Industrial & Engineering Chemistry Research

Process Systems Engineering

Improved Vine Copula-Based Dependence Description for Multivariate Process Monitoring based on Ensemble Learning

Yang Zhou, Shaojun Li, and Ning Xiong

Ind. Eng. Chem. Res., **Just Accepted Manuscript** • DOI: 10.1021/acs.iecr.8b04081 • Publication Date (Web): 12 Feb 2019

ACS Publications

is published by the American Chemical Society. 1155 Sixteenth Street N.W., Washington, DC 20036

Improved Vine Copula-Based Dependence Description for Multivariate Process Monitoring based on Ensemble Learning

Yang Zhou[1], Shaojun Li[1*], Ning Xiong[2]

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

[2]School of Innovation, Design and Engineering, Mälardalen University, SE-72123 Västeras, Sweden

Abstract: This paper proposes a boosting Vine Copula-Based Dependence Description (BVCDD) method for multivariate and multimode process monitoring. The BVCDD aims to improve the standard Vine Copula-Based Dependence Description (VCDD) method by establishing an ensemble of sub-models from sample directions based on a boosting strategy. The generalized Bayesian inference-based probability (GBIP) index is introduced to assess the degrees of a VCDD model (sub-model) to depict different samples, which means how likely an observation is under the probabilistic model for the system. Every sample is weighted individually according to the depiction degree. The weights are then used to choose a certain number of samples for each succeeding sub-model. In this way, the samples with large error in the preceding model can be selected for training the next sub-model. Moreover, the number of sub-models as well as the number of training samples chosen for every sub-model are determined adaptively in the ensemble learning process. The proposed BVCDD method can not only solve weak sample problems but also remove redundant information in samples. To examine the performance, empirical evaluations have been conducted in comparing BVCDD with some other state-of-the-art methods in a numerical example, the Tennessee Eastman (TE) process, and an acetic acid dehydration process. The results show that the developed BVCDD models are superior to those obtained by the counterparts on weak samples in both

[*] Corresponding author. Tel.: 86-021-64253820
E-mail addresses: lishaojun@ecust.edu.cn (S.J. Li)

accuracy and stability.

1. Introduction

Modern industry is being advanced towards complex, automated, and even intellectualized levels, enabling large-scale manufacturing systems with high production efficiency[1-3]. However, large-scale systems are also subject to high complexity and uncertainty, which may bring hidden risks to the process causing severe faults and catastrophic accidents[4-6]. It is imperative to develop more effective methods and technologies for safety assurance in process industry to eliminate the occurrence of disasters[7].

Process monitoring, as one of the powerful means for safety and quality assurance, has received much attention and research effort in the last few decades. Particularly for process industry, it is important to ensure the safe and stable operation, reduce energy consumption and pollution, as well as ensure that the product quality meets the demand of customers. According to Venkatsubramanian et al.[8-10], current methods of process monitoring can be divided into three categories: model-based, knowledge driven and data driven. Data driven methods have become most popular in recent years, which include multivariate statistical analysis methods[11,12], machine learning methods[13-15], and signal processing methods[16].

Two traditional methods for multivariate statistical analysis are principal component analysis (PCA)[17] and partial least squares (PLS)[12]. But they are restricted to linear and Gaussian systems. To handle nonlinear systems, Kramer[18] proposed a nonlinear PCA method based on a five-layer neural network, whose input layer extracts nonlinear principal components from the nonlinear function. Schölkopf[19] proposed kernel PCA (KPCA) method to transform the original input space to a higher dimensional feature space in process monitoring. Moreover independent component analysis (ICA)[20] was proposed to solve non-Gaussian problems. It has also been proved that the non-Gaussian property of variables can reflect the independence of variables[21], and consequently the non-Gaussian property plays an important role in the estimation of ICA models[22].

Although the aforementioned multivariate statistical analysis techniques are efficient in getting information based on their own features, they may lose some information when extracting features for reducing the dimensionality. To address this problem, Ren[23] et al. used the vine copula-based dependence description (VCDD) model to reduce the information loss in analyzing nonlinear and non-Gaussian systems. However, the VCDD method makes assumption that the marginal distribution function of the data obeys the Gaussian distribution. Some methods based on the VCDD method are proposed to improve the performance of the VCDD method. Yu[24] et al. applied rolling pin into multivariate copula models to deal with the non-monotonicity issue and used the copula function item in the joint distribution function to diagnosis which variable resulted in the fault. But to simplify the model, Yu only used the Gaussian copula depicting the sample, which resulted in bad precision. To solve the problem of missing partial samples of a variable, Zhou[25] et.al. used active learning strategy and the generalized Bayesian inference-based probability (GBIP) index based on VCDD method to choose samples that can provide the most significant information for the process monitoring model. To solve the Gaussian assumption and fault diagnosis problems, more recently a two-subspace method[26] was proposed, in which the copula function was divided into the marginal distribution subspace and dependence structure subspace for achieving robust performance on nonlinear and non-Gaussian systems. This method used the kernel density estimation method (KDEM) to depict the edge distribution function, which do not make Gaussian assumption. However, this will be time-consuming and reduce the robustness of the VCDD model. To address the Gaussian assumption of the marginal distribution function and weak model problem[27], this manuscript combines a boosting strategy and the VCDD method, which can also improve the robustness of the VCDD model.

Indeed, the fundamental idea for process monitoring is to construct a distribution model to represent all possible normal operating condition (NOC) data[26, 28]. A fault is then detected if measurements are deviated from the model of normal behavior. Nevertheless, there exist two hidden obstacles in the way of practicing this. The first is data integrity which is difficult to guarantee given the high cost of collecting ample samples with various fluctuations. The second issue is how to detect small-scale faults[29] causing merely minor deviation (from normal data). It concerns appropriateness of the decision boundary as well as precision of the distribution model. In many cases, a single model is not sufficient to achieve high accuracy, particularly when the faults to be detected are of different types.

Ensemble learning (EL)[30], built upon the probably approximately correct learning model[31, 32], attempts to construct multiple sub-models to increase the accuracy of the total system. Different sub-models are built by applying different sampling schemes on the same sample space, and the results from the individual sub-models are usually integrated by voting to obtain the final result. Boosting[33-35] presents a group of commonly used techniques in EL, where samples are weighted to determine their probabilities of being selected for each round of training. In recent years, some researchers combined EL with soft sensing giving rise to the new methods such as PLS[36, 37], KPCA[38], and generalized global-local structure analysis model[39] for improved non-linear and non-Gaussian process monitoring.

This paper proposes to combine the boosting strategy and VCDD methods for process monitoring. The samples are weighted according to the GBIP index, which assesses the fitting degrees of different samples with respect to the existing model. Essentially the core issue of boosting is to decide which samples to select in the next sub-model for performance improvement. Our basic idea is to exploit probabilistic models, the VCDD method and GBIP index, to tackle this decision problem with uncertainty. The training samples are selected based on a probabilistic view to optimize the performance of the system. The weight of each sample is updated for the next sub-model based on the depiction degree of the sample in the current VCDD model. In this way, the samples with large errors in the current model will be largely reused next time for training the next sub-model. Integration of multiple sub-models can solve the weak sample[40, 41] problem, making the ensemble model diversified, stable and competent to identify different kinds of faults.

The rest of the paper is organized as follows. The preliminaries are introduced in

Section 2, including ensemble learning, copula and vine copula as well as an overview of the VCDD methods. Section 3 elaborates the proposed method of boosting VCDD. Sections 4 and 5 present the results of evaluation in a numerical example, the TE chemical process, and an acetic acid dehydration process respectively. Finally, concluding remarks are given in Section 6.

2. Preliminaries

In this section, we will briefly introduce boosting method, copula and vine copula theory as well as the VCDD method for multivariate multimode process monitoring.

2.1 Boosting Method

Usually EL method is used for classification in process monitoring in four ways: (1) dealing with training data: (2) dealing with input features (3) dealing with labels of samples (4) dealing with learning algorithms. The boosting method, which focuses on dealing with training data[42], seems to be the most representative and promising one among the alternatives.

The task of boosting method in classification is as follows: Given a training set $\left\{(\mathrm{x}_1,\mathrm{y}_1),(\mathrm{x}_2,\mathrm{y}_2),\ldots,(\mathrm{x}_m,\mathrm{y}_m)\right\}$ with $\mathrm{x}_i \in \mathrm{R}^n, \mathrm{y}_i \in \{-1,1\}$, constructing a set of classification models through an iterative learning process. In the first-round training, the weight of every sample in the training set is the same. Afterwards, the weights of the training samples are updated according to the training effect, i.e., the samples with poor training results are given big weights such that they gain more importance in the next round of training. It follows that, after *T* iterations, a sequence of estimations (sub-models) $\mathrm{h}_{1,}$ h_2, … , h_T are obtained, with each estimation having its own weight. The final estimation *H* is decided by a weighted voting scheme. The steps of the boosting algorithm are listed below.

Step 1. Give the number of sub-models *T*, initialize iteration t=1, the weak learner *h* and weights of the samples according to Eq (1)

$$\mathrm{D}_1(\mathrm{i}) = \frac{1}{\mathrm{m}},\ \mathrm{i}=1,\ldots,\mathrm{m}\ . \tag{1}$$

Step 2. Construct the estimation $\mathrm{h}_t : \mathrm{x} \to \{\text{-}1,+1\}$ on the training set with the current weight distribution.

Step 3. Calculate the training error of h_t

$$\varepsilon_t = \Pr_{i\sim D_t}[\mathrm{h}_t(\mathrm{x}_i) \neq \mathrm{y}_i]. \tag{2}$$

If $\varepsilon_t=0$ or $\varepsilon_t \geq \frac{1}{2}$, terminate the iteration and go to Step 8.

Step 4. Define α_t by Eq.(3)

$$\alpha_t = \frac{1}{2}\ln(\frac{1-\varepsilon_t}{\varepsilon_t}). \tag{3}$$

Step 5. Update the weights of samples

$$\mathrm{D}_{t+1}(\mathrm{i}) = (\mathrm{D}_t(\mathrm{i})\ \exp[\text{-}\alpha_t \mathrm{y}_i \mathrm{h}_t(\mathrm{x}_i)]) / Z_t, \tag{4}$$

where the expression of Z_t is Eq. 5

$$\mathrm{Z}_t = \sum_i \mathrm{D}_t(\mathrm{i})\ \exp[\text{-}\alpha_t \mathrm{y}_i \mathrm{h}_t(\mathrm{x}_i)]. \tag{5}$$

Step 6. Update the iteration t=t+1.

Step 7. If t=T+1, go to Step 8; otherwise go to Step 2.

Step 8. Make the final estimation function

$$\mathrm{H}(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(\mathbf{x})\right). \tag{6}$$

Compared with boosting classification problems[43-45], regression and process monitoring represent a class of more complex problems based on a continuous space. The training errors on samples and the weights of samples determined upon these errors have a significant impact on the selection of the next iteration's training samples[46-47]. As mentioned in Section1, it is obvious that the essential purpose of the boosting method is to reach the optimal solution to the decision problem of which samples to select for the next sub-model learning. A probabilistic model would offer an effective means to handle this problem. It's very appropriate to combine the boosting strategy with related probabilistic models. This paper introduces the VCDD method and GBIP index, both of which are built upon probabilistic computing. The training samples are selected in the probabilistic view in this paper to optimize the performance of the ensemble system.

