Deep Learning of Latent Variable Models for Industrial Process Monitoring

Xiangyin Kong, Zhiqiang Ge, *Senior Member, IEEE*

***Abstract*—Data-driven process monitoring based on latent variable models are widely employed in industry. This paper proposes a novel monitoring framework for latent variable models using hierarchical feature extraction, Bayesian inference and weighting strategy. We first establish a deep structure to implement hierarchical latent variables extraction, the extracted features are used to construct diverse monitoring statistics. Then we utilize Bayesian inference and proper weighting strategy to fuse various useful information. In line with the different characteristics of principal component analysis (PCA) and independent component analysis (ICA), we construct a deep PCA-ICA model for process monitoring according to the proposed framework. The deep PCA-ICA model performs hierarchical feature extraction, which can simultaneously extract deep Gaussian information and deep non-Gaussian information. The features extracted by different layers are then transformed to posterior probabilities through Bayesian inference. After that, different posterior probabilities are combined through appropriate weighting strategy to build new probabilistic statistic which can give more synthetic monitoring results. Moreover, the Bayesian inference and weighting strategy are further used to integrate the advantages of different models by transforming various probabilistic statistics into an overall monitoring index, which can comprehensively indicate the process status. The Tennessee Eastman process is used to validate the superiority of the proposed model over the existing methods. Besides, the extracted features are further analyzed to show the effectiveness and benefits of the deep hierarchical feature extraction structure. The source code of this paper is available at https://github.com/kxytim/DLVM_for_process_monitoring.**

***Index Terms*—Process monitoring, data-driven, latent variable models, principal component analysis (PCA), independent component analysis (ICA).**

I. Introduction

With the increasing scale of modern industrial systems, process monitoring is playing a progressively important role in reducing production cost, improving production quality and ameliorating safety. The huge scale of production and the advances in sensing technology have led to the emergence of a large amount of process data. As a result, data-driven process monitoring and fault detection methods have become more and more popular over the past two decades [1]–[4]. Among them, latent variable models may be the most widely used ones in data-driven process monitoring [5], [6]. Kresta *et al.* first [7] apply principal component analysis (PCA) in multivariate process monitoring. Chen *et al.* [8] use canonical correlation analysis (CCA) to monitor plant-wide process. Wang *et al.* [9] utilize the modified partial least squares (PLS) algorithm to detect quality-related faults. Lee *et al.* [10] first introduce independent component analysis (ICA) to process monitoring and achieved better results than PCA. Based on the different characteristics of PCA and ICA, Ge *et al.* [11] propose a model named of ICA-PCA, Huang *et al.* [12] propose a Gaussian and non-Gaussian double subspace monitoring method based on PCA and ICA, and Xu *et al.* [13] develop a hybrid method integrating ICA-PCA with relevant vector machine. To address the dynamic problem in actual industrial systems, Dong *et al.* [14] propose a novel dynamic PCA method for modeling dynamic data. Yu *et al.* [15] improve the dynamic monitoring performance of the original canonical variate analysis (CVA) model by introducing a variational Bayesian CVA method. In order to deal with the problem that practical industrial processes are mostly nonlinear, kernel trick is introduced to build nonlinear latent variable monitoring models [16], [17].

Despite the successful applications of latent variable models, we may ask a common question: could the models we studied sufficiently capture the inherent characteristics in data? Recently, with the vigorous development of deep learning theories and practices, a number of latent variable models based on deep learning have emerged. Lu *et al.* [18] utilize stacked denoising autoencoder to improve the fault diagnosis accuracy of rotary machinery components. Yu *et al.* [19] use the hidden layers of deep belief network for whole process monitoring. Recently, monitoring methods based on the emerging variational autoencoder also achieve outstanding results [20]. In addition to these common neural-network based deep models, some novel methods apply the idea of hierarchical feature extraction to traditional latent variable models to build new-style deep models. The most popular new-style deep model may be the deep principal component analysis (deep PCA). Liong *et al.* [21] use zero-phase component analysis whitening and PCA to build a two-layer deep PCA model, albeit PCA technique only involves linear transformations, Liong *et al.* still demonstrate that applying PCA transformation in series is effective in improving the feature extraction ability. Chen *et al.* [22] construct a novel deep PCA model to achieve real-time incipient fault detection. Deng *et al.* [23] build a layerwise feature extraction model based on PCA and kernel PCA, and the proposed model can learn deep nonlinear features to effectively improve the accuracy of fault detection.

