

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

Kai Wang¹, Michael G. Forbes², Bhushan Gopaluni³, Junghui Chen⁴ and Zhihuan Song¹

¹State Key Laboratory of Industrial Control Technology, Zhejiang University, Hangzhou, 310027 Zhejiang, China

²Honeywell Process Solutions, North Vancouver, BC V7J 3S4, Canada

³Department of Chemical and Biological Engineering, The University of British Columbia, Vancouver, BC, Canada.

⁴Department 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 instead 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 uncertainties

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.

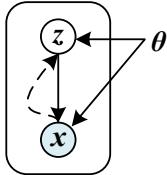


Fig. 1. Illustration of the generative model and the inference model

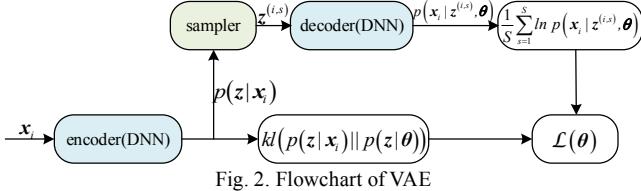


Fig. 2. Flowchart of VAE

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 \mathbf{x} is generated by a random process $p(\mathbf{x}, \mathbf{z}|\boldsymbol{\theta}) = p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})p(\mathbf{z}|\boldsymbol{\theta})$, where \mathbf{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 \mathbf{x} is generated by the conditional distribution $p(\mathbf{x}|\mathbf{z}, \boldsymbol{\theta})$, in which \mathbf{z} is sampled from the prior distribution $p(\mathbf{z}|\boldsymbol{\theta})$. 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 $\mathbf{X} = \{\mathbf{x}_i \in R^m, i = 1, 2, \dots, N\}$ with N independent observations. The goal is to estimate unknown parameters ($\boldsymbol{\theta}$)

and LVs by maximizing the log-likelihood function given by

$$\ln p(\mathbf{X}) = \sum_{i=1}^N \ln p(\mathbf{x}_i) \quad (1)$$

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),

$$\ln p(\mathbf{x}_i) = kl(q(z) || p(z|\mathbf{x}_i)) + L(q(z), \boldsymbol{\theta}) \quad (2)$$

where $q(z)$ 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 $q(z)$ and the posterior distribution $p(z|\mathbf{x}_i)$ given by

$$kl(q(z) || p(z|\mathbf{x}_i)) = \int q(z) \ln \frac{q(z)}{p(z|\mathbf{x}_i)} dz \quad (3)$$

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

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

$$L_i(q(z), \boldsymbol{\theta}) = \int q(z) \ln \left\{ \frac{p(\mathbf{x}_i, z | \boldsymbol{\theta})}{q(z)} \right\} dz \quad (4)$$

In most cases, the marginal distribution $p(\mathbf{x})$ 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 $q(z)$ is with $p(z|\mathbf{x}_i)$, the smaller the approximation error is. Especially, the approximation error will be zero when $q(z)$ is equal to $p(z|\mathbf{x}_i)$. Hence, at the maximum of the lower bound $L_i(q(z), \boldsymbol{\theta})$, $q(z)$ is chosen to be $p(z|\mathbf{x}_i)$. Substituting $q(z)$ with $p(z|\mathbf{x}_i)$, the lower bound of Eq.(4) can be rewritten as

$$L_i(\boldsymbol{\theta}) = E_{p(z|\mathbf{x}_i)}(\ln p(\mathbf{x}_i | z, \boldsymbol{\theta})) - kl(p(z|\mathbf{x}_i) || p(z|\boldsymbol{\theta})) \quad (5)$$

where $E_{p(z|\mathbf{x}_i)}(\ln p(\mathbf{x}_i | z, \boldsymbol{\theta}))$ denotes the expectation of $\ln p(\mathbf{x}_i | z, \boldsymbol{\theta})$ w.r.t $p(z|\mathbf{x}_i)$. To evaluate the loss function as Eq.(5), it first needs the estimation of the posterior distribution $p(z|\mathbf{x}_i)$, called an encoding process, which is used to infer the LVs (codes) z given an observation \mathbf{x}_i . Then the conditional likelihood $\ln p(\mathbf{x}_i | z, \boldsymbol{\theta})$ in Eq.(5), called a decoding process, should be evaluated. It is used to generate the observation given the codes. The second term $kl(p(z|\mathbf{x}_i) || p(z|\boldsymbol{\theta}))$ 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.

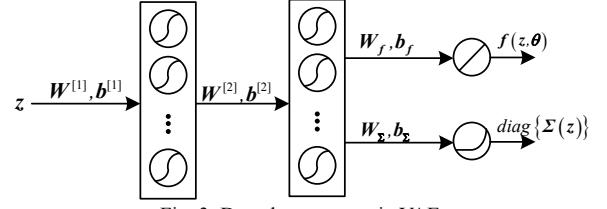


Fig. 3. Decoder structure in VAE

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

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

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

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

$$\mathbf{x} = \mathbf{f}(z, \boldsymbol{\theta}) + \mathbf{w}(z) \quad (7)$$

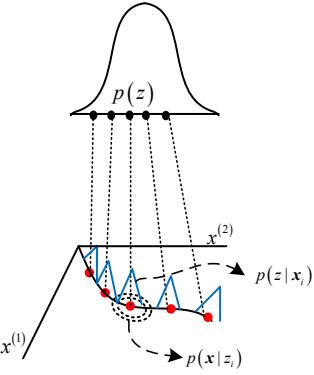


Fig. 4. Illustration of distributions in VAE

where $\mathbf{w} = N(\boldsymbol{\theta}, \Sigma(z))$ 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. $f(z, \theta) \in \mathbb{R}^n \rightarrow \mathbb{R}^m$ 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:

$$p(\mathbf{x} | z, \theta) = N(f(z, \theta), \Sigma(z)) \quad (8)$$

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

$$f(z, \theta) = \mathbf{W}_f \tanh(\mathbf{W}^{[2]} \tanh(\mathbf{W}^{[1]} z + \mathbf{b}^{[1]}) + \mathbf{b}^{[2]}) + \mathbf{b}_f \quad (9)$$