2.2 Copula and Vine Copula Theory

Sklar[48] first proposed the Copula theory in 1959, which is a statistical theory to describe the correlation between variables. According to the Sklar theorem, Copula is a multivariate distribution function consisting of marginal distribution functions obeying the uniform distribution *U*[0, 1], and the joint distribution function is replaced by the marginal distribution function of each variable and their correlation structures (copula function). For *d* dimensional random variables, the joint distribution function can be expressed as follows

$$F(x_1,\ldots,x_d)=C(u_1,u_2,\ldots,u_d), \tag{7}$$

where u_i is the i$^{\text{th}}$ variable of the edge cumulative distribution function.

$$u_i = F_i(x_i) = \int_0^{x_i} f_i(t)\,dt, u_i \in [0,1] \tag{8}$$

where $f_i(x_i)$ is the edge cumulative density function of the i$^{\text{th}}$ variable . C represents multiple copula. If the edge cumulative distribution function of each variable is continuous, then the copula function is unique[49]. Further, if c is differentiable, the joint probability density function (PDF) $f(x_1, x_2, \dots, x_d)$ corresponding to each random variable can be written as follows

$$f(x_1, \dots, x_d) = c(u_1, u_2, \dots, u_d) \prod_{i=1}^{d} f_i(x_i), \tag{9}$$

where the density function c is defined by Eq. 10

$$c(u_1, u_2, \dots, u_d) = \frac{\partial^d C(u_1, \dots, u_d)}{\partial u_1, \dots, \partial u_d}. \tag{10}$$

Once the structure[50, 51] and parameters[52-54] of copula are determined, the joint density distribution of multidimensional data can be obtained according to Eq.(9). However, with the increase of variables, fitting the sample data according to Eq.(10) will become more difficult and the computational complexity of the traditional multiple copula will also increase.

As the most common and simplest approach in the copula family, bivariate copula is used to depict the correlation structure of bivariate random variables, and its optimization is easy to realize. The vine copula model can decompose a multivariate joint distribution into a series of bivariate copula functions and the marginal distributions of individual variables. In this way, the optimization problem of high-dimensional variables can be transformed into a series of bivariate copula estimation problems. For a d-dimensional random vector $\boldsymbol{x} = (x_1, \dots, x_d)^{\mathrm{T}}$, its PDF can be decomposed into the following form of conditional distribution[55, 56]

$$f(x_1,\ldots,x_d) = f(x_d)\prod_{t=1}^{d-1} f(x_t \mid x_{t+1},\ldots,x_d). \tag{11}$$

It can be further decomposed as follow

$$f(x_i \mid \boldsymbol{v}) = c_{i,j|\boldsymbol{v}_{-j}}\left[F(x_i \mid \boldsymbol{v}_{-j}), F(x_j \mid \boldsymbol{v}_{-j})\right] f(x_i \mid \boldsymbol{v}_{-j}), \tag{12}$$

where, $\boldsymbol{v} = x_{-i}$ represents the vector that excludes x_i from $\boldsymbol{x}$, v_j is the j$^{\text{th}}$ element in vector $\boldsymbol{v}$, $v_{\text{-}j}$ denotes the vector reproduced from $\boldsymbol{v}$ but excluding the element v_j, and $c_{i,j|\boldsymbol{v}_{-j}}$ is the corresponding conditional bivariate copula density function.

The common bivariate copula includes Elliptical family (including Gaussian bivariate copula, Student-t bivariate copula, etc.) and Archimedean family (including Clayton bivariate copula, Gumbel bivariate copula, Frank bivariate copula, etc.)[57]. The D-vine copula is a convenient and typical one among the vine copulas, and the joint density function of the D-vine copula structure is expressed as Eq. (13)[55]

$$f(\boldsymbol{x}) = \prod_{t=1}^{d} f_t(x_t) \times \prod_{i=1}^{d-1}\prod_{j=1}^{d-i} c_{j,j+i|j+1:j+i-1}\left[F(x_j \mid x_{j+1},\ldots,x_{j+i-1})F(x_{j+i} \mid x_{j+1},\ldots,x_{j+i-1});\theta_{j,j+i|j+1:j+i-1}\right], \tag{13}$$

where θ represents the copula parameters. Compared with Eq.(9), the d dimensional copula density function $c(u_1,u_2,\ldots,u_d)$ is transformed into the product of $d(d-1)/2$ bivariate copulas $c_{j,j+i|j+1:j+i-1}$.

The bivariate copula in Eq. (13) contains the conditional distribution functions. How to analyze these conditional distribution functions is directly related to the ability to implement the model. Aas et al.[51] put forward a function *h* to build the above conditional distribution functions

$$h_{x_i,x_j|\tilde{\boldsymbol{x}}}\left[F(x_i \mid \tilde{\boldsymbol{x}}) \mid F(x_j \mid \tilde{\boldsymbol{x}});\theta_{x_i,x_j|\tilde{\boldsymbol{x}}}\right] \triangleq F(x_i \mid x_j, \tilde{\boldsymbol{x}}) = \frac{\partial C_{x_i,x_j|\tilde{\boldsymbol{x}}}\left[F(x_i \mid \tilde{\boldsymbol{x}}), F(x_j \mid \tilde{\boldsymbol{x}});\theta_{x_i,x_j|\tilde{\boldsymbol{x}}}\right]}{\partial F(x_j \mid \tilde{\boldsymbol{x}})}. \tag{14}$$

A series of conditional distribution functions in D-vine copula are obtained by iterative computation based on Eq.(14). The detailed description and deduction can be found in the references[50-54].

2.3 The Vine Copula-Based Dependence Description Model

The VCDD method, proposed by Ren et.al.[23], is a multi-mode fault detection method based on vine copula correlation description. The fault detection process based on the D-vine copula model is divided into two stages: offline modeling and online monitoring. To measure the distance of the process data from each non-Gaussian mode, a generalized local probability (GLP) index is defined. To calculate the marginal densities of the joint density *f (k)* in GLP index, Ren et.al. defined *l* as the step size for discretization, and the density quantile table. Consequently, the GBIP index under a given control limit, which is set as 0.95 in most situation, can be further calculated in real time via searching the density quantile table created offline. The definition of GBIP index is as follow: The GBIP index, built upon the Bayesian inference-based probability (BIP)[58], was designed for nonlinear and non-Gaussian systems. Assume that there are K modes $C_k(k=1,2,\ldots,K)$ in a system and $f^{(k)}(\boldsymbol{x})$ represents the joint PDF of the kth mode, where $\boldsymbol{x}\in\Re^m$ stands for the variable vector. And $P(C_k)$ is the prior probability of the monitored data belonging to $f^{(k)}(\boldsymbol{x}_t)$. The GBIP index is given in Eq.(15) and Eq.(16) with the purpose to measure the distance between a training sample and the present model

$$GBIP(x_t)=\sum_{k=1}^{K}P(C_k\,|\,x_t)P_L^{(k)}(x_t) \tag{15}$$

$$P(C_k|x_t) = \frac{P(C_k)f^{(k)}(x_t)}{\sum_{i=1}^{K} P(C_i)f^{(i)}(x_t)}, \tag{16}$$

where $P_L^{(k)}(x_t)$ stands for the GLP between x_t and mode C_k , $P(C_k|x_t)$ represents the posterior probability of the monitored data x_t belonging to $f^{(k)}(\boldsymbol{x}_t)$, and $P(x_t|C_k)$ represents a posteriori conditional probability that the test data x_t belongs to the mode C_k. $P(C_k)$ is the prior probability of the monitored data belonging to $f^{(k)}(\boldsymbol{x})$. The informative prior or the prior information extracted from the previous monitored data in a time window is suggested. More practically, if n_k (k = 1,2,...,K) training data are clustered into each of the K modes, respectively, then we can simply set as $P(C_k)=\frac{n_k}{\sum_{i=1}^{K} n_i}$ $(k=1,2,...,K)$. For non-Gaussian systems, the highest density region (HDR)[59, 60] is introduced into the GLP index instead of the Mahalanobis distance based local probability index. In this way, the GLP index is defined as follows

$$P_L^{(k)}(x_t) = \Pr\left(f^{(k)}(\mathbf{X}) \geq f^{(k)}(x_t)\right), \tag{17}$$

where $f^{(k)}(\boldsymbol{X})$ represents the one-dimensional random vector $\boldsymbol{x}$ of mode C_k after its PDF mapping. y_i denotes the corresponding joint PDF value, where $y_i = f^{(k)}(\boldsymbol{x}_t)$ and $\mathbf{y}=(y_1,y_2,\ldots,y_n)$, and the lower quantiles of $\mathbf{y}$ under the confidence level α is $q_{\hat{\alpha}}^{\boldsymbol{y}}$, if

$$q_{\hat{\alpha}}^{y} = f^{(k)}(x_t). \tag{18}$$

Then

$$\hat{\alpha}=\lim_{n\to\infty} 1-P_L^{(k)}(\boldsymbol{x}_t), \tag{19}$$

where $\hat{\alpha}$ represents the estimation of the confidence level α corresponding to the PDF values $\mathbf{y}$. Eq.(19) indicates that the constructed GLP index is consistent with a certain confidence level of the sample's density quantile. The details of the complete method for process monitoring were given in[23]. The simple steps of offline modeling and online monitoring are summarized as follows.

Offline modeling

(1) Collect a set of historical training data under all possible modes. The information on data belonging to the real operating modes can be obtained via expert knowledge or performing clustering methods.

(2) Establish D-vine model for each mode, which includes (a) determining variable orders; (b) constructing D-vine models; (c) calculating conditional distribution functions and optimizing pair copulas layer by layer; (d) performing goodness-of-fit testing.

(3) Obtain the samples of the D-vine model for each mode using MCMC method. This step can be eliminated when the historical training data are large enough to cover the most information of each operating mode.

(4) Specify a control limit *CL*, choose the step size for discretization according to $l = \frac{1}{1-CL}$, then estimate the discrete density quantiles with different confidence levels and create the corresponding static density quantile table.

Online monitoring

(1) Calculate the posterior probability $P(C_k \mid \boldsymbol{x}_{\mathrm{t}}^{\mathrm{monitor}})$ that the monitored data $\boldsymbol{x}_{\mathrm{t}}^{\mathrm{monitor}}$ belong to each mode $C_k (k = 1, 2, \ldots, K)$ (the precise definition of $P(C_k \mid \boldsymbol{x}_{\mathrm{t}}^{\mathrm{monitor}})$ will be given in Section 3). The non-informative prior or the prior information extracted from the previous monitored data in a time window is suggested.

(2) Compute the joint PDF values of $\boldsymbol{x}_t^{\mathrm{monitor}}$ under all modes (D-vine models) and estimate the GLP index of each mode through a searching strategy.

(3) Estimate the GBIP index. According to the given control limit *CL* in the GBIP control chart, detect the abnormal operating conditions for $\boldsymbol{x}_t^{\text{monitor}}$ satisfying GBIP>*CL*.

3. The Boosting Vine Copula-Based Dependence Description (BVCDD) Method

The EL calls a simple learning algorithm repeatedly and obtains different basic learners by changing the learning samples. By combining the basic learners we acquire the ensemble model with synergistic effect. The ensemble model can achieve a higher accuracy than what is obtainable by single basic learners. Usually, the computational complexity of the ensemble model is much higher than that of basic learning models. However, if the improvement of performance is significant, the ensemble learner can not only reduce the error of the model but also improve the generalization ability of the learning algorithm.