However, existing methods are all aimed at a specific latent

This work was supported in part by the National Natural Science Foundation of China (NSFC) (61833014 and 92167106), and in part by the Natural Science Foundation of Zhejiang Province (LR18F030001). *(Corresponding author: Zhiqiang Ge.)*

X. Kong and Z. Ge are with the State Key Laboratory of Industrial Control Technology, Institute of Industrial Process Control, College of Control Science and Engineering, Zhejiang University, Hangzhou 310027, China. Z. Ge is also with the Peng Cheng Laboratory, Shenzhen 518000, China (e-mail: xiangyinkong@zju.edu.cn; gezhiqiang@zju.edu.cn).

variable model by using the idea of deep learning. So far, there is no such method that can be generally applied to all latent variable models. For the purpose of process monitoring, this paper firstly builds a generalized deep latent variable model based on hierarchical feature extraction, Bayesian inference and weighting strategy. Using the proposed method, we can deepen most traditional single-layer latent variable models. In addition, we also utilize Bayesian inference and weighting strategy to construct a monitoring index that can integrate the advantages of different latent variable models. Specifically, several monitoring statistics with different properties are first converted into posterior probabilities through Bayesian inference, then an overall probability-based monitoring index are constructed by weighting these posterior probabilities. Moreover, the features extracted by our deep model are further analyzed to evaluate the effectiveness of the proposed method. By contrasting the features at different layers, we found that the extracted features become more and more capable of reflecting the inherent information in data, and the monitoring performance increases layer by layer.

The rest of this paper is organized as follows. In Section II, we detailedly derive the monitoring framework of the proposed deep latent variable model. In Section III, the proposed deep model structure is applied to PCA and ICA, and successfully transforms them from shallow structure to deep structure. The Tennessee Eastman benchmark case study is conducted in Section IV to evaluate the superiority of the proposed method. Finally, conclusions are made in Section V.

II. The Proposed Monitoring Framework

Suppose $f(\cdot)$ represents an arbitrary latent variable model (such as PCA or ICA), and $\boldsymbol{X} \in \mathbb{R}^{N\times M}$ is the training data after preprocessing, where N is the number of samples, and M is the variables. By applying the model $f(\cdot)$ to $\boldsymbol{X}$, we have $f(\boldsymbol{X}) = \boldsymbol{Z}^1$, where $\boldsymbol{Z}^1$ is a latent variables matrix extracted by the model. Consider $\boldsymbol{Z}^1$ as the input data for the 2nd layer, we get the features $\boldsymbol{Z}^2 = f(\boldsymbol{Z}^1)$. After L times mapping, a hierarchical and cascaded structure with the depth equals to L layers can be obtained. At each layer, there is a latent variables matrix, the above total L matrices are computed by

$$\begin{aligned} &\boldsymbol{Z}^1 = f(\boldsymbol{X}) \\ &\boldsymbol{Z}^l = f(\boldsymbol{Z}^{l-1}),\ 2 \leqslant l \leqslant L \end{aligned} \tag{1}$$

For process monitoring, we can certainly build several monitoring statistics from the extracted latent variables according to some rules. A simple way is to calculate the 2-Norm of the latent variables vector for each sample. Assume S is a monitoring statistic formulated by the function $g(\cdot)$, i.e.

$$S^l(n) = g(\boldsymbol{z}^l(n)), 1 \leqslant n \leqslant N,\ 1 \leqslant l \leqslant L \tag{2}$$

where $\boldsymbol{z}^l(n)$ is the latent representation for a specific sample $\boldsymbol{x}_n$ at the l-th layer. There are total N samples and L layers.

