Journal of Process Control 75 (2019) 136–155

Contents lists available at ScienceDirect

Journal of Process Control

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

Gaussian feature learning based on variational autoencoder for improving nonlinear process monitoring

Zehan Zhang*, Teng Jiang, Chengjun Zhan, Yupu Yang

Key Laboratory of Ministry of Education in System Control and Information Processing, Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China

ARTICLE INFO

Article history:
Received 23 March 2018
Received in revised form 23 July 2018
Accepted 15 January 2019

Keywords:
Nonlinear process monitoring
Deep learning
Variational autoencoder
Gaussian feature learning

ABSTRACT

Deep learning algorithms, especially the autoencoders, have been applied in nonlinear process monitoring recently. However, the features extracted by the autoencoders can hardly follow the Gaussian distribution, consequently, the control limit of the corresponding monitoring statistic can not be determined by an F or χ^2 distribution. Recent improvements in the unsupervised learning domain of deep learning offer opportunities to avoid the problem. In this paper, a novel nonlinear process monitoring method based on variational autoencoder (VAE) is proposed to tackle the Gaussian assumption problem. Due to the Gaussian distribution limitation added in the hidden layer of the VAE, it can not only automatically learn the key features of the nonlinear system, but also learn features that follow the Gaussian distribution. The Gaussian feature representations obtained from VAE are then provided to construct a new statistic H^2 whose control limit can be easily determined by a χ^2 distribution. A nonlinear numerical study and the TE benchmark process have verified the effectiveness of the proposed method.

1. Introduction

The scale of the modern industrial systems has become more and more complex. Timely detection of faults in these systems is critical to ensuring the safety of people and property, and improving product quality [1–3]. Traditional methods based on process mechanistic or models are not always available or may be difficult to construct due to the high complexity of the system [4]. With the wide application of distributed control systems (DCS) and advances in computer technology, a large amount of industrial data is collected and stored. As a result, alternative monitoring methods based on data have received a great deal of attention, especially methods based on multivariate statistical process monitoring (MSPM). MSPM mines the correlations among the system variables in historical normal data, generates key feature sets, and eventually builds the monitoring model [5–9].

Among many MSPM methods, principal component analysis (PCA) serves as a basic monitoring method [4,10,11]. In general, PCA transforms data into two parts: one converts the relevant variables into a series of orthogonal variables (principal components) and the other stores the residual information (residual space). The principal components retain as much data variability as possible, and the Hotelling's T^2 statistic is typically constructed in this space. The Q-statistic, also known as the squared prediction error (SPE), is used in residual space to make up the overly sensitive to inaccuracies for T^2 statistic. Because of its simplicity and effectiveness, PCA has been used in many industrial processes. However, the conventional PCA-based monitoring methods are based on the assumption that process variables are linear and Gaussian distributed. In a real complicated industrial process, characteristics of the collected data are very complex, and the internal variables are highly nonlinear and do not satisfy the Gaussian distribution. In this case, PCA cannot fully characterize the data and thus has poor monitoring performance.

To address the nonlinear problem mentioned above, different types of nonlinear methods have been proposed. The first representative nonlinear method is based on earlier neural network (NN) approaches. For example, Kramer [12] proposed the autoassociative neural network as a nonlinear PCA method, Dong and McAvoy [13] combined the principal curve and the neural network to tackle the nonlinear problem, and Geng and Zhu [14] proposed the NLPCA method that is also based on the neural network. These methods based on the traditional neural network need to develop the model offline and train the model through some optimization methods. At that time, due to the poor performance of computers, the small amount of data, and the limitations of the traditional neural network technology, fewer and fewer researchers paid attention to the NN-based methods. In contrast, the kernel learning method, especially the method based on kernel PCA (KPCA), has attracted the attention of many researchers [15–20]. By introducing a non-

* Corresponding author.
E-mail addresses: zehanzhang@126.com (Z. Zhang), jtengyp@163.com (T. Jiang), wszhancj@sjtu.edu.cn (C. Zhan), ypyang@sjtu.edu.cn (Y. Yang).

https://doi.org/10.1016/j.jprocont.2019.01.008
0959-1524/

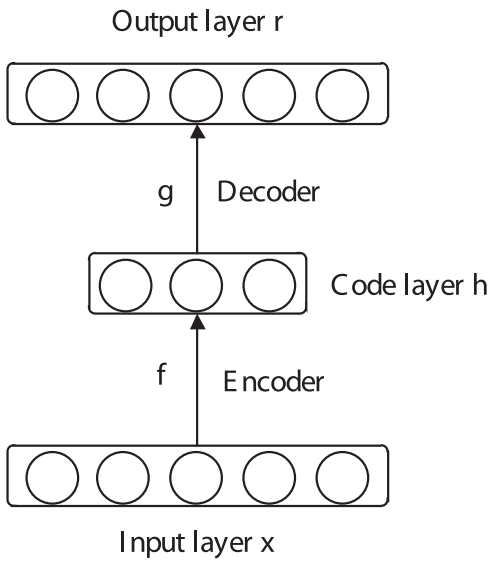

Fig. 1. The whole architecture of autoencoder.

linear kernel function, KPCA first maps the original data to a high-dimensional feature space, and then uses the PCA algorithm to set up the process monitoring model in this space. Similar to PCA, two monitoring statistics are constructed separately for monitoring the main information part and the noisy part. KPCA's model is easy to implement and various types of nonlinearity can be modeled, so it is used in many process monitoring applications. However, it is very difficult to choose a suitable kernel function, and the inappropriate kernel function will not correctly reflect the characteristics of the process data [21,22]. Another type of nonlinear method is linear approximation approach (LAA) [23,24]. In this method, several local linear models are used to approximate the entire nonlinear space. When the local linear models are established, Bayesian inference is used to integrate the results of local models to perform fault detection for the entire process. LAA is easier to implement than NN and KPCA, but the disadvantages are obvious. First, the number of local models is difficult to determine; second, it may not be able to fully reflect the data's nonlinearity.

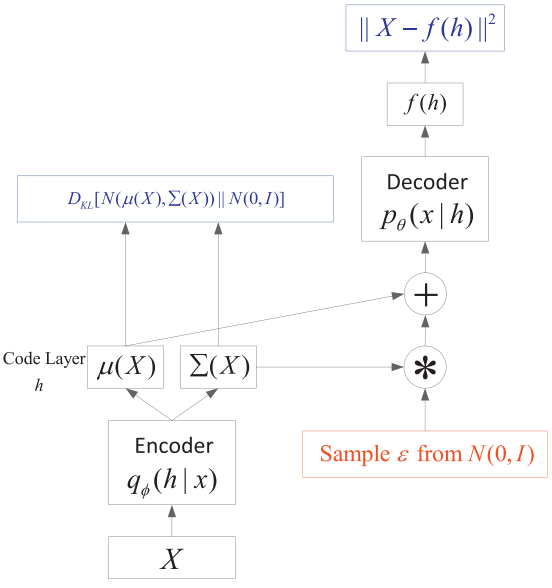

Fig. 2. The architecture of variational autoencoder at training time with reparameterization trick. Blue shows loss layers. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

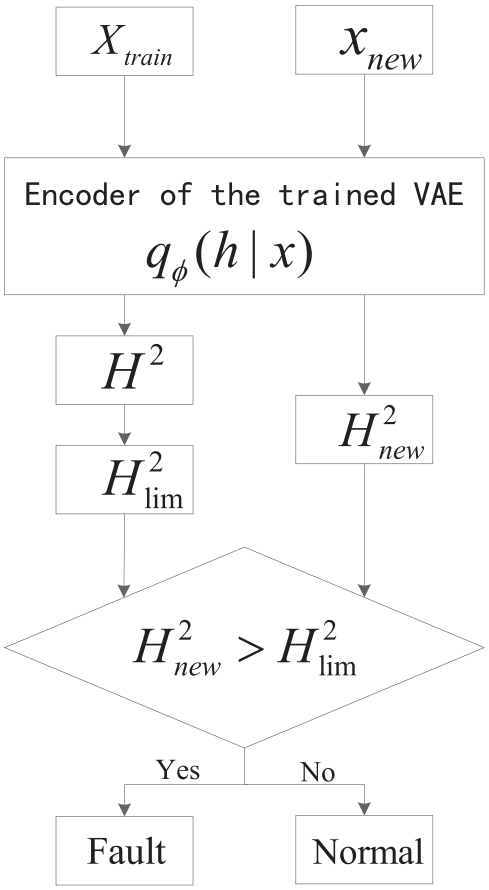

Fig. 3. The procedure of VAE-based fault detection.

More recently, due to the advances in computer technology, the development of deep neural network algorithms, and the growth of collected and stored data, a new powerful machine learning algorithm called deep learning (DL) has achieved great success in many applications [25–30]. Deep learning technology abstracts the original data into low-level feature representations through multi-level feature representation, and uses these features to mine the intricacies of the data. Up to now, some scholars have already applied deep learning to the field of nonlinear process monitoring and achieved good results [31]. Yan et al. [32] proposed a nonlinear monitoring method based on variant autoencoders. Lv et al. [33] used stacked sparse autoencoders to perform nonlinear fault detection and diagnosis. Jiang et al. [34] proposed a semi-supervised fault classification method based on sparse autoencoders to tackle the dynamic nonlinear problem. Among various of DL-based nonlinear process monitoring methods, autoencoder (AE) plays a central role. It has the main advantages:

(1) Its model is completely trained on process data and able to learn nonlinear features from the data automatically, which is very helpful for discovering intricate information inside high-dimensional nonlinear data.
(2) It provides an initial way for deep neural networks to make models deeper through the stacking way.
(3) It is suitable for mining big data, generally speaking, the more data, the better the generalization ability of the model.

When performing fault detection, the T^2 statistic, also called H^2 [32], is constructed in the AE's feature space. However, the control

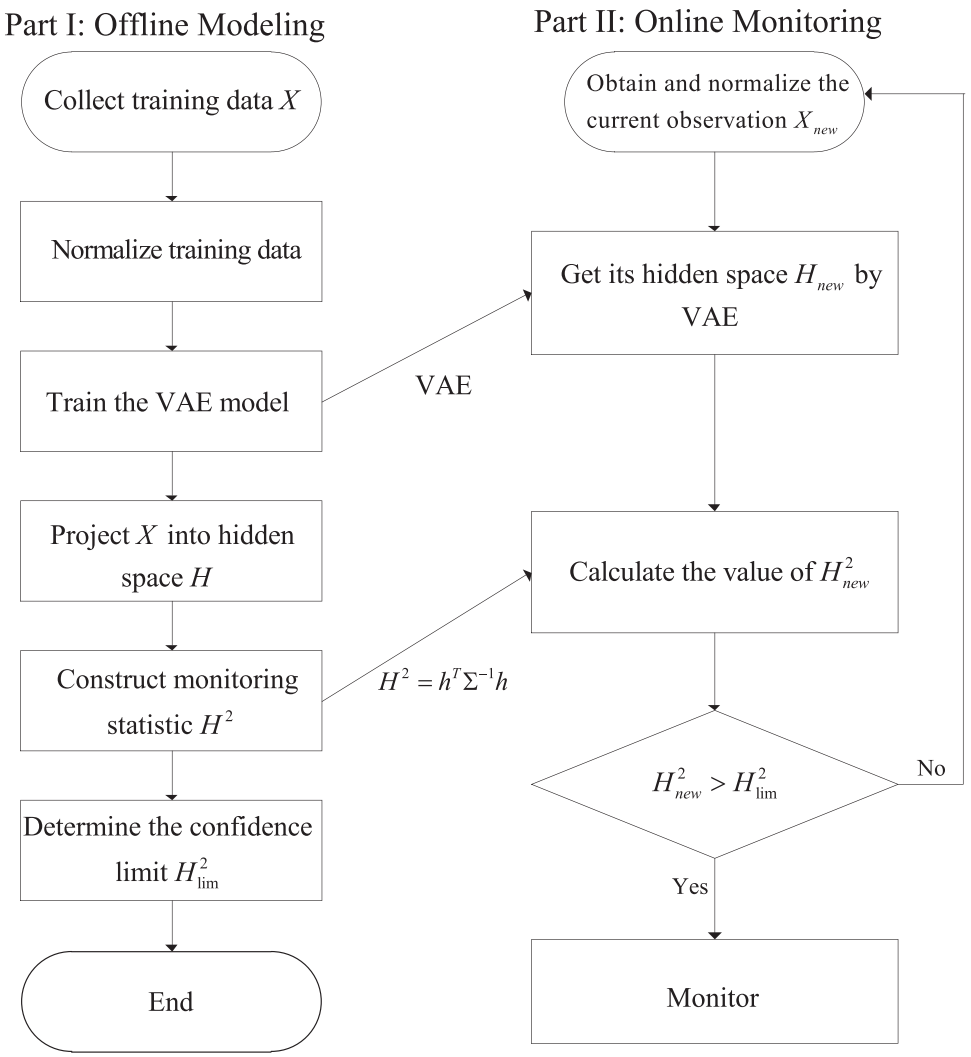

Fig. 4. Flow chart of VAE based process monitoring.

limit of the T^2 statistic is estimated under the assumption that the features follow Gaussian distribution. In AE, due to nonlinear transformation, it is difficult to ensure that the extracted features follow Gaussian distribution. Therefore, the control limit of T^2 statistic cannot be determined by the known F distribution.

To address the problem stated above, this paper proposes a novel nonlinear process monitoring method based on variational autoencoder (VAE) [35]. VAE is a powerful generative model that has been successfully used in many applications [35–38]. It comes from Bayesian inference, and wants to model the potential probability distribution of data so that it can generate new samples from this distribution. The Gaussian distribution restriction is added to the hidden layer of VAE so that the features learned by VAE follow the Gaussian distribution. Therefore, regardless of the distribution of the raw data, the features extracted by VAE follow the Gaussian distribution. Moreover, the control limit of the newly constructed statistic H^2 in the feature space can be easily determined by a χ^2 distribution. In this paper, the VAE model is first trained on the normal dataset to extract key Gaussian features. Next, a new monitoring statistic H^2 is constructed based on the Gaussian feature representations with the corresponding control limit determined by a χ^2 distribution.

The remainder of this paper is organized as follows. First, a brief review of AE is given in Section 2. Section 3 details the proposed method for nonlinear process monitoring, including the development of the VAE model, the construction and control limit estimation of the H^2 statistic, and the entire monitoring strategy based on VAE. In Section 4, a nonlinear numerical system and the TE benchmark process are provided to demonstrate the efficiency of the proposed method. Some in-depth discussion of the proposed method is then given in Section 5. Finally, some conclusions are drawn.

2. Preliminaries

This section provides an overview of the basic autoencoder.

2.1. The basic autoencoder (AE)

The basic autoencoder, such as the one used in [26], is a neural network that is trained to try to copy its input to its output, which is an unsupervised machine learning technique. Specially, the network can be thought of as containing of two parts: an encoder function $\mathbf{h} = f(\mathbf{x})$ and a decoder function $\mathbf{r} = g(\mathbf{h})$ that produces a reconstruction of $\mathbf{x}$. Fig. 1 presents the whole architecture.