$$\text{diag}\{\Sigma(z)\} = \zeta(\mathbf{W}_{\Sigma} \tanh(\mathbf{W}^{[2]} \tanh(\mathbf{W}^{[1]} z + \mathbf{b}^{[1]}) + \mathbf{b}^{[2]}) + \mathbf{b}_{\Sigma}) \quad (10)$$

where $\text{diag}\{\Sigma(z)\}$ 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 $\{\mathbf{W}^{[1]}, \mathbf{b}^{[1]}, \mathbf{W}^{[2]}, \mathbf{b}^{[2]}\}$. $f(z, \theta)$ is outputted through a linear unit with parameters $\{\mathbf{W}_f, \mathbf{b}_f\}$ because of the unlimited range of expectations. In contrast, $\text{diag}\{\Sigma(z)\}$ 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 \mathbf{W}_{Σ} and the bias \mathbf{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.,

$$p(z) = N(\boldsymbol{\theta}, \mathbf{I}) \quad (11)$$

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 \mathbf{x} conditioned in z . It is obvious that the posterior distribution $p(z|x)$ 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 ($p(z|x_i)$) 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.

$$p(z|x_i) = N(\boldsymbol{\mu}(x_i), \mathbf{V}(x_i)) \quad (12)$$

where $\boldsymbol{\mu}(x_i)$ and $\mathbf{V}(x_i)$ 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, $\mathbf{V}(x_i)$ 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:

$$\begin{aligned} & \text{kl}(p(z|x_i) || p(z)) = \\ & \frac{1}{2} \left\{ \boldsymbol{\mu}(x_i)^T \boldsymbol{\mu}(x_i) - \ln(|\mathbf{V}(x_i)|) + \text{tr}(\mathbf{V}(x_i)) - n \right\} \end{aligned} \quad (13)$$

where $\text{tr}(\bullet)$ refers to the trace of one squared matrix. As shown in Fig. 2, the output of the encoder network will be $\boldsymbol{\mu}(x_i)$ and the diagonal elements of $\mathbf{V}(x_i)$. 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 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

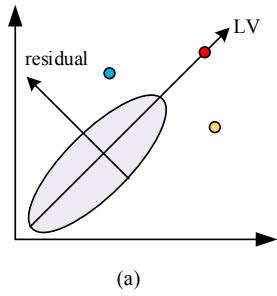
End For

End For

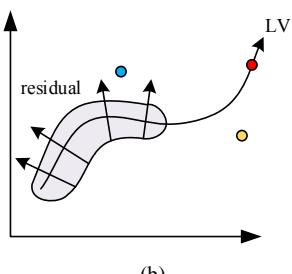
End

3behind reparameterization is that the distribution in Eq.(12)
4can be regarded as an affine transformation of the normal
5distribution $p(\boldsymbol{\varepsilon}) = N(\boldsymbol{\theta}, \mathbf{I})$, i.e.

$$6 \quad z = V(\mathbf{x}_i)^{-\frac{1}{2}} \boldsymbol{\varepsilon} + \boldsymbol{\mu}(\mathbf{x}_i) \quad (14)$$

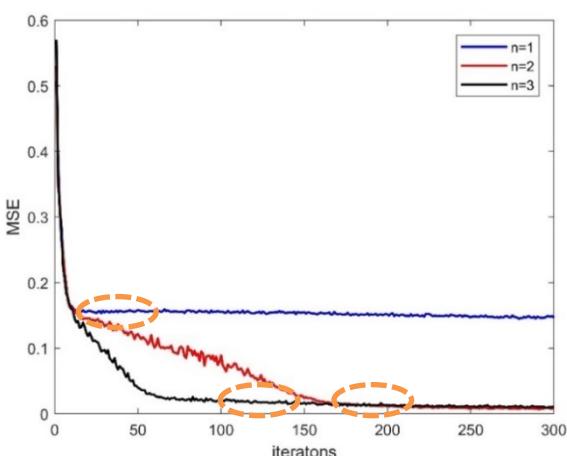


(a)



(b)

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



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

17 which is differentiable. Each point $z^{(i,s)}$ required in VAE
18 shown in Fig. 2 is given by $z^{(i,s)} = V(\mathbf{x}_i)^{-\frac{1}{2}} \boldsymbol{\varepsilon}^{(i,s)} + \boldsymbol{\mu}(\mathbf{x}_i)$,
19 $s = 1, \dots, S$, where $\boldsymbol{\varepsilon}^{(i,s)}$ represents a sample from the normal
20 distribution. By reparameterization, the orthogonal latent
21 variables become learnable.

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

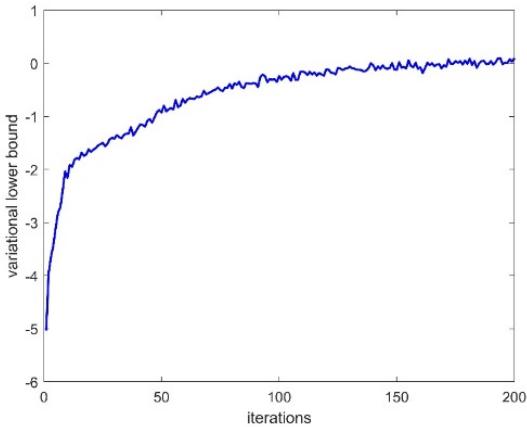
$$42 \quad MSE = \frac{1}{N_v} \sum_{i=1}^{N_v} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2 \quad (15)$$

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

$$45 \quad \hat{\mathbf{x}}_i = f(z^{(i,s)}, \boldsymbol{\theta}) \quad (16)$$

46 where $z^{(i,s)}$ is one of the sampling points of LVs based on the
47 encoder network. The optimal number of iterations denoted as
48 $n = n^{opt}$ is obtained when MSE tends to be stationary or has
49 not significantly improved. By incrementing n gradually,
50 the one with a minimum MSE of the validation dataset is
51 considered to be the optimal number of LVs. In this paper, the
52 upper bound of n can be heuristically determined by PCA,

1 therefore about 80% cumulative variance contribution is
2 chosen for the number of principal



3 Fig. 7. The trend of the variational lower bound in the numerical example.
4

5 components. The complete learning algorithm is given in
6 Algorithm 1.