By doing so, at each layer, N training samples can be turned to N monitoring statistics. For each layer, after collecting all the N monitoring statistics S, we can use any non-parametric estimation method, such as kernel density estimation (KDE), to estimate the probability density function (PDF) of this statistic. When the PDF is obtained, we can determine the confidence limit or control limit $S_{\lim}$ corresponding to a preset significance level δ. Briefly, the point which occupies the $1-\delta$ area of the PDF is the confidence limit. Significance level δ is usually set to 1% or 5%.

Given $\boldsymbol{x}_{new} \in \mathbb{R}^M$ as a new test sample, using (1) and (2), L layer monitoring statistics $S^l(new)$ can be acquired, where $1 \leqslant l \leqslant L$. To combine the monitoring statistics at different layers, we can use Bayesian inference [24] to generate a synthetic probability-based monitoring statistic. The specific steps are as follows.

Denote the conditional probabilities of test sample $\boldsymbol{x}_{new}$ under fault condition F and normal condition N at each layer as $P_S^l(\boldsymbol{x}_{new}|F)$ and $P_S^l(\boldsymbol{x}_{new}|N)$, respectively. The two conditional probabilities can be expressed as

$$P_S^l(\boldsymbol{x}_{new}|F) = \exp(-\mu S_{\lim}^l / S^l(new)),\ 1 \leqslant l \leqslant L \tag{3}$$

$$P_S^l(\boldsymbol{x}_{new}|N) = \exp(-\mu S^l(new) / S_{\lim}^l),\ 1 \leqslant l \leqslant L \tag{4}$$

where μ is a tuning parameter, which may decrease the sensitivity to data outliers by selecting an appropriate value.

According to Bayesian inference, the monitoring statistic value S^l can be transformed to the posterior fault probability by

$$P_S^l(F|\boldsymbol{x}_{new}) = \frac{P_S^l(\boldsymbol{x}_{new}|F)P_S^l(F)}{P_S^l(\boldsymbol{x}_{new})},\ 1 \leqslant l \leqslant L \tag{5}$$

where $P_S^l(F)$ is the prior fault probability of the system under fault operation, to our knowledge, which should be equal to the significance level δ. And $P_S^l(\boldsymbol{x}_{new})$ is the occurrence probability of sample $\boldsymbol{x}_{new}$, which is defined by the total probability theorem

$$P_S^l(\boldsymbol{x}_{new}) = P_S^l(\boldsymbol{x}_{new}|N)P_S^l(N) + P_S^l(\boldsymbol{x}_{new}|F)P_S^l(F),\quad 1 \leqslant l \leqslant L \tag{6}$$

where $P_S^l(N)$, similar to $P_S^l(F)$, is the prior probability of the system under normal condition, which is equal to confidence level $1-\delta$.

Under appropriate prior conditions, the Bayesian inference formula (5) transforms the monitoring statistics at different layers into posterior probabilities, thus provides us L probability-based statistics. Each of them represents the monitoring result of one specific layer, unlike the real monitoring statistics whose values are extremely various from each others, the values of the L transformed probability-based statistics are all between 0 and 1, and their confidence limits are all directly equivalent to δ. This provides convenience for fusing the features of different layers, after collecting these probability values, by weighting them in the light of suitable rules (i.e. (7) and (10)), we can combine the information extracted by different layers to construct a synthetic probability-based monitoring statistic. Since the statistic is generated by using the hierarchical feature extraction idea of deep learning and