The boosting algorithm, as the most common EL technique, has achieved a great success in both theoretical analysis and practical applications. For a boosting algorithm, there are two alternative ways to select training samples[58]. One is to use the weighted samples directly. This method is useful only when the basic learning mechanism can directly use the samples with weights. The other way is to treat the weight of a sample as the probability of selection and thereby samples are randomly selected based on their probabilities into the basic learning modules. This approach is effective for those basic learning algorithms that can acquire an additive effect when learning with multiple copies of the same sample.

The method proposed in this paper is based on the boosting algorithm. The key idea of the boosting method is to update the weights of the samples according to the depiction degrees of the learner model on individual samples. It is very appropriate to select the samples from the view of probabilistic decision making. The VCDD method and GBIP index are therefore introduced in this paper based on probabilistic computing.

The GBIP index based on HDR enables quantitative estimation of any distributions and determines the depiction degrees between the current model and different samples. The proposed method uses the GBIP index to measure the depiction degrees between the current model and different samples. The GBIP index is a distance measure, it represents the distance between the sample and the modal center. The sample that has a large GBIP index is far away from the center of the mode, it means that the current VCDD model cannot describe this sample very well. In other word, if a sample stays within the VCDD model, its GBIP index will remain small, otherwise, it will be a large value. Compared to the samples that stay within the VCDD model, those that are found far from the VCDD model have relatively large GBIP index values. Therefore, this paper give the sample that has the largest GBIP value the biggest weight, which can make the sample easier to be selected and retrained in the next model.

Within the procedure of the BVCDD method, there are two parameters affecting the monitoring performance, namely the maximum number (denoted as T) of sub-models and the number of samples (denoted as m) chosen from the training set in building every sub-model. This paper puts forward a new way to set the two parameters adaptively.

There is a specific relationship between the performance of the BVCDD model and the number of training samples chosen for training every sub-model. The initial samples will affect the performance and give randomness to the final model, so this paper selects the initial m samples with the Kennard-Stone (K-S)[61] method, which can select the maximum range of the training samples. Initially increasing the training with weak samples will help the model get a better performance. However, at later stages overtraining will gradually happen with some strong samples, which will degrade the accuracy of the learning results, because repetitive trainings with strong samples merely change the model to fit the coincidence rather than improve the accuracy of the ensemble system on the entire application domain. With the increase of training samples in sub-models, the computational complexity will become higher with ensemble learning, which is not desired. This paper defines an index θ to determine the number of training samples chosen for training each sub-model:

$$\theta_k = \mathrm{FAR}_{k-1} - \mathrm{FAR}_k, \tag{20}$$

where FAR_k denotes the false alarm rate (FAR) of the VCDD model with first k$^{\mathrm{th}}$ training samples on all the training data. The number of training samples is successively increased until reaching a negative value of the θ index, at which we terminate the procedure of picking samples from the training set.

The precision of the BVCDD model is somehow determined by the number of sub-models. The more sub-models, the better the ensemble model precision and the higher the computational complexity. Generally, the generalization accuracy of the model depends on the complexity of the combination function, and a simpler combination function can lead to stronger generalization ability. The computational complexity of the combination function depends on the basic learning model. The boosting method can balance the sensitivity of the basic learner and the computational complexity as measured by the number of sub-models. If increasing the number of sub-models contributes to model accuracy improvement which is more important than the cost of prolonged computational time, the combination algorithm will balance the training and generalization accuracy. This paper uses a parameter ρ_t to determine the number of sub-models as given in Eq (21).

$$\rho_t\text{=FAR}_t \tag{21}$$

where, t is the number of sub-models. FAR_t is the FAR value of the BVCDD model by using all the training samples when the number of sub-modes is t. Supposed the confidence level is set at *CL*(%), the minimum value of t is set as the number of sub-models when ρ_t is less than (1-*CL*). The reason why we set (1-*CL*) as the control boundary is that normally it can be accepted, if (1-*CL*) of the training samples are out of the control boundary. So when the FARt is less than (1-CL), it means the model is reliable.

There are two parts in the procedure of the boosting vine copula-based dependence description (BVCDD) method. The detailed procedures of offline modeling and the online monitoring are presented as follows.

Offline Modeling:

Step 1. Determine the number of training samples *m* chosen from the training set for every sub-model through the index θ in Eq.20. In order to show the variation of θ with the two parameters clearly, the number of sub-models is fixed. While the number of training samples is being increased, stop selecting more samples once the index θ gets negative. The number of training samples *m* chosen from the training set is thereby determined for each sub-model. And select the initial m samples with K-S method.

Step 2. Set the index of the current sub-model as t = 1 and initialize the training sample weights as:

$$w_i^{(1)} = \frac{1}{n},\ i = 1, 2, \ldots, n. \tag{22}$$

Step 3. Calculate the selection probability of each sample in the training set according to Eq.(23) and if this is the first loop to choose the initial samples, go to step 4; else chose *m* training samples from training set based on these selection probabilities

$$p_i^{(t)} = \frac{w_i^{(t)}}{\sum_{j=1}^{n} w_j^{(t)}} \quad \text{i=1,2,…,n.} \tag{23}$$

Step 4. Construct the VCDD model with the *m* selected training samples of loop t.

Step 5. Calculate the GBIP index for every sample with respect to the t^th VCDD model.

Step 6. Calculate the weighted GBIP index of sub-model t:

$$\bar{G}_t = \sum_{i=1}^{n} G_i^{(t)} p_i^{(t)}, \tag{24}$$

where the $G_i^{(t)}$ means the GBIP index of i^th sample is calculated with the t^th VCDD model.

Step 7. Calculate the sub-model weight:

$$\beta_t = \frac{\bar{G}_t}{1 - \bar{G}_t}. \tag{25}$$

Step 8. Update the weights of training samples:

$$w_i^{(t+1)} = w_i^{(t)} \beta_t^{[1 - G_i^{(t)}]}. \tag{26}$$

Step 9. Calculate the index ρ_t. If ρ_t<(1-*CL*), set the number of sub-models as T = t and go to Step 10; otherwise set t= t+1 and go to Step 3.

Step 10. Integrate all the T sub-models into the BVCDD model:

$$\mathrm{BGBIP}_{\mathrm{total}} = \sum_{t=1}^{T} \frac{1}{\beta_t} \times \mathrm{GBIP}_t. \tag{27}$$

where the $\mathrm{BGBIP}_{\mathrm{total}}$ index means the final GBIP index of boosting all the t VCDD models, and the GBIP_t index means the GBIP index of the t^{th} VCDD model.

Online Monitoring:

Step 1. Calculate the posterior probability of the monitored samples and GBIP_t index for t = 1, 2, …, T.

Step 2. Calculate the ensemble $\mathrm{BGBIP}_{\mathrm{total}}$ index.

Step 3. Judge whether the $\mathrm{BGBIP}_{\mathrm{total}}$ index is inside the control limit to fulfill the requirement in real-time monitoring.

The flowchart, as shown in Figure 1, depicts the overall monitoring process.

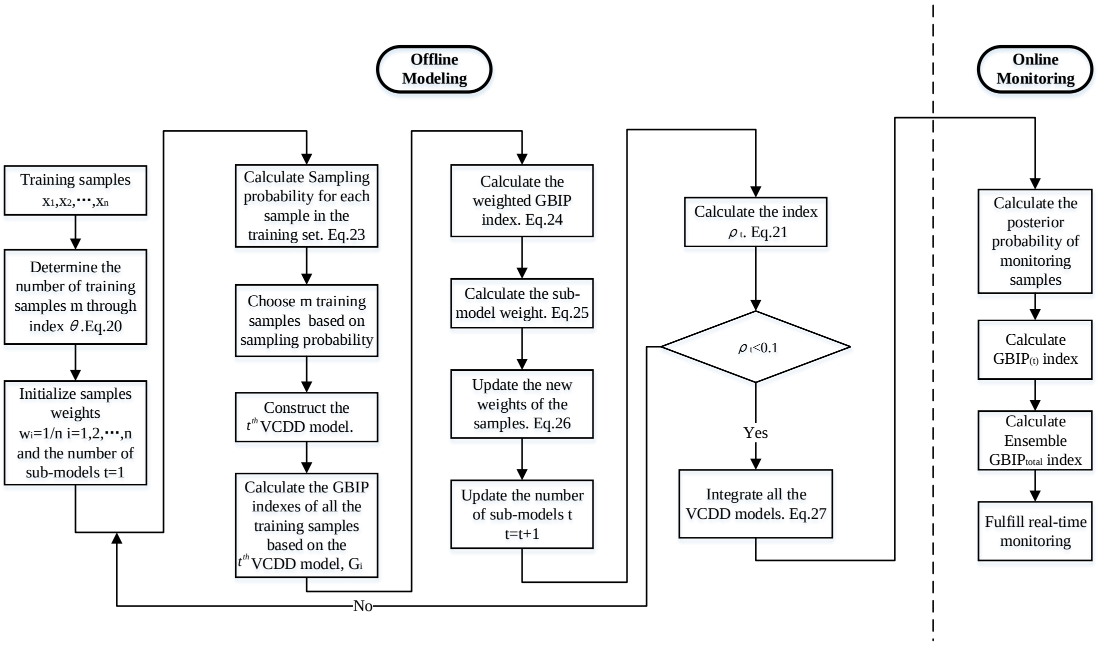

Offline Modeling
Online Monitoring
Training samples x1,x2,⋯,xn
Determine the number of training samples m through index θ.Eq.20
Initialize samples weights wi=1/n i=1,2,⋯,n and the number of sub-models t=1
Calculate Sampling probability for each sample in the training set. Eq.23
Choose m training samples based on sampling probability
Construct the t^th VCDD model.
Calculate the GBIP indexes of all the training samples based on the t^th VCDD model, Gi
Calculate the weighted GBIP index. Eq.24
Calculate the sub-model weight. Eq.25
Update the new weights of the samples. Eq.26
Update the number of sub-models t t=t+1
Calculate the index ρt. Eq.21
ρt<0.1
Yes
No
Integrate all the VCDD models. Eq.27
Calculate the posterior probability of monitoring samples
Calculate GBIP(t) index
Calculate Ensemble GBIPtotal index
Fulfill real-time monitoring

Figure 1. Flowchart of the BVCDD process monitoring method.

By integrating multiple VCDD models, the BVCDD method can depict the non-Gaussian marginal distribution data and describe data more accurately by integrating multiple VCDD model. To be more specific, Figure 2, which depicts the boundary of BVCDD (solid line) and the boundary of VCDD (dashed line), is used to show the advantage of the BVCDD method. Evidently, the VCDD model fails to capture the structure in region 1, partly due to the Gaussian assumption of the edge distribution function. Once the testing data appear in region 1, they will be mistakenly considered as faults. In addition, due to accuracy limitation of modeling, the given confidence level contour hardly keeps in align with the exact boundary. While the BVCDD model can depict the non-Gaussian marginal distribution data with multiple VCDD model and describe data more accurately especially for fuzzy samples on the boundary. In addition, the integrated model with the boosting strategy can improve the robustness of the single model. The robustness of VCDD method is not very good when the sample data is incomplete, such as uneven data. For example, in Figure 2 if most of the training samples are located in low left quarter, and few training samples are located in top right quarter such as area 2, the VCDD model will fail to capture the structure in area 2. For this kind of data with uneven distribution, it is nearly impossible to describe the edge accurately with VCDD alone, while the BVCDD model can construct a more robust model by integrating multiple VCDD models. When the training samples are not even, the pattern of the training samples and the importance of every training samples will be destroyed, and a single model cannot depict the samples accurately. And boosting strategy can solve this kind of weak model problem well.