8B. PROCESS MONITORING

9 After the VAE-based process model is developed, a latent
10 space and a residual space can be obtained by the encoder and
11 the decoder, respectively. Instead of point estimates of LV and
12 the residual for a specific observation, VAE offers distribution
13 descriptions, giving more information than a point estimate.
14 To make the full use of the distribution information, the
15 monitoring indices should be constructed by the posterior
16 PDF ($p(z|x_i)$) and the conditional PDF ($p(x_i|z)$).

17 1) MONITORING INDEX IN LATENT SPACE

18 According to the lower bound of the likelihood in Eq.(5), the
19 objective with respect to the KL divergence
20 $kl(p(z|x_i)||p(z))$ makes the posterior distribution as
21 close to the prior distribution as possible. Therefore, an
22 abnormal sample will have a large KL divergence because the
23 posterior will be dissimilar to the prior. The KL divergence
24 can be considered as a monitoring index D_i in the latent
25 space, as in Eq.(13). To define the normal operating region or
26 control limits, PDF of D_i described by a known density
27 function such as Gaussian or normal PDF cannot be
28 guaranteed. To overcome the limitation, KDE is used to
29 estimate the distribution of the monitoring index.

$$30 \quad p(D) = \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{D-D_i}{h}\right) \quad (17)$$

31 where $K(\cdot)$ is a kernel function with the constraints
32 $\int K(x)dx=1$ and $K(x)\geq 0$. h is a hyperparameter
33 known as the bandwidth to adjust the smoothness of the
34 kernel function. The Gaussian kernel function is used in this
35 paper and the bandwidth is determined by the empirical
36 criterion as follows, derived by Mugdadi and Ahmad [30]

$$37 \quad h = 1.06\sigma N^{-0.2}$$

38(18)
39 where σ is the standard deviation of the sample. Based on
40 the distribution, a control limit D_{lim} can be designed under a

41 given confidence level α such as 95%; i.e., the minimum
42 D_{lim} satisfies

$$43 \quad P(D) = \int_{-\infty}^{D_{lim}} p(D)dD \geq \alpha \quad (19)$$

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

47 2) MONITORING INDEX IN RESIDUAL SPACE

48 Similarly, the expectation of the conditional log-likelihood
49 $E_{p(z|x_i)}(\ln p(x_i|z,\theta))$ in Eq.(5) can represent an index
50 measuring the possibility of the observation drawn from the
51 conditional distribution. The larger $\ln p(x_i|z,\theta)$ indicates
52 the observation highly follows the distribution $p(x_i|z,\theta)$.
53 According to Eq.(8), there is

$$54 \quad \begin{aligned} \ln p(x_i|z,\theta) = \\ -\frac{1}{2}(x_i - f(z))\Sigma^{-1}(z)(x_i - f(z)) - \frac{1}{2}|\Sigma(z)| - m\pi \end{aligned} \quad (20)$$

55 One can see $\ln p(x_i|z,\theta)$ is similar to the indices in the
56 residual space and $\ln p(x_i|z,\theta)$ represents the distance
57 between the observations x_i and the reconstructed values
58 $f(z)$. Moreover, it simultaneously considers the uncertainty
59 measured by the covariance matrix $\Sigma(z)$. Hence,
60 $\ln p(x_i|z,\theta)$ is able to play the role of detecting the
61 anomaly related to residuals. To make $f(z)$ and $\Sigma(z)$
62 tractable, the point $z^{(i,s)}$ sampled from $p(z|x_i)$ is used to
63 calculate the final index, given by

$$64 \quad R_{i,s} = -\ln p(x_i|z^{(i,s)},\theta) - m\pi = \\ \frac{1}{2}(x_i - f(z^{(i,s)}))\Sigma^{-1}(z^{(i,s)})(x_i - f(z^{(i,s)})) + \frac{1}{2}|\Sigma(z^{(i,s)})| \quad (21)$$

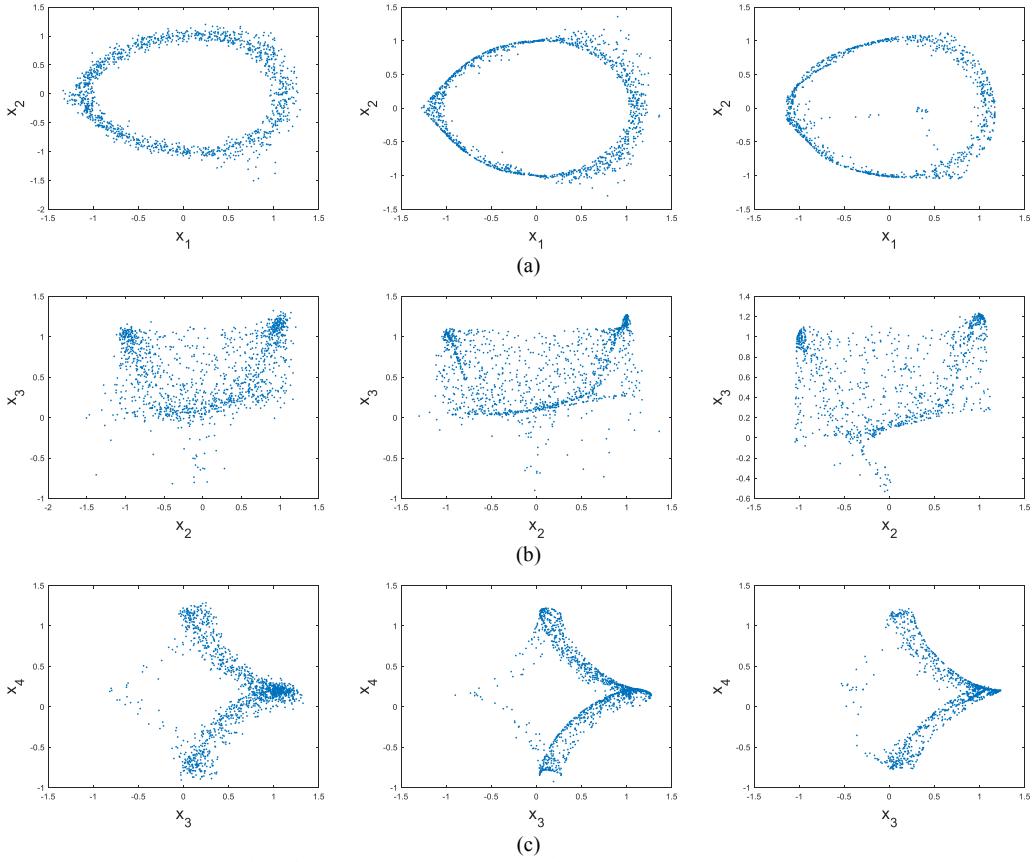
65 The larger R_i is, the more likely x_i is an abnormal point,
66 but it is hard to produce a closed form of the expectation
67 because of the nonlinear representation. Here an empirical