Bayesian inference, we named it as deep Bayesian statistic (DBS)

$$DBS = \frac{\sum_{l=1}^{L} w_S^l P_S^l(F|\boldsymbol{x}_{new})}{\sum_{l=1}^{L} w_S^l} \tag{7}$$

where w_S^l is the weighting coefficient. Calculate it separately, a standardized weighting coefficient is obtained by $\bar{w}_S^l = w_S^l \Big/ \sum_{l=1}^{L} w_S^l$. Therefore, (7) can be rewritten as

$$DBS = \sum_{l=1}^{L} \bar{w}_S^l P_S^l(F|\boldsymbol{x}_{new}) \tag{8}$$

The purpose of process monitoring is to precisely detect the fault condition, therefore, the weighting coefficient should be designed to stress the fault information. As we already know, $P_S^l(F|\boldsymbol{x}_{new})$ in (5) indicates the fault probability at l-th layer. Under normal operation condition, it should be smaller than the significance level δ. Thus, if $P_S^l(F|\boldsymbol{x}_{new})$ exceeds the significance level, it may indicate a fault state, then its corresponding weighting coefficient should be assigned with a large value. Otherwise, the weighting coefficient should be given a small value.

However, in order to minimize the impact of data outliers. The assignment of weighting coefficients should consider not only the state of the current point, but also the states of the previous several points. Therefore, a mean posterior fault probability based on the past Y test samples around sample $\boldsymbol{x}_{new}$ can be defined by

$$\bar{P}_S^l(F|\boldsymbol{x}_{new}) = \frac{\sum_{j=new-Y+1}^{new} P_S^l(F|\boldsymbol{x}_j)}{Y} \tag{9}$$

$P_S^l(F|\boldsymbol{x}_{new})$ in (5) is the judgement criterion based on the current sample, while $\bar{P}_S^l(F|\boldsymbol{x}_{new})$ in (9) is the judgement criterion in view of the past Y test samples. When both these two probability value go beyond of the significance level, we can account the current sample $\boldsymbol{x}_{new}$ is under fault condition in a more persuasive way. At the same time, its corresponding weighting coefficient should be given a large value. Otherwise, the process is regarded as under normal condition, while the weighting coefficient should have a small value. Specifically, the weighting rules are detailed by

$$w_S^l = \begin{cases} \dfrac{1}{\eta}, \text{ if } P_S^l(F|\boldsymbol{x}_{new}) \geqslant \delta \& \bar{P}_S^l(F|\boldsymbol{x}_{new}) \geqslant \delta \\ \eta, \text{ otherwise} \end{cases} \tag{10}$$

where η is a small positive number and δ is the significance level.

It is worth noting that, the value of η or Y is problem-dependent. Taking η for example, a sufficiently small η would generate a relatively large weighting coefficient, which means our model can properly emphasize the fault information in the process system. Relatively, a large η number may result in no difference between fault state and normal state, which causes the fault information cannot be highlighted. Similarly, an appropriately large Y would help to eliminate the harmful effect of the noise in process data. But a very large Y may introduce too much unnecessary information, which will blanket the status of the current sample and lead to wrong decisions. Generally, the necessity of choosing η and Y suitably lies in the balance between false positive rate (FPR) and false negative rate (also known as missing detection rate (MDR)). The FPR is defined as the percentage of the samples that exceed the confidence limit under the normal operation condition, while the MDR is defined as the percentage of the samples do not exceed the confidence limit over all the faulty samples. η and Y provide a certain degree of freedom for the model adjustment, the values of η and Y can be determined by cross validation or trial and error. The purpose of adjusting η and Y is to achieve a low MDR while maintain a proper FPR.

By establishing the weighting rules (10) in detail, a synthetic probability-based monitoring statistic DBS in (8) is built, which has the advantage of combining all the L layer features extracted by our deep latent variable model. On the basis of previous analyses, if $DBS \geqslant \delta$, the process is considered under fault condition, otherwise, no fault occurs.

Through equations (3)-(10), we firstly turn the L layers numerical value S_l into L posterior fault probabilities $P_S^l(F|\boldsymbol{x}_{new})$. Then, weighted summation of these posterior probabilities provides a probability-based value DBS that combines the information at each layer. There is a question worth considering: could the above fusion strategy be further utilized?