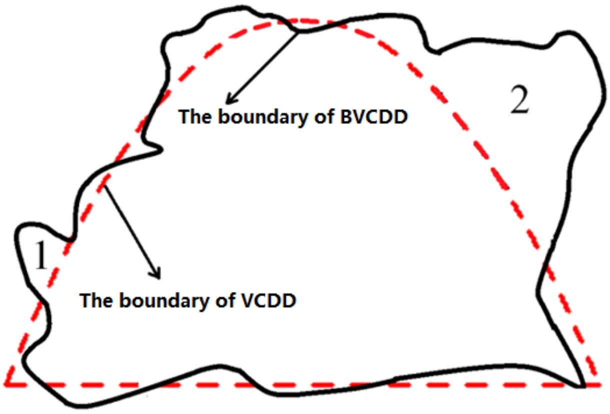


Figure 2. The boundary of BVCDD (solid line) and the boundary of VCDD (dashed line)

4. An Illustrative Example

In this section, a numerical case is used to illustrate the effectiveness of the proposed method, with the samples taken from[26]. Two operating modes are analyzed, each of which has 200 training samples in three dimensions. So there are 400 training samples for training. Two test data sets are used to test different methods. The two test data sets were generated in the following ways respectively: 1) The process ran normally at mode 1 for the first 50 moments, followed by the next 100 moments in which variable X_2 was given an offset of 0.5 and a nonlinear drift term (fault 1) at mode 1, and then the process was restored to the normal condition at mode 2; 2) The process ran normally at mode 1 for the first 50 moments, in time periods 51-100, the system works at mode 2, followed by the next 100 moments in which variable X_1 was given an offset of 4 (fault 2) at mode 2, while variable X_3 was always in the normal state. Figure 3 shows a variable scatter plot of the samples with each other (black dots represent normal training samples, blue dots show test samples under normal operating conditions, and green dots represent faulty test samples).

Figure 3 shows that the training samples are highly non-Gaussian. There is heavy lower tail dependence, which means that the training samples in lower region are more relevant, between variables 1 and 2, and heavy upper tail, which means that the training samples in upper region are more relevant, dependence between variables 1 and 3. Fault 1 adds a small offset and a non-linear dynamic drift, which is not too large at the beginning. Thus, the mean value only changes a little, while the correlation is abnormal obviously. So Fault 1 is a typical dependence variation corresponding to nonlinear drifting error. Fault 2 is that a variable directly adds a fixed value offset. Compared with normal data, the mean value of that variable distribution changes, but the correlation remains unchanged. So the Fault 2 is just the margin deviation and the most common fault.

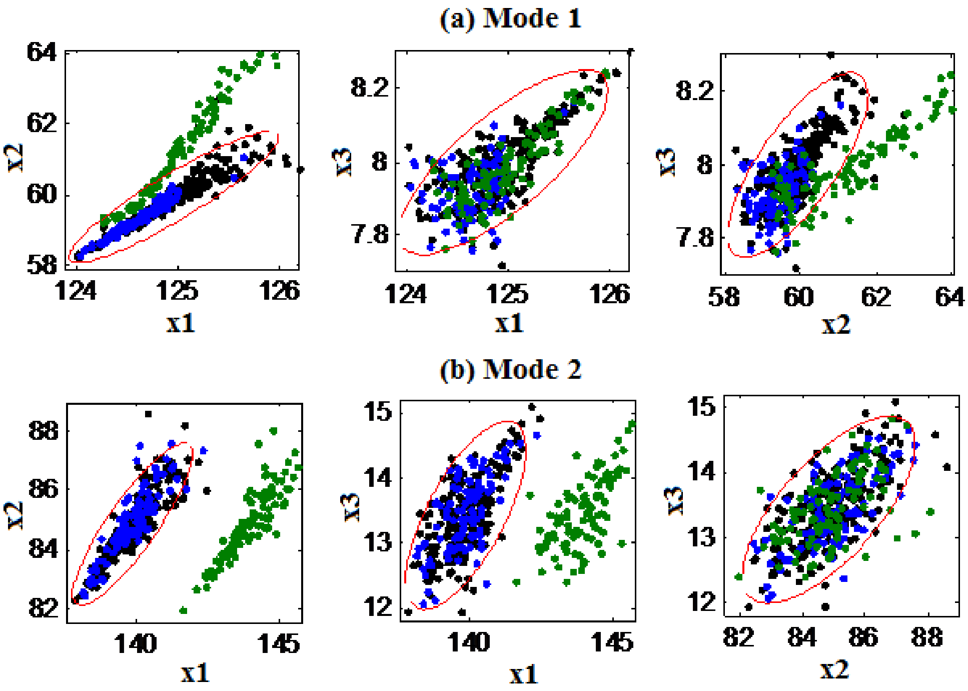


Figure 3. Scatter plots of training data and testing data

To examine the performance of different methods, two monitoring evaluation indexes, fault detection rate (FDR) and FAR, are used here. The FDR index is based on the faulty samples, which is defined as the ratio of the number of faulty samples that can be detected to the number of all faulty samples. The FAR index represents the ratio of the number of normal samples that are classified as faulty to the number of all normal samples. To avoid randomness, a total of 20 simulations were conducted in evaluating each of the methods. The CL is set as 0.99 in the numeral example.

There is a specific relationship between the performance of BVCDD model and the number of training samples chosen from the training set in every sub-model. And this number of training samples was determined given that the ensemble contains 20 sub-models. A lot of experiments have indicated that the number of sub-models does not present a sensitive factor for the decision of the number of training samples for sub-model learning. We experimented with different numbers of training samples (ranging from 50 to 400) chosen from the training set for ensemble learning. The resultant performance of the BVCDD model under certain variations is given in Table 1.

Table 1. The FDR and the FAR values of the BVCDD model with different numbers of training samples

The number of training samples	θ	FDR of fault 1	FDR of fault 2	FAR
300	2.0	100	84.6	11.5
310	1.8	100	87.6	9.5
320	1.3	100	91.7	7.5
330	0.8	100	92.5	6.5
340	0.5	100	94.6	4
346	0.4	100	95.9	3.5
347	0.4	100	96.4	3.5
348	0.3	100	96.8	3.5
349	0.3	100	97.2	3.5
350	0.2	100	97.5	3.5
351	0.2	100	97.7	3
352	0.1	100	97.9	3
353	**0.1**	**100**	**98.1**	**3**
354	-0.1	100	97.8	3.5
355	-0.1	100	97.5	3.5
360	-0.2	100	96.5	3.5
370	-0.2	100	96.0	4
380	-0.2	100	94.4	4
390	-0.3	100	93.2	4.5
400	-0.3	100	90.1	5.5

In order to examine the impact of the number of training samples, we tracked the results while the number of training samples increased to 400. As can be seen from Table. 1, while the number of training samples increased, the performance of the

BVCDD model started to get better until reaching an optimum point. Afterwards, as the number of training samples continued to increase, the performance of the BVCDD model began to become worse. This indicates that increasing training with weak samples in the initial stage will help improving the performance of the model. However, over-training, which will lead to the bad generalization ability of the model, will gradually happen with some strong samples, which will deteriorate the accuracy of the learning algorithm. The reason is that over-training on strong samples only focus on confidence rather than the true characteristic of the process. Moreover, training samples for every sub-model have to be constrained due to the computational complexity. As can be seen from Table 1, when the number of training samples were set to 354, the index θ changed the sign. Therefore the number of training samples chosen for every sub-model was decided as 353.

The precision of the BVCDD model is determined by the number of sub-models in some way. To exemplify this relation, Table.2 illustrates the varied performance of the BVCDD model under different numbers of sub-models. In the Table.2 the index ρ is the FAR of BVCDD model with all the training samples, and FAR index is the FAR of BVCDD model with all the testing samples. To examine the possibility of performance improvement with more sub-models, Table 2 gives the resultant FDR and FAR values while the number of sub-models was increased from 2 to 70. The BVCDD model was greatly improved with the increase of accuracy mainly achieved in the first several rounds. The reason is that the training samples were chosen according to the training error of the preceding sub-model such that the model performance was reinforced observably in the early stage. However, after the number of sub-models reached 20, the performance of the BVCDD model gradually converged. In particular, when the number of sub-models reached 44, the index ρ became less than 0.01. It can be seen that when ρt is less than 1%, as the number of sub-model rise, the FDR and FAR indexes hardly get better. Although when the number of sub-models reaches 60, the FDR of fault 2 get better 0.1%, while the number of sub-models increases 16.This means that further increasing sub-models would make little contribution to performance improvement. Therefore, the number of sub-models in this example was chosen as 44.

Table 2. The FDR and FAR indexes of the BVCDD model with different numbers of sub-models (0.99 Confidence Level)

different number of sub-models	ρ	FDR of fault 1	FDR of fault 2	FAR
2	8.25	100	83.5	10.5
5	5.75	100	86.1	7.5
10	3.75	100	89.6	6
15	2.75	100	92.5	5
20	2.25	100	94.8	4
25	1.75	100	96.3	3
30	1.5	100	98.7	2.5
35	1	100	99.0	2
40	1	100	99.4	2
41	1	100	99.5	2
42	1	100	99.6	2
43	1	100	99.6	1.5
44	**0.75**	100	**99.7**	**1.5**
45	0.75	100	99.7	1.5
50	0.75	100	99.7	1.5
60	0.75	100	99.8	1.5
70	0.75	100	99.8	1.5

Finite Gaussian Mixture models (FGMM) [57] and KMPCA [19] are commonly used effective methods for processing non-Gaussian data and nonlinear data. In this paper, the proposed BVCDD method is compared with the FGMM, KMPCA, and VCDD methods to illustrate the effectiveness of the BVCDD method. The FAR index is shown as the rate for Fault 0. Figure 4 and Table 3 show the monitoring chart and results of the KMPCA, FGMM, VCDD and BVCDD methods respectively.