The encoder is used to find a latent layer h that can represent the input. Internally, it transforms an input $\mathbf{x} \in R^d$ into a hidden representation $\mathbf{h} \in R^{d'}$ with the deterministic mapping:

$$\mathbf{h} = f(\mathbf{x}) = s(\mathbf{W}\mathbf{x} + \mathbf{b}) \tag{1}$$

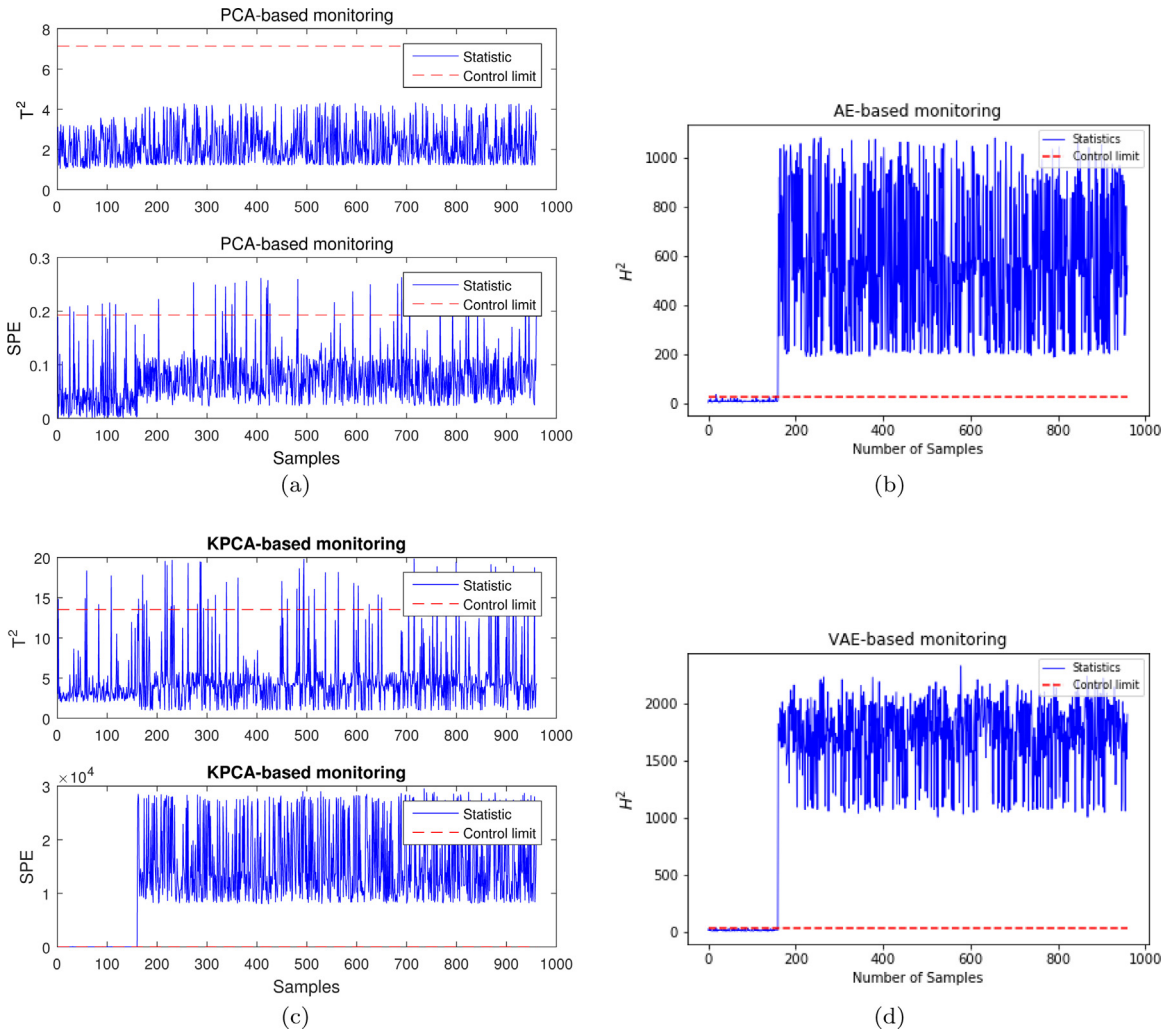

Fig. 5. Monitoring charts of fault 1 in the numerical system: (a)PCA, (b) AE, (c) KPCA and (d) VAE.

where s is an activation function, usually a nonlinear function such as the sigmoid function $s(x) = \frac{1}{1+e^{-x}}$, $\mathbf{W}$ is a $d' \times d$ weight matrix, and $\mathbf{b}$ is the corresponding bias vector.

The decoder is the opposite of the encoder, whose input is the latent vector produced by encoder and output is the reconstruction of the original input. In detail, a transformation g maps back the resulting hidden representation $\mathbf{h}$ to a d-dimensional reconstruction of $\mathbf{x}$ denoted by $\mathbf{r}$:

$$\mathbf{r} = g(\mathbf{h}) = s(\mathrm{W}'\mathrm{h} + \mathrm{b}') \tag{2}$$

where s is also a nonlinear function, $\mathbf{W}'$ is a $d \times d'$ weight matrix, and $\mathbf{b}'$ is the corresponding bias vector.

In an autoencoder, a training data $\mathbf{x}$ is transformed to a corresponding representation $\mathbf{h}$ and a reconstruction $\mathbf{r}$, parameterized by $\theta = \{\mathbf{W}, \mathbf{b}\}$ and $\phi = \{\mathbf{W}', \mathbf{b}'\}$. The object of the autoencoder is to minimize the reconstruction error between the input $\mathbf{x}$ and reconstruction $\mathbf{r}$. The parameters of the two networks (encoder and decoder) can be trained together to minimize the objective function:

$$\begin{aligned} \theta^*, \phi^* &= \mathrm{argmin}_{\theta,\phi} \frac{1}{n} \sum_{i=1}^{n} L(\mathbf{x}^i, \mathbf{r}^i) \\ &= \mathrm{argmin}_{\theta,\phi} \frac{1}{n} \sum_{i=1}^{n} L(\mathbf{x}^i, g(f(\mathbf{x}^i))) \end{aligned} \tag{3}$$

where n represents the number of training samples, L is a loss function that can take many forms according to the distribution assumptions on the input. The conventional one is squared error:

$$\mathbf{L}(\mathrm{x}, \mathrm{r}) = \|\mathbf{x} - \mathbf{r}\|^2 \tag{4}$$

If the interpretation of $\mathbf{x}$ and $\mathbf{r}$ is either bit vector or vector of bit probabilities, the cross-entropy can be an alternative:

$$\mathbf{L}(\mathrm{x}, \mathrm{r}) = -\sum_{k=1}^{d} [\mathbf{x}_k \log \mathbf{r}_k + (1 - \mathbf{x}_k) \log(1 - \mathbf{r}_k)] \tag{5}$$

Stochastic gradient descent is typically applied to the above optimization, and the required gradients are easily obtained by using the backpropagation algorithm [25].

3. VAE-based nonlinear process monitoring

This section introduces the detail of the proposed VAE-based process monitoring method. The original VAE technique from the perspective of generated model is firstly described. Then, in order to illustrate the essential difference between the VAE used in this study and the original VAE, we have newly introduced the VAE from the perspective of feature extraction. The construction of the monitoring statistic for the VAE is then given, followed by its estimation of the control limit. Finally, the whole monitoring scheme of the proposed method is given.

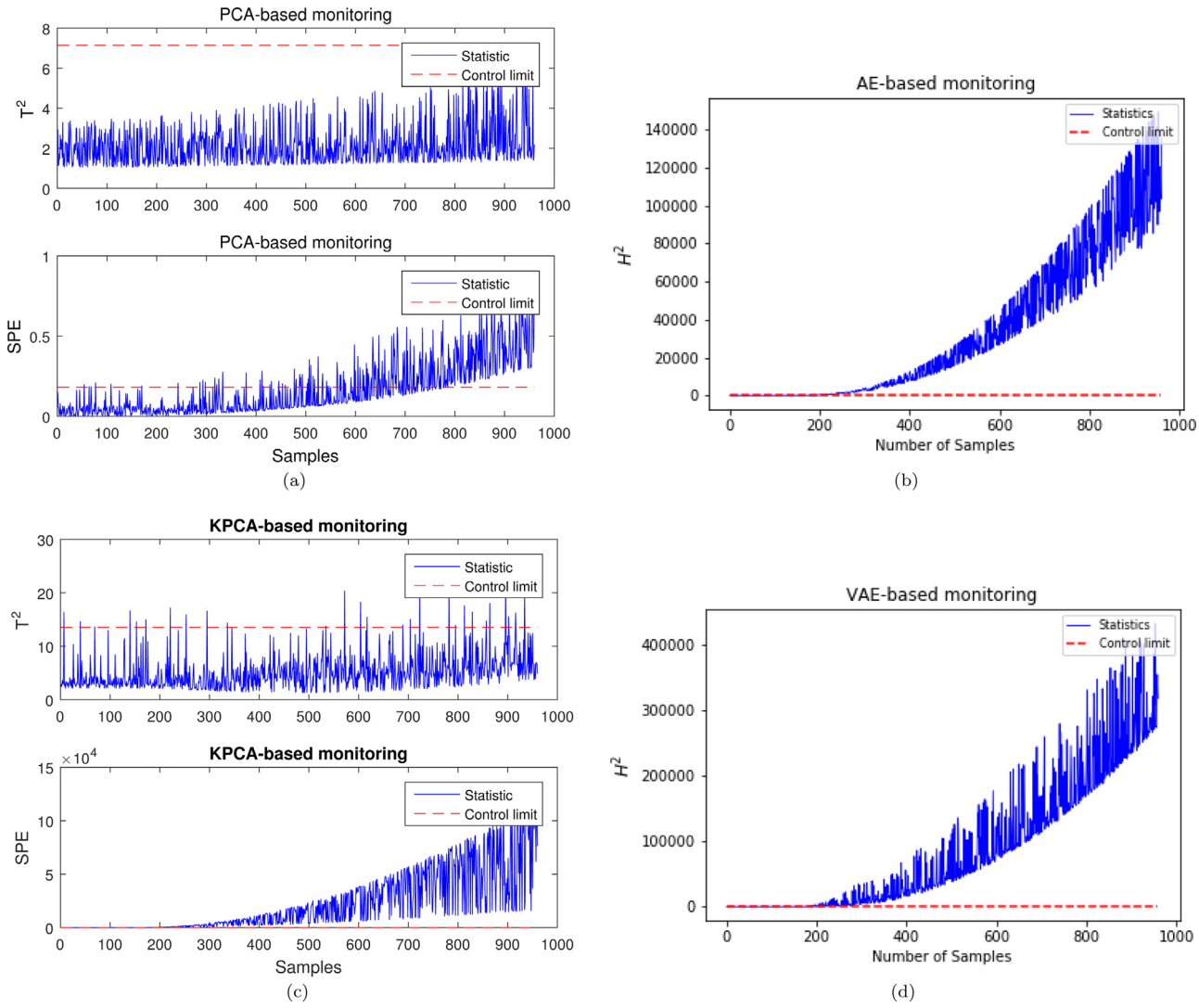

Fig. 6. Monitoring charts of fault 2 in the numerical system: (a) PCA, (b) AE, (c) KPCA and (d) VAE.

3.1. Original variational autoencoders (VAE): generated model

The original VAE was proposed to solve the inference problem in generative models. Therefore, this section first introduces VAE from the perspective of generative models, thus explaining the essential difference between the original VAE application and the VAE used in this paper.

The goal of the generative model is to generate the required data $\{\mathbf{x}^i\}_{i=1}^N$ from underlying unobserved random latent variable $\mathbf{h}$, where N is the number of samples. This process consists of two steps: (1) generate $\mathbf{h}^i$ from the prior distribution $p_{\boldsymbol{\theta}}(\mathbf{h})$; (2) generate $\mathbf{x}^i$ from some conditional distribution $p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{h})$[35]. θ is the parameter of the above generated model. The optimal parameter is then obtained by maximizing the likelihood $p_\theta(\mathbf{x})$. There is a very serious problem when maximizing likelihood $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{h})p_\theta(\mathbf{x}|\mathbf{h})d\mathbf{h}$: because humans cannot exhaust all $\mathbf{h}$s, the above calculation process is intractable. However, the VAE has subtly avoided the above problem.

The log likelihood $\log p_\theta(\mathbf{x}^i)$ can be written in the following form:

$$\log p_\theta(\mathbf{x}^i) = \mathbb{E}_{q_\phi(\mathbf{h}|\mathbf{x}^i)}[\log p_\theta(\mathbf{x}^i|\mathbf{h})] - D_{KL}(q_\phi(\mathbf{h}|\mathbf{x}^i)||p_\theta(\mathbf{h})) + D_{KL}(q_\phi(\mathbf{h}|\mathbf{x}^i)||p_\theta(\mathbf{h}||\mathbf{x}^i)) \quad (6)$$

where $p_\theta(\mathbf{x}^i|\mathbf{h})$ is the decoder network, $q_\phi(\mathbf{h}|\mathbf{x}^i)$ is the encoder network, θ and ϕ are corresponding parameters, and $D_{KL}(P||Q)$ is the Kullback–Leibler (KL) divergence [39]. For continuous probability distributions P and Q, the form of the Kullback–Leibler divergence is as follows:

$$D_{KL}(P||Q) = \int_{-\infty}^{\infty} p(x) \log \frac{p(x)}{q(x)} dx \quad (7)$$

where p and q indicate probability densities of P and Q. This divergence is the measure of the difference between two probability distributions.

Now, in Eq. (6), the first two RHS terms are tractable and the third RHS term is non-negative since KL-divergence is non-negative. Therefore, maximizing $\log p_\theta(\mathbf{x}^i)$ is equivalent to maximizing the first two RHS terms of Eq. (6). Write these two terms as follows:

$$\mathcal{L}(\theta, \phi; \mathbf{x}^i) = \mathbb{E}_{q_\phi(\mathbf{h}|\mathbf{x}^i)}[\log p_\theta(\mathbf{x}^i|\mathbf{h})] - D_{KL}(q_\phi(\mathbf{h}|\mathbf{x}^i)||p_\theta(\mathbf{h})) \quad (8)$$

Eq. (8) is called the variational lower bound(ELBO) on the log likelihood $\log p_\theta(\mathbf{x}^i)$. Now, as long as the ELBO is maximized, the optimal parameters (θ^* and ϕ^*) of the VAE can be obtained. Training the VAE can use the stochastic gradient descent algorithm as well as training traditional neural networks. After training the VAE, the trained decoder $p_\theta(\mathbf{x}^i|\mathbf{h})$ is used to generate the required data.