In fact, following the steps (3)-(10), we can not only construct a synthetic probability-based index for one specific monitoring statistic by combining the information at each layer, but also establish an overall statistic by integrating several different types of monitoring statistics. As mentioned before, more than one monitoring statistic could be built based on a latent variable model. Taking PCA for example, T^2 and Q can be built for monitoring principal component subspace and residual subspace, respectively. Firstly, build a deep PCA model, then by replacing S in (3)-(10) with T^2 and Q, we get two probability-based index DBT^2 and DBQ. Again, by substituting DBT^2 and DBQ into the fusion strategy (3)-(10), unlike what we do to T^2 or Q, this time the total number of statistics I is regarded as the depth of layers L in (3)-(10). Here, we denote the number of statistics involved in integration as I, which, in the PCA case, is equivalent to 2. After this, an overall probability-based statistic $ODBS$ (Overall Deep Bayesian Statistic) can be built based on DBT^2 and DBQ.

Remark: Define the fusion strategy used for integrating L layers of features as method 1, while the one for integrating I statistics is recorded as method 2. The functions of these two methods are obviously different. Method 1 performs Bayesian inference on a type of specific monitoring statistic under and only under one latent variable model, it integrates the information in different layers, but all this information belongs to a particular type. While method 2 aims to combine various monitoring statistics with different properties, such as the statistic in principal component subspace and the one in residual subspace of PCA. Actually, the input of method 2 is

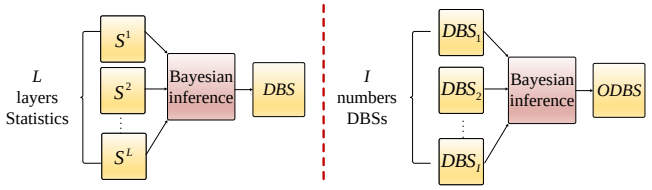


Fig. 1. Difference between the two methods. Method 1 on the leftside, while method 2 on the rightside.

the output of method 1. Moreover, method 2 can be utilized to different latent variable models. Actually, the greater the diversity between models, the better results of process monitoring may achieve. In detail, assume we integrate a total of H latent variable models, K monitoring statistics are built based on these models ($K \geqslant H$). For each statistic S_k, it has a layer of L_k ($1 \leqslant k \leqslant K$). Define the whole fusion procedures as $R(\cdot)$. The above two methods can be represented by:

$$\begin{aligned} &R(S_1^1, S_1^2, \cdots, S_1^{L_1}) = DBS_1 \\ &R(S_2^1, S_2^2, \cdots, S_2^{L2}) = DBS_2 \\ &\cdots \\ &R(S_K^1, S_K^2, \cdots, S_K^{L_K}) = DBS_K \\ &R(DBS_1, DBS_2, \cdots, DBS_K) = ODBS \end{aligned} \tag{11}$$

The difference between the above two methods is shown in Fig. 1.

To demonstrate the effectiveness of the proposed framework, we adopt two classical latent variable models — PCA and ICA, to develop a deep model for process monitoring.

III. Deep PCA-ICA Model for Process Monitoring

Process monitoring through deep PCA-ICA relies on four monitoring statistics — T^2 and Q_T in PCA, I^2 and Q_I in ICA. A brief overview for PCA and ICA, as well as the computation of T^2, Q_T, I^2 and Q_I are presented in Supplementary Material, Section I.

ICA has a non-Gaussian assumption on the distribution of independent components, therefore the stronger the non-Gaussianity of the original observed data is, the better result the ICA algorithm may achieve. On the other hand, PCA is looking for the Gaussian information. Since PCA can extract the Gaussian information in data, we can assume that after extracting information from original data matrix through PCA, the residual matrix has strong non-Gaussianity. At this time, applying the ICA algorithm to the residual matrix may achieve better results than directly applying ICA to original data. Based on the above characteristics, we first propose a single-layer PCA-ICA model, whose schematic is shown in Fig. 2.

