Systematic Development of a New Variational Autoencoder Model Based on Uncertain Data for Monitoring Nonlinear Processes

Kai Wang[1], Michael G. Forbes[2], Bhushan Gopaluni[3], Junghui Chen[4] and Zhihuan Song[1]

[1]State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, 310027 Zhejiang, China
[2]Honeywell Process Solutions, North Vancouver, BC V7J 3S4, Canada
[3]Department of Chemical and Biological Engineering, The University of British Columbia, Vancouver, BC, Canada.
[4]Department of Chemical Engineering, Chung-Yuan Christian University, Chungli, Taoyuan 32023, Taiwan, ROC

Corresponding authors: Junghui Chen (jason@wavenet.cycu.edu.tw); Zhihuan Song (songzhihuan@zju.edu.cn)

ABSTRACT Deep learning models have been applied to industrial process fault detection because of their ability to approximate complex nonlinear dynamic behavior. They have been proven to outperform shallow neural network models. However, there are no good guidelines on how to build these deep models. Therefore, a good deep model is often constructed through a trial and error exercise. It is not easy to interpret the model because of features that do not have any physical interpretation. In addition, latent variables (or features) in a deep model are not independent. This causes features to overlap with each other, resulting in challenges in evaluating distributions of features and designing suitable monitoring indices. Lastly, typical deep learning models in process monitoring are used in a deterministic manner and do not automatically provide confidence levels for each decision. In this paper, a variational auto-encoder is utilized to develop a framework for monitoring uncertain nonlinear processes. The learned latent variables are guaranteed to be independent (or orthogonal) of each other under a specific optimization objective with constraints. The proposed method provides density estimates of latent variables and residuals intead of point estimates. The density functions are used to design appropriate indices for monitoring. A simulation example and an industrial paper machine example are presented to validate the effectiveness of the proposed method.

INDEX TERMS fault detection, latent variables, probability, variational auto-encoder

I. INTRODUCTION

Process complexity and high demands for process safety have driven the development of data-based process monitoring techniques, in particular, multivariate statistical process monitoring [1, 2]. Among them, the continuous latent variable (LV) models have been applied to fault detection for several decades [3-5]. These LV models are proved to be effective because they are able to decompose the observation space into the LV subspace and the residual subspace. The LV subspace describes process mapping, known as a generative model, from LVs to the observed variables. On the other hand, the residual subspace represents the space spanned by measurement noises[6].

Initially, an LV model called principal component analysis (PCA) [7, 8] was widely used for monitoring linear processes with Gaussian observations. However, most processes are characterized by complex nonlinearities and uncertainities and therefore can not be accurately approximated by PCA. To handle these practical issues, advanced LV models for monitoring nonlinear processes have been widely studied. Kernel PCA (KPCA)[9-12] is one of the effective and widely used extensions of PCA for nonlinear processes. With KPCA, original observation variables with nonlinear correlations are nonlinearly mapped onto a high-dimensional space; then the mapped data in high-dimensional space is decomposed into the latent subspace and the residual subspace. Compared with other linear approximation methods which use several linear subspaces to approximate nonlinear dynamics, KPCA is a transformation from the low-dimensional nonlinear observation space into the high-dimensional linear feature space without any approximation error. However, the monitoring performance of KPCA critically depends on the selection of kernel functions and it is also very sensitive to some hyper-parameters required for kernel functions.

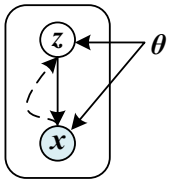


Fig. 1. Illustration of the generative model and the inference model

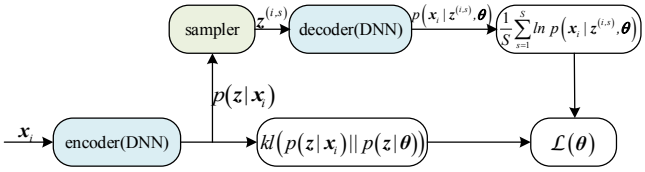


Fig. 2. Flowchart of VAE

Considering these drawbacks, approaches from manifold learning like maximum variance unfolding [13] and neighborhood preserving embedding (NPE) [14] can be used by directly learning the kernel space from observation variables. Nevertheless, approaches resorting to kernel tricks can not deal with large-scale datasets without compromising their performance. The dimension of the kernel matrix is same as the number of samples, and therefore for large data sets, algorithms involving kernel tricks require time-consuming matrix decomposition and large memory for storage. The computational and memory storage challenges are too prohibitive to apply KPCA and manifold learning on large data sets.

In the past modeling approaches, the structure of nonlinear models has limited flexibility and therefore the models were considered to be shallow. They do not have enough flexibility in the models to represent strongly nonlinear systems. Recently, deep learning [15-18] has received a lot of attention, especially in the process system engineering community, because of its high model flexibility. In particular, Zhang et al. [19] proposed a nonlinear process monitoring method based on the stacked denoising auto-encoder (SDAE) that maps observations into LVs through a deep forward network (encoder) and reconstruct observations with LVs through another deep forward network (decoder). Compared with the shallow models, current deep models, when applied to process monitoring, have shown superior performance, but they lack good model interpretability [17]. Specifically, it is hard to explore what kind of manifold in the LVs forms a low-dimensional space because of no explicit local preserving constraints. Besides, unlike LVs in PCA and KPCA, LVs in SDAE are not orthogonal. Their relative process variabilities are unknown in advance. Moreover, observations are driven by the randomly varying LVs and contaminated by random measurement noises. Process variables intrinsically follow a stochastic path while SDAE is constructed through mapping and reconstruction in a deterministic manner. Thus, from the stochastic perspective, SDAE lacks a good probabilistic interpretation about how observations are generated from a distribution. In contrast, many multivariate statistical analysis methods have their probabilistic counterparts such as probabilistic PCA(PPCA) [20], factor analysis(FA) [21], probabilistic KPCA [22] and probabilistic weighted PCA [23], and so on, but these methods are a class of shallow models.

In this paper, an algorithm for process monitoring based on variational autoencoders (VAE) [24, 25], also known as auto-encoding variational Bayes, is developed. VAE is one of the deep learning models and it can infer LVs and generate reconstructed observations with complex posterior and conditional distributions, respectively. VAE can be regarded as a nonlinear version of PPCA or FA. PPCA and FA are based on linear Gaussian models and therefore the posteriors from observations to LVs and the emission distributions from LVs to observations are Gaussian. In addition, the PPCA and FA solutions are analytical. In VAE, complex nonlinearities are taken into account so that deep neural networks can be used to approximate corresponding posteriors and emission distributions. LVs in an industrial system include those variables that make contributions excite the process systems [26]. These exciting signals generally consist of unmeasured disturbance changes, measured disturbance change, and possible setpoint changes, all of which vary independently [7]. The variational Bayes framework brings about a probabilistic interpretation by regarding the industrial plant as a stochastic process. This approach has the benefit of providing estimates of distributions unlike the shallow models. As the complexity of probability distributions evolves with the strong process nonlinearity, process knowledge is easy to incorporate into VAE by designing a proper structure in data distributions. In this article, we propose a fault detection algorithm for complex nonlinear processes using a deep orthogonal LV model. Based on LV and noise distributions, two detection indices in the LV space and the residual space are designed, respectively. The index in the LV space is able to capture the main process variability while the index in the residual space is used to interpret the breakdown of process correlations. The control limits of these two detection indices are determined by kernel density estimation (KDE)[27]. The proposed design algorithm is detailed in the rest of the sections. Section 2 reviews the basic ideas of VAE. Then based on VAE, extraction of orthogonal LVs and their application to fault detection are developed in Section 3. Section 4 presents case studies to illustrate the effectiveness of the proposed framework and conclusions are drawn in the final section.