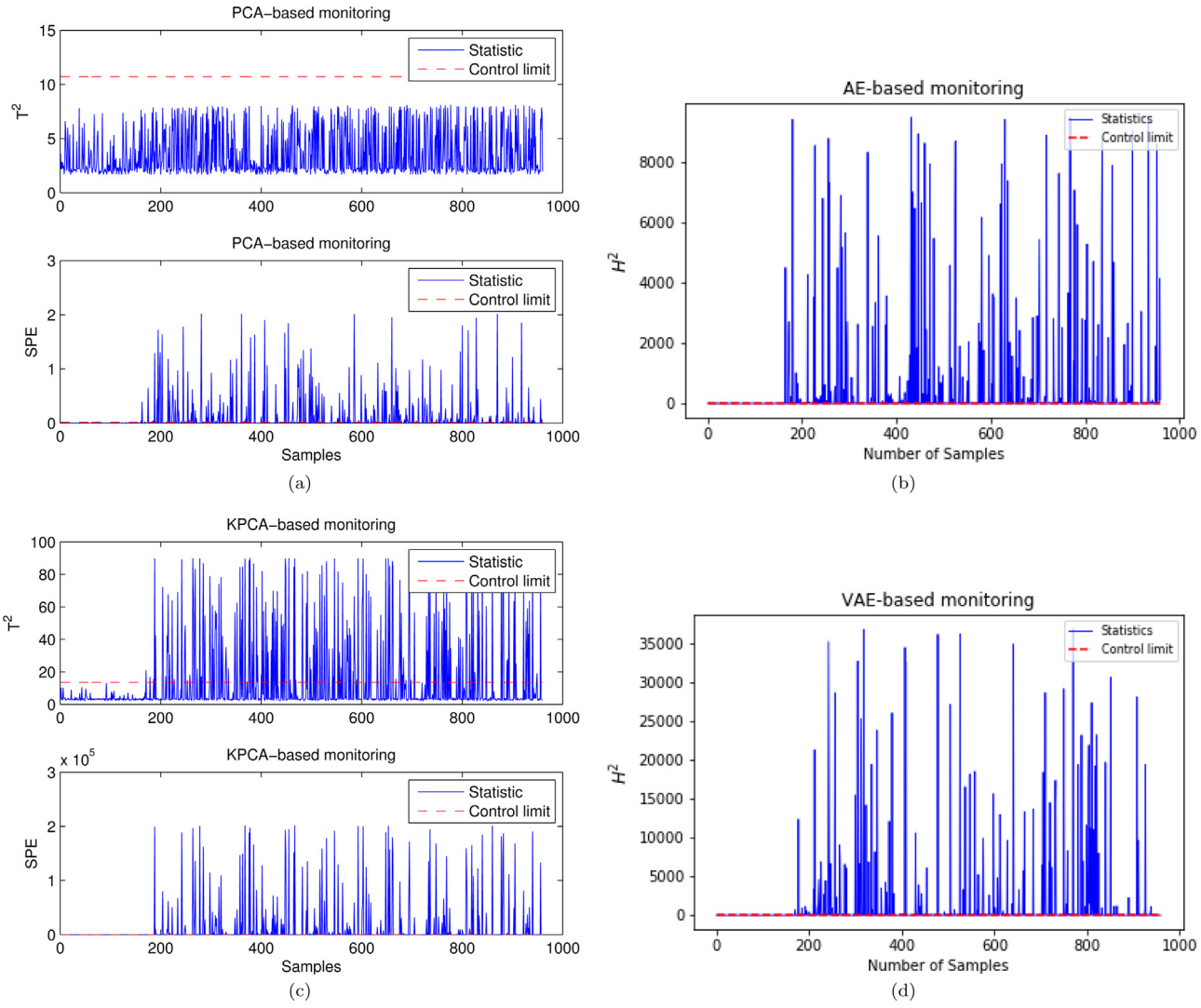

Fig. 7. Monitoring charts of fault 3 in the numerical system: (a) PCA (b) AE (c) KPCA (d) VAE.

Currently, VAEs are used to generate many complex data [35–38], but the use of VAEs to extract key Gaussian features and applying these features to corresponding applications has rarely been studied.

3.2. *Extracting key Gaussian features using VAE for nonlinear process monitoring*

As mentioned in the previous section, VAEs are heavily used to generate complex data, but are rarely used to extract key Gaussian features. But in nonlinear process monitoring, the key Gaussian features are what we most want to get. Therefore, this section reinterprets the VAE from the perspective of feature extraction, and then derives a nonlinear process monitoring algorithm based on VAE.

To facilitate comparison with previous sections, the loss function of the AE is written as a probabilistic form here:

$$\mathbf{L}(\theta, \phi; \mathbf{x}^i) = \mathbb{E}_{q_\phi(\mathbf{h}|\mathbf{x}^i)}[\log p_\theta(\mathbf{x}^i|\mathbf{h})] \tag{9}$$

where $q_\phi(\mathbf{h}|\mathbf{x}^i)$ is the encoder network and $p_\theta(\mathbf{x}^i|\mathbf{h})$ is the decoder network. By maximizing the above loss function, the required AE can be trained.

In nonlinear process monitoring, we want to get features that can reconstruct the original data and follow the Gaussian distribution. It is only necessary to add one restriction to the output (code layer) of the encoder: let the distribution of the code layer be close to the given distribution. To accomplish this, we only need to add the KL divergence regularization term to the loss function of the AE, namely:

$$\mathbf{L}(\theta, \phi; \mathbf{x}^i) = \mathbb{E}_{q_\phi(\mathbf{h}|\mathbf{x}^i)}[\log p_\theta(\mathbf{x}^i|\mathbf{h})] - D_{KL}(q_\phi(\mathbf{h}|\mathbf{x}^i)||p_\theta(\mathbf{h})) \tag{10}$$

When maximizing the above equation, the first RHS term encourages the model to reconstruct the original data, while the second RHS term makes features learned by the encoder more and more close to the given prior distribution $p_\theta(\mathbf{h})$. The goal of nonlinear process monitoring can be completed when the given distribution is a Gaussian distribution. Comparing Eqs. (8) and (10), we can find that they are exactly the same. So this kind of AE that adds some kind of distribution restriction in the feature layer is called VAE. But the starting point of these two perspectives is completely different: the original VAE is to use the decoder to generate data, and in nonlinear process monitoring is to use the encoder to extract the key Gaussian distribution features.

In this paper, the key Gaussian features extracted by the VAE are used in nonlinear process monitoring. And, the form of the prior distribution $p_\theta(\mathbf{h})$ is specified as a standard normal distribution, or in other words $p_\theta(\mathbf{h}) = Normal(0, 1)$. If the distribution of the code layer $q_\theta(\mathbf{h}|\mathbf{x})$ is different from the standard normal distribution, the

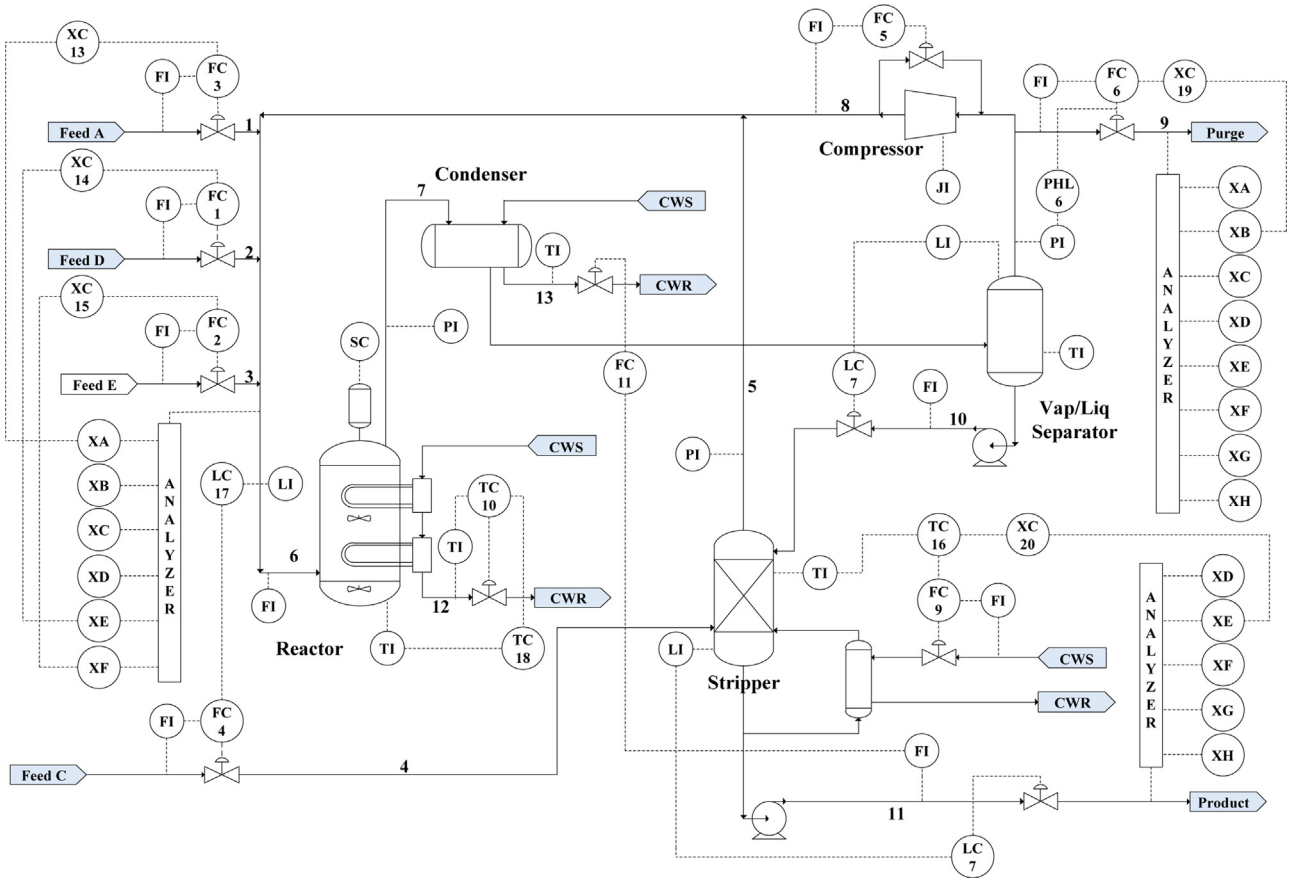

Fig. 8. Layout of the TE benchmark process.

loss will receive a penalty. Therefore, when the VAE is trained well, we can not only obtain the effective reconstruction of the raw data, but also get the useful hidden layer features that approximate the normal distribution well. Fig. 2 shows the complete architecture of the VAE model. In Fig. 2, the form of $KL[N(\mu(\mathbf{X}), \Sigma(\mathbf{X})) \| N(0, I)]$ can be computed as follows:

$$D_{KL}[N(\mu(\mathbf{X}), \Sigma(\mathbf{X}))\|N(0, I)] = \frac{1}{2}(\mathrm{tr}(\Sigma(\mathbf{X})) + \mu(\mathbf{X})^T\mu(\mathbf{X}) - k - \log\det(\Sigma(\mathbf{X}))) \quad (11)$$

where k is the dimension of the expected Gaussian distribution, $\mathrm{tr}(A)$ represents the trace of the matrix A, and $\det(A)$ represents the determinant of the matrix A. Finally, square error is used as the reconstruction loss function for the VAE, thus, the form of the loss function for the VAE can be simply written as:

$$\mathbf{L}=||\mathbf{X} - f(\mathbf{h})||^2 + \frac{1}{2}(\mathrm{tr}(\Sigma(\mathbf{X})) + \mu(\mathbf{X})^T\mu(\mathbf{X}) - k - \log\det(\Sigma(\mathbf{X}))) \quad (12)$$

Stochastic gradient descent and its variants are also adapted to train the VAE. However, to easily optimize the whole loss function, a simple reparameterization trick is used (the detail is presented in the paper [35]). As with the AE, the gradients are also obtained by the backpropagation algorithm.

3.3. Construction and control limit estimation of the monitoring statistic

In nonlinear process monitoring, as mentioned in the Introduction, the features extracted by many nonlinear methods cannot be guaranteed to satisfy the Gaussian distribution. Thus, if T^2 statistics are still used to monitor abnormal parts in those nonlinear methods, monitoring performance will be worsened. More detailed proof can be found in the literature [16].

However, as mentioned in the previous section, one of the advantages of VAE is that the learned hidden layer features follow the Gaussian distribution. Coupled with the nonlinear nature of the neural network, VAE is very suitable for nonlinear process monitoring. Similar to many process monitoring methods, the VAE model is firstly trained with normal condition data. After this, the feature space that represents the normal condition is built by mapping the normal data to hidden representation $\mathbf{h}$ through the encoder model in VAE. Since $\mathbf{h}$ follows the multivariate Gaussian distribution, the H^2 statistic can be constructed in this feature space by the following formula, which is similar to the T^2 statistic:

$$H^2 = \mathbf{h}^T\Sigma^{-1}\mathbf{h} \quad (13)$$

where Σ is the covariance matrix of the training set in the hidden representation space (also called feature space).

To detect faults, the confidence limit of the new statistic H^2 should be predefined. Note that $\mathbf{h}$ follows the Gaussian distribution, hence the confidence limit H^2_{lim} of the H^2 statistic can be easily determined by an F or χ^2 distribution for which the degrees of freedom are equal to the dimension of the hidden space [40,41]. When a new datapoint $\mathbf{x}_{new}$ arrives, $\mathbf{x}_{new}$ is firstly mapped to the hidden representation $\mathbf{h}_{new}$ through the trained encoder network. Then, the statistic H^2_{new} can be calculated by Eq. (13). If $H^2_{new} > H^2_{\mathrm{lim}}$, $\mathbf{x}_{new}$ is abnormal; otherwise, it is normal. Fig. 3 shows the flowchart of the VAE-based fault detection method.

It can be seen from the above that due to the fact that the features extracted by the VAE satisfy the Gaussian distribution, the construction of the process monitoring statistic and the estimation of the confidence limit for the corresponding statistic become

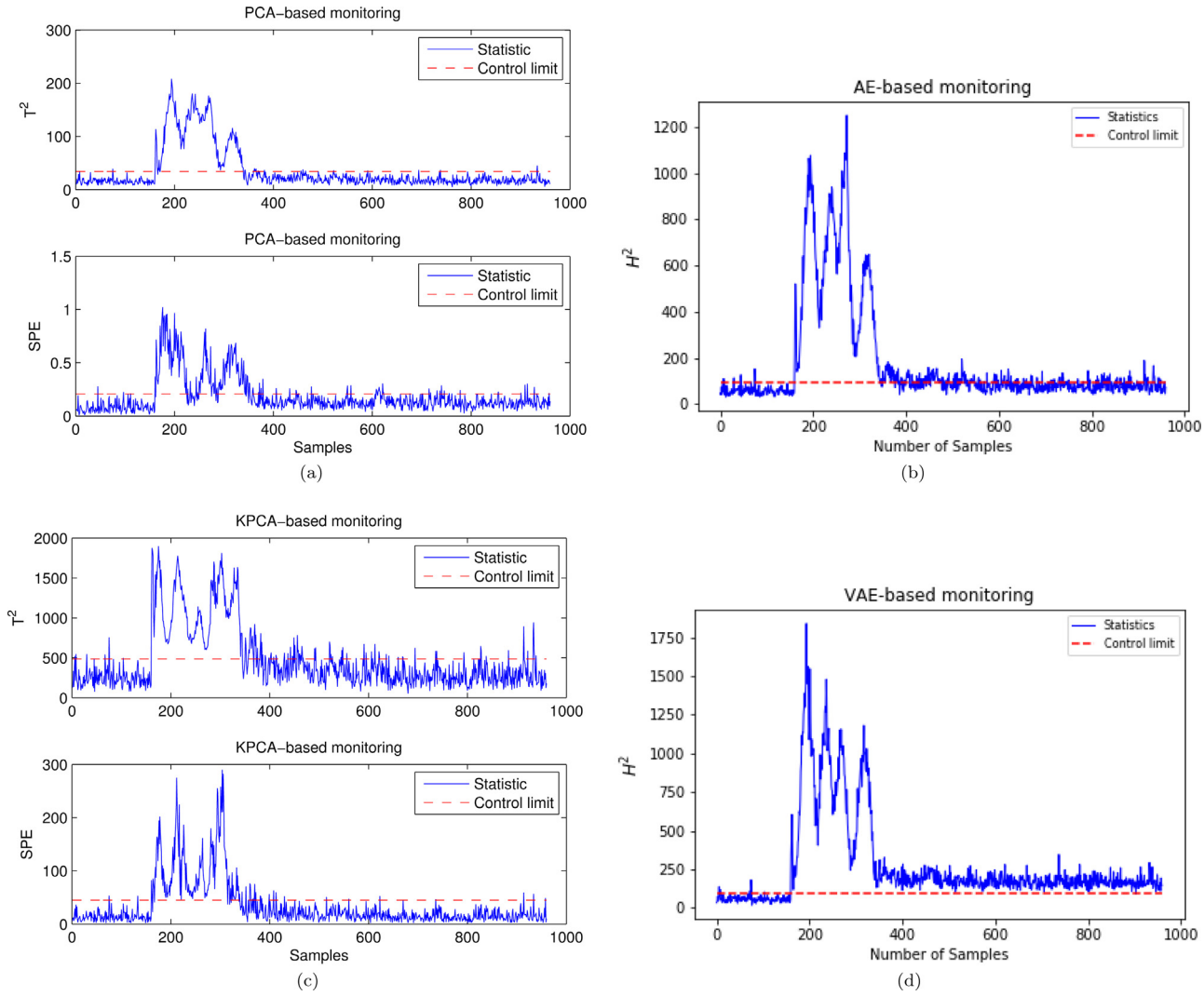

Fig. 9. Monitoring charts of fault 5 in the TE process: (a) PCA, (b) AE, (c) KPCA and (d) VAE.

simple and valid, which is of great practical significance for online monitoring.