Based on the deep latent variable model proposed in Section II, we can easily transform the single-layer PCA-ICA model to the deep architecture. This raises a question: how deep should we go? Actually, the number of layers is influenced by many factors and is certainly problem-dependent. An effective way to determine this parameter is through trial and error. For a specific problem, before using the proposed model for online monitoring, the technicians are suggested to first test the

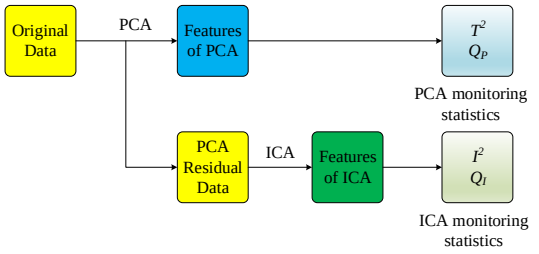


Fig. 2. Schematic of the single-layer PCA-ICA model.

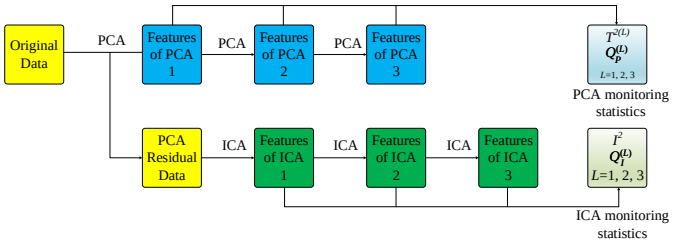


Fig. 3. Offline modeling schematic of DPI-3L.

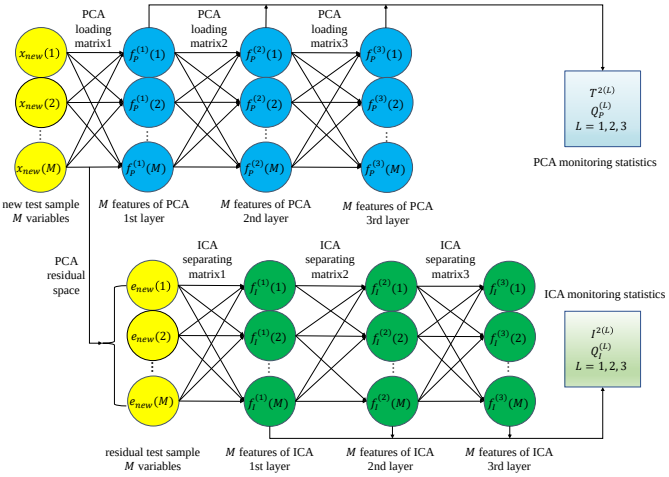


Fig. 4. Computing monitoring statistics for a new test sample through DPI-3L.

model with some offline data to determine the best parameters, especially the number of layers. In this paper, we build a two-layers deep PCA-ICA model (abbreviation: DPI-2L) and a three-layers deep PCA-ICA model (abbreviation: DPI-3L). The offline modeling procedures of DPI-3L is shown in Fig. 3, and the corresponding process of calculating monitoring statistics at different layers for a new test sample is depicted in Fig. 4. According to this structure, other models with different layers can also be established. Noted that without ICA model, the proposed deep PCA-ICA will converge to deep PCA model. Similarly, if the ICA algorithm acts directly on the original data instead of on the residual data, then deep PCA-ICA will degenerate to deep ICA model.

Specifically, the features of PCA at each layer are computed by

$$\begin{aligned} &\boldsymbol{T}^{(1)} = \boldsymbol{X}\boldsymbol{P}^{(1)} \\ &\boldsymbol{T}^{(l)} = \boldsymbol{T}^{(l-1)}\boldsymbol{P}^{(l)}, \ l = 2, 3, \ldots \end{aligned} \tag{12}$$

Similarly, the features of ICA at each layer are calculated by

$$\begin{aligned} \boldsymbol{S}^{(1)} &= \boldsymbol{W}^{(1)}\boldsymbol{E} \\ \boldsymbol{S}^{(l)} &= \boldsymbol{W}^{(l)}\boldsymbol{S}^{(l-1)},\ l = 2, 3, \dots \end{aligned} \tag{13}$$

where $\boldsymbol{E}$ is the PCA residual matrix in the first layer.