II. OVERVIEW OF VARIATIONAL AUTOENCODERS

As shown in Fig. 1, assume an *m*-dimensional observation $\boldsymbol{x}$ is generated by a random process $p(\boldsymbol{x},\boldsymbol{z}|\boldsymbol{\theta})=p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{\theta})p(\boldsymbol{z}|\boldsymbol{\theta})$, where $\boldsymbol{z}$ is a vector of *n*-dimensional continuous LVs that is unobserved and $\boldsymbol{\theta}$ is a group of unknown parameters governing the generative process. That means $\boldsymbol{x}$ is generated by the conditional distribution $p(\boldsymbol{x}|\boldsymbol{z},\boldsymbol{\theta})$, in which $\boldsymbol{z}$ is sampled from the prior distribution $p(\boldsymbol{z}|\boldsymbol{\theta})$. VAE is a realization of variational Bayes with deep learning, especially when performing efficient inference and learning in directed probabilistic models in the presence of continuous LVs with intractable posterior distributions and large datasets (Fig. 1) [24]. Generally, LVs are unobservable. What can be obtained are the independent observation samples organized as a dataset $\boldsymbol{X}=\{\boldsymbol{x}_i\in R^{\boldsymbol{m}},i=1,2,\cdots,N\}$ with N independent observations. The goal is to estimate unknown parameters ($\boldsymbol{\theta}$)

and LVs by maximizing the log-likelihood function given by

$$ln\,p(\boldsymbol{X}) = \sum_{i=1}^{N} ln\,p(\boldsymbol{x}_i) \tag{1}$$

where *ln* refers to the natural logarithm and the equality in Eq.(1) follows from the assumption of independent observations. For each term on the right hand side in Eq.(1),

$$ln\,p(\boldsymbol{x}_i) = kl\left(q(\boldsymbol{z}) \,||\, p(\boldsymbol{z}\,|\,\boldsymbol{x}_i)\right) + \mathrm{L}\left(q(\boldsymbol{z}),\boldsymbol{\theta}\right) \tag{2}$$

where $q(\boldsymbol{z})$ is the distribution of LVs. The first term in the right-hand side of Eq.(2) is a Kullback-Leibler (KL) divergence measuring the dissimilarity between the defined distribution $q(\boldsymbol{z})$ and the posterior distribution $p(\boldsymbol{z}\,|\,\boldsymbol{x}_i)$ given by

$$kl\left(q(\boldsymbol{z}) \,||\, p(\boldsymbol{z}\,|\,\boldsymbol{x}_i)\right) = \int q(\boldsymbol{z}) ln \frac{q(\boldsymbol{z})}{p(\boldsymbol{z}\,|\,\boldsymbol{x}_i)} d\,\boldsymbol{z} \tag{3}$$

$p(\boldsymbol{z}\,|\,\boldsymbol{x}_i)$ is also known as an inference model for inferring $\boldsymbol{z}$ given by the observation.

Because of the non-negativity of KL divergence, $\mathrm{L}_i\left(q(\boldsymbol{z}),\boldsymbol{\theta}\right)$ becomes a variational lower bound of $ln\,p(\boldsymbol{x}_i)$ given by

$$\mathrm{L}_i\left(q(\boldsymbol{z}),\boldsymbol{\theta}\right) = \int q(\boldsymbol{z}) ln \left\{ \frac{p(\boldsymbol{x}_i,\boldsymbol{z}\,|\,\boldsymbol{\theta})}{q(\boldsymbol{z})} \right\} d\boldsymbol{z} \tag{4}$$

In most cases, the marginal distribution $p(\boldsymbol{x})$ is so complex that directly maximizing Eq.(2) is difficult and even intractable. Instead, the lower bound in Eq.(4) is maximized to approximate the marginal likelihood. Note that the KL divergence in Eq.(3) plays the role of measuring the approximation error when the lower bound is used to approximate the marginal log-likelihood. It is obvious that the more similar $q(\boldsymbol{z})$ is with $p(\boldsymbol{z}\,|\,\boldsymbol{x}_i)$, the smaller the approximation error is. Especially, the approximation error will be zero when $q(\boldsymbol{z})$ is equal to $p(\boldsymbol{z}\,|\,\boldsymbol{x}_i)$. Hence, at the maximum of the lower bound $\mathrm{L}_i\left(q(\boldsymbol{z}),\boldsymbol{\theta}\right)$, $q(\boldsymbol{z})$ is chosen to be $p(\boldsymbol{z}\,|\,\boldsymbol{x}_i)$. Substituting $q(\boldsymbol{z})$ with $p(\boldsymbol{z}\,|\,\boldsymbol{x}_i)$, the lower bound of Eq.(4) can be rewritten as

$$\mathrm{L}_i\left(\boldsymbol{\theta}\right) = \mathrm{E}_{p(\boldsymbol{z}|\boldsymbol{x}_i)}\left(ln\,p(\boldsymbol{x}_i\,|\,\boldsymbol{z},\boldsymbol{\theta})\right) - kl\left(p(\boldsymbol{z}\,|\,\boldsymbol{x}_i)\,||\,p(\boldsymbol{z}\,|\,\boldsymbol{\theta})\right) \tag{5}$$

where $\mathrm{E}_{p(\boldsymbol{z}|\boldsymbol{x}_i)}\left(ln\,p(\boldsymbol{x}_i\,|\,\boldsymbol{z},\boldsymbol{\theta})\right)$ denotes the expectation of $ln\,p(\boldsymbol{x}_i\,|\,\boldsymbol{z},\boldsymbol{\theta})$ w.r.t $p(\boldsymbol{z}\,|\,\boldsymbol{x}_i)$. To evaluate the loss function as Eq.(5), it first needs the estimation of the posterior distribution $p(\boldsymbol{z}\,|\,\boldsymbol{x}_i)$, called an encoding process, which is used to infer the LVs (codes) $\boldsymbol{z}$ given an observation $\boldsymbol{x}_i$. Then the conditional likelihood $ln\,p(\boldsymbol{x}_i\,|\,\boldsymbol{z},\boldsymbol{\theta})$ in Eq.(5), called a decoding process, should be evaluated. It is used to generate the observation given the codes. The second term $kl\left(p(\boldsymbol{z}\,|\,\boldsymbol{x}_i)\,||\,p(\boldsymbol{z}\,|\,\boldsymbol{\theta})\right)$ in the loss function Eq.(5) sets up a regularization that ensures that the posterior distribution is not too ``far'' from the prior distribution. Hence, it potentially puts a constraint on the posterior distribution that is determined by the structure of prior distribution.

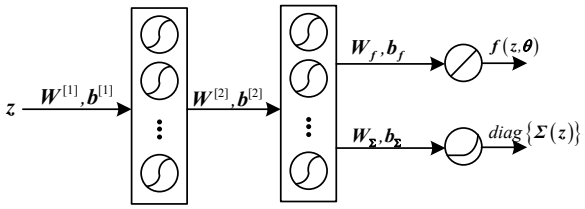


Fig. 3. Decoder structure in VAE

Notice that there is an integral in Eq.(5) for calculating the expectation $\mathrm{E}_{p(\boldsymbol{z}|\boldsymbol{x}_i)}\left(ln\,p(\boldsymbol{x}_i\,|\,\boldsymbol{z},\boldsymbol{\theta})\right)$. It is often difficult to derive the integral analytically. A simple alternative way is a reparative sampling strategy. The true expectation with an empirical average can be estimated using the samplings. Specifically, *S* samples are drawn from $p(\boldsymbol{z}\,|\,\boldsymbol{x}_i)$, denoted as $\boldsymbol{z}^{(i,1)},\boldsymbol{z}^{(i,2)},\cdots,\boldsymbol{z}^{(i,S)}$; then the empirical average is given by $\frac{1}{S}\sum_{s=1}^{S} ln\,p\left(\boldsymbol{x}_i\,|\,\boldsymbol{z}^{(i,s)},\boldsymbol{\theta}\right)$.

The various steps in the implementation of VAE are depicted in Fig. 2. Using the mini-batch stochastic gradient optimization for training a deep neural network (DNN), the cost function for a mini-batch with N_m samples is

$$\mathrm{L}\left(\boldsymbol{\theta}\right) = \frac{1}{S}\sum_{i=1}^{N_m}\sum_{s=1}^{S} ln\,p\left(\boldsymbol{x}_i\,|\,\boldsymbol{z}^{(i,s)},\boldsymbol{\theta}\right) - \sum_{i=1}^{N_m} kl\left(p(\boldsymbol{z}\,|\,\boldsymbol{x}_i)\,||\,p(\boldsymbol{z}\,|\,\boldsymbol{\theta})\right) \tag{6}$$

Note that the sampling number *S* can be chosen as 1 when the mini-batch number is large. This idea is analogous to the stochastic gradient descent algorithms; just one sampling point is used to update the network parameters in each iteration. Similarly, *S*=1 is equivalent to picking up one point from the distribution to evaluate the gradient and update the networks in each iteration. After several iterations, the network would finally converge. [25]

III. PROCESS MODELING AND MONITORING WITH VAE

A. PROCESS MODELING

Observation variables are measured process variables which are often high-dimensional in large-scale systems. They are correlated with each other in a complex fashion because of highly complex nonlinearities in real industrial processes. As described in Introduction Section, in the high-dimensional process variables of an industrial system, LVs representing the essential features of observation variables are assumed to be independently corrupted by noise signals. They can be assumed to be uncorrelated with each other. Based on these assumptions, a process model is given by

$$\boldsymbol{x} = \boldsymbol{f}\left(\boldsymbol{z},\boldsymbol{\theta}\right) + \boldsymbol{w}(\boldsymbol{z}) \tag{7}$$