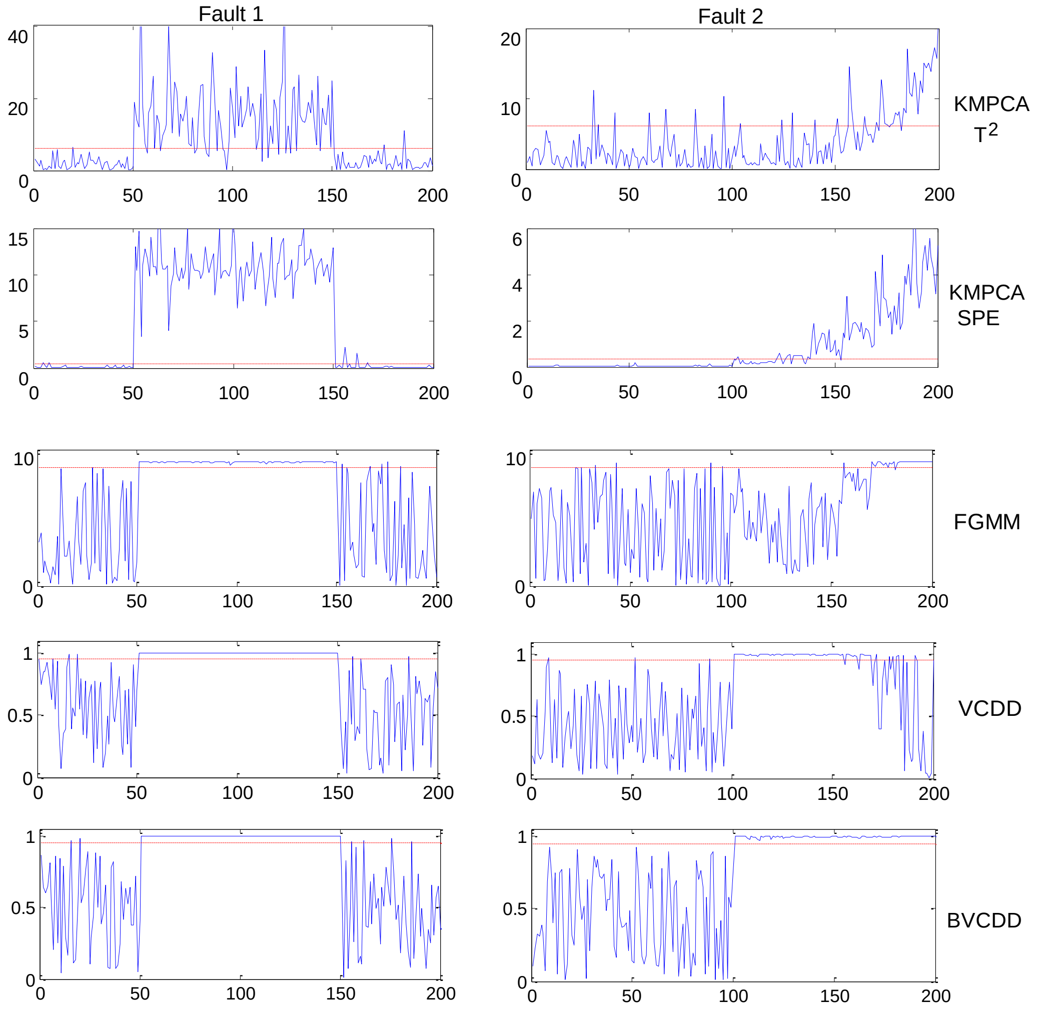


Figure 4. The real-time monitoring chart of the KMPCA, FGMM, VCDD and BVCDD methods

Table 3. Fault detection performance for 2 faults with KMPCA, FGMM, VCDD and BVCDD methods

Fault	FGMM	KPCA		VCDD	BVCDD
		T^2	SPE		
0	6.5	9	6.5	2.5	**1.5**
1	**100**	88.27	**100**	**100**	**100**
2	39.3	30.7	63.9	78.4	**99.7**

Note that the faults in this numeral example only change in one dimension, which can be regarded as a simulation of sensor faults for an open-loop system. For those widely discussed process faults, abnormality of Variable 1 can lead to the deviation of Variable 2 or 3, due to the multivariate dependence behaviors. These process faults can be simulated by adjusting the state variables of a state equation model. Analog sensor faults are well considered here given that the change and mathematical meaning of the proposed indices can be observed clearly. The results show that all the methods can well detect the occurrence of Fault 1 because of the large magnitude of deviation. Neither KMPCA-T^2, KMPCA-SPE nor FGMM can detect the initial anomalies of Fault 2 (e.g., 101~140). Interesting is that both VCDD and BVCDD models have high FDR values. Compared with the KMPCA and FGMM methods, VCDD and BVCDD can build more accurate models. Figure 5 shows the HDR of different dimensions in the numerical example, and also shows the distribution of the samples between different dimensions. So it can be seen from the Figure 5 that the distribution of different dimensions are not the Gaussian distribution. The remarkable detection performance of VCDD method comes from the fact that the D-Vine model can accurately describe the distribution of data with different type of copulas. FGMM method cannot describe this kind of normal samples well. Because FGMM method combines multiple Gaussian component to describe samples, there is a blank area between Gaussian component edge of FGMM and the actual training data edge, for example, there must be a blank area between hyper ellipsoid edge of FGMM method and real red line and blue line. Fault points in these blank areas are often mistaken for normal state data (such as Fault 2). In contrast, the VCDD method can describe the complex correlation structure between different variables and construct a more accurate distribution function of each variable, which is the main reason why VCDD has higher FDR. The KMPCA method is very limited in describing complex processes, due to the Gaussian hypothesis of data. But it is obviously seen in the Figure 5 that the samples do not obey Gaussian hypothesis, which will cause a lot of misjudgments. So the KMPCA and FGMM method cannot set up accurate models for the numeral example.

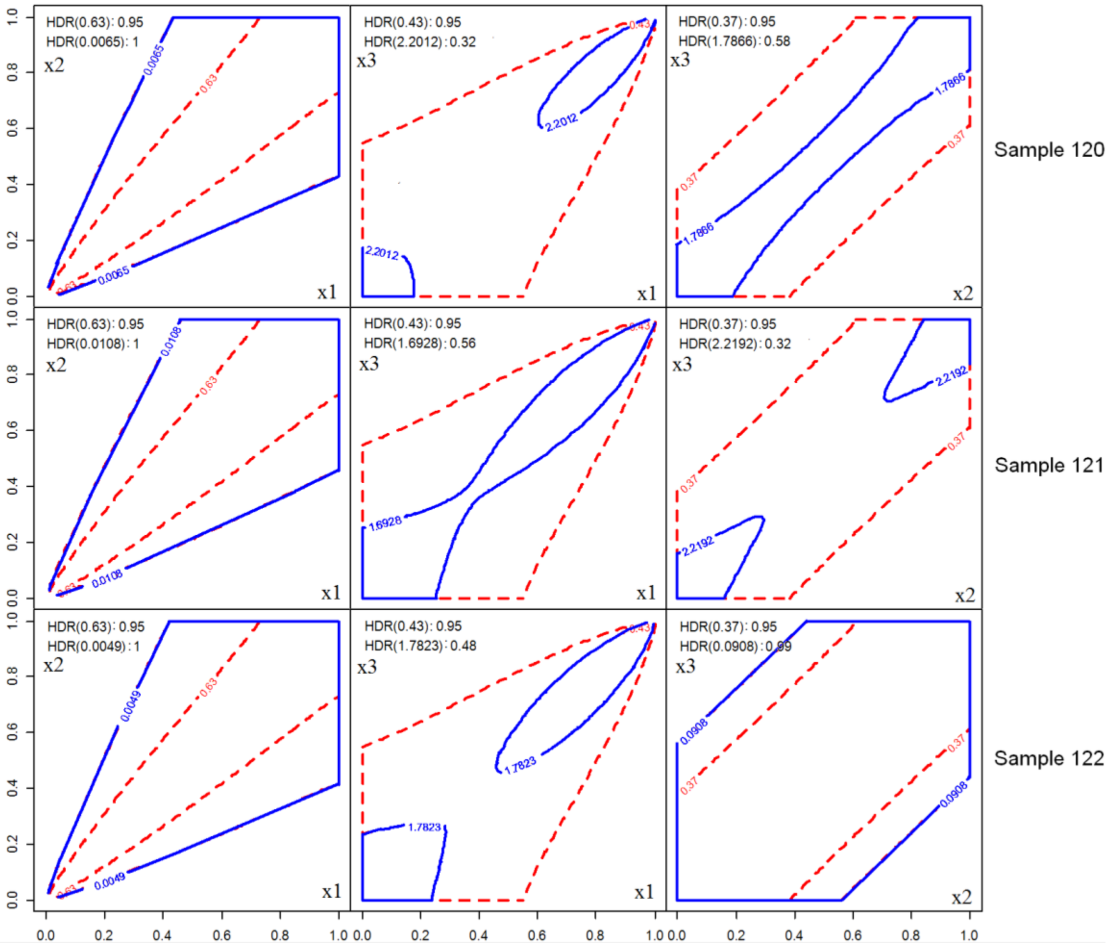


Figure 5. The HDR values of the bivariate variables in the PDF

For systems with different types of faults, a single VCDD model cannot effectively monitor all of them, and the BVCDD method can make up for this deficiency. Because the BVCDD method aims to learn to fit each sample, multiple sub-models can be combined therein to well detect different faults. In the real-time monitoring chart, the

BVCDD model can very well monitor two faults at the same time.

5. Industrial Cases

5.1 TE Benchmark Case Study

The TE process is a simulation system established by Eastman Chemical Company to evaluate process control and monitoring methods. It can well simulate actual complex industrial systems. As a simulation system, it has been widely used in the process monitoring field. It consists of five basic units: reactor, condenser, compressor, stripper, and separation tower. Each unit constructs a series of algebraic equations and differential equations based on material balance, gas-liquid equilibrium and energy balance[62]. Figure 6 is a process flow chart of the TE process.

The TE process consists of 53 process variables and 21 process faults. Among the 53 process variables there are 22 continuous variables, 19 component variables and 12 operating variables. In the 21 faults, faults 1~7 are step changes, faults 8~12 are random faults, fault 13 is the slow drift in the reaction kinetics, faults 14, 15 and 21 are related to valve viscosity, and faults 16~20 are unknown. In this paper, the models were constructed based on 22 continuous process variables, and monitoring was conducted to detect all the 21 faults[63]. There were 500 training samples and 960 testing samples in each test setting, the sampling period was 3 min, and all the faults were introduced at the 161st sampling instant. The simulation samples were obtained from the website http://web.mit.edu/braatzgroup/links.html.

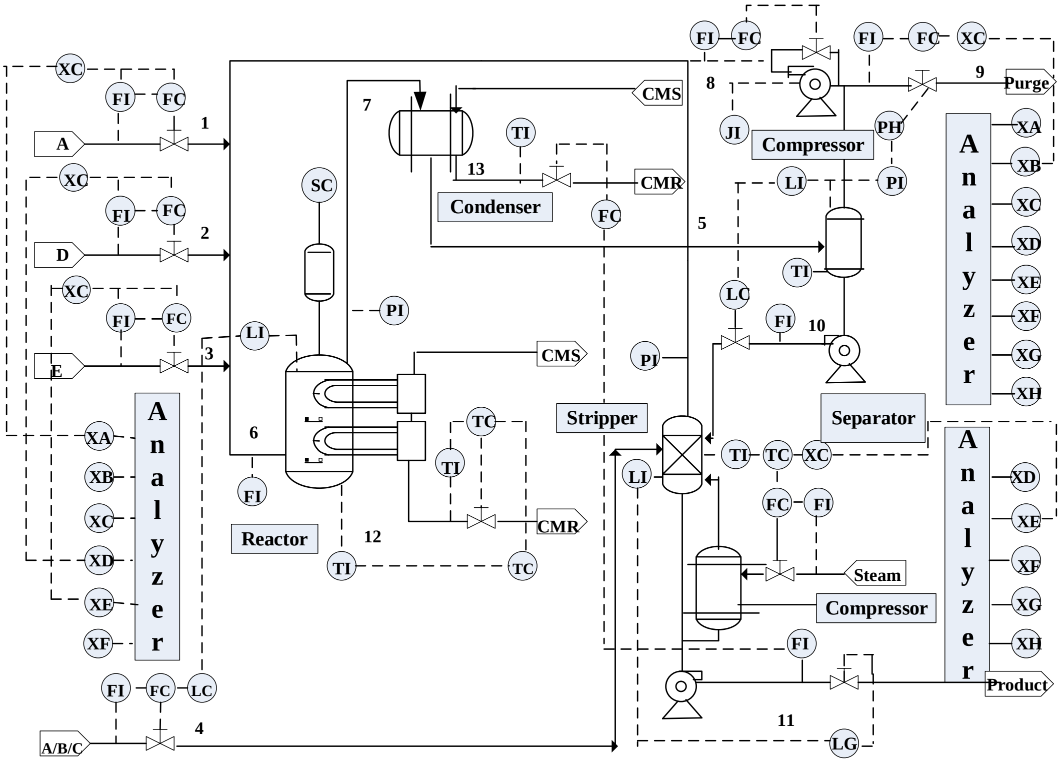


Figure 6. Process flow chart of the TE process

Table 4 shows the monitoring results of the KMPCA, FGMM, VCDD and BVCDD methods. Principal components of KMPCA was selected such that 85% variance was explained[19]. For fair comparison 0.99 was set as the confidence level for all methods. The results in Table 4 show that the performance of BVCDD is better than that of the other three methods. Usually different methods have different ability to extract features. KMPCA mainly extracts nonlinear information, so it can more prominently detect faults 7 and 11. FGMM, mainly extracting non-Gaussian information, is more competent for faults 10 and 20. VCDD is superior to FGMM and KMPCA except for only small parts of faults. Since both VCDD and BVCDD can well capture nonlinear and non-Gaussian information, they achieve better monitoring results than KMPCA and FGMM. Besides, by integrating multiple sub-models, BVCDD appears more powerful to detect different faults than VCDD.