3.4. Outline of the monitoring strategy based on VAE

This subsection gives the specific VAE-based process monitoring strategy. The whole monitoring scheme is illustrated in Fig. 4 and the concrete description in each step is as follow:

(1) Offline Modeling
- Step 1: Collect a training dataset $\mathbf{X}$ that runs under normal operating condition.
- Step 2: Normalize the training set and save the normalization parameters for online monitoring.
- Step 3: Design the architecture of VAE, tune some associated hyperparameters and train the VAE model.
- Step 4: Project $\mathbf{X}$ into the hidden space using the trained VAE.
- Step 5: Use Eq. (13) to construct the monitoring statistic H^2.
- Step 6: Determine the confidence limit H^2_{lim} of the H^2 statistic.

(2) Online Monitoring
- Step 1: Obtain the current observation $\mathbf{x}_{new}$.
- Step 2: Normalize $\mathbf{x}_{new}$ by saved normalization parameters in offline modeling.
- Step 3: Map $\mathbf{x}_{new}$ to the hidden representation $\mathbf{h}_{new}$ by the trained VAE.
- Step 4: Calculate the value of $H^2_{\mathbf{x}_{new}}$ by Eq. (13).
- Step 5: Detect whether $\mathbf{x}_{new}$ is faulty according to values of $H^2_{\mathbf{x}_{new}}$ and H^2_{lim}.

4. Case studies

In this section, two case studies are employed to verify the monitoring performance of the proposed method. The first one is a nonlinear numerical system originally suggested by Ge et al. [24]. The other one is the TE benchmark process, which has been widely used as an experiment platform for process monitoring.

4.1. Numerical system

The numerical system is a nonlinear system containing five variables that can be generated by the following equation [24]:

$$\begin{aligned} x_1 &= z + e_1, \\ x_2 &= z^2 - 3z + e_2, \\ x_3 &= -z^3 + 3z^2 + e_3, \\ x_4 &= z^4 - 4z^3 + 2z + e_4, \\ x_5 &= -2z^5 + 6z^4 - 3z^3 + z + e_5, \end{aligned} \tag{14}$$

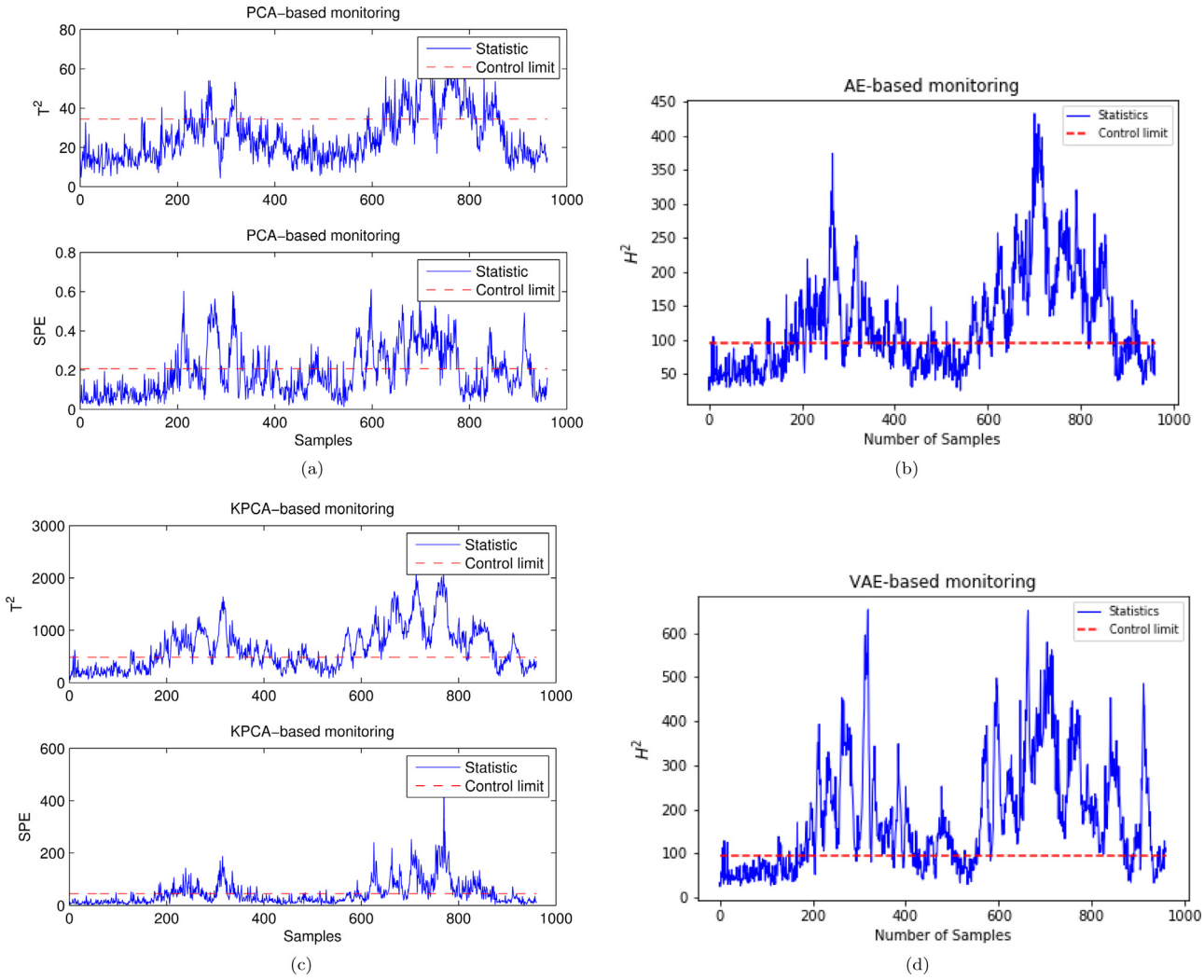

Fig. 10. Monitoring charts of fault 10 in the TE process: (a) PCA, (b) AE, (c) KPCA and (d) VAE.

Table 1
REs of VAE model with different hidden nodes in validation set.

n	2	4	6	8	10	15	20
Validation error	0.262	0.192	0.188	0.207	0.142	0.150	0.195

where z belongs to the uniform distribution of [0.01, 2], and e_1, e_2, e_3, e_4 and e_5 are independent noise variables subject to Gaussian distribution with the mean of 0 and variance of 0.01.

For the sake of comparison, in addition to the proposed method, the PCA, KPCA, and AE methods are also used. To develop the model of the proposed method, two normal datasets are first generated. One containing 500 samples is used to train the model. The other one containing 960 samples, also called validation dataset, is used to validate the trained model to tune the parameters of the model. In this case study, the reconstruction errors (REs) are used as an indicator of tuning parameters. Table 1 lists the REs of VAE with different hidden nodes (denoted as n). The smaller the REs, the better. Therefore, the number of hidden nodes is set as 10 according to the REs in the validation set and the architecture of VAE is [5, 10, 10, 10, 5]. Furthermore, some other model parameters are also listed in Table 2. For the sake of fairness, AE has the same parameters as VAE. Meanwhile, the bandwidth parameter of the KPCA is set to 25 (five times the data dimension). The process statistic and confidence limit for AE are constructed similarly to VAE.

To evaluate the monitoring performance of the proposed method, three additional fault datasets are provided, which all consist of 960 data samples and are generated separately as follows:

- a step change of x_3 by 1 is introduced starting from sample 161;
- a ramp change of x_5 by $0.05(k-160)$ is added to each sample between sample 161 and 960, in which k is the sample number;
- a step change of z by 0.5 is introduced starting from sample 161.

Fault detection rates (FDRs) and detection delays (FDDs) of the four methods for three faults are compared in Table 3. Figs. 5–7 show the monitoring results of four methods for three faults. All three types of faults have occurred since the 161st sample. Therefore, starting from the 161st sample, if the value of the monitoring statistic is greater than the corresponding control limit, the fault is successfully detected, otherwise, the fault detection fails. From the monitoring results in Table 3 and Fig. 5–7, it can be seen that KPCA, AE and VAE all have achieved good monitoring results for the first two faults because most of the fault points have been detected. In both cases, VAE achieves the best results with the lowest detection delay. For fault 3, although the test results of the four methods are unsatisfactory, the VAE still has the highest fault detection rate. In

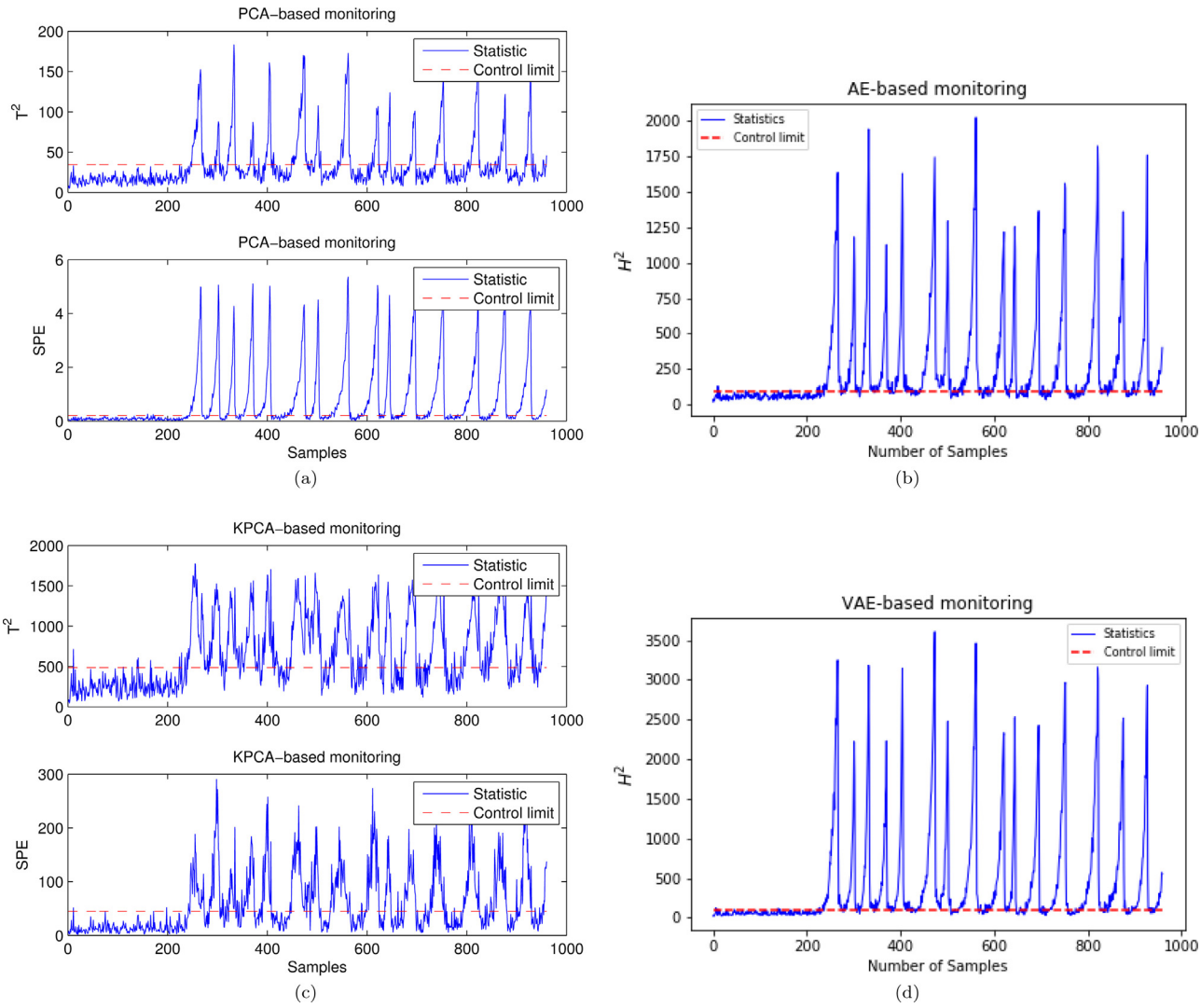

Fig. 11. Monitoring charts of fault 20 in the TE process: (a) PCA, (b) AE, (c) KPCA and (d) VAE.

Table 2
Parameters for VAE and AE.

Algorithm	n	Learning rate	Number of epochs	Batch size	Activation	Optimizer
VAE	10	0.0001	50	10	Relu	Adam
AE	10	0.0001	50	10	Relu	Adam

Table 3
Fault detection rates and detection delays (number of samples) of three faults in the nonlinear numerical system. The good performances are highlighted in bold.

Fault No.	PCA		KPCA		AE	VAE
	T^2	SPE	T^2	SPE	H^2	H^2
1	0.000/–	0.337/43	0.061/–	0.995/1	1.000/0	**1.000/0**
2	0.000/–	0.408/10	0.027/–	0.982/12	0.985/10	**0.9875/2**
3	0.000/–	0.268/1	0.262/11	0.283/9	0.273/5	**0.295/2**

addition, it is clear from Table 3 that KPCA's T^2 monitoring results of the first two faults are very bad and the AE monitoring result of the third fault is significantly lower than the VAE method. This is because the features extracted by KPCA and AE violate the Gaussian distribution assumption. In contrast, the features extracted by VAE are closer to the Gaussian distribution, which gives it the best monitoring results. In this regard, Section 5 will conduct further analysis. In summary, VAE-based fault detection shows its superiority both in detection rate and sensitivity.