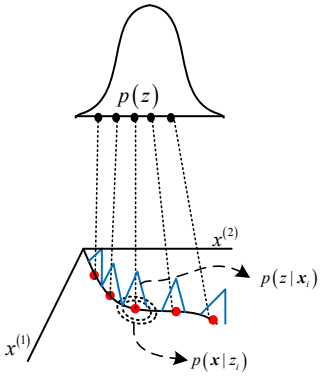


Fig. 4. Illustration of distributions in VAE

where $w = N(\theta, \Sigma(z))$ is zero-mean Gaussian white noise standing for measurement errors. Here $\Sigma(z)$ can vary with z considering the nonlinearities in measurements. In this case we relaxed the assumption that the covariance matrix is constant as in the conventional model representation. And $\Sigma(z)$ can be a diagonal covariance matrix because measurement errors from different sensors are assumed to be independent. $f(z,\theta) \in \mathrm{R}^n \to \mathrm{R}^m$ is a nonlinear function with process parameters θ, representing a complex process model mapping from the latent space onto the observation space. From Eq.(7), a conditional Gaussian distribution can be derived as:

$$p(x \mid z,\theta) = N\left(f(z,\theta), \Sigma(z)\right) \tag{8}$$

where a decoder in VAE is used to model the nonlinear functions $f(z,\theta)$ and $\Sigma(z)$. Taking the neural network with three hidden layers as a decoder, for example, $f(z,\theta)$ and $\Sigma(z)$ are calculated by

$$f(z,\theta) = W_f \, tanh\left(W^{[2]} \, tanh\left(W^{[1]} z + b^{[1]}\right) + b^{[2]}\right) + b_f \tag{9}$$

$$diag\{\Sigma(z)\} = \zeta\left(W_\Sigma \, tanh\left(W^{[2]} \, tanh\left(W^{[1]} z + b^{[1]}\right) + b^{[2]}\right) + b_\Sigma\right) \tag{10}$$

where $diag\{\Sigma(z)\}$ stands for the vector consisting of diagonal elements of $\Sigma(z)$. Fig. 3 illustrates the network structure presented by Eqs.(9) and (10). In the decoder, the expectation and the covariance share the parameters of the hidden neurons $\{W^{[1]}, b^{[1]}, W^{[2]}, b^{[2]}\}$. $f(z,\theta)$ is outputted through a linear unit with parameters $\{W_f, b_f\}$ because of the unlimited range of expectations. In contrast, $diag\{\Sigma(z)\}$ should be larger than zero so that the softplus activation function $\zeta(x) = ln(1+e^x)$ is utilized in the corresponding output layer with the weight W_Σ and the bias b_Σ.

Since $f(z,\theta)$ has formally embraced complexity related to processes, the prior distribution of z can be chosen to be a simple distribution. Moreover, as the components in z are uncorrelated with each other, the prior distribution $p(z)$ is chosen to be normal; i.e.,

$$p(z) = N(0, I) \tag{11}$$

This means that the points in the latent space are assumed to be drawn from the normal distribution. Taking two-dimensional observation variables in Fig. 4 for example, assume that there is a one-dimensional LV. Firstly, the samples in the latent space (black nodes) are randomly generated from the unit Gaussian distribution. By a nonlinear mapping, the black nodes are projected onto the red nodes in the two-dimensional observation space. The ellipses surrounding the red nodes refer to the uncertainty caused by data quality, denoting the distribution of x conditioned in z. It is obvious that the posterior distribution $p(z \mid x)$ is not a linear Gaussian model under the nonlinear mapping function $f(z,\theta)$. In Fig. 4, one of the contours related to the posterior distribution ($p(z \mid x_i)$) is denoted by the black smooth curve, which indicates there is a manifold embedded in the lower dimensional space. Because of the complexity of posterior distribution, it is difficult to formalize the true posterior with several parameters. Instead, local description is used as an approximation to the true posterior, i.e., the probability density function (PDF) is given at each observation. As shown in Fig. 4, the blue curve denotes the specific posterior PDF at x_i, which is also chosen to be a normal distribution. But the expectation and covariance describing the local PDF vary with x_i as follows.

$$p(z \mid x_i) = N\left(\mu(x_i), V(x_i)\right) \tag{12}$$

where $\mu(x_i)$ and $V(x_i)$ are the posterior expectation and the posterior covariance matrices, respectively, both of which are nonlinear functions of x_i. To guarantee LVs to be orthogonal with each other, $V(x_i)$ is constrained to be a diagonal matrix. By introducing the Gaussian distribution as a local estimator, the KL divergence of the last term in Eq.(5) involving two Gaussian distributions (Eqs.(11) and (12)) has a closed form as follows:

$$kl\left(p(z \mid x_i) \,||\, p(z)\right) = \frac{1}{2}\left\{\mu(x_i)^T \mu(x_i) - ln\left(\left|V(x_i)\right|\right) + tr\left(V(x_i)\right) - n\right\} \tag{13}$$

where $tr(\bullet)$ refers to the trace of one squared matrix. As shown in Fig. 2, the output of the encoder network will be $\mu(x_i)$ and the diagonal elements of $V(x_i)$. Following the same design logic of the decoder in Eqs.(9) and (10), in the encoder, the expectation in the output layer can be activated by a linear unit while the output activation function related to the covariance is chosen to be the softplus function.

So far, the process model has been constructed based on VAE with these specific distributions. The gradient back-propagation along the network in Fig. 2 is used to learn the network weights and biases. However, sampling is not differentiable so that the backpropagation is blocked from the decoder back-propagating to the encoder. Reparameterization trick of Gaussian distribution [25] makes the network learnable without any extra cost or compromise. The idea

Algorithm 1. Training VAE with orthogonal LV constraints

Input: Dataset $\boldsymbol{X}$ with M mini-batch; the number of latent variables n; maximum iteration K; network structure (the numbers of hidden layer and units in each layer); learning rate η.

Output: Network parameters (weights and biases)

Start:

Initialize weights and biases

For k=1:K

For m=1:M

Encoder: Calculate the posterior means and posterior variances (Eq.(12))

Sample data from the unit Gaussian distribution and transform these samples (Eq.(14))

Decoder: Calculate conditional means and conditional variances (Eq.(8))

Calculate the lower bound of the likelihood (Eq.(6))

Update weights and biases by the gradient descent with the learning rate η

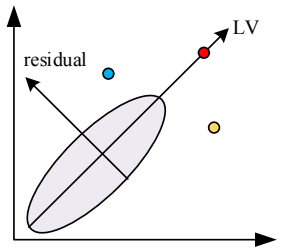
End For

End For

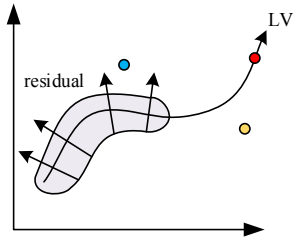
End

behind reparameterization is that the distribution in Eq.(12) can be regarded as an affine transformation of the normal distribution $p(\boldsymbol{\varepsilon}) = N(\boldsymbol{0}, \boldsymbol{I})$, i.e.

$$\boldsymbol{z} = \boldsymbol{V}(\boldsymbol{x}_i)^{-\frac{1}{2}}\boldsymbol{\varepsilon} + \boldsymbol{\mu}(\boldsymbol{x}_i) \tag{14}$$



(a)



(b)

Fig. 5. Illustrations of different kinds of anomaly patterns located at the LV and the residual spaces in (a) a linear system; (b) a nonlinear system.

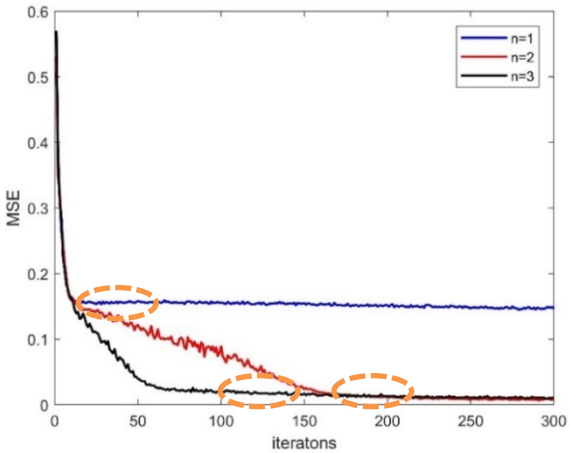


Fig. 6. Validation errors for different numbers of LVs in the numerical example

which is differentiable. Each point $\boldsymbol{z}^{(i,s)}$ required in VAE shown in Fig. 2 is given by $\boldsymbol{z}^{(i,s)} = \boldsymbol{V}(\boldsymbol{x}_i)^{-\frac{1}{2}}\boldsymbol{\varepsilon}^{(i,s)} + \boldsymbol{\mu}(\boldsymbol{x}_i)$, $s = 1, \cdots, S$, where $\boldsymbol{\varepsilon}^{(i,s)}$ represents a sample from the normal distribution. By reparameterization, the orthogonal latent variables become learnable.