Table 4. Fault detection performance for 21 faults with different approaches (0.99 Confidence Level)

Fault	FGMM	KMPCA		VCDD	CDS-KDEM		BVCDD
		T^2	SPE		MDP	DVP	
0	8.69	9.03	9.66	8.95	7.12	7.22	**6.71**
1	**99.88**	99.75	99.75	99.75	99.63	45.75	**99.88**
2	98.43	98.13	98.25	98.25	98.50	91.00	**99.88**
3	1.76	4.38	5.00	3.25	5.13	3.63	**12.25**
4	2.23	2.00	2.25	2.25	2.50	3.00	**8.00**
5	24.50	27.00	27.00	26.38	25.88	12.50	**32.75**
6	**100**	**100**	**100**	**100**	100	97.75	**100**
7	35.78	42.38	42.63	40.00	43.38	17.25	**50.25**
8	97.62	97.38	97.75	98.50	98.00	38.38	**98.63**
9	3.71	3.38	4.88	3.13	5.13	3.25	**12.13**
10	70.87	45.00	60.00	76.13	54.25	54.00	**83.75**
11	19.75	34.50	40.88	36.50	42.75	15.88	**51.88**
12	98.50	99.50	99.13	99.25	98.75	29.88	**99.68**
13	94.63	94.75	94.63	94.75	94.63	45.00	**95.13**
14	99.88	99.88	99.88	99.88	100	79.63	**100**
15	8.30	10.13	7.13	6.75	11.88	5.13	**21.50**
16	21.75	32.38	35.38	31.13	34.50	21.38	**51.63**
17	93.37	95.38	94.63	96.38	93.63	77.25	**97.50**
18	89.75	89.88	89.88	90.38	89.88	81.38	**91.88**
19	10.88	4.13	16.63	24.13	18.50	37.88	**52.88**
20	70.25	45.00	50.63	74.50	45.75	69.50	**83.38**
21	38.25	44.63	49.75	46.13	48.25	16.50	**57.38**

It can be easily seen from Table 4 that, for all the faults, the BVCDD method has the best performance. For nonlinear and non-Gaussian systems with different faults, the FGMM and KMPCA methods cannot establish very accurate models, hence their performance in fault detection is not promising. Although a single VCDD model cannot effectively handle all types of faults, the combination of multiple sub-models (BVCDD model) can well depict each sample so as to detect different kinds of faults.

Next, for further analysis, we focused on two typical faults (Faults 6 and 10) in the experiments. The results show that all the methods can detect Fault 6 with the rate of 100%. This means that the margin deviation information is sufficient for detecting Fault 6, which results in a step change in a feed. The scatter plots with this fault are shown in Figure 7, in which black and red points represent training data and test data respectively. From the view of mathematics, a single VCDD model can separate faulty data from original NOC data. The level of global accuracy may be intermediate or low for faults that lead to a large deviation. The single VCDD model can achieve a satisfying performance in process monitoring.

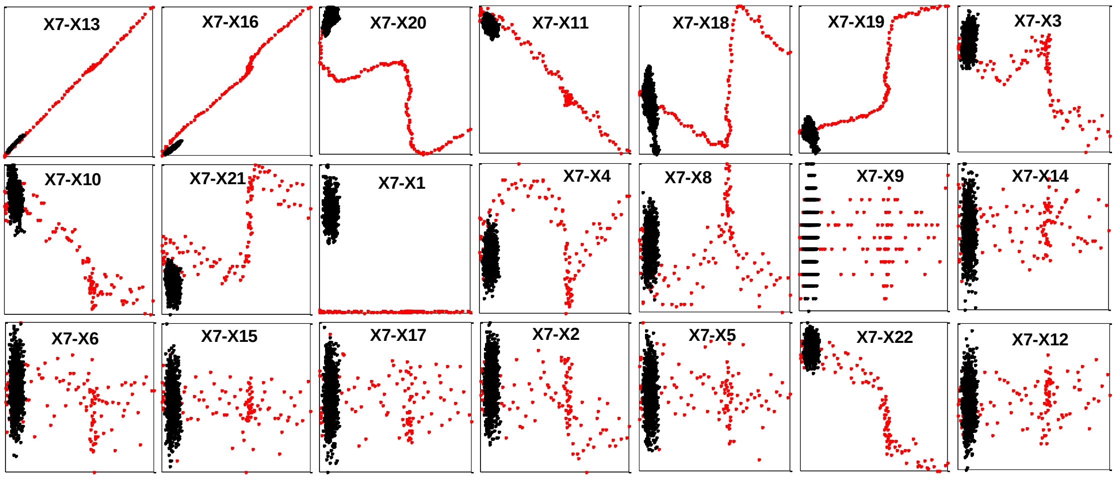


Figure 7. Scatter plots of the Tennessee Eastman process with Fault 6.

Figure 8 shows the control boundaries of VCDD (solid line) and BVCDD (dashed line) in the scatter plot of X_7 and X_{20}. Dot denotes training data, and plus denotes anomalies that BVCDD successfully detects and VCDD fails in fault 5. Because of the improvement of global accuracy, BVCDD detects an additional 44 anomalies while no additional anomalies are detected by VCDD. The small-scale fault means that the magnitude of the fault is small. It can be seen from the Figure 8 that this fault is a small-scale fault, because there are a lot of fault samples that are very close to the normal samples. The BVCDD model has a better performance than VCDD model. In addition, for the complex process the control boundary of BVCDD model is more accurate than the control boundary of VCDD model. The control boundary of BVCDD, calculated by the multiple VCDD models, may deviate the normal samples from fault samples more accurately than a single VCDD model. Due to the accurate control boundary of BVCDD model, some fault test samples are inside the control boundary of VCDD model while outside the control boundary of BVCDD model, which cannot be detected with VCDD model but alarmed with BVCDD model.

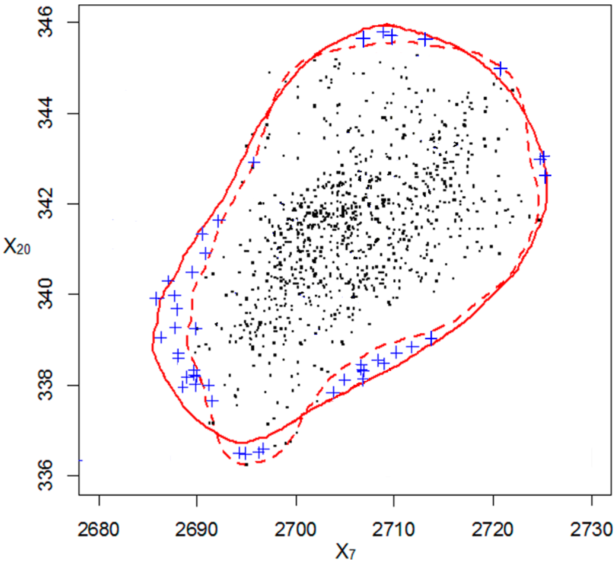


Figure 8. The control boundaries of VCDD (solid line) and BVCDD (dashed line) in the scatter plot of X_7 and X_{20}.

Figure 9 shows the fault detection charts of the TE process with Fault 10, which results in an intermediate bias due to controller adjustment. It can be seen from Table 4 that KMPCA only achieved 60% detection rate due to the invalidity of the Gaussian assumption. With BVCDD, the model was improved by integrating multiple sub-models and its detection rate climbed up to 83.75%. In summary, BVCDD acquired the best results on the TE process compared with the other methods.