4.2. Tennessee Eastman benchmark process

The Tennessee Eastman (TE) process has been widely used as a benchmark simulation platform for examining the performance of the various monitoring algorithms [42,43]. This process was first proposed by Downs and Vogel [44] and reconstructed by Lyman and Georgakis [45] later. Fig. 8 shows the control structure of the TE process schematically that was listed in [45]. The simulation data used in this study come from the TE process. These data include

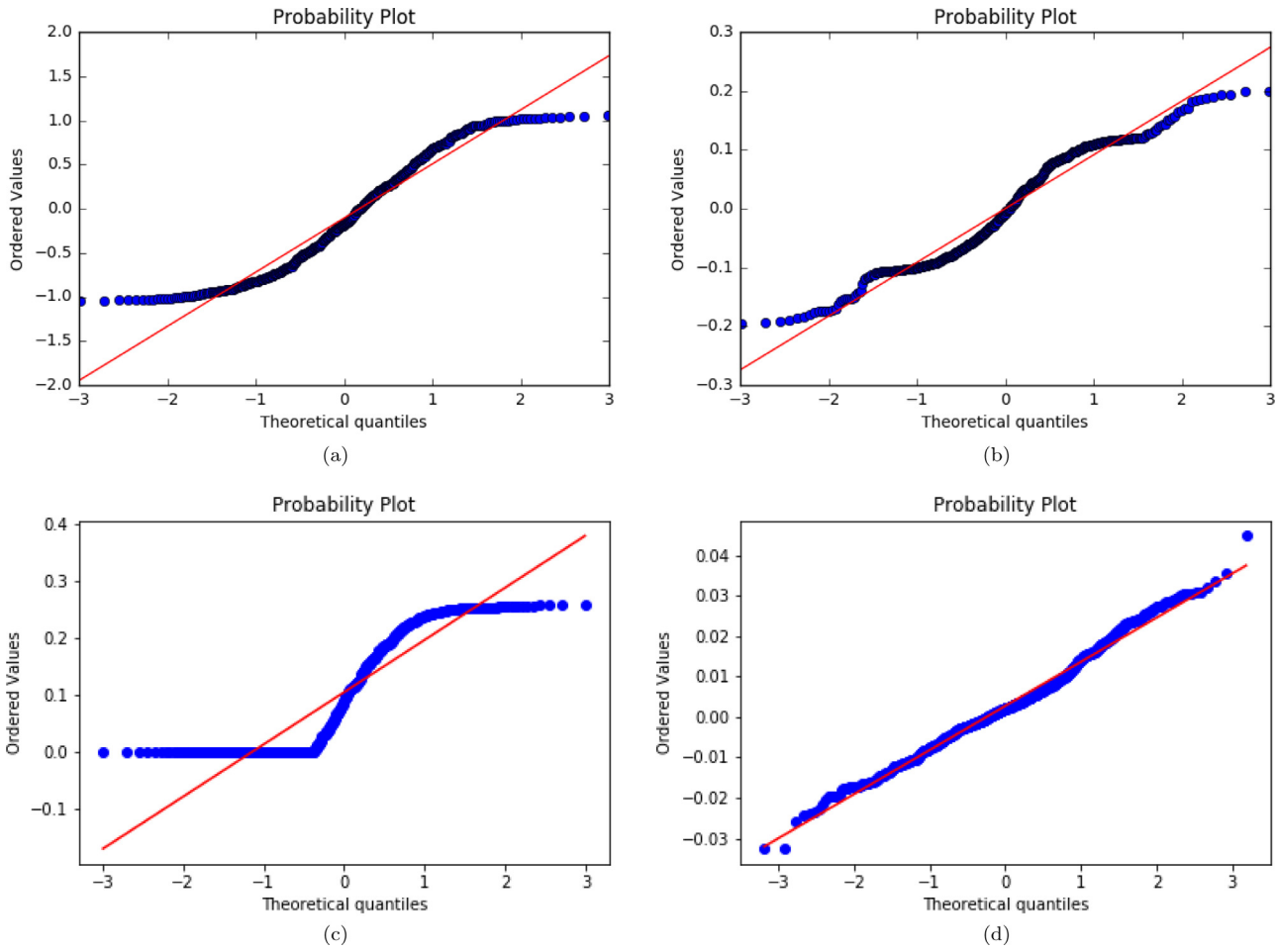

Fig. 12. Normal probability plots for the normal data in the numerical system: (a) PCA (b) KPCA (c) AE (d) VAE.

Table 4
Monitoring variables in TE process.

No.	Measurements	No.	Measurements
1	A feed	18	Stripper temperature
2	D feed	19	Stripper steam flow
3	E feed	20	Compressor work
4	Total feed	21	Reactor cooling water outlet temperature
5	Recycle flow	22	Separator cooling water outlet temperature
6	Reactor feed rate	23	D feed flow valve
7	Reactor pressure	24	E feed flow valve
8	Reactor level	25	A feed flow valve
9	Reactor temperature	26	Total feed flow valve
10	Purge rate	27	Compressor recycle valve
11	Product separator temperature	28	Purge valve
12	Product separator level	29	Separator pot liquid flow valve
13	Product separator pressure	30	Stripper liquid product flow valve
14	Product separator underflow	31	Stripper steam valve
15	Stripper level	32	Reactor cooling water flow
16	Stripper pressure	33	Condenser cooling water flow
17	Stripper underflow		

two normal datasets for training and validating models, and 21 fault sets for testing performance of the trained models [46]. There are 22 continuous measurement variables, 19 composition measurement variables and 12 manipulated variables. In this study, 11 manipulated and all 22 continuous variables are selected to conduct process monitoring. Table 4 lists the details of the above variables. Moreover, Table 5 lists 21 simulated faults.

In this case study, three methods including PCA, KPCA, and AE are constructed to compare with the VAE-based method to verify the effectiveness of the proposed method. Six principal components are selected in PCA model. The bandwidth parameter in KPCA model is chosen as $5m$, where m is the dimension of the input data, i.e., $m = 33$. The AE's monitoring statistic and confidence limit are constructed and calculated in the same way as the VAE uses. All four methods first develop the models in the training set and then test the monitoring performance in the fault sets. The training set that we use contains 960 samples, and another validation set containing 500 samples is used to tune the parameters of the proposed method. Table 6 shows the REs of the VAE method using the different number of hidden nodes (denoted as

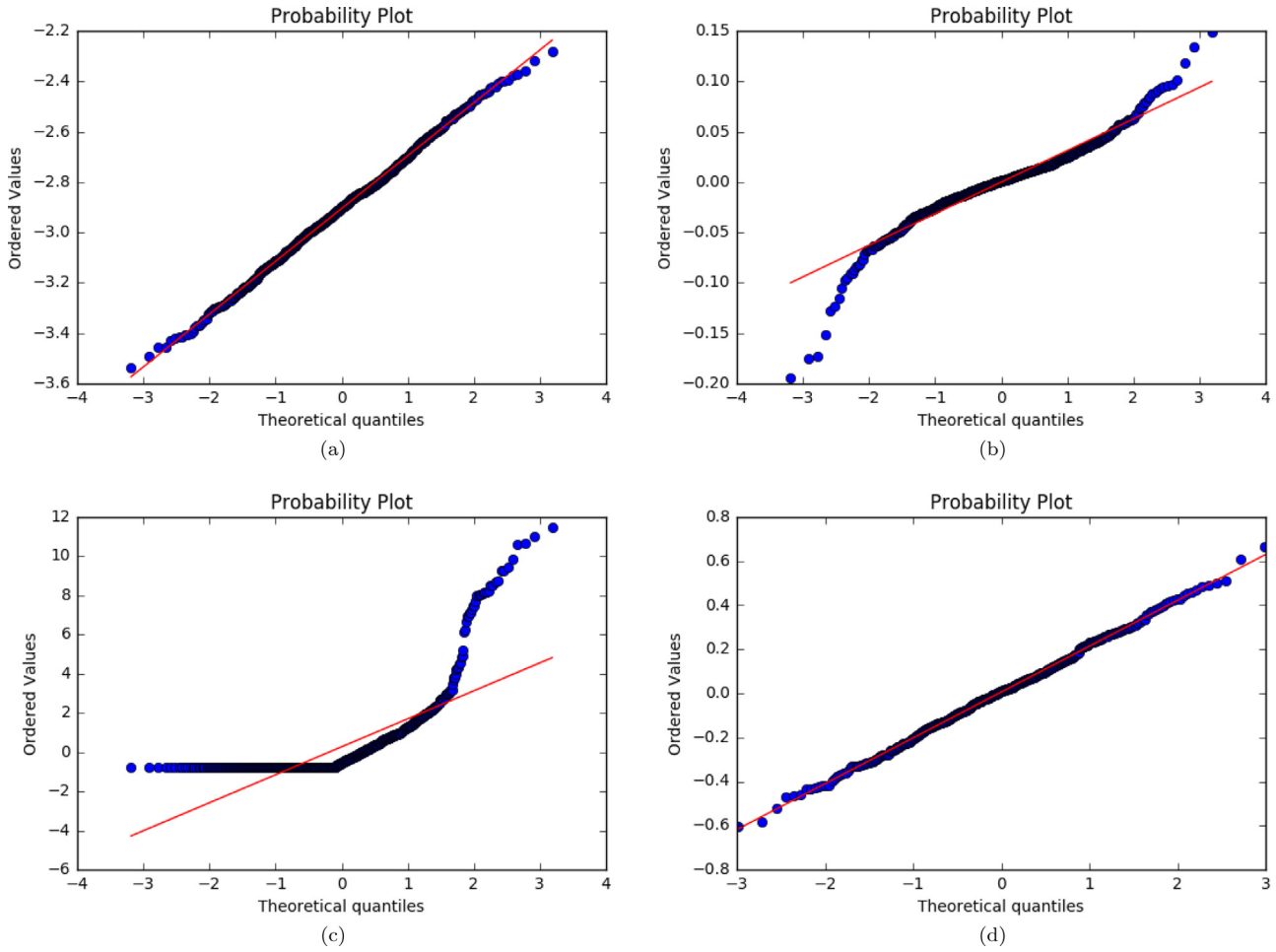

Fig. 13. Normal probability plots for the normal data in the TE process: (a) PCA, (b) KPCA, (c) AE and (d) VAE.

Table 5
Process faults in TE process.

Fault No.	Process variable	Type
1	A/C feed ratio, B composition constant (stream 4)	Step
2	B composition, A/C ratio constant (stream 4)	Step
3	D feed temperature (stream 2)	Step
4	Reactor cooling water inlet temperature	Step
5	Condenser cooling water inlet temperature	Step
6	A feed loss (stream 1)	Step
7	C header pressure loss-reduced availability (stream 4)	Step
8	A, B, C feed composition (stream 4)	Random variation
9	D feed temperature (stream 2)	Random variation
10	C feed temperature (stream 4)	Random variation
11	Reactor cooling water inlet temperature	Random variation
12	Condenser cooling water inlet temperature	Random variation
13	Reaction kinetics	Slow drift
14	Reactor cooling water valve	Sticking
15	Condenser cooling water valve	Sticking
16	Unknown	Unknown
17	Unknown	Unknown
18	Unknown	Unknown
19	Unknown	Unknown
20	Unknown	Unknown
21	The valve for stream 4 was fixed at the steady state position	Step constant position

n) in the validation set. From the table, 60 corresponding to the smallest validating error is selected as the number of hidden layers for our model and the architecture of VAE is [33, 60, 60, 60, 33]. Some other parameters used in VAE and AE are also listed in Table 7.

After developing models, two assessment criteria are used to measure the monitoring performance including fault detection rates (FDRs) and fault detection delays (FDDs). The FDRs of four methods for all faults in the TE process are listed in Table 8. For some small fault types, such as 3, 9, and 15, whose monitoring vari-

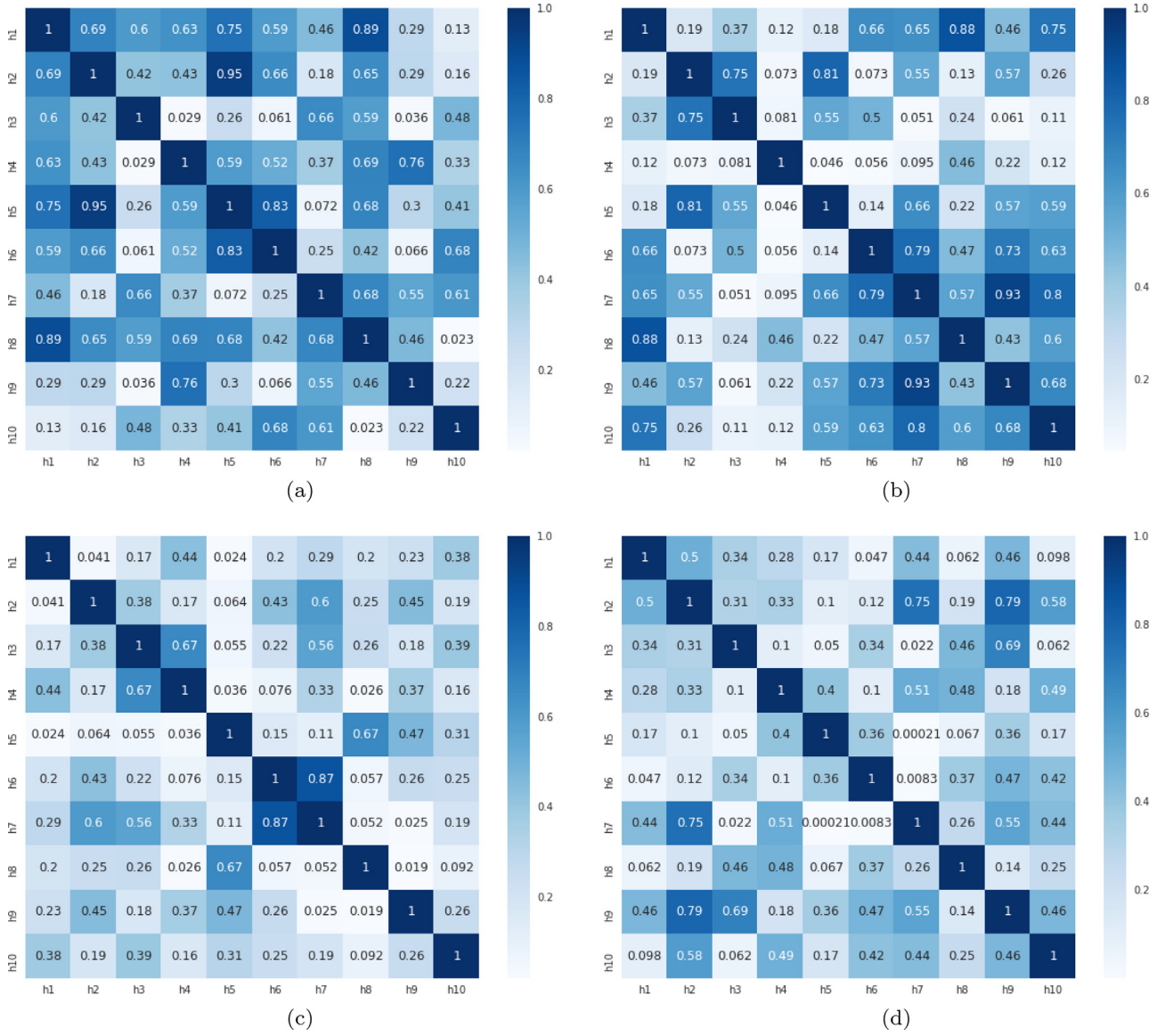

Fig. 14. Heat maps of correlation coefficients among 10 features extracted by VAE in the numerical system: (a) VAE1, (b) VAE2, (c) VAE3 and (d) VAE4.