$$68 \quad \text{average } R_i = \frac{1}{S} \sum_{s=1}^S \ln p(x_i|z^{(i,s)},\theta) \text{ introduced in Section 2}$$

69 can be used for monitoring the anomaly in the residual space.
70 As mentioned before, S can be chosen as 1. With the
71 estimation of PDF of the R index, like the determination of
72 the control limit in the D index, the corresponding control
73 limit in the R index can be determined.

74

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



1

3

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

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

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

22methods, just learn the given training set. Nonlinear methods
 23may not learn these patterns caused by larger LVs which do
 24not occur in the training sets. Take Fig. 5(b) for example. The
 25nonlinear model is a local description of a training set. For the
 26red point, a larger LV would be captured by T^2 but SPE may
 27also respond to it because the LV model with the collected
 28training set cannot cover all the other unknown patterns.
 29Hence, there is probably a significant reconstruction error
 30caught by SPE. In Fig. 5(b), the behaviors of the other two
 31kinds of anomalies (the blue point and the yellow point) look
 32like those in Fig. 5(a) of the linear system.
 33

34To sum up, the algorithm with the proposed VAE based
 35process monitoring algorithm are listed as follows:

36Step 1. Normalize process variables with sample means
 37and standard deviations.

38Step 2. Organize the training set by randomly sampling 80%
 39from the dataset and the remaining 20% of the
 40dataset is taken as the validation set.

41Step 3. Determine the upper bound of the number of LVs
 42by PCA.

43Step 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
 45training set.

47iii. Evaluate the validation error in each iteration.

48iv. Record the optimal number of iterations in the
 49current n through an early stopping strategy.
 50Record the validation error under the optimal
 51number of iterations.

- 1 v. Increment n and return back to ii until n is up to
2 the upper bound.
- 3 Step 5. Determine the optimal n and the corresponding
4 number of iterations based on the minimum
5 validation error.
- 6 Step 6. Retrain the model with Algorithm 1 using the
7 whole dataset under the predefined optimal
8 hyperparameters.
- 9 Step 7. Output the two defined detection indices using the
10 normal data set and determine the control limits.
- 11 Step 8. For any new sample, normalize it using the means
12 and standard deviations of the normal dataset.
13 Calculate the two indices by feeding it into the
14 trained model and compare the indices with the
15 corresponding control limits. If yes, keep
16 monitoring the next new data points; otherwise,
17 further analyze what caused the abnormal situation.
18

54

$$x_1 = 0.1z_1 + z_1 / \sqrt{z_1^2 + z_2^2} + w_1$$

$$x_2 = 0.1z_1 z_2 + z_2 / \sqrt{z_1^2 + z_2^2} + w_2$$

$$x_3 = t \cos^3 z_1 + 0.1 e^{i \pi z_2} + w_3$$

$$x_4 = \sin^3 z_1 + 0.2 \ln(2 + \cos z_2) + w_4$$

55 where t is an adjustable coefficient and is 1 when the
56 system is normal. z_1 and z_2 are LVs subject to unit
57 Gaussian distributions. $w_i, i=1,2,3,4$ are zero-mean
58 Gaussian measurement noises with standard deviations
59 0.05, 0.06, 0.05, 0.04, respectively. A total of 1,200 normal
60 observations are generated and normalized. Among them,
61,000 observations are randomly selected as the training set
62 and the remaining 200 points are organized as the validation
63 set. With PCA, the singular values of the covariance matrix of
64 the training set are [1.83, 1.14, 0.84, 0.18]. Therefore, the
65 maximum number of LVs is 3 using the cumulative variance
66 of 80%. To construct the VAE model, both the encoder and
67 the decoder are built by three hidden-layer feedforward
68 networks, in which each layer contains 30 neurons. The
69 maximum iterations are set up as 300. Fig. 6 presents the
70 trends of MSE as the iteration time evolves with the three
71 different numbers of LVs from 1 to 3. According to the early
72 stopping strategy, the optimal iteration number is marked by a
73 circle point in Fig. 6. Among the three different LVs, the
74 minimum MSE is obtained at $n=2$ and with the
75 corresponding 200 iterations. Then, the 1,000 training
76 samples and the 200 validation samples are concatenated and
77 used to retrain the network with these determined
78 hyperparameters. The trend of the variational lower bound is
79 illustrated in Fig. 7. One can see the lower bound gradually
80 increases as the iteration proceeds and it tends to be steady
81 when the number of iterations is close to 200. Taking the
82 parameters updated at the 200th iterations as the final network
83 parameters, the posterior distribution and the conditional
84 distribution can be obtained by feeding each normal sample to
85 the network. In this paper, the means of the posterior play the
86 role of point estimates of LVs for each observation. The
87 covariance matrix of the means of the posterior for all the
88 observations are $\begin{bmatrix} 0.87 & 0.08 \\ 0.08 & 0.87 \end{bmatrix}$, which implies there is little
89 correlation between the two LVs and the learned LVs are
90 orthogonal to each other. This satisfies the required VAE
91 model; there is no overlapped and redundant information
92 among different LVs. To visualize the ability of signal
93 reconstruction of VAE, the scatter plots between two different
94 observed variables are shown in Fig. 8. Fig. 8 presents the
95 scatter plots of (x_1, x_2) , (x_2, x_3) , and (x_3, x_4) from top to
96 the bottom. The left column shows the observed values
97 contaminated by measurement noises. The middle column
98 presents the true observation values without considering
99 noises so that the contour in the middle column is more
100 distinct than the left column. The right column gives the
101 reconstructed values with the LVs. The reconstructed values
102 in the right column are close to the true values in the middle