After projecting the train data to the proposed model, based on (3)-(10), the T^2, Q_P, I^2 and Q_I statistics can be turned to probability-based statistics DBT^2, DBQ_P, DBI^2 and DBQ_I, respectively. As discussed earlier, PCA looks for the Gaussian information, while ICA is interested in the non-Gaussian information. The features obtained by these two models have completely different properties, thus the proposed model can extract deep Gaussian information and deep non-Gaussian information simultaneously. To combine these diversities, an overall index, which is based on (11), can be obtained by integrating the above four probability-based statistics:

$$ODBS = R(DBT^2, DBQ_P, DBI^2, DBQ_I) \tag{14}$$

According to (3) and (4), a key step of using the proposed fusion theory is to obtain the confidence limits of all statistics. In this paper, the control limits of all statistics are determined through kernel density estimation (KDE) [25].

Process monitoring using deep PCA-ICA requires two stages: offline modeling and online monitoring. Detailed procedures for the two stages are described in Supplementary Material, Section II. According to the procedures, we conduct a comprehensive case study in Section IV.

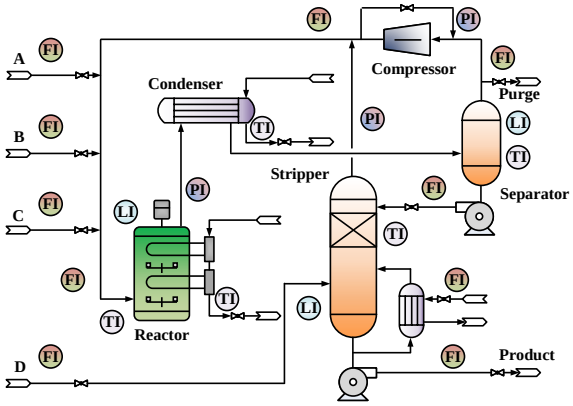


Fig. 5. Flowchart of the TE process.

TABLE I
AVERAGE MDR (%) ON 21 TE PROCESS FAULTS

Statistic		Avg	Statistic		Avg
PCA	T^2	39.726	DPI-2L	DBT^2	21.066
	Q_P	31.887		DBQ_P	20.000
ICA	I^2	22.590		DBI^2	19.500
	Q_I	23.649		DBQ_I	19.785
PCA-ICA	I^2	20.435		$ODBS$	17.185
	Q_I	23.203	DPI-3L	DBT^2	19.185
DAE	Q_D	20.476		DBQ_P	18.869
CAE	Q_C	33.774		DBI^2	19.155
RCAE	Q_R	31.112		DBQ_I	18.928
ARCAE	Q_A	23.344		$ODBS$	16.708
DePCA	PT^2	18.196			

Note: This table only displays the average MDR. The detailed MDRs are shown in ***Supplementary Material, Table S II***.

IV. CASE STUDY

A. TE Process Experiments and Results

The Tennessee Eastman (TE) process [26] is a benchmark process. For the past two decades, it has been widely used to evaluate the performance of different process monitoring models [27]. The flowchart of TE process is depicted in Fig. 5. In this paper, we select a total of 52 variables for process monitoring. For more information about the 52 monitoring variables, please refer to [26]. One normal condition and 21 faults are studied. The details of the 21 process faults are listed in Supplementary Material, Table S I. The normal condition dataset involves 1460 samples, 500 of them are used as the training dataset, the rest 960 samples are used as the validating dataset to calculate the confidence limits of all statistics. Each fault dataset consists of 960 samples, and each of the faults is introduced at 161th sample.

For PCA monitoring, the number of the retained principal components (PCs) is determined by the average eigenvalue method. As for the independent components (ICs) in ICA monitoring, we first sort them based on the magnitude of their negentropy, then select the top 15 ICs to construct monitoring statistics. Through trial and error, the tuning parameter μ is set to 1.3, the small number η which determines the weighting coefficient is set to 0.01, and Y is set to 5. The significance level δ is set to 5%, and the confidence limit is computed by KDE.