Table 6
REs of VAE model with different hidden nodes in validation set.

n	Validating error	n	Validating error
20	0.5078	80	0.4115
30	0.4590	90	0.4249
40	0.4299	100	0.4211
50	0.4187	150	0.4456
60	0.4110	200	0.4728
70	0.4142		

Table 7
Parameters for VAE and AE.

Algorithm	n	Learning rate	Number of epochs	Batch size	Activation	Optimizer
VAE	60	0.0001	110	20	Relu	Adam
AE	60	0.0001	110	20	Relu	Adam

ables are applied with very little perturbation [21], almost all four methods can not detect these faults because their FDRs are very low. However, the VAE-based method gives the highest FDRs on detecting these three faults compared with the other three methods. When monitoring those easily detected 1, 2, 4, 6, 7, 8, 12, 13, 14, 17 and 18 faults, all four methods show similar detection results. The detection results based on the VAE method have been significantly improved in faults 5, 10, 16, 19, 20 and 21. The FDRs of VAE are much higher than the other three methods for these faults. Table 9 shows the FDDs for the TE process. The detection delay reflects the sensitivity of the monitoring method. It can be seen from Table 9 that VAE's FDDs are almost very low and even many FDDs are zero. This shows that the VAE-based method can detect faults quickly, which has a great effect on online fault warning. Overall, VAE provides a very good monitoring results and has a very fast detection capability.

In order to demonstrate the effectiveness of the proposed method more concretely, three representative faults 5, 10, and 20 are selected for further explanation.

Fault 5 involves a step change in the condenser cooling water inlet temperature, which results in a step change in the condenser cooling water flowrate. The difficulty of fault 5 is that the temperature in the separator will return to the set value because the step change can be compensated by the control loop after a period of time. Fig. 9 shows the monitoring results of four methods for

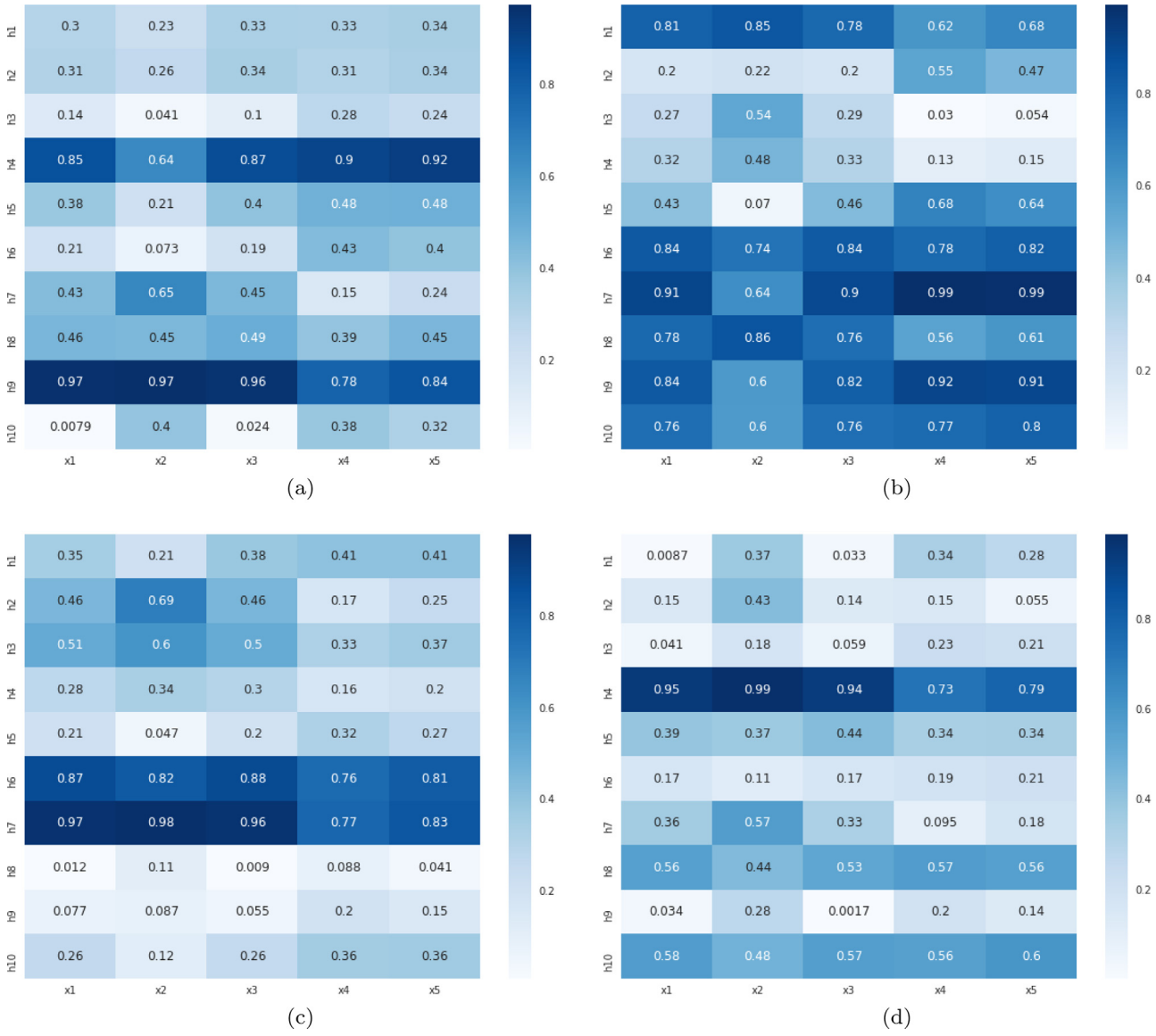

Fig. 15. Heat maps of correlation coefficients between 10 features and original 5 variables in the numerical system: (a) VAE1, (b) VAE2, (c) VAE3 and (d) VAE4.

this fault. It can be clearly seen from the figure that the values of the monitoring statistics for all methods have obviously increased when the fault has just occurred. After a period time, the statistics suddenly dropped dramatically. From this moment on, PCA, KPCA, and AE can no longer detect faults. However, the VAE is still able to detect the fault until the last moment as the values of the statistic H^2 for VAE are almost all above the confidence limit.

Fault 10 is related to a random change in the temperature of stream 4. The monitoring results of this fault are presented in Fig. 10. It can be seen from the figure that after the fault occurs, the differences of the detection results for the four methods come from the sample 350 to 650 as well as the last samples of the process. In the above two intervals, the statistics of the PCA, KPCA, and AE are all basically under the corresponding control limits. Conversely, as shown in Fig. 10(d), the values of the H^2 statistic for VAE are mostly above the control limit in the above interval. This is also the reason that VAE is better than the other three methods. At the same time, the detection delay of the VAE is also very low indicating that there is also a quick response to the Fault 10.

Fault 20 is an unknown type. Fig. 11 presents the monitoring charts of four methods. Fig. 11 shows that the statistics for all methods have similar upward and downward oscillations. When the values of the statistics are at the bottom of the shock, it is difficult to detect faults in the PCA, KPCA and AE methods. However, at these points, VAE detects most of the failures, which leads to a higher detection rate of VAE than the other methods.

5. Discussion

This section discusses the results of simulation experiments for the proposed method in more detail. First, the verification process of the Gaussian distribution of features extracted by the VAE is performed. We then explore the relationship among features themselves and the relationship between features and original variables. Finally, a preliminary study on fault identification based on VAE is given.

5.1. Verification of Gaussian distributions

From the principle of VAE, it is known that the features learned by VAE follow the Gaussian distribution, thus making subsequent process monitoring easier. To verify whether the features learned

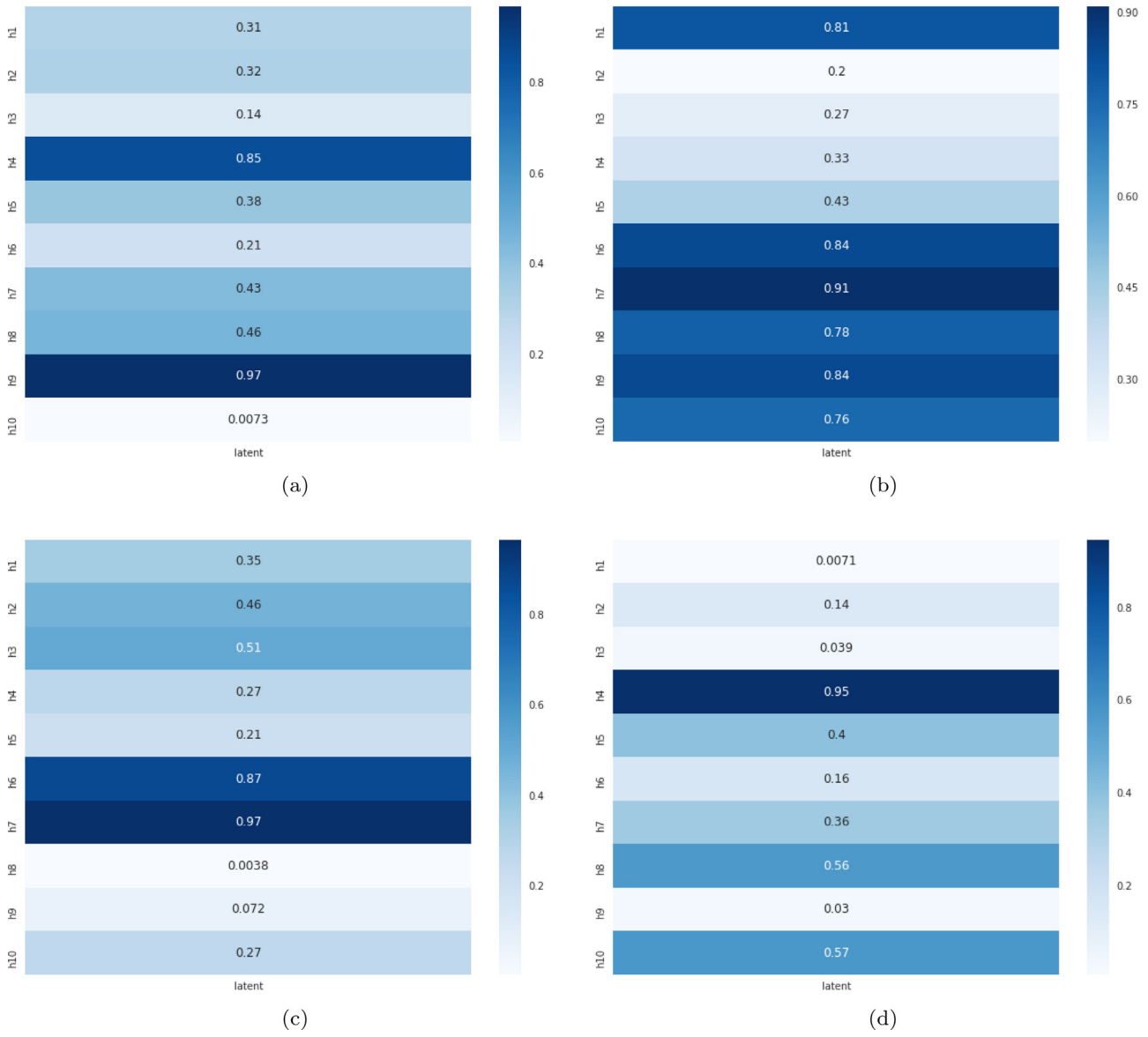

Fig. 16. Heat maps of correlation coefficients between features extracted by VAE in the numerical system: (a) VAE1, (b) VAE2, (c) VAE3 and (d) VAE4.

by the VAE are close to the Gaussian distribution, the normal data of the Numerical system and the TE process are both used to illustrate it. First, for comparison, the trained models of PCA, KPCA, AE, and VAE project the normal data to feature spaces, respectively. Then, one of the features in the feature space of the above models is individually selected to plot the normal probability plot. Normal probability plots are conducted to test the normality of features learned by above models. If data are subject to the Gaussian distribution, the resulting points will be well attached to the $y = x$ line. Figs. 12 and 13 show normal probability plots for the Numerical system and the TE process, respectively. As can be seen from Figs. 12 and 13, features learned by the VAE are basically attached to the line $y = x$, which proves that features learned by the VAE are subject to the Gaussian distribution.

5.2. Discussion on the correlation of features

Exploring the physical meaning of deep learning has always attracted the attention of the academic community. The high degree of nonlinearity of neural networks makes it difficult to study the specific physical interpretation of neural networks, which also hinders the development of deep learning physical structure research. This section is only a preliminary study for features extracted by the VAE. The actual physical meaning will be the focus of our future researches.

To explore the relationship among features extracted by the VAE themselves and the relationship between extracted features and original variables, we use Pearson correlation coefficient [47] as the reference index. The larger the absolute value of the correlation coefficient, the more relevant the two variables are. Since the number of variables and features in the TE process is too large, the Numerical system is used as the research object in this section.