19 IV. CASE STUDIES

20 In this section, the feasibility and efficiency of the proposed
21 method are evaluated by two examples, including a numerical
22 example and an industrial process example. The numerical
23 example is created artificially. Then the proposed method is
24 applied to a real industrial process, a more challenging test
25 bed for process monitoring. The proposed VAE in this paper
26 is compared with several conventional data-driven fault
27 detection methods, including PCA, KPCA, NPE, and SDAE.
28 Among them, PCA is a benchmark approach to process
29 monitoring. KPCA is a popular representative of kernel
30 methods for process monitoring. Attempting different
31 commonly used kernels for KPCA, the sigmoid kernel
32 $k(\mathbf{x}, \mathbf{y}) = \tanh(\beta_0 \mathbf{x}^T \mathbf{y} + \beta_1)$ has better monitoring
33 performance than the other kernels such as the polynomial
34 kernel and the radial basis kernel in this example. Hence,
35 KPCA with the sigmoid kernel ($\beta_0 = 1$ and $\beta_1 = 0$) is used.
36 NPE is one of the manifold learning methods. Here the
37 number of LVs in the three comparative methods is
38 determined by the cumulative variance of 80%. Regarding
39 SDAE, it is a deep feature extraction method based on deep
40 learning, and the number of LVs is set up to be the same as
41 the proposed method. Note that the detection indices of
42 different methods in the LV space play a similar role of
43 measuring the variability in LVs, though different methods
44 may nominate different indices such as T² in PCA, HD in
45 SDAE[19] and D in VAE. For convenience, in this paper, all
46 these indices are commonly referred to as T². Likewise, all the
47 indices in the residual space are commonly referred to as SPE.
48 The deep models were trained in the environment of NVIDIA
49 GeForce GTX 1060.

50 A. NUMERICAL EXAMPLE

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

103
104

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

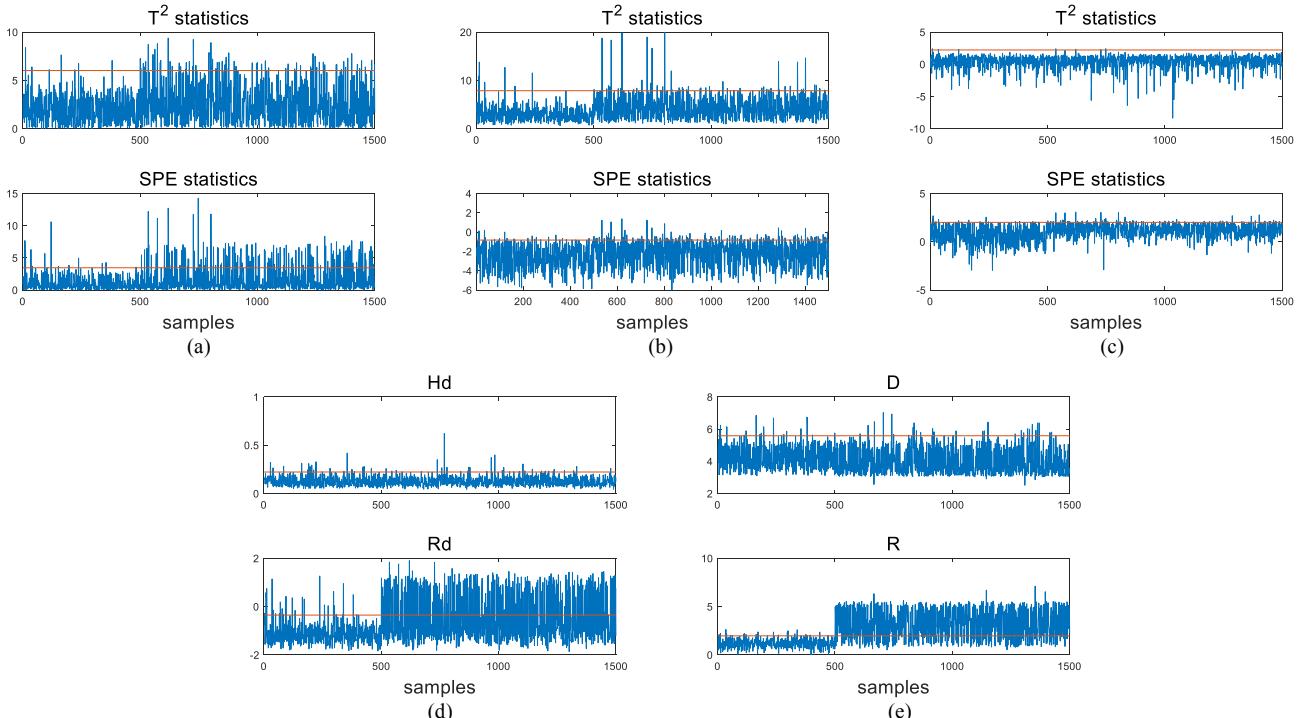


Fig. 9. The control charts of (a) PCA, (b) KPCA, (c) NPE, (d) SDAE, and (e) VAE for F10 in the numerical example.

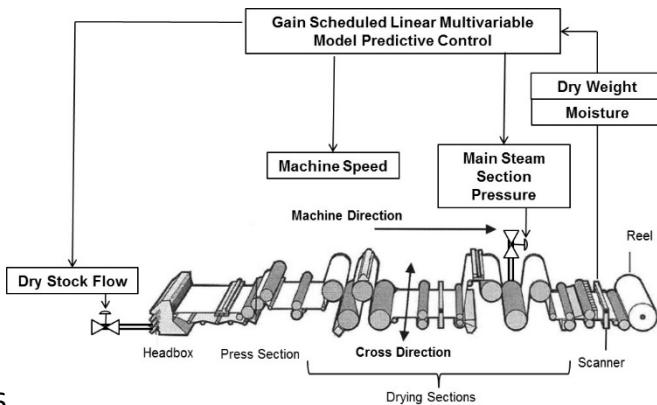


Fig. 10. Structure of a typical industrial paper machine.

