processes," *AIChE Journal*, vol. 42, no. 4, pp. 995–1009, 1996.

[10] L. H. Chiang, E. L. Russell, and R. D. Braatz, "Fault diagnosis in chemical processes using Fisher discriminant analysis, discriminant partial least squares, and principal component analysis," *Chemometrics and Intelligent Laboratory Systems*, vol. 50, no. 2, pp. 243–252, 2000.

[11] D. R. Baughman, *Neural Networks in Bioprocessing and Chemical Engineering*, Academic Press, New York, NY, USA, 1995.

[12] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, NY, USA, 2009.

[13] A. S. Naik, S. Yin, S. X. Ding, and P. Zhang, "Recursive identification algorithms to design fault detection systems," *Journal of Process Control*, vol. 20, no. 8, pp. 957–965, 2010.

[14] S. X. Ding, S. Yin, P. Zhang, E. L. Ding, and A. Naik, "An approach to data-driven adaptive residual generator design and implementation," in *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, pp. 941–946, Barcelona, Spain, July 2009.

[15] S. Ding, P. Zhang, S. Yin, and E. Ding, "An integrated design framework of fault-tolerant wireless networked control systems for industrial automatic control applications," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 462–471, 2013.

[16] H. Zhang, Y. Shi, and A. Mehr, "On H_∞ filtering for discrete-time takagi-sugeno fuzzy systems," *IEEE Transactions on Fuzzy Systems*, vol. 20, no. 2, pp. 396–401, 2012.

[17] H. Zhang and Y. Shi, "Parameter dependent H_∞ filtering for linear time-varying systems," *Journal of Dynamic Systems, Measurement, and Control*, vol. 135, no. 2, Article ID 0210067, 7 pages, 2012.

[18] Y. W. Zhang, "Enhanced statistical analysis of nonlinear process using KPCA, KICA and SVM," *Chemical Engineering Science*, vol. 64, pp. 800–801, 2009.

[19] M. Misra, H. H. Yue, S. J. Qin, and C. Ling, "Multivariate process monitoring and fault diagnosis by multi-scale PCA," *Computers and Chemical Engineering*, vol. 26, no. 9, pp. 1281–1293, 2002.

[20] S. Ding, S. Yin, K. Peng, H. Hao, and B. Shen, "A novel scheme for key performance indicator prediction and diagnosis with application to an industrial hot strip mill," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2239–2247, 2013.

[21] B. Scholkopf and A. J. Smola, *Learning with Kernels, Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, New York, NY, USA, 2002.

[22] N. F. Thornhill and A. Horch, "Advances and new directions in plant-wide disturbance detection and diagnosis," *Control Engineering Practice*, vol. 15, no. 10, pp. 1196–1206, 2007.

[23] C. W. Hsu, C. C. Chang, and C. Lin, *A Practical Guide to Support Vector Classification*, Department of Computer Science, National Taiwan University, 2010.

[24] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, Springer, New York, NY, USA, 2001.

[25] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright, "Convergence properties of the Nelder-Mead simplex method in low dimensions," *SIAM Journal on Optimization*, vol. 9, no. 1, pp. 112–147, 1998.

[26] A. Kulkarni, V. K. Jayaraman, and B. D. Kulkarni, "Knowledge incorporated support vector machines to detect faults in Tennessee Eastman process," *Computers and Chemical Engineering*, vol. 29, no. 10, pp. 2128–2133, 2005.

[27] S. R. Gunn, *Support Vector Machines for Classification and Regression*, Faculty of Engineering, Science and Mathematics School of Electronics and Computer Science, 1998.

[28] A. Widodo and B. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 21, no. 6, pp. 2560–2574, 2007.

[29] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[30] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, NY, USA, 1998.

[31] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, 2000.

[32] B. S. Dayal and J. F. Macgregor, "Improved PLS algorithms," *Journal of Chemometrics*, vol. 11, no. 1, pp. 73–85, 1997.

[33] P. R. Lyman and C. Georgakis, "Plant-wide control of the Tennessee Eastman problem," *Computers and Chemical Engineering*, vol. 19, no. 3, pp. 321–331, 1995.

[34] J. J. Downs and E. F. Vogel, "A plant-wide industrial process control problem," *Computers and Chemical Engineering*, vol. 17, no. 3, pp. 245–255, 1993.

[35] S. Yin, S. X. Ding, A. Haghani, H. Hao, and P. Zhang, "A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process," *Journal of Process Control*, vol. 22, pp. 1567–1581, 2012.