Before the network is trained, the number of latent variables n and the number of iterations K are two important hyperparameters to be predefined. A poor choice of n that is different from the true value can cause a severe model bias resulting in an estimated model structure that deviates from the true model structure. In terms of the number of iterations, a small K may also result in model bias as the network would not have converged. On the other hand, a large K will induce a high model variance known as overfitting. In this paper, the early stopping strategy [28, 29] is used to find the hyperparameters. Consequently, the original dataset is divided into mutually exclusive training and validation datasets. For a specific n, set up a large K and observe the validation error on the validation set. The validation error should be defined to reflect the underfitting and overfitting of the networks. The mean squared error (MSE) is used to evaluate the model error because it is a trade-off index that considers the model bias and the model variance simultaneously. The corresponding MSE is given by

$$MSE = \frac{1}{N_v}\sum_{i=1}^{N_v}\left\|\boldsymbol{x}_i - \hat{\boldsymbol{x}}_i\right\|_2 \tag{15}$$

where N_v is the number of validation set. $\hat{\boldsymbol{x}}_i$ is the reconstructed observations defined as

$$\hat{\boldsymbol{x}}_i = \boldsymbol{f}\left(\boldsymbol{z}^{(i,s)}, \boldsymbol{\theta}\right) \tag{16}$$

where $\boldsymbol{z}^{(i,s)}$ is one of the sampling points of LVs based on the encoder network. The optimal number of iterations denoted as $n = n^{opt}$ is obtained when MSE tends to be stationary or has not significantly improved. By incrementing n gradually, the one with a minimum MSE of the validation dataset is considered to be the optimal number of LVs. In this paper, the upper bound of n can be heuristically determined by PCA,

therefore about 80% cumulative variance contribution is chosen for the number of principal components. The complete learning algorithm is given in Algorithm 1.

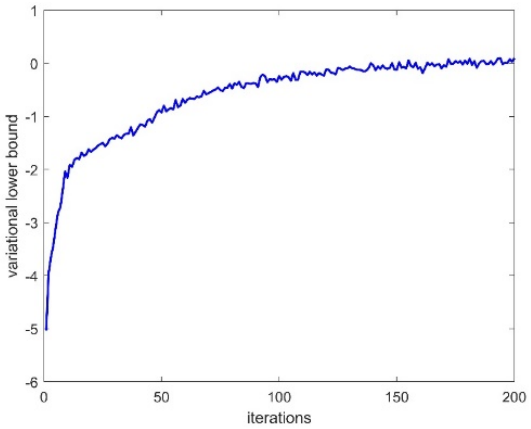


Fig. 7. The trend of the variational lower bound in the numerical example.

B. PROCESS MONITORING

After the VAE-based process model is developed, a latent space and a residual space can be obtained by the encoder and the decoder, respectively. Instead of point estimates of LV and the residual for a specific observation, VAE offers distribution descriptions, giving more information than a point estimate. To make the full use of the distribution information, the monitoring indices should be constructed by the posterior PDF ($p(\boldsymbol{z}|\boldsymbol{x}_i)$) and the conditional PDF ($p(\boldsymbol{x}_i|\boldsymbol{z})$).

1) MONITORING INDEX IN LATENT SPACE

According to the lower bound of the likelihood in Eq.(5), the objective with respect to the KL divergence $kl\left(p(\boldsymbol{z}|\boldsymbol{x}_i)||p(\boldsymbol{z})\right)$ makes the posterior distribution as close to the prior distribution as possible. Therefore, an abnormal sample will have a large KL divergence because the posterior will be dissimilar to the prior. The KL divergence can be considered as a monitoring index D_i in the latent space, as in Eq.(13). To define the normal operating region or control limits, PDF of D_i described by a known density function such as Gaussian or normal PDF cannot be guaranteed. To overcome the limitation, KDE is used to estimate the distribution of the monitoring index.

$$p(D)=\frac{1}{Nh}\sum_{i=1}^{N}K\left(\frac{D-D_i}{h}\right) \tag{17}$$

where $K(\cdot)$ is a kernel function with the constraints $\int K(\boldsymbol{x})d\boldsymbol{x}=1$ and $K(\boldsymbol{x})\geq 0$. h is a hyperparameter known as the bandwidth to adjust the smoothness of the kernel function. The Gaussian kernel function is used in this paper and the bandwidth is determined by the empirical criterion as follows, derived by Mugdadi and Ahmad [30]

$$h=1.06\sigma N^{-0.2} \tag{18}$$

where σ is the standard deviation of the sample. Based on the distribution, a control limit D_{lim} can be designed under a given confidence level α such as 95%; i.e., the minimum D_{lim} satisfies

$$P(D)=\int_{-\infty}^{D_{lim}}p(D)dD\geq a \tag{19}$$

For a new sample, one should first derive the posterior using the encoder of VAE. Then calculate the KL divergence, and finally judge whether it exceeds the control limit.

2) MONITORING INDEX IN RESIDUAL SPACE

Similarly, the expectation of the conditional log-likelihood $\mathrm{E}_{p(\boldsymbol{z}|\boldsymbol{x}_i)}\left(ln\ p(\boldsymbol{x}_i|\boldsymbol{z},\boldsymbol{\theta})\right)$ in Eq.(5) can represent an index measuring the possibility of the observation drawn from the conditional distribution. The larger $ln\ p(\boldsymbol{x}_i|\boldsymbol{z},\boldsymbol{\theta})$ indicates the observation highly follows the distribution $p(\boldsymbol{x}_i|\boldsymbol{z},\boldsymbol{\theta})$. According to Eq.(8), there is

$$\begin{aligned}&ln\ p(\boldsymbol{x}_i|\boldsymbol{z},\theta)=\\&-\frac{1}{2}\left(\boldsymbol{x}_i-\boldsymbol{f}(\boldsymbol{z})\right)\boldsymbol{\Sigma}^{-1}(\boldsymbol{z})\left(\boldsymbol{x}_i-\boldsymbol{f}(\boldsymbol{z})\right)-\frac{1}{2}\left|\boldsymbol{\Sigma}(\boldsymbol{z})\right|-m\pi\end{aligned} \tag{20}$$

One can see $ln\ p(\boldsymbol{x}_i|\boldsymbol{z},\boldsymbol{\theta})$ is similar to the indices in the residual space and $ln\ p(\boldsymbol{x}_i|\boldsymbol{z},\boldsymbol{\theta})$ represents the distance between the observations $\boldsymbol{x}_i$ and the reconstructed values $\boldsymbol{f}(\boldsymbol{z})$. Moreover, it simultaneously considers the uncertainty measured by the covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{z})$. Hence, $ln\ p(\boldsymbol{x}_i|\boldsymbol{z},\boldsymbol{\theta})$ is able to play the role of detecting the anomaly related to residuals. To make $\boldsymbol{f}(\boldsymbol{z})$ and $\boldsymbol{\Sigma}(\boldsymbol{z})$ tractable, the point $\boldsymbol{z}^{(i,s)}$ sampled from $\boldsymbol{p}(\boldsymbol{z}|\boldsymbol{x}_i)$ is used to calculate the final index, given by

$$\begin{aligned}&R_{i,s}=-ln\ p\left(\boldsymbol{x}_i|\boldsymbol{z}^{(i,s)},\theta\right)-m\pi=\\&\frac{1}{2}\left(\boldsymbol{x}_i-\boldsymbol{f}\left(\boldsymbol{z}^{(i,s)}\right)\right)\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{z}^{(i,s)}\right)\left(\boldsymbol{x}_i-\boldsymbol{f}\left(\boldsymbol{z}^{(i,s)}\right)\right)+\frac{1}{2}\left|\boldsymbol{\Sigma}\left(\boldsymbol{z}^{(i,s)}\right)\right|\end{aligned} \tag{21}$$

The larger R_i is, the more likely $\boldsymbol{x}_i$ is an abnormal point, but it is hard to produce a closed form of the expectation because of the nonlinear representation. Here an empirical average $R_i=\frac{1}{S}\sum_{s=1}^{S}ln\ p\left(\boldsymbol{x}_i|\boldsymbol{z}^{(i,s)},\boldsymbol{\theta}\right)$ introduced in Section 2 can be used for monitoring the anomaly in the residual space. As mentioned before, S can be chosen as 1. With the estimation of PDF of the R index, like the determination of the control limit in the D index, the corresponding control limit in the R index can be determined.