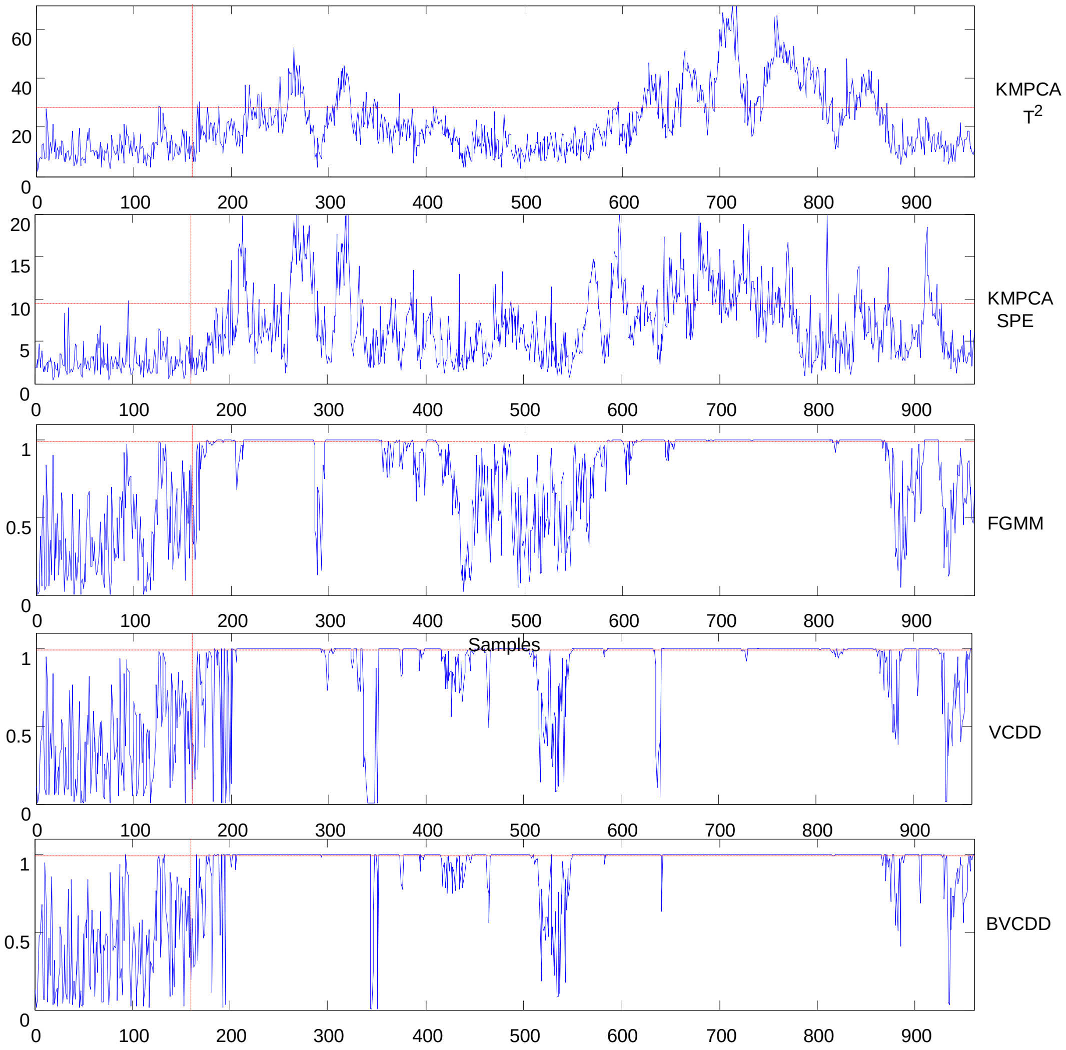


Figure 9. The fault detection charts of the TE process with Fault 10

5.2 An Acetic Acid Dehydration Process

Purified terephthalic acid is an important raw material for polyester production widely used in textile and packaging industries. It is produced by catalytic oxidization of p-xylene followed by subsequent purification of the crude terephthalic acid by selective hydrogenation. Acetic acid is the solvent in the oxidization process. Usually a part of acetic acid leaves the top of the reactor with water. Solvent dehydration column is used to recover acetic acid from waste water. A plant in China applies ordinary distillation method to separate acetic acid and water. Under the normal circumstances, the top and the bottom products of the distillate column are water and acetic acid, respectively. Although acetic acid and water do not form an azeotrope at atmospheric pressure, conducting simple distillation to separate these two components would require many equilibrium stages and biggish reflux ratio since the system has a tangent pinch on the pure water end. Usually the top product contains acetic acid less than 1.15% and the acetic acid product contains water from 5% to 8%, when the acetic acid returns the oxidation reactor after purification.

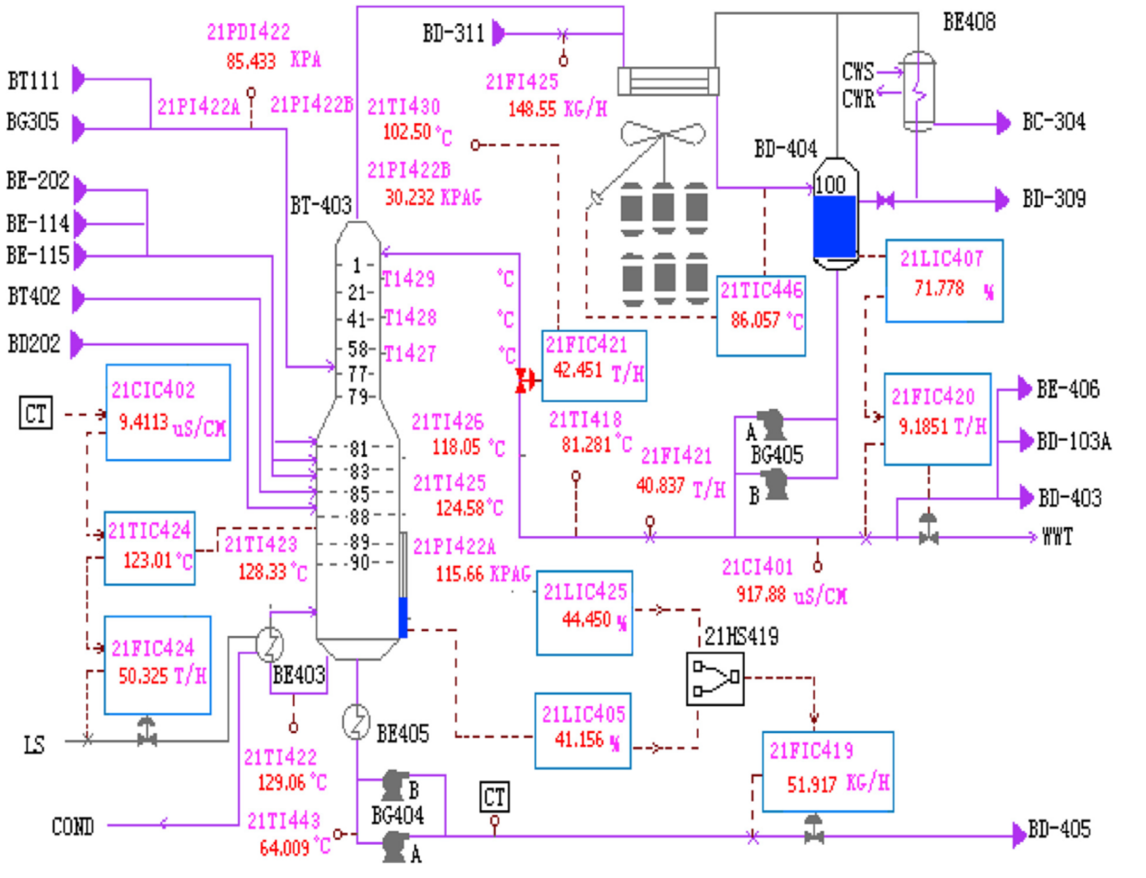


Figure 10. The control system of the dehydration process.

Figure 10 illustrates the control system of the dehydration process. The column has 90 stages and four feeds. The mixture of acetic acid and water comes from the oxidation reactor. In this study, 21 continuous process variables, including temperatures, pressures, flowrates and conductivities and so on, were used in real-time monitoring for identifying the top acetic acid faults. Additionally, 300 samples and 300 testing cases were chosen from the DC (dehydration control) system for evaluation of different approaches.

The monitored data contain 300 samples, in which the first, second and third parts (each having 100 samples) represent normal condition data, fault data, and normal condition data, respectively. The fault data were caused by the step change of acetic acid concentration from below 1% to overpass 1.2%. Table 5 presents the FDRs and FARs of the BVCDD, KMPCA, FGMM, and VCDD methods. It can be seen that the BVCDD method has the best performance among the four methods.

Table 5. Fault detection performance with different approaches

Detection Index	FGMM	KMPCA		VCDD	BVCDD
		T^2	SPE		
FDR	0.60	0.61	0.62	0.61	**0.98**
FAR	0.09	0.11	0.07	0.055	**0.02**

Figure 11 shows the control boundaries of VCDD (dashed line) and BVCDD (solid line) in the scatter plot of X_9 and X_{16}. Dot denotes training data, plus denotes anomalies that BVCDD successfully detects and VCDD fails, and the cross denotes normal sample that BVCDD successfully detects but VCDD fails. Because of the improvement of global accuracy, BVCDD detects an additional 37 anomalies while no additional anomalies are detected by the VCDD, and the VCDD wrongly detects an additional 7 anomalies while the BVCDD detects correctly. It can be seen from the Figure 11 that the fault samples are close to the normal samples, and the BVCDD model has a more accurate control boundary than the VCDD model. Due to the accurate control boundary of the BVCDD model, some fault test samples are inside the control boundary of the VCDD model while outside the control boundary of the BVCDD model, which cannot be detected with the VCDD model but alarmed with the BVCDD model.

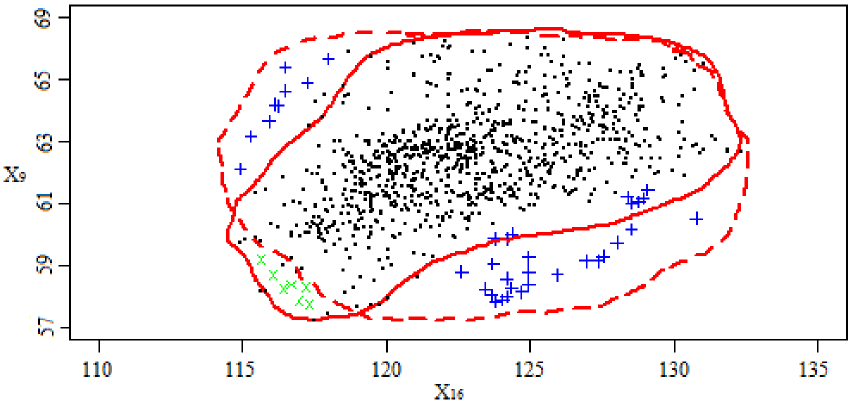


Figure 11. The control boundaries of VCDD (dashed line) and BVCDD (solid line) in the scatter plot of X_9 and X_{16}.

The detailed monitoring results using the four different methods are also shown in Table 5. For the normal condition data, the BVCDD approach can offer the lowest FAR; while with fault condition data, it can hold 98% FDR, indicating that the proposed BVCDD approach performs very well in the acetic acid hydration process monitoring.

6. Conclusions

This paper proposes the BVCDD method which combines the VCDD and boosting techniques in process monitoring. Using the GBIP index to measure the depiction degrees of the existing model, each sample is given an individual weight to determine its probability of being selected for the next sub-model learning. In this way, the samples with bigger error in the current sub-model can get more utilized in training the next sub-model to depict all samples more precisely. As for the number of training samples chosen for each sub-model and the number of sub-models, this paper proposes a new method to set their values adaptively. Finally, the BVCDD model has been successfully applied to monitor nonlinear and non-Gaussian systems with multiple faults. The comparison with the FGMM, KMPCA and VCDD methods demonstrates that the proposed BVCDD model has a higher FDR index and a lower FAR index. This indicates that the proposed BVCDD method is highly promising in practice with extensive application opportunities. Especially for the TE process and the real acetic acid dehydration process, which are the nonlinear and non-Gaussian systems with multiple faults, the proposed BVCDD method exhibits a very prominent advantage.

Acknowledgments

The authors of this paper appreciate the National Natural Science Foundation of China (under Project No. 21676086) for their financial support.

References

[1] Pariyani, A.; Seider, W. D.; Oktem, U. G.; Soroush, M. Dynamic risk analysis using alarming databases to improve process safety and product quality: part II. Bayesian analysis. *AIChE J*. **2012**, 58, 826−841.

[2] Khakzad, N.; Khan, F.; Amyotte, P. Dynamic risk analysis using bow-tie approach. *Reliab. Eng. Syst. Safe*. **2012**, 104, 36−44.

[3] Hashemi, S. J.; Ahmed, S.; Khan, F. I. Risk-based operational performance analysis using loss functions. *Chem. Eng. Sci.* **2014**, 116, 99−108.

[4] Adhitya, A.; Cheng, S.; Lee, Z.; Srinivasan, R. Quantifying the effectiveness of an alarm management system through human factors studies. *Comput. Chem. Eng.* **2014**, 67, 1−12.

[5] Cheng, Y.; Izadi, I.; Chen, T. Pattern matching of alarm flood sequences by a modified Smith. Waterman algorithm. *Chem. Eng. Res. Des.* **2013,** 91, 1085−1094.

[6] Li, D.; Hu, J.; Wang, H.; Huang, W. A distributed parallel alarm management strategy for alarm reduction in chemical plants. *J. Process Control*. **2015**, 34, 117−125.

[7] Khan, F.; Wang, H.; Yang, M. Application of loss functions in process economic risk assessment. *Chem. Eng. Res. Des*. **2016**, 111, 371−386.