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

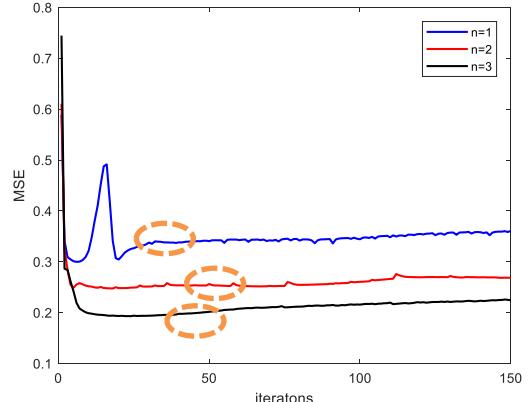


Fig. 11. The validation errors for different numbers of LVs in the paper machine.

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

13

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

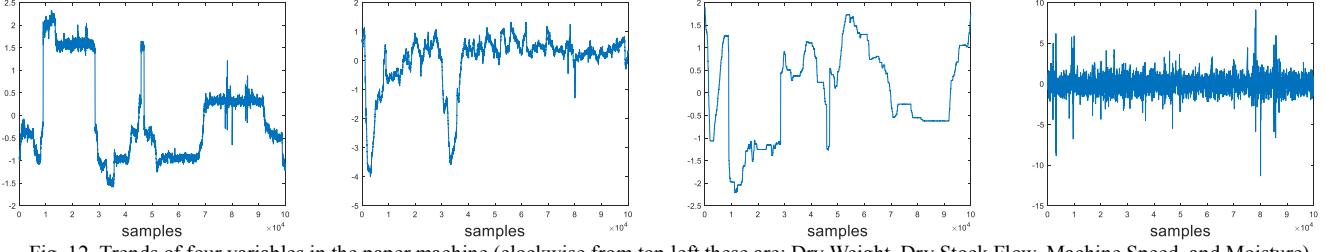


Fig. 12. Trends of four variables in the paper machine (clockwise from top left these are: Dry Weight, Dry Stock Flow, Machine Speed, and Moisture)

1
2
3
4

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

5magnitudes; one wants to test the sensitivity of the fault
6detection methods. With the 95% confidence level, Table II
7summarizes the false alarm rates (FARs), referring to the ratio
8of false alarm numbers to the total numbers for normal data. It
9is reasonable when FARs are less than 5% because of the 95%
10confidence level. One can see from Table II that all the
11methods have an eligible FAR. The fault detection rate (FDR)
12is the ratio of the samples with detection indices beyond the
13control limits to the total samples. A decisive performance of
14FDR that evaluates these methods for all the fault data is
15listed in Table III. In the table, the largest FDRs in the two
16detection indices for each fault are bold. F1-F6 are faults in
17LVs. Among them, F1, F2, and F3 are bias faults in the first
18LV while F4, F5 and F6 are noise faults in the second LV. T^2
19and SPE should catch these two kinds of anomalies based on
20the analysis in the remark. It is found that VAE substantially
21behaves better than the other methods. Specifically, T^2 in VAE
22for F1-F4 outperforms the other methods. SDAE presents a
23smaller preponderance of F5 and F6 over VAE. Especially, for
24the three noise faults (F4-F6), FDRs of other shallow methods
25in T^2 even cannot exceed 5%, causing a misleading statement
26that the root cause does not come from LVs. The reason is
27that shallow methods provide a very poor estimate for true
28data distribution. F7-F12 are structural faults, F7-F9 suffer a
29sensor precision degradation, and F10-F12 simulate a varying
30variable correlation. In these situations, SPE would be
31sensitive while T^2 is immune to these kinds of faults because
32these faults will not result in a wide range of fluctuations. It is
33clear that both NPE and VAE give correct judgment because
34SPE in the two methods is out of control while T^2 is still
35under control. Moreover, SPE of VAE has larger FDRs than
36that of NPE. In contrast, PCA, KPCA and SDAE would
37mislead engineers into considering a change of external
38exciting signals because there are some out-of-control points
39in T^2 . Substantially, the proposed process model based on
40VAE outperforms the other methods in inferring fault
41locations and sensitivity of detecting faults. Taking F10 for
42example, the control chart for each method is plotted in Fig. 8.
43In the figure, the former 500 samples are normal and the latter
441,000 samples are abnormal.

45B. INDUSTRIAL EXAMPLE

46An industrial paper machine process is studied in this work.
47Paper machines, such as the one shown in Fig. 10, transform

48stock, which is a suspension of wood cellulose fibres in a
49water solution, into a web of paper which is wound onto a reel.
50The direction the paper moves along the paper machine is
51known as machine direction (MD), while the direction
52perpendicular to this is known as cross direction (CD).
53Quality variables of the paper must be kept uniform along
54both the machine and cross directions, but typically the MD
55and CD control problems are handled independently. In this
56study, data from the MD process is examined. Paper quality
57measurements of average dry weight (the weight of the paper
58less any remaining moisture on a per area basis) and the
59average moisture content across the web are taken by a
60scanning sensor at the end of the machine. These quality
61variables are controlled by adjusting the dry stock flow (the
62volumetric flow of the solids in the stock), the steam
63pressures in heated metal cylinders in the drying sections, and
64the machine speed. The MD process is a multivariable process
65with sufficiently nonlinear behavior that a common industrial
66practice is to apply a gain scheduled linear model predictive
67control with different models for different operating regions.
68In this study, 9 process variables are examined, including
69actuator signals, setpoints, sensor signals and mode signals. A
70dataset with 100,000 points is collected and judged as normal
71data by engineers. To show the complexity of the process, the
72trends of 4 normalized variables in the training set are plotted
73in Fig. 11. It is found that most of the variables present strong
74nonlinear fluctuations. To construct a VAE model, likewise,
75PCA is used to determine the upper bound of the number of
76LVs. The singular values of the 9 variables in the covariance
77matrix are [4.3, 3.2, 1.0, 0.7, 0.3, 0.05, 0.007, 0.0007, 0.0003].
78Based on the 80% cumulative variance contribution, the first
79three LVs are used as the candidates. To verify the training
80models with different selected LVs, the collected dataset is
81separated into a training set and a validation set. The MSEs on
82the validation set for the chosen three different number of LVs
83are shown in Fig. 12. Based on the early stopping strategy, the
84optimal iterations are marked in circle points (Fig. 12).
85Among the LV models, the model with three LVs is selected
86as the minimum MSE is obtained for this paper machine
87process after 50 iterations. With these trained
88hyperparameters, the posterior distribution and the conditional
89distribution for each sample can be obtained and the means of
90the posterior distribution are regarded as a point estimate of
91LVs. The covariance matrix of the means of all the training