Remark. Like the T^2 statistic in PCA based process monitoring, the D detection index in VAE based process monitoring is applied to the latent space while the negative R detection index in VAE is an analogy to the SPE statistic in PCA. In a linear time-invariant system, an identified model still works for normal data patterns even though the scope of variables is beyond the training set because the model for linear systems would not vary with variables. Hence, when a large external fluctuation happens in process systems without a breakdown of the process model, only T^2 is out of control, like the red point in Fig. 5 (a). In the figure, the gray area is

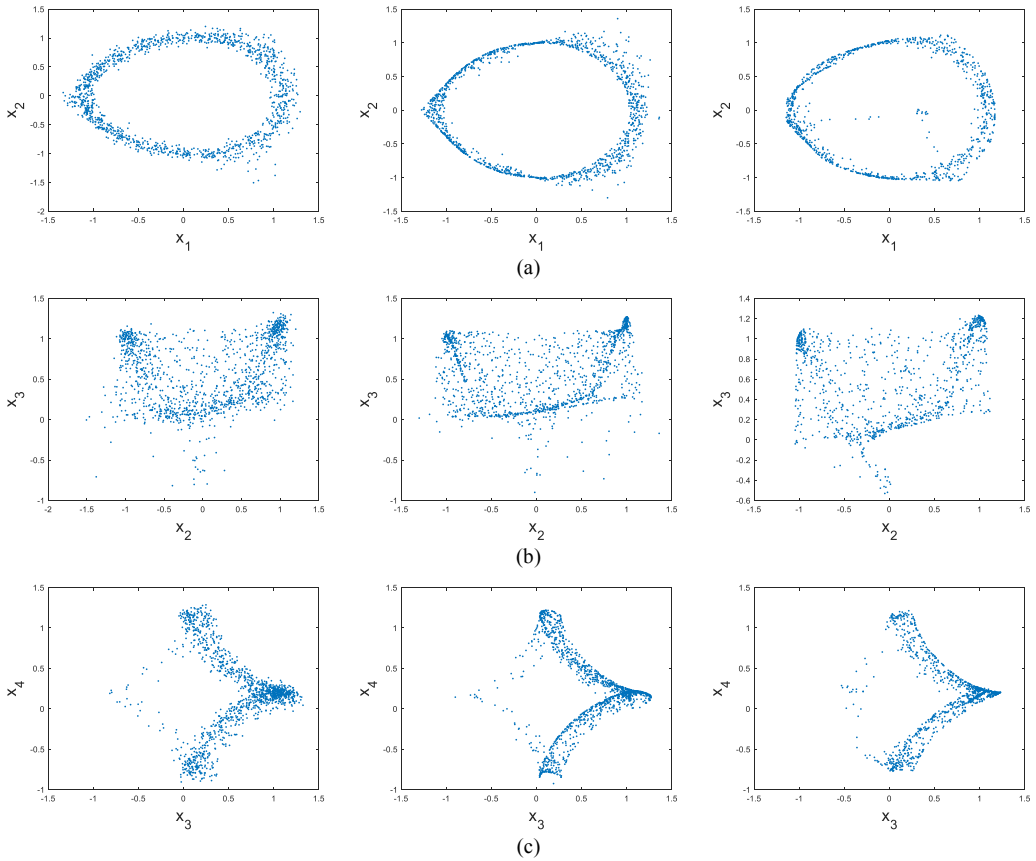


Fig. 8. Scatter plots of the variable pairs of (a) (x_1, x_2), (b) (x_2, x_3), and (c) (x_3, x_4). The left column represents the measured observations. The middle column shows the true values of observations. The right column displays the reconstructed values by the model.

Table I. The descriptions of fault scenarios in the numerical example

Fault No.	Location	Type	Magnitude
F1	z_1	Mean (bias fault)	1
F2			2
F3			3
F4	z_2	standard deviation (noise fault)	1.5
F5			2.5
F6			3.5
F7	w_1	standard deviation (sensor precision degradation)	0.15
F8			0.25
F9			0.35
F10	t	process fault	1.5
F11			2
F12			2.5

Table II. The FARs in the numerical example (%)

Methods	T^2	SPE
PCA	1.2	3.2
KPCA	1.4	4.8
NPE	0.5	4.2
SDAE	4.8	4.8
VAE	4.2	4.0

the control limit determined by the training set. The blue point in Fig. 5(a) indicates T^2 is in a normal region. SPE is out of control, so there is a change of an inner system structure. It implies a real fault. But for the yellow point in Fig. 5 (a), the anomaly can be detected in both spaces (T^2 and SPE). Nevertheless, the situation becomes more complex when it comes to nonlinear processes. As one knows, nonlinear models, no matter whether they are shallow methods or deep methods, just learn the given training set. Nonlinear methods may not learn these patterns caused by larger LVs which do not occur in the training sets. Take Fig. 5(b) for example. The nonlinear model is a local description of a training set. For the red point, a larger LV would be captured by T^2 but SPE may also respond to it because the LV model with the collected training set cannot cover all the other unknown patterns. Hence, there is probably a significant reconstruction error caught by SPE. In Fig. 5(b), the behaviors of the other two kinds of anomalies (the blue point and the yellow point) look like those in Fig. 5(a) of the linear system.

To sum up, the algorithm with the proposed VAE based process monitoring algorithm are listed as follows:

Step 1. Normalize process variables with sample means and standard deviations.

Step 2. Organize the training set by randomly sampling 80% from the dataset and the remaining 20% of the dataset is taken as the validation set.

Step 3. Determine the upper bound of the number of LVs by PCA.

Step 4. Train the VAE based process monitoring model

i. Initialize the number of LVs n as 1.

ii. Train the model with Algorithm 1 using the training set.

iii. Evaluate the validation error in each iteration.

iv. Record the optimal number of iterations in the current n through an early stopping strategy. Record the validation error under the optimal number of iterations.

v. Increment n and return back to ii until n is up to the upper bound.

Step 5. Determine the optimal n and the corresponding number of iterations based on the minimum validation error.

Step 6. Retrain the model with Algorithm 1 using the whole dataset under the predefined optimal hyperparameters.

Step 7. Output the two defined detection indices using the normal data set and determine the control limits.

Step 8. For any new sample, normalize it using the means and standard deviations of the normal dataset. Calculate the two indices by feeding it into the trained model and compare the indices with the corresponding control limits. If yes, keep monitoring the next new data points; otherwise, further analyze what caused the abnormal situation.

IV. CASE STUDIES

In this section, the feasibility and efficiency of the proposed method are evaluated by two examples, including a numerical example and an industrial process example. The numerical example is created artificially. Then the proposed method is applied to a real industrial process, a more challenging test bed for process monitoring. The proposed VAE in this paper is compared with several conventional data-driven fault detection methods, including PCA, KPCA, NPE, and SDAE. Among them, PCA is a benchmark approach to process monitoring. KPCA is a popular representative of kernel methods for process monitoring. Attempting different commonly used kernels for KPCA, the sigmoid kernel $k(\boldsymbol{x},\boldsymbol{y}) = tanh(\boldsymbol{\beta}_0 \boldsymbol{x}^T \boldsymbol{y} + \boldsymbol{\beta}_1)$ has better monitoring performance than the other kernels such as the polynomial kernel and the radial basis kernel in this example. Hence, KPCA with the sigmoid kernel ($\boldsymbol{\beta}_0 = 1$ and $\boldsymbol{\beta}_1 = 0$) is used. NPE is one of the manifold learning methods. Here the number of LVs in the three comparative methods is determined by the cumulative variance of 80%. Regarding SDAE, it is a deep feature extraction method based on deep learning, and the number of LVs is set up to be the same as the proposed method. Note that the detection indices of different methods in the LV space play a similar role of measuring the variability in LVs, though different methods may nominate different indices such as T^2 in PCA, HD in SDAE[19] and D in VAE. For convenience, in this paper, all these indices are commonly referred to as T^2. Likewise, all the indices in the residual space are commonly referred to as SPE. The deep models were trained in the environment of NVIDIA GeForce GTX 1060.