First, the relationship among 10 features extracted by the VAE is explored. Fig. 14 shows heat maps of the absolute value of the correlation coefficient among 10 features. In Fig. 14, four VAE models are trained using the normal data in the Numerical system. The reason for training four models instead of just one is that when training the model of the VAE, each training process will not get the same

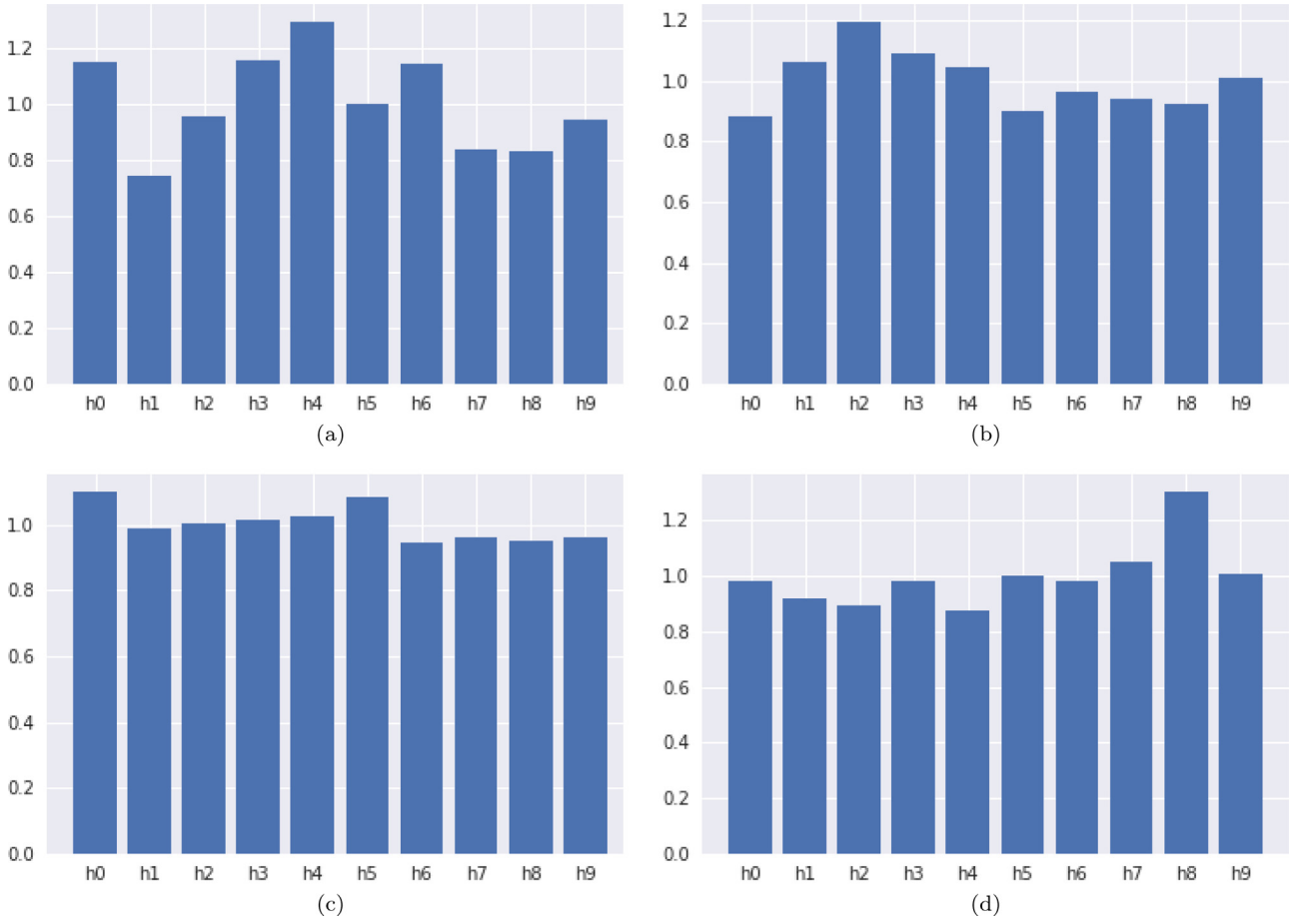

Fig. 17. Contributions based on PDC index for normal data in the numerical system: (a) VAE1, (b) VAE2, (c) VAE3 and (d) VAE4.

Table 8
Fault detection rates for 21 faults in the TE process. The good performances are highlighted in bold.

Fault No.	PCA		KPCA		AE	VAE
	T^2	*SPE*	T^2	*SPE*	H^2	H^2
1	0.992	0.997	1.000	0.416	1.000	0.996
2	0.960	0.985	0.987	0.030	0.987	0.986
3	0.004	0.007	0.135	0.026	0.141	**0.176**
4	0.043	0.969	1.000	0.973	0.999	0.996
5	0.236	0.177	0.336	0.247	0.495	**0.999**
6	0.992	1.000	1.000	0.018	1.000	**1.000**
7	0.383	1.000	1.000	0.801	1.000	**1.000**
8	0.926	0.955	0.981	0.609	0.983	**0.983**
9	0.002	0.014	0.114	0.029	0.113	**0.163**
10	0.316	0.166	0.682	0.406	0.643	**0.813**
11	0.161	0.658	0.826	0.701	0.803	0.800
12	0.934	0.950	0.995	0.601	0.993	0.994
13	0.929	0.949	0.955	0.264	0.955	**0.956**
14	0.875	1.000	1.000	0.788	1.000	**1.000**
15	0.001	0.011	0.152	0.006	0.145	**0.178**
16	0.150	0.153	0.685	0.470	0.608	**0.801**
17	0.751	0.891	0.962	0.438	0.958	0.944
18	0.879	0.899	0.910	0.004	0.911	**0.915**
19	0.004	0.209	0.498	0.205	0.425	**0.758**
20	0.265	0.439	0.657	0.516	0.656	**0.734**
21	0.269	0.439	0.568	0.445	0.526	**0.578**
Average	0.481	0.612	0.735	0.385	0.731	**0.800**

Table 9
Fault detection delays (number of samples) for 21 faults in the TE process. The good performances are highlighted in bold.

Fault No.	PCA		KPCA		AE	VAE
	T^2	*SPE*	T^2	*SPE*	H^2	H^2
1	7	3	1	2	0	3
2	26	13	5	12	4	11
3	46	98	15	18	14	**14**
4	1	1	1	1	0	**0**
5	4	1	1	1	0	**0**
6	7	1	1	1	0	**0**
7	1	1	1	1	0	**0**
8	27	21	10	3	6	14
9	3	1	1	12	0	**0**
10	16	8	6	15	5	**4**
11	7	6	6	7	5	**5**
12	8	3	3	3	2	**2**
13	47	39	26	38	25	**7**
14	2	1	1	1	0	**0**
15	578	301	57	67	56	**56**
16	174	22	2	17	1	**1**
17	26	20	1	1	0	**0**
18	96	61	13	15	14	14
19	208	11	11	11	7	**1**
20	81	68	6	38	5	**5**
21	257	251	21	41	39	**20**

model. For the sake of fairness, four models are trained separately and fault detection effects of four models are not much different. It can be seen from Fig. 14: (1) the model of each training process is different; (2) each feature extracted by VAE has some relationships with the other features. For example, correlation coefficients of feature 1 and feature 2, 3, 4, 5, 6, and 8 in Fig. 14(a) all exceed 0.5.

Although the above study shows that there are correlations among extracted features, this does not mean that the 10 features

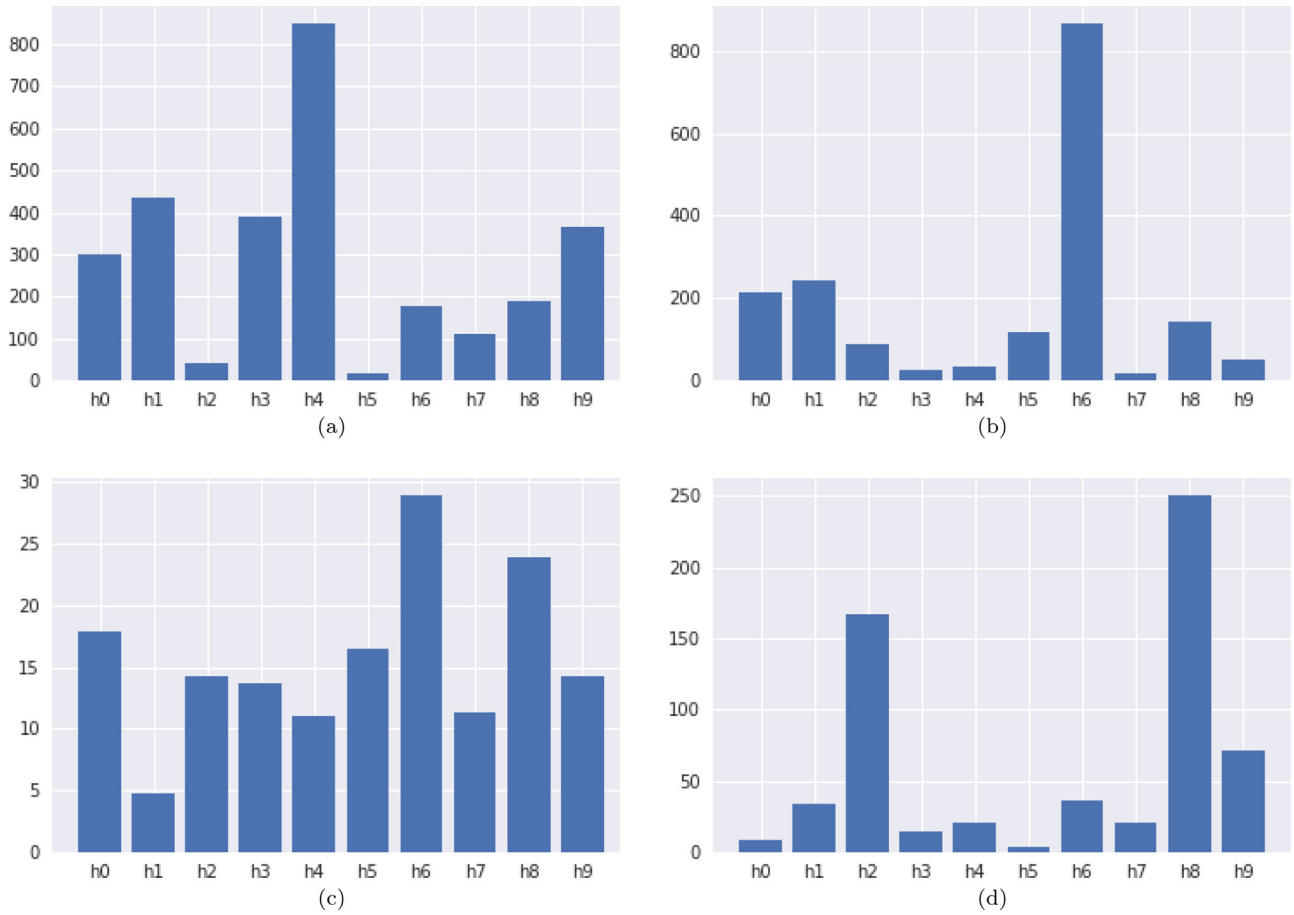

Fig. 18. Contributions based on DC index for normal data in the numerical system: (a) VAE1, (b) VAE2, (c) VAE3 and (d) VAE4.

are redundant. This can be explained from the following: Firstly, extracted features of VAE are approximate Gaussian distribution, which means that we want to use multiple Gaussian distributions to represent the original data space through a nonlinear function (decoder). To achieve this, two steps need to be completed: one is to obtain key features that can reconstruct data, and the other is to make features follow Gaussian distributions. Therefore, in order to satisfy both of the above two conditions, it is common for the number of features (the number of Gaussian distributions) learned by VAE to be greater than the number of original variables. Secondly, the restriction we have added is that the distribution of each feature is as close as possible to the standard Gaussian distribution, so it is inevitable that there will be a certain correlation between features. However, although it is found that the features are related to each other, it is still difficult to express this relationship into the exact mathematical expression. At the same time, extracting independent key and Gaussian-distributed features is one of the future research directions.

Next, the relationship between extracted features and original variables is also studied. Fig. 15 shows heat maps of the correlation coefficient between 10 features and original 5 variables; Fig. 16 shows the relationship between 10 features and the initial latent variable. Similarly, the above four VAE models are also used in this study. As can be seen from Fig. 15, each feature has some relationships with some original variables, and this point is also illustrated in Fig. 16, in which each feature has relationship with the latent variable. The above phenomenon is also easy to understand: these 10 features are obtained from the original 5 variables through multiple weighted and nonlinear transformations, and the original 5 variables are obtained by nonlinear transformation of the latent variable, so there must be some correlations among them.

Although we try to study the relationship between features and original variables through correlation coefficients, the specific physical meanings and exact mathematical expressions are still difficult to find. Of course, due to the highly nonlinear structure of the neural network, the specific physical meaning of features extracted by neural networks and the exact mathematical relationship between features and original variables are still difficult problems in the world. The authors have also tried their best to study the specific physical meaning of neural networks, but the current progress is still not ideal.

5.3. Research on VAE-based fault identification

In this section, we have done some researches on VAE-based fault identification. Similarly, for the convenience of this study, we still use the numerical system.

Since $\mathbf{h}=f(\mathbf{x})$, f is a highly nonlinear function fitted by a neural network, so the exact mathematical expression of f is not known, and some fault identification indices such as the complete decomposition contribution(CDC) and the reconstruction-based contribution (RBC) are temporarily unable to be constructed. But we have made some attempts to fault identification, and these attempts are currently conducted at the feature level. According to Ref. [48], we can construct two contribution graphs of the partial

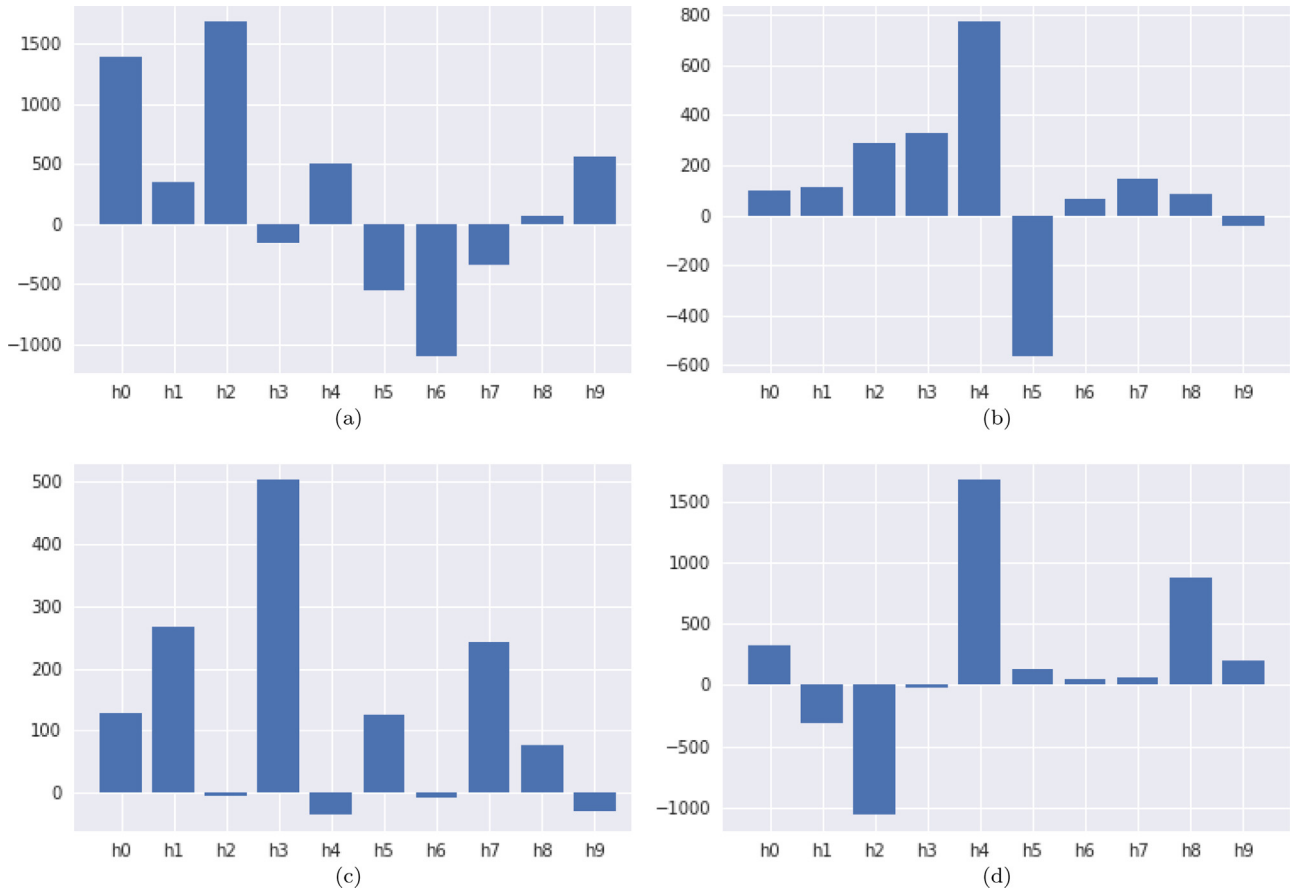

Fig. 19. Contributions based on PDC index for fault 1 in the numerical system: (a) VAE1, (b) VAE2, (c) VAE3 and (d) VAE4.

decomposition contribution (PDC) and the diagonal contributions (DC). Specific expressions of these two indices based on VAE are as follows:

$$\begin{aligned} PDC_i^{H^2} &= \mathbf{h}^T \Sigma^{-1} \xi_i \xi_i^T \mathbf{h}, \\ DC_i^{H^2} &= \mathbf{h}^T \xi_i \xi_i^T \Sigma^{-1} \xi_i \xi_i^T \mathbf{h}, \\ \xi_i &= [0, 0, \ldots, 1, \ldots, 0]^T, \\ i &\in [1, n], \end{aligned} \tag{15}$$

where n is the number of features. According to Ref. [48], when no faults are present, all variable contributions should have statistically the same mean. So we first draw contributions of two indices on the normal data, as shown in Figs. 17 and 18 . As before, we also use four VAE models.