[8] Venkatasubramanian, V.; Rengaswamy, R.; Yin, K.; Kavuri, S. N. A Review of Process Fault and Diagnosis, Part I: Quantitative Model-Based Methods. *Comput. Chem. Eng*. **2003**, 27, 293−311

[9] Venkatasubramanian, V.; Rengaswamy, R.; Kavuri, S. N. A Review of Process Fault and Diagnosis, Part II: Qualitative models and Search Strategies. *Comput. Chem. Eng*. **2003**, 27, 313−326

[10] Venkatasubramanian, V.; Rengaswamy, R.; Kavuri, S. N.; Yin, K. A Review of Process Fault and Diagnosis, Part III: Process History Based-Methods. *Comput. Chem. Eng*. **2003**, 27, 327−346.

[11] Qin, S.J. Survey on data-driven industrial process monitoring and diagnosis. *Annual Reviews in Control.* **2012**, 36, 220-234.

[12] Ge, Z.; Song, Z. Review of recent research on data-based process monitoring. *Ind. Eng. Chem. Res.* **2013**, 52, 3543-3562.

[13] Zhang, Y. Fault detection and diagnosis of nonlinear processes using improved kernel independent component analysis (kica) and support vector machine (svm). *Ind. Eng. Chem. Res.* **2008**, 47, 6961-6971

[14] Zhang, Y. Enhanced statistical analysis of nonlinear processes using kpca, kica and svm. *Chem. Eng. Sci.* **2009**, 64, 801-811.

[15] Hadad, K.; Pourahmadi, M.; Majidi-Maraghi, H. Fault diagnosis and classification based on wavelet transform and neural network. *Prog. Nucl. Energ.* **2011**, 53, 41-47.

[16] Li, S.; Wen, J. A model-based fault detection and diagnostic methodology based

on PCA method and wavelet transform. *Energ. Buildings.* **2014**, 68, 63-71.

[17] Cai, B.; Liu, Y.; Fan, Q. et al. Multi-source information fusion based fault diagnosis of ground-source heat pump using Bayesian network. *Appl. Energ.* **2014**, 114, 1-9.

[18] Kramer, M.A. Nonlinear principal component analysis using auto associative neural networks. *AIChE Journal*. **1991**, 37, 233-243.

[19] Schölkopf, B.; Smola, A.J.; Müller, K. Nonlinear component analysis as a kernel eigenvalue problem. *Neural. Comput.* **1998**, 10, 1299-1399.

[20] Hyvärinen, A.; Oja, E. Independent component analysis: algorithms and applications. *Neural Networks* **2000**, 13, 411-430.

[21] Bingham, E.; Hyvärinen, A. A fast fixed-point algorithm for independent component analysis of complex valued signals. *Int. J. Neural. Syst.* **2000**, 10, 1-8.

[22] Hyvärinen, A. Fast and robust fixed-point algorithms for independent component analysis. *IEEE T. Neural Networ.* **1999**, 10, 626-634.

[23] Ren, X.; Tian, Y.; Li, S. Vine Copula-Based Dependence Description for Multivariate Multimode Process Monitoring. *Ind. Eng. Chem. Res.* **2015**, 54, 10001-10019.

[24] Yu, H.; Khan, F.; Garaniya, V. Modified Independent Component Analysis and Bayesian Network based Two-stage Fault Diagnosis of Process Operations. *Ind. Eng. Chem. Res.* **2015**, 54, 1-19.

[25] Zhou, Y.; Ren, X.; Li S. Enhancing Quality of Multivariate Process Monitoring Based on Vine Copula and Active Learning Strategy. *Ind. Eng. Chem. Res.* **2017**, 57 (23), 7961–7974.

[26] Ren, X.; Zhu, K.; Cai, T. Fault Detection and Diagnosis for Nonlinear and Non-Gaussian Processes Based on Copula Subspace Division. *Ind. Eng. Chem. Res.* **2017**, 56, 11545-11564.

[27] Zhou Z. Ensemble Methods Foundations and Algorithms. *CRC Press*. 2012.

[28] Yu, H.; Khan, F.; Garaniya, V. Modified Independent Component Analysis and Bayesian Network based Two-stage Fault Diagnosis of Process Operations. *Ind. Eng. Chem. Res.* **2015**, 54(10), 1-19.

[29] Zadakbar, O.; Imtiaz, S.; Khan, F. Dynamic Risk Assessment and Fault Detection

Using Principal Component Analysis. *Ind. Eng. Chem. Res.* **2013**, 52:809-816.

[30] Valiant, L. A Theory of Learnable. *Commun. ACM*. **1984**, 27, 1134-1142.

[31] Gubala, A. M.; Schmitz, J. F.; Kearns, M. J., et al. The goddard and saturn genes are essential for Drosophila male fertility and may have arisen de novo. *Mol. Biol. Evol.* **2017**, 34, 1066-1082.

[32] Valiant, L. G. What must a global theory of cortex explain?. *Curr. Opin. Neurobiol.* **2014**, 25, 15-19.

[33] Schapire, R. E. The strength of weak learnability. *Mach. Learn.* **1990**, 5, 197-227.

[34] Freund, Y. Boosting a Weak Algorithm by Majority. *Inform. Comput.* **1995**, 121, 256-285.

[35] Breiman, L. Bagging Predictors. *Mach. Learn.* **1996**, 24, 123-140.

[36] Ge, Z.; Song, Z. Subspace partial least squares model for multivariate spectroscopic calibration. *Chemometr. Intell. Lab.* **2013**, 125, 51-57.

[37] Ge, Z.; Song, Z. Performance-driven ensemble learning ICA model for improved non-Gaussian process monitoring. *Chemometr. Intell. Lab.* **2013**, 123, 1-8.

[38] Li, N.; Yang, Y. Ensemble Kernel Principal Component Analysis for Improved Nonlinear Process Monitoring. *Ind. Eng. Chem. Res.* **2015**, 54, 318-329.

[39] Zhang, M.; Ge, Z.; Song, Z.; Fu, R. Global-Local Structure Analysis Model and Its Application for Fault Detection and Identification. *Ind. Eng. Chem. Res.* **2011**, 50, 6837-6848.

[40] Kearns M., Valiant L. G. Learning Boolean Formulae or Factoring. Technical Report TR-1448, *Cambridge, MA: Harvard University Aiken Computation laboratory*, **1988**.

[41] Kearns M.; Valiant L. G. Cryptographic Limitation on Learning Boolean Formulae and Finite Automata. *Proceedings of the 21st Annual ACM Symposium on Theory of Computing* **1989**, 433-444.

[42] Freund, Y.; Schapire, R. E. A Decision-Theoretic Generalization of Online Learning and an Application to Boosting. *J. Comput. Syst. Sci.* **1997**, 55, 119-139.

[43] Hu, J.; Li, Y.; Zhang, M. Predicting Protein-DNA Binding Residues by Weightedly Combining Sequence-Based Features and Boosting Multiple SVMs. *IEEE-ACM T.*

Comput. Bi. **2017**, 14, 1389-1398.

[44] Fernandez-Baldera, A.; Buenaposada, J. M.; Baumela, L. BAdaCost: Multi-class Boosting with Costs. *Pattern Recogn.* **2018**, 79, 467-479.

[45] Soleymani, R.; Granger, E.; Fumera, G. Progressive boosting for class imbalance and its application to face re-identification. *Expert Syst. Appl.* **2018**, 101, 271-291.

[46] Schapire, E. R.; Freund, Y.; Barlett, P.; Lee, W. S. Boosting the margin: A new explanation for the effectiveness of voting methods. *Ann. Stat.* **1998**, 26, 1651-1686.

[47] Schwenk, H.; Bengio, Y. Boosting neural networks. *Neural. Comput.* **2000**, 12, 1869-1887.

[48] Sklar, A. Fonctions dé repartition à n dimensions et Leurs Marges. *Publ. Inst. Statist. Univ. Paris.* **1959**, 8, 229-231.

[49] Nelsen, R. B. An introduction to Copulas. *Springer: Berlin*, **2006**.

[50] Kurowicka, D.; Joe, H. Dependence modeling: vine copula handbook. *World Scientific: NJ*, **2011**.

[51] Joe, H. Multivariate models and multivariate dependence concepts. *CRC Press*, **1997**.

[52] Aas, K.; Czado, C.; Frigessi, A. et al. Pair-Copula constructions of multiple dependence. *Insur. Math. Econ.* **2009**, 44, 182-198.

[53]Genest, C.; Favre, A. C. Everything you always wanted to know about Copula modeling but were afraid to ask. *J. Hydrol. Eng.* **2007**, 12(4), 347-368.

[54] Kim, D.; Kim, J. M.; Liao, S. M. et al. Mixture of D-Vine Copulas for modeling dependence. *Comput. Stat. Data. An.* **2013**, 64, 1-19.

[55] Smith, M.; Min, A.; Czado, C. et al. Modelling longitudinal data using a Pair-Copula decomposition of serial dependence. *J. Am. Stat. Assoc.* **2010**, 105, 1467-1479.

[56] Alexander, J. M. Sampling nested Archimedean copulas. *J. Stat. Comput. Sim.* **2008**, 78, 567-581.

[57] Yu, J.; Qin, S. J. Multimode process monitoring with Bayesian inference-based finite Gaussian mixture models. *AIChE Journal.* **2008**, 54, 1811-1829.

[58] Gutta, S.; Huang, J. R. J.; Phillips, P. J. Mixture of experts for classification of gender, ethnic origin, and pose of human faces. *IEEE T. Neural Networ.* **2000**, 11, 948-

960.

[59] Hyndman, R. J. Computing and graphing highest density regions. *Am. Stat.* **1996**, 50, 120−126.

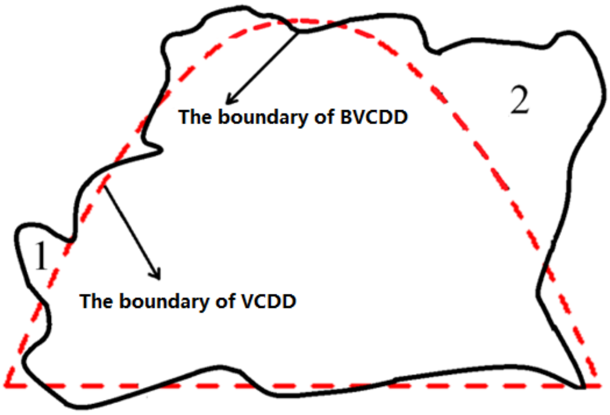
[60] Duong, T.; Koch, I.; Wand, M. P. Highest Density Difference Region Estimation with Application to Flow Cytometric Data. *Biometrical J*. **2009**, 51, 504-521.

[61]Claeys D. D.; Verstraelen T.; Pauwels E. Conformational Sampling of Macrocyclic Alkenes Using a Kennard−Stone-Based Algorithm. *J. Phys. Chem. A* **2010**, 114, 6879-6887.

[62] Downs, J.J.; Vogel, E.F. A plant-wide industrial process control problem. *Comput. Chem. Eng.* **1993**, 17, 245-255.

[63] Li, G.; Alcala, C. F.; Qin, S. J.; Zhou, D. Generalized reconstruction-based contributions for output-relevant fault diagnosis with application to the Tennessee Eastman process. *IEEE Trans. Control Syst. Technol.* **2011**, 19, 1114−1127.

TOC graphic:


2
The boundary of BVCDD
1
The boundary of VCDD