A. NUMERICAL EXAMPLE

A nonlinear system with 4 observation variables is considered in this example. There are 2 LVs generating the observations contaminated by Gaussian measurement noises, given by

$$x_1 = 0.1 z_1 + z_1 / \sqrt{z_1^2 + z_2^2} + w_1$$
$$x_2 = 0.1 z_1 z_2 + z_2 / \sqrt{z_1^2 + z_2^2} + w_2$$
$$x_3 = t\cos^3 z_1 + 0.1 e^{\sin z_2} + w_3$$
$$x_4 = \sin^3 z_1 + 0.2 \ln(2 + \cos z_2) + w_4$$

where t is an adjustable coefficient and is 1 when the system is normal. z_1 and z_2 are LVs subject to unit Gaussian distributions. $w_i, i = 1,2,3,4$ are zero-mean Gaussian measurement noises with standard deviations $0.05, 0.06, 0.05, 0.04$, respectively. A total of 1,200 normal observations are generated and normalized. Among them, 1,000 observations are randomly selected as the training set and the remaining 200 points are organized as the validation set. With PCA, the singular values of the covariance matrix of the training set are $[1.83, 1.14, 0.84, 0.18]$. Therefore, the maximum number of LVs is 3 using the cumulative variance of 80%. To construct the VAE model, both the encoder and the decoder are built by three hidden-layer feedforward networks, in which each layer contains 30 neurons. The maximum iterations are set up as 300. Fig. 6 presents the trends of MSE as the iteration time evolves with the three different numbers of LVs from 1 to 3. According to the early stopping strategy, the optimal iteration number is marked by a circle point in Fig. 6. Among the three different LVs, the minimum MSE is obtained at $n = 2$ and with the corresponding 200 iterations. Then, the 1,000 training samples and the 200 validation samples are concatenated and used to retrain the network with these determined hyperparameters. The trend of the variational lower bound is illustrated in Fig. 7. One can see the lower bound gradually increases as the iteration proceeds and it tends to be steady when the number of iterations is close to 200. Taking the parameters updated at the 200th iterations as the final network parameters, the posterior distribution and the conditional distribution can be obtained by feeding each normal sample to the network. In this paper, the means of the posterior play the role of point estimates of LVs for each observation. The covariance matrix of the means of the posterior for all the observations are $\begin{bmatrix} 0.87 & 0.08 \\ 0.08 & 0.87 \end{bmatrix}$, which implies there is little correlation between the two LVs and the learned LVs are orthogonal to each other. This satisfies the required VAE model; there is no overlapped and redundant information among different LVs. To visualize the ability of signal reconstruction of VAE, the scatter plots between two different observed variables are shown in Fig. 8. Fig. 8 presents the scatter plots of (x_1, x_2), (x_2, x_3), and (x_3, x_4) from top to the bottom. The left column shows the observed values contaminated by measurement noises. The middle column presents the true observation values without considering noises so that the contour in the middle column is more distinct than the left column. The right column gives the reconstructed values with the LVs. The reconstructed values in the right column are close to the true values in the middle

Table III. The FDRs in the numerical example (%)

Fault No.	T^2	SPE

	PCA	KPCA	NPE	SDAE	VAE	PCA	KPCA	NPE	SDAE	VAE
F1	4.8	6.6	1.5	7.5	**13.0**	9.1	7.9	**14.1**	10.9	10.0
F2	12.1	24.1	8.6	11.3	**32.1**	29.6	27.7	31.4	**34.8**	34.6
F3	20.8	44.2	35.4	14.8	**49.2**	65.7	67.5	36.6	68.8	**72.2**
F4	1.2	0.9	0.5	5.1	**7.9**	3.5	3.7	3.8	4.9	**9.3**
F5	1.7	2.1	0.8	**14.4**	13.7	6.0	8.0	2.3	13.2	**26.3**
F6	2.1	3.5	2.2	**24.0**	16.6	8.4	11.8	2.0	16	**34.4**
F7	0.8	0.7	0.2	**8.2**	3.8	2.3	4.4	4.8	4.4	**11.2**
F8	0.9	1.0	0.8	**15.2**	4.5	3.1	7.3	6.7	10.6	**25.1**
F9	1.6	1.1	1.9	**22.2**	4.6	3.9	12.3	7.0	14.5	**36.0**
F10	**7.8**	6.1	0.4	3.7	2.7	17.6	12.9	9.2	37.2	**43.1**
F11	18	**35.7**	0.0	7.6	3.9	29.5	41.9	36.3	53.5	**67.0**
F12	25.2	**44.0**	0.3	13.0	3.2	42.1	47.6	47.4	56.4	**72.9**

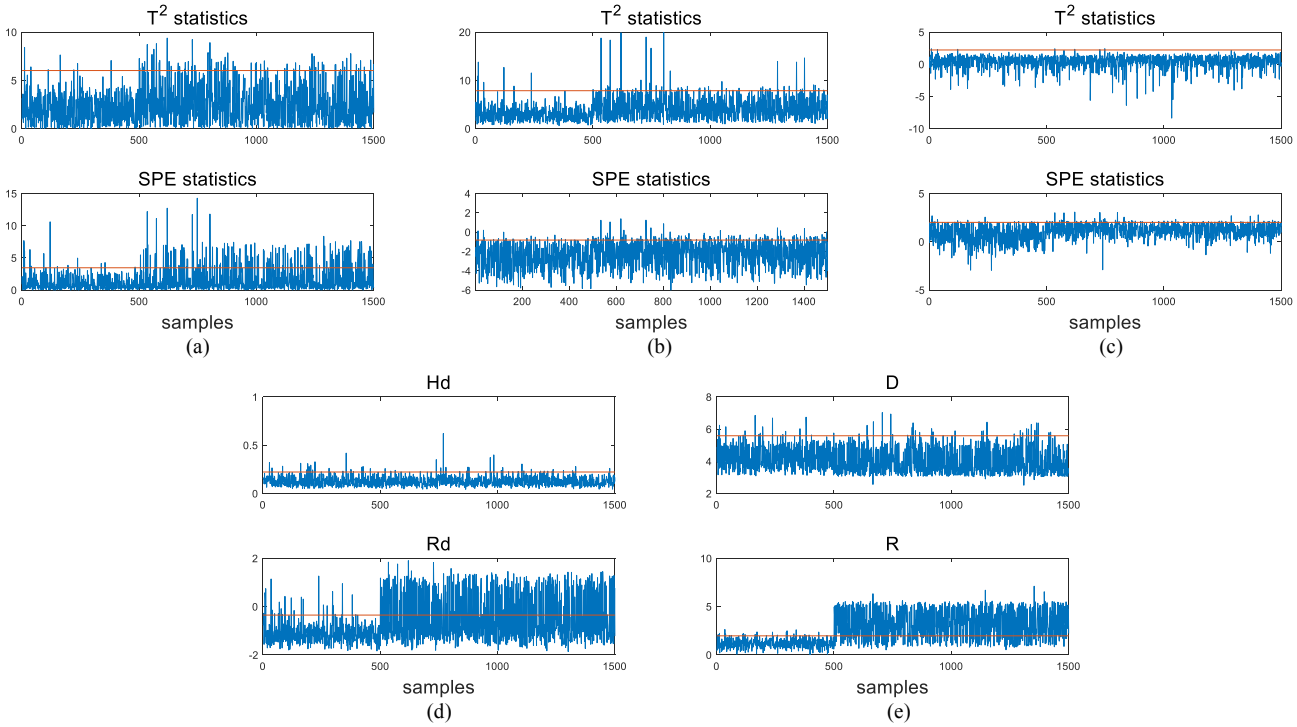


Fig. 9. The control charts of (a) PCA, (b) KPCA, (c) NPE, (d) SDAE, and (e) VAE for F10 in the numerical example.

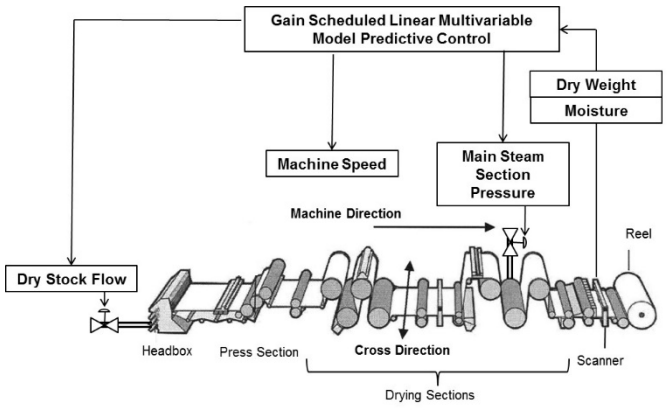


Fig. 10. Structure of a typical industrial paper machine.