1data is

$$2 \begin{bmatrix} 1.01 & -0.09 & 0.03 \\ -0.09 & 0.8 & -0.08 \\ 0.03 & -0.08 & 0.8 \end{bmatrix}$$

3As the off-diagonal entries of the matrix are close to zero, all
4the LVs can be considered as being orthogonal to each other.
5This satisfies the required VAE model.

6

7The proposed method, as well as several comparative
8methods (PCA, KPCA, NPE, and SDAE), is applied to
9detecting the anomalies of this paper machine. As the
10estimated models of KPCA and NPE are computationally
11expensive for large-scale data, the size of the dataset is
12beyond their tractability in the configuration of our own
13computer system in this example. Hence, a downsampling
14strategy is performed for KPCA and NPE. Three kinds of
15testing data are collected and described as follows:

- 16 F1: A controlled variable is added with an additional
17 sensor noise.
18 F2: A sequence of data with the operation modes which
19 are not included in the normal training data.
20 F3: A sequence of data produced with a changed
21 controller.

22Each fault scenario contains 30,000 samples. Given a 95%
23confidence level, control limits can be calculated for different
24methods. The results of the FARs listed in Table IV indicate
25the shallow methods, PCA, KPCA, and NPE, cannot
26sufficiently fit the data distribution as FARs in both T^2 and/or
27SPE statistics exceed 5%. Further, FDRs for the three testing
28data are given in Table V. For F1, it is a scenario of sensor
29precision degradation. All the FDRs in T^2 are just over the
30critical value given by their FARs. These detected points
31mainly come from outliers or produce severe fluctuations. In
32fact, the sensor fault occurs because of the change of the
33measured device; and the fault points should be distributed in
34the residual space. The results of SPE indicate VAE gives the
35highest FDR while PCA and NPE cannot identify this fault. In
36F2, mode changes can be considered as a change of LVs
37because an external adjustment occurs. Considering the
38nonlinearity of this process, T^2 and SPE would
39simultaneously detect the fault points. Even though SPE in
40KPCA is slightly larger than VAE, T^2 in KPCA cannot present
41a correct conclusion. It is clear that the comprehensive
42performance of VAE in the two detection indices precedes
43those of the other methods while SDAE is in the second place.
44Regarding F3, T^2 in VAE is the smallest while SPE is the
45largest. This mostly conforms to the reality because the
46changed controller in F3 would cause the adjustment of
47variable correlations. Based on these three representative fault
48patterns, it is validated that VAE is able to achieve better
49monitoring performance.

50V. CONCLUSIONS

51In this paper, a novel VAE based process fault detection
52algorithm is proposed. VAE is constructed under a
53probabilistic deep learning framework for inferring LVs and
54generating observations. Simultaneously, by formalizing
55posterior distributions and conditional distributions, the
56orthogonal constraints of the latent variables are effectively
57incorporated into the VAE models based on the available

58process knowledge. It is particularly good for complex
59nonlinear systems. Compared with the past models, the
60proposed model has the following merits:

- 61● It automatically extracts LVs through a deep neural
62 network. It is more powerful than shallow methods,
63 especially when handling complex nonlinear processes.
64● Unlike most multivariate statistical analysis methods,
65 there are no steps resorting to matrix decomposition in
66 the proposed VAE method, so the large-scale data can
67 still be applied and online monitoring is also highly
68 efficient.
69● Unlike general deep learning models, VAE can learn
70 independent LVs easily and avoid information overlap.
71● Since VAE provides a distribution estimate for LVs and
72 residuals, more comprehensive detection indices instead
73 of a point estimate can be designed for fault detection.
74The better fault detection results of the proposed method have
75been proved by the numerical example and the paper machine
76process, both of which contain complex nonlinear elements.
77Even though this paper solves the most fundamental issue in
78learning orthogonal LVs under the assumption that LVs and
79noises are Gaussian, it actually formulates a framework for
80more complex problems. For example, Student t distribution
81can be considered as a prior distribution for data with outliers.
82To obtain sparse LVs, Laplace distribution for LVs is a good
83choice. These deductions should be further validated in the
84future.