As can be seen from Figs. 17 and 18, in the normal data set, contributions based on the PDC index of all 10 features are roughly the same, but contributions based on the DC index are obviously different. Thus, we adapted the PDC index to identify the fault 1 and fault 2 in the Numerical system. Figs. 19 and 20 show PDC-based contribution plots for fault 1 and fault 2 for four VAEs, respectively. It can be seen from two figures that different faults have different effects on different features when the fault occurs. If we know the exact relationship between the feature and the original variable at this time, then we can infer the specific contribution of the original variable to the fault, and thus identify the fault. However, the physical interpretation of neural networks is still at a very early stage, which still requires us to do a lot of efforts to solve this problem in the future.

In summary, the VAE-based nonlinear process monitoring method can extract the key features of Gaussian distribution and provide an effective and simple method for the process monitoring field, which not only has important significance in this field, but also has practical value.

6. Conclusion

In this paper, a novel nonlinear process monitoring method based on VAE is proposed to address the Gaussian assumption problem. Different from the traditional AE that cannot guarantee that the features satisfy Gaussian distribution, VAE makes the feature representations follow the Gaussian distribution by adding a K-L divergence to hidden layer output. Therefore, VAE is adapted and trained in normal data to extract key Gaussian features in the proposed method. Based on these features, a new statistic H^2 is constructed. Then, the control limit of the H^2 is easily determined by a χ^2 distribution. Two case studies including a nonlinear numerical system and the TE benchmark process have been applied to evaluate the superior performance of the proposed VAE-based method over the PCA, KPCA, and AE approaches. In the future, one of our works is to combine VAE with denoising and sparsity to extract more robust and effective features. At the same time, extending VAE to the monitoring of batch processes and dynamic processes is also an important part of our future work.

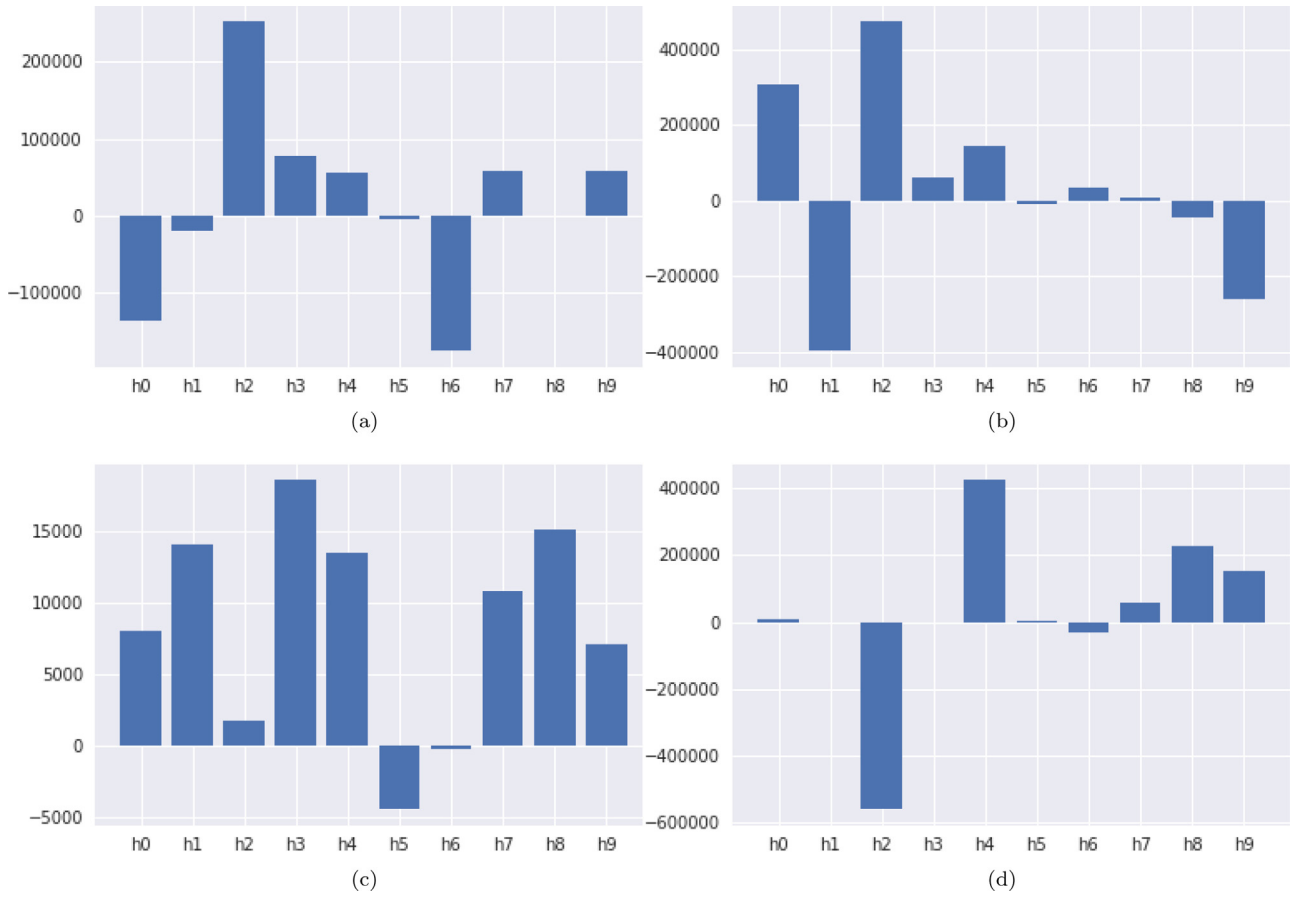

Fig. 20. Contributions based on PDC index for fault 2 in the numerical system: (a) VAE1, (b) VAE2, (c) VAE3 and (d) VAE4.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (51777122).

References

[1] S.J. Qin, Survey on data-driven industrial process monitoring and diagnosis, Annu. Rev. Control 36 (2) (2012) 220–234.
[2] Z. Ge, Z. Song, F. Gao, Review of recent research on data-based process monitoring, Ind. Eng. Chem. Res. 52 (10) (2013) 3543–3562.
[3] S. Yin, S.X. Ding, A. Haghani, H. Hao, P. Zhang, A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process, J. Process Control 22 (9) (2012) 1567–1581.
[4] S. Joe Qin, Statistical process monitoring: basics and beyond, J. Chemom. 17 (8–9) (2003) 480–502.
[5] E.L. Russell, L.H.C. Ms, R.D. Braatz, Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes, Springer, London, 2000.
[6] S. Yin, S.X. Ding, X. Xie, H. Luo, A review on basic data-driven approaches for industrial process monitoring, IEEE Trans. Ind. Electron. 61 (11) (2014) 6418–6428.
[7] Z. Ge, Z. Song, S.X. Ding, B. Huang, Data mining and analytics in the process industry: the role of machine learning, IEEE Access 5 (2017) 20590–20616.
[8] C. Aldrich, L. Auret, Overview of process fault diagnosis, in: Unsupervised Process Monitoring and Fault Diagnosis with Machine Learning Methods, Springer, 2013, pp. 17–70.
[9] Z. Ge, Review on data-driven modeling and monitoring for plant-wide industrial processes, Chemomet. Intell. Lab. Syst. 171 (2017) 16–25.
[10] J. Huang, X. Yan, Gaussian and non-Gaussian double subspace statistical process monitoring based on principal component analysis and independent component analysis, Ind. Eng. Chem. Res. 54 (3) (2015) 1015–1027.
[11] Q. Jiang, X. Yan, Just-in-time reorganized PCA integrated with SVDD for chemical process monitoring, AIChE J. 60 (3) (2014) 949–965.
[12] M.A. Kramer, Nonlinear principal component analysis using autoassociative neural networks, AIChE J. 37 (2) (1991) 233–243.
[13] D. Dong, T.J. McAvoy, Nonlinear principal component analysis-based on principal curves and neural networks, Comput. Chem. Eng. 20 (1) (1996) 65–78.
[14] Z. Geng, Q. Zhu, Multiscale nonlinear principal component analysis (NLPCA) and its application for chemical process monitoring, Ind. Eng. Chem. Res. 44 (10) (2005) 3585–3593.
[15] J.-M. Lee, C. Yoo, S.W. Choi, P.A. Vanrolleghem, I.-B. Lee, Nonlinear process monitoring using kernel principal component analysis, Chem. Eng. Sci. 59 (1) (2004) 223–234.
[16] Z. Ge, C. Yang, Z. Song, Improved kernel PCA-based monitoring approach for nonlinear processes, Chem. Eng. Sci. 64 (9) (2009) 2245–2255.
[17] C.F. Alcala, S.J. Qin, Reconstruction-based contribution for process monitoring with kernel principal component analysis, Ind. Eng. Chem. Res. 49 (17) (2010) 7849–7857.
[18] C.-Y. Cheng, C.-C. Hsu, M.-C. Chen, Adaptive kernel principal component analysis (KPCA) for monitoring small disturbances of nonlinear processes, Ind. Eng. Chem. Res. 49 (5) (2010) 2254–2262.
[19] Q. Jiang, X. Yan, Weighted kernel principal component analysis based on probability density estimation and moving window and its application in nonlinear chemical process monitoring, Chemom. Intell. Lab. Syst. 127 (2013) 121–131.
[20] G.R. Ji-Dong Shao, Nonlinear process monitoring based on maximum variance unfolding projections, Expert Syst. Appl. 36 (8) (2009) 11332–11340.
[21] L. Luo, S. Bao, J. Mao, D. Tang, Nonlinear process monitoring using data-dependent kernel global–local preserving projections, Ind. Eng. Chem. Res. 54 (44) (2015) 11126–11138.
[22] N. Li, Y. Yang, Ensemble kernel principal component analysis for improved nonlinear process monitoring, Ind. Eng. Chem. Res. 54 (1) (2014) 318–329.
[23] G. Kerschen, J.-C. Golinval, Non-linear generalization of principal component analysis: from a global to a local approach, J. Sound Vib. 254 (5) (2002) 867–876.
[24] Z. Ge, M. Zhang, Z. Song, Nonlinear process monitoring based on linear subspace and Bayesian inference, J. Process Control 20 (5) (2010) 676–688.
[25] G.E. Hinton, R.R. Salakhutdinov, Reducing the dimensionality of data with neural networks, Science 313 (5786) (2006) 504–507.
[26] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, in: Advances in Neural Information Processing Systems, 2007, pp. 153–160.
[27] Y. Xiao, J. Wu, Z. Lin, X. Zhao, A deep learning-based multi-model ensemble method for cancer prediction, Comput. Methods Programs Biomed. 153 (2018) 1–9.
[28] Q. Wang, W. Guo, K. Zhang, A.G. Ororbia II, X. Xing, X. Liu, C.L. Giles, Adversary resistant deep neural networks with an application to malware detection, in:

Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2017, pp. 1145–1153.
[29] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[30] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W.-k. Wong, W.-c. Woo, Deep learning for precipitation nowcasting: a benchmark and a new model, in: Advances in Neural Information Processing Systems, vol. 30, 2017, pp. 5622–5632.
[31] Z. Zhang, T. Jiang, S. Li, Y. Yang, Automated feature learning for nonlinear process monitoring – an approach using stacked denoising autoencoder and k-nearest neighbor rule, J. Process Control 64 (2018) 49–61.
[32] W. Yan, P. Guo, G. Liang, Z. Li, Nonlinear and robust statistical process monitoring based on variant autoencoders, Chemom. Intell. Lab. Syst. 158 (2016) 31–40.
[33] F. Lv, C. Wen, Z. Bao, M. Liu, Fault diagnosis based on deep learning, in: American Control Conference (ACC), IEEE, 2016, pp. 6851–6856.
[34] L. Jiang, Z. Ge, Z. Song, Semi-supervised fault classification based on dynamic sparse stacked auto-encoders model, Chemom. Intell. Lab. Syst. 168 (2017) 72–83.
[35] D.P. Kingma, M. Welling, Stochastic gradient VB and the variational auto-encoder, in: Second International Conference on Learning Representations, ICLR, 2014.
[36] Y. Pu, Z. Gan, R. Henao, X. Yuan, C. Li, A. Stevens, L. Carin, Variational autoencoder for deep learning of images, labels and captions, in: Advances in Neural Information Processing Systems, 2016, pp. 2352–2360.
[37] T.D. Kulkarni, W. Whitney, P. Kohli, J.B. Tenenbaum, Deep Convolut. Inverse Graph. Netw. 71 (2) (2015) 2539–2547.
[38] K. Gregor, I. Danihelka, A. Graves, D.J. Rezende, D. Wierstra, Draw: a recurrent neural network for image generation, Comput. Sci. (2015) 1462–1471.
[39] S. Kullback, R.A. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1) (1951) 79–86.
[40] U. Kruger, S. Kumar, T. Littler, Improved principal component monitoring using the local approach, Automatica 43 (9) (2007) 1532–1542.
[41] E.L. Russell, L.H. Chiang, R.D. Braatz, Data-driven Methods for Fault Detection and Diagnosis in Chemical Processes, Springer Science & Business Media, 2012.
[42] L.H. Chiang, E.L. Russell, R.D. Braatz, Fault Detection and Diagnosis in Industrial Systems, Springer Science & Business Media, 2000.
[43] C. Zhan, S. Li, Y. Yang, Enhanced fault detection based on ensemble global–local preserving projections with quantitative global-local structure analysis, Ind. Eng. Chem. Res. 56 (38) (2017) 10743–10755.
[44] J.J. Downs, E.F. Vogel, A plant-wide industrial process control problem, Comput. Chem. Eng. 17 (3) (1993) 245–255.
[45] P.R. Lyman, C. Georgakis, Plant-wide control of the Tennessee Eastman problem, Comput. Chem. Eng. 19 (3) (1995) 321–331.
[46] Z. Ge, Z. Song, Nonlinear probabilistic monitoring based on the Gaussian process latent variable model, Ind. Eng. Chem. Res. 49 (10) (2010) 4792–4799.
[47] J.L. Rodgers, W.A. Nicewander, Thirteen ways to look at the correlation coefficient, Am. Stat. 42 (1) (1988) 59–66.
[48] C.F. Alcala, S.J. Qin, Analysis and generalization of fault diagnosis methods for process monitoring, J. Process Control 21 (3) (2011) 322–330.