Table IV. The FARs in the paper machine (%)

Methods	T^2	SPE
PCA	5.8	5.9
KPCA	6.8	6.3
NPE	5.7	3.4
SDAE	5.0	5.0
VAE	4.9	5.0

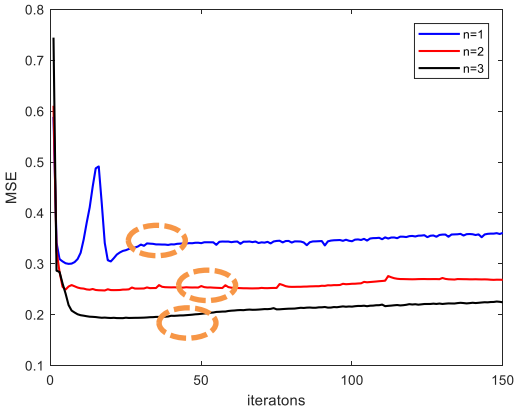


Fig. 11. The validation errors for different numbers of LVs in the paper machine.

column, which implies overfitting is suppressed significantly and the model precision is satisfactory.

For fault detection, 12 fault scenarios, listed in Table I, are designed with several different magnitudes and fault locations. A total of 1,000 samples of each fault is collected. In the 12 fault scenarios, each fault type contains three different fault

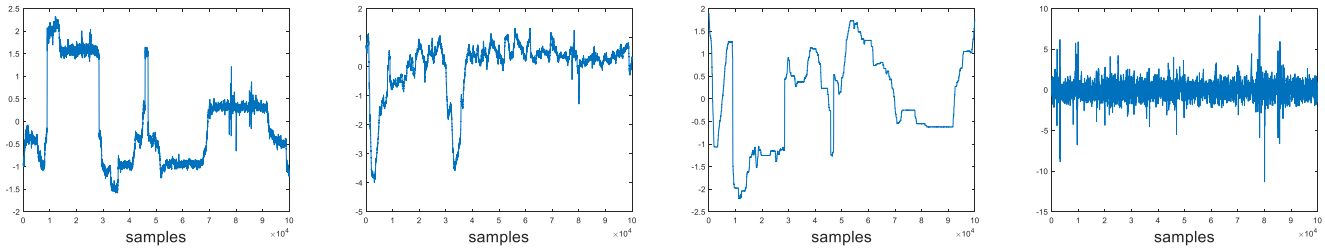


Fig. 12. Trends of four variables in the paper machine (clockwise from top left these are: Dry Weight, Dry Stock Flow, Machine Speed, and Moisture)

Table V. The FDRs in the paper machine (%).

Fault No.	T^2					SPE				
	PCA	KPCA	NPE	SDAE	VAE	PCA	KPCA	NPE	SDAE	VAE
F1	6.0	6.2	7.5	6.7	**11.7**	2.4	8.7	0.1	44.2	**68.6**
F2	**51.7**	4.3	32.0	43.1	31.3	26.5	**88.3**	23.8	44.1	86.9
F3	25.2	37.6	**45.5**	15.3	7.9	11.7	41.6	8.4	14.4	**59.2**

magnitudes; one wants to test the sensitivity of the fault detection methods. With the 95% confidence level, Table II summarizes the false alarm rates (FARs), referring to the ratio of false alarm numbers to the total numbers for normal data. It is reasonable when FARs are less than 5% because of the 95% confidence level. One can see from Table II that all the methods have an eligible FAR. The fault detection rate (FDR) is the ratio of the samples with detection indices beyond the control limits to the total samples. A decisive performance of FDR that evaluates these methods for all the fault data is listed in Table III. In the table, the largest FDRs in the two detection indices for each fault are bold. F1-F6 are faults in LVs. Among them, F1, F2, and F3 are bias faults in the first LV while F4, F5 and F6 are noise faults in the second LV. T^2 and SPE should catch these two kinds of anomalies based on the analysis in the remark. It is found that VAE substantially behaves better than the other methods. Specifically, T^2 in VAE for F1-F4 outperforms the other methods. SDAE presents a smaller preponderance of F5 and F6 over VAE. Especially, for the three noise faults (F4-F6), FDRs of other shallow methods in T^2 even cannot exceed 5%, causing a misleading statement that the root cause does not come from LVs. The reason is that shallow methods provide a very poor estimate for true data distribution. F7-F12 are structural faults, F7-F9 suffer a sensor precision degradation, and F10-F12 simulate a varying variable correlation. In these situations, SPE would be sensitive while T^2 is immune to these kinds of faults because these faults will not result in a wide range of fluctuations. It is clear that both NPE and VAE give correct judgment because SPE in the two methods is out of control while T^2 is still under control. Moreover, SPE of VAE has larger FDRs than that of NPE. In contrast, PCA, KPCA and SDAE would mislead engineers into considering a change of external exciting signals because there are some out-of-control points in T^2. Substantially, the proposed process model based on VAE outperforms the other methods in inferring fault locations and sensitivity of detecting faults. Taking F10 for example, the control chart for each method is plotted in Fig. 8. In the figure, the former 500 samples are normal and the latter 1,000 samples are abnormal.

B. INDUSTRIAL EXAMPLE

An industrial paper machine process is studied in this work. Paper machines, such as the one shown in Fig. 10, transform stock, which is a suspension of wood cellulose fibres in a water solution, into a web of paper which is wound onto a reel. The direction the paper moves along the paper machine is known as machine direction (MD), while the direction perpendicular to this is known as cross direction (CD). Quality variables of the paper must be kept uniform along both the machine and cross directions, but typically the MD and CD control problems are handled independently. In this study, data from the MD process is examined. Paper quality measurements of average dry weight (the weight of the paper less any remaining moisture on a per area basis) and the average moisture content across the web are taken by a scanning sensor at the end of the machine. These quality variables are controlled by adjusting the dry stock flow (the volumetric flow of the solids in the stock), the steam pressures in heated metal cylinders in the drying sections, and the machine speed. The MD process is a multivariable process with sufficiently nonlinear behavior that a common industrial practice is to apply a gain scheduled linear model predictive control with different models for different operating regions. In this study, 9 process variables are examined, including actuator signals, setpoints, sensor signals and mode signals. A dataset with 100,000 points is collected and judged as normal data by engineers. To show the complexity of the process, the trends of 4 normalized variables in the training set are plotted in Fig. 11. It is found that most of the variables present strong nonlinear fluctuations. To construct a VAE model, likewise, PCA is used to determine the upper bound of the number of LVs. The singular values of the 9 variables in the covariance matrix are [4.3, 3.2, 1.0, 0.7, 0.3, 0.05, 0.007, 0.0007, 0.0003]. Based on the 80% cumulative variance contribution, the first three LVs are used as the candidates. To verify the training models with different selected LVs, the collected dataset is separated into a training set and a validation set. The MSEs on the validation set for the chosen three different number of LVs are shown in Fig. 12. Based on the early stopping strategy, the optimal iterations are marked in circle points (Fig. 12). Among the LV models, the model with three LVs is selected as the minimum MSE is obtained for this paper machine process after 50 iterations. With these trained hyperparameters, the posterior distribution and the conditional distribution for each sample can be obtained and the means of the posterior distribution are regarded as a point estimate of LVs. The covariance matrix of the means of all the training

data is

$$\begin{bmatrix} 1.01 & -0.09 & 0.03 \\ -0.09 & 0.8 & -0.08 \\ 0.03 & -0.08 & 0.8 \end{bmatrix}$$

As the off-diagonal entries of the matrix are close to zero, all the LVs can be considered as being orthogonal to each other. This satisfies the required VAE model.

The proposed method, as well as several comparative methods (PCA, KPCA, NPE, and SDAE), is applied to detecting the anomalies of this paper machine. As the estimated models of KPCA and NPE are computationally expensive for large-scale data, the size of the dataset is beyond their tractability in the configuration of our own computer system in this example. Hence, a downsampling strategy is performed for KPCA and NPE. Three kinds of testing data are collected and described as follows:

F1: A controlled variable is added with an additional sensor noise.

F2: A sequence of data with the operation modes which are not included in the normal training data.

F3: A sequence of data produced with a changed controller.