85REFERENCES

- 86[1] Z. Ge, Z. Song, S. X. Ding *et al.*, “Data mining and analytics in the
87 process industry: the role of machine learning,” *IEEE Access*, vol. 5, pp.
88 20590-20616, 2017.
89[2] S. X. Ding, “Data-driven design of monitoring and diagnosis systems for
90 dynamic processes: A review of subspace technique based schemes and
91 some recent results,” *Journal of Process Control*, vol. 24, no. 2, pp.
92 431-449, Feb, 2014.
93[3] A. Bakdi, A. Kouadri, and A. Bensmail, “Fault detection and diagnosis
94 in a cement rotary kiln using PCA with EWMA-based adaptive
95 threshold monitoring scheme,” *Control Engineering Practice*, vol. 66,
96 pp. 64-75, 2017.
97[4] U. Kruger, Y. Zhou, and G. W. Irwin, “Improved principal component
98 monitoring of large-scale processes,” *Journal of Process Control*, vol.
99 14, no. 8, pp. 879-888, 2004.
100[5] L. Zhou, J. Zheng, Z. Ge *et al.*, “Multimode process monitoring based
101 on switching autoregressive dynamic latent variable model,” *IEEE
102 Transactions on Industrial Electronics*, vol. 65, no. 10, pp. 8184-8194,
103 2018.
104[6] X. Yuan, Y. Wang, C. Yang *et al.*, “Weighted linear dynamic system for
105 feature representation and soft sensor application in nonlinear dynamic
106 industrial processes,” *IEEE Transactions on Industrial Electronics*, vol.
107 65, no. 2, pp. 1508-1517, 2018.
108[7] S. Joe Qin, “Statistical process monitoring: basics and beyond,” *Journal
109 of chemometrics*, vol. 17, no. 8-9, pp. 480-502, 2003.
110[8] K. Wang, J. Chen, and Z. Song, “Performance Analysis of Dynamic
111 PCA for Closed-Loop Process Monitoring and Its Improvement by
112 Output Oversampling Scheme,” *IEEE Transactions on Control Systems
113 Technology*, vol. 27, no. 1, pp. 378-385, 2019.
114[9] J.-M. Lee, C. Yoo, S. W. Choi *et al.*, “Nonlinear process monitoring
115 using kernel principal component analysis,” *Chemical Engineering
116 Science*, vol. 59, no. 1, pp. 223-234, 2004.
117[10] Q. Jiang, and X. Yan, “Parallel PCA-KPCA for nonlinear process
118 monitoring,” *Control Engineering Practice*, vol. 80, pp. 17-25, 2018.
119[11] X. Deng, X. Tian, S. Chen *et al.*, “Fault discriminant enhanced kernel
120 principal component analysis incorporating prior fault information for
121 monitoring nonlinear processes,” *Chemometrics and Intelligent
122 Laboratory Systems*, vol. 162, pp. 21-34, 2017.
123[12] X. Deng, X. Tian, S. Chen *et al.*, “Nonlinear Process Fault Diagnosis
124 Based on Serial Principal Component Analysis,” *IEEE Transactions on*

- 1 Neural Networks and Learning Systems, vol. 29, no. 3, pp. 560-572,
 2 2018.
- 3 [13] C. Wei, J. Chen, and Z. Song, "Multilevel MVU models with localized
 4 construction for monitoring processes with large scale data," *Journal of
 5 Process Control*, vol. 67, pp. 176-196, 2018.
- 6 [14] B. Song, S. Tan, and H. Shi, "Process monitoring via enhanced
 7 neighborhood preserving embedding," *Control Engineering Practice*,
 8 vol. 50, pp. 48-56, 2016.
- 9 [15] X. Li, F. Duan, P. Loukopoulos *et al.*, "Canonical variable analysis and
 10 long short-term memory for fault diagnosis and performance estimation
 11 of a centrifugal compressor," *Control Engineering Practice*, vol. 72, pp.
 12 177-191, 2018.
- 13 [16] L. Jiang, Z. Song, Z. Ge *et al.*, "Robust self-supervised model and its
 14 application for fault detection," *Industrial & Engineering Chemistry
 15 Research*, vol. 56, no. 26, pp. 7503-7515, 2017.
- 16 [17] X. Yuan, B. Huang, Y. Wang *et al.*, "Deep Learning-Based Feature
 17 Representation and Its Application for Soft Sensor Modeling With
 18 Variable-Wise Weighted SAE," *IEEE Transactions on Industrial
 19 Informatics*, vol. 14, no. 7, pp. 3235-3243, 2018.
- 20 [18] Q. Jiang, and X. Yan, "Learning Deep Correlated Representations for
 21 Nonlinear Process Monitoring," *IEEE Transactions on Industrial
 22 Informatics*, pp. 1-1, 2018.
- 23 [19] Z. Zhang, T. Jiang, S. Li *et al.*, "Automated feature learning for
 24 nonlinear process monitoring – An approach using stacked denoising
 25 autoencoder and k-nearest neighbor rule," *Journal of Process Control*,
 26 vol. 64, pp. 49-61, 2018.
- 27 [20] D. Kim, and I.-B. Lee, "Process monitoring based on probabilistic PCA,"
 28 *Chemometrics and intelligent laboratory systems*, vol. 67, no. 2, pp.
 29 109-123, 2003.
- 30 [21] Z. Ge, and Z. Song, "Maximum-likelihood mixture factor analysis
 31 model and its application for process monitoring," *Chemometrics and
 32 Intelligent Laboratory Systems*, vol. 102, no. 1, pp. 53-61, 2010.
- 33 [22] H. Lu, Y. Meng, K. Yan *et al.*, "Kernel principal component analysis
 34 combining rotation forest method for linearly inseparable data,"
 35 *Cognitive Systems Research*, 2018.
- 36 [23] X. Yuan, L. Ye, L. Bao *et al.*, "Nonlinear feature extraction for soft
 37 sensor modeling based on weighted probabilistic PCA," *Chemometrics
 38 and Intelligent Laboratory Systems*, vol. 147, no. Supplement C, pp.
 39 167-175, 2015.
- 40 [24] D. P. Kingma, and M. Welling, "Auto-encoding variational bayes,"
 41 *arXiv preprint arXiv:1312.6114*, 2013.
- 42 [25] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint
 43 arXiv:1606.05908*, 2016.
- 44 [26] K. Wang, J. Chen, and Z. Song, "Fault diagnosis for processes with
 45 feedback control loops by shifted output sampling approach," *Journal of
 46 the Franklin Institute*, vol. 355, no. 7, pp. 3249-3273, 2018.
- 47 [27] E. Parzen, "On estimation of a probability density function and mode,"
 48 *The annals of mathematical statistics*, vol. 33, no. 3, pp. 1065-1076,
 49 1962.
- 50 [28] I. Goodfellow, Y. Bengio, A. Courville *et al.*, *Deep learning*: MIT press
 51 Cambridge, 2016.
- 52 [29] R. Caruana, S. Lawrence, and C. L. Giles, "Overfitting in neural nets:
 53 Backpropagation, conjugate gradient, and early stopping." pp. 402-408.
- 54 [30] A. R. Mugdadi, and I. A. Ahmad, "A bandwidth selection for kernel
 55 density estimation of functions of random variables," *Computational
 56 Statistics & Data Analysis*, vol. 47, no. 1, pp. 49-62, 2004.
- 57
- 58
- 59