Each fault scenario contains 30,000 samples. Given a 95% confidence level, control limits can be calculated for different methods. The results of the FARs listed in Table IV indicate the shallow methods, PCA, KPCA, and NPE, cannot sufficiently fit the data distribution as FARs in both T^2 and/or SPE statistics exceed 5%. Further, FDRs for the three testing data are given in Table V. For F1, it is a scenario of sensor precision degradation. All the FDRs in T^2 are just over the critical value given by their FARs. These detected points mainly come from outliers or produce severe fluctuations. In fact, the sensor fault occurs because of the change of the measured device; and the fault points should be distributed in the residual space. The results of SPE indicate VAE gives the highest FDR while PCA and NPE cannot identify this fault. In F2, mode changes can be considered as a change of LVs because an external adjustment occurs. Considering the nonlinearity of this process, T^2 and SPE would simultaneously detect the fault points. Even though SPE in KPCA is slightly larger than VAE, T^2 in KPCA cannot present a correct conclusion. It is clear that the comprehensive performance of VAE in the two detection indices precedes those of the other methods while SDAE is in the second place. Regarding F3, T^2 in VAE is the smallest while SPE is the largest. This mostly conforms to the reality because the changed controller in F3 would cause the adjustment of variable correlations. Based on these three representative fault patterns, it is validated that VAE is able to achieve better monitoring performance.

V. CONCLUSIONS

In this paper, a novel VAE based process fault detection algorithm is proposed. VAE is constructed under a probabilistic deep learning framework for inferring LVs and generating observations. Simultaneously, by formalizing posterior distributions and conditional distributions, the orthogonal constraints of the latent variables are effectively incorporated into the VAE models based on the available process knowledge. It is particularly good for complex nonlinear systems. Compared with the past models, the proposed model has the following merits:

- It automatically extracts LVs through a deep neural network. It is more powerful than shallow methods, especially when handling complex nonlinear processes.
- Unlike most multivariate statistical analysis methods, there are no steps resorting to matrix decomposition in the proposed VAE method, so the large-scale data can still be applied and online monitoring is also highly efficient.
- Unlike general deep learning models, VAE can learn independent LVs easily and avoid information overlap.
- Since VAE provides a distribution estimate for LVs and residuals, more comprehensive detection indices instead of a point estimate can be designed for fault detection.

The better fault detection results of the proposed method have been proved by the numerical example and the paper machine process, both of which contain complex nonlinear elements. Even though this paper solves the most fundamental issue in learning orthogonal LVs under the assumption that LVs and noises are Gaussian, it actually formulizes a framework for more complex problems. For example, Student t distribution can be considered as a prior distribution for data with outliers. To obtain sparse LVs, Laplace distribution for LVs is a good choice. These deductions should be further validated in the future.

REFERENCES

[1] Z. Ge, Z. Song, S. X. Ding *et al.*, "Data mining and analytics in the process industry: the role of machine learning," *IEEE Access,* vol. 5, pp. 20590-20616, 2017.

[2] S. X. Ding, "Data-driven design of monitoring and diagnosis systems for dynamic processes: A review of subspace technique based schemes and some recent results," *Journal of Process Control,* vol. 24, no. 2, pp. 431-449, Feb, 2014.

[3] A. Bakdi, A. Kouadri, and A. Bensmail, "Fault detection and diagnosis in a cement rotary kiln using PCA with EWMA-based adaptive threshold monitoring scheme," *Control Engineering Practice,* vol. 66, pp. 64-75, 2017.

[4] U. Kruger, Y. Zhou, and G. W. Irwin, "Improved principal component monitoring of large-scale processes," *Journal of Process Control,* vol. 14, no. 8, pp. 879-888, 2004.

[5] L. Zhou, J. Zheng, Z. Ge *et al.*, "Multimode process monitoring based on switching autoregressive dynamic latent variable model," *IEEE Transactions on Industrial Electronics,* vol. 65, no. 10, pp. 8184-8194, 2018.

[6] X. Yuan, Y. Wang, C. Yang *et al.*, "Weighted linear dynamic system for feature representation and soft sensor application in nonlinear dynamic industrial processes," *IEEE Transactions on Industrial Electronics,* vol. 65, no. 2, pp. 1508-1517, 2018.

[7] S. Joe Qin, "Statistical process monitoring: basics and beyond," *Journal of chemometrics,* vol. 17, no. 8-9, pp. 480-502, 2003.

[8] K. Wang, J. Chen, and Z. Song, "Performance Analysis of Dynamic PCA for Closed-Loop Process Monitoring and Its Improvement by Output Oversampling Scheme," *IEEE Transactions on Control Systems Technology,* vol. 27, no. 1, pp. 378-385, 2019.

[9] J.-M. Lee, C. Yoo, S. W. Choi *et al.*, "Nonlinear process monitoring using kernel principal component analysis," *Chemical Engineering Science,* vol. 59, no. 1, pp. 223-234, 2004.

[10] Q. Jiang, and X. Yan, "Parallel PCA–KPCA for nonlinear process monitoring," *Control Engineering Practice,* vol. 80, pp. 17-25, 2018.

[11] X. Deng, X. Tian, S. Chen *et al.*, "Fault discriminant enhanced kernel principal component analysis incorporating prior fault information for monitoring nonlinear processes," *Chemometrics and Intelligent Laboratory Systems,* vol. 162, pp. 21-34, 2017.

[12] X. Deng, X. Tian, S. Chen *et al.*, "Nonlinear Process Fault Diagnosis Based on Serial Principal Component Analysis," *IEEE Transactions on*

Neural Networks and Learning Systems, vol. 29, no. 3, pp. 560-572, 2018.
[13] C. Wei, J. Chen, and Z. Song, "Multilevel MVU models with localized construction for monitoring processes with large scale data," *Journal of Process Control,* vol. 67, pp. 176-196, 2018.
[14] B. Song, S. Tan, and H. Shi, "Process monitoring via enhanced neighborhood preserving embedding," *Control Engineering Practice,* vol. 50, pp. 48-56, 2016.
[15] X. Li, F. Duan, P. Loukopoulos *et al.*, "Canonical variable analysis and long short-term memory for fault diagnosis and performance estimation of a centrifugal compressor," *Control Engineering Practice,* vol. 72, pp. 177-191, 2018.
[16] L. Jiang, Z. Song, Z. Ge *et al.*, "Robust self-supervised model and its application for fault detection," *Industrial & Engineering Chemistry Research,* vol. 56, no. 26, pp. 7503-7515, 2017.
[17] X. Yuan, B. Huang, Y. Wang *et al.*, "Deep Learning-Based Feature Representation and Its Application for Soft Sensor Modeling With Variable-Wise Weighted SAE," *IEEE Transactions on Industrial Informatics,* vol. 14, no. 7, pp. 3235-3243, 2018.
[18] Q. Jiang, and X. Yan, "Learning Deep Correlated Representations for Nonlinear Process Monitoring," *IEEE Transactions on Industrial Informatics*, pp. 1-1, 2018.
[19] Z. Zhang, T. Jiang, S. Li *et al.*, "Automated feature learning for nonlinear process monitoring – An approach using stacked denoising autoencoder and k-nearest neighbor rule," *Journal of Process Control,* vol. 64, pp. 49-61, 2018.
[20]D. Kim, and I.-B. Lee, "Process monitoring based on probabilistic PCA," *Chemometrics and intelligent laboratory systems,* vol. 67, no. 2, pp. 109-123, 2003.
[21] Z. Ge, and Z. Song, "Maximum-likelihood mixture factor analysis model and its application for process monitoring," *Chemometrics and Intelligent Laboratory Systems,* vol. 102, no. 1, pp. 53-61, 2010.
[22] H. Lu, Y. Meng, K. Yan *et al.*, "Kernel principal component analysis combining rotation forest method for linearly inseparable data," *Cognitive Systems Research*, 2018.
[23] X. Yuan, L. Ye, L. Bao *et al.*, "Nonlinear feature extraction for soft sensor modeling based on weighted probabilistic PCA," *Chemometrics and Intelligent Laboratory Systems,* vol. 147, no. Supplement C, pp. 167-175, 2015.
[24] D. P. Kingma, and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
[25] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.
[26] K. Wang, J. Chen, and Z. Song, "Fault diagnosis for processes with feedback control loops by shifted output sampling approach," *Journal of the Franklin Institute,* vol. 355, no. 7, pp. 3249-3273, 2018.
[27] E. Parzen, "On estimation of a probability density function and mode," *The annals of mathematical statistics,* vol. 33, no. 3, pp. 1065-1076, 1962.
[28] I. Goodfellow, Y. Bengio, A. Courville *et al.*, *Deep learning*: MIT press Cambridge, 2016.
[29] R. Caruana, S. Lawrence, and C. L. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping." pp. 402-408.
[30] A. R. Mugdadi, and I. A. Ahmad, "A bandwidth selection for kernel density estimation of functions of random variables," *Computational Statistics & Data Analysis,* vol. 47, no. 1, pp. 49-62, 2004.