The monitoring performance of PCA, ICA, PCA-ICA, deep principal component analysis (DePCA) [23], DPI-2L and DPI-3L are investigated in this paper. The monitoring results of denoising autoencoders (DAE) are also reported. DAE is an neural networks-based deep learning model, which is demonstrated to be an excellent feature extraction approach and has unique advantages for extracting robust features [28]. Besides, we also build a convolutional autoencoders (CAE) based on one-dimensional convolution and deconvolution to realize process monitoring, whose network architecture is shown in Supplementary Material, Table S III. Moreover, the monitoring results obtained by two novel CAE-based models — residual connected CAE (RCAE) and self-attention based RCAE (ARCAE)[1] [29] are also included here. The monitoring statistic of neural networks-based monitoring model is usually built by the reconstruction error. And for DAE, CAE, RCAE and ARCAE, their statistics are denoted as Q_D, Q_C, Q_R and Q_A, respectively. The average MDR on 21 faults obtained by various models are summarized in Table I. Besides, the average MDR obtained by T^2, Q_P, I^2 and Q_I at each layer in DPL-3L are shown in Table II. To provide more information, the detailed MDRs on 21 faults obtained by several methods are presented in ***Supplementary Material, Table S II***.

The following conclusions can be drawn from the results:

1) The single layer PCA-ICA model (Fig. 2) has four

[1]In [29], *RCAE* and *ARCAE* are called *CNN* and *NPMNet*, respectively.

TABLE II
AVERAGE MDR (%) ON 21 TE PROCESS FAULTS OBTAINED BY EACH LAYER OF DPL-3L

Statistic \ Layer	1st Layer	2nd Layer	3rd Layer
T^2	39.726	21.232	19.696
Q_P	31.887	20.322	20.184
I^2	20.435	20.518	20.810
Q_I	23.203	20.107	20.095

monitoring statistics T^2, Q_P, I^2 and Q_I. The average MDR achieved by them are 39.726%, 31.887%, 20.435% and 23.203%, respectively. In Section III, we made an assumption that the PCA residual matrix has strong non-Gaussianity. The monitoring statistic Q_P of PCA is used to monitor the data variation in the residual space. In PCA-ICA, the ICA method is also applied to monitor the residual space. On the monitoring of PCA residual space, the average MDR achieved by the statistic I^2 is much lower than Q_P. The reason is that the monitoring statistic Q_P of PCA is based on Gaussian distribution. In the previous assumption, the Gaussianity of the residual space is weak while the non-Gaussianity is strong. Therefore, the I^2 statistic based on the non-Gaussian distribution performs much better than the Q_P statistic based on the Gaussian distribution. This confirms our previous assumption.

2) One may guess that since the PCA residual matrix has strong non-Gaussianity, applying the ICA algorithm to the PCA residual space may achieve better results than directly applying it to the original data. Observe the monitoring results of I^2 and Q_I in ICA model and PCA-ICA method, we can see the these two statistics do achieve better results in the PCA-ICA model than in the ICA model. This is also consistent with our hypothesis.

3) One of the purposes of deep learning is to capture the deep intrinsic features in data. Observe Table II, we can see that the MDR of I^2 is good and stable at each layer. While for the other three statistics, with the increase of layers, the average MDR decrease significantly. This indicates that the hierarchical feature extraction strategy is effective and able to extract deep inherent features in data.

4) In DPI-2L and DPI-3L, according to (3)-(10) , the four statistics T^2, Q_P, I^2 and Q_I are turned into DBT^2, DBQ_P, DBI^2 and DBQ_I. Table I indicates that from single layer PCA-ICA to DPI-2L and then to DPI-3L, the MDR of each monitoring statistic decreases gradually. For example, the average MDR obtained by T^2 in PCA-ICA, DBT^2 in DPI-2L and DPI-3L are: 39.726%→21.066%→19.185%. The decrease of the MDR for each statistic validates the effectiveness of the proposed fusion strategy. This is because the strategy can combine the useful information at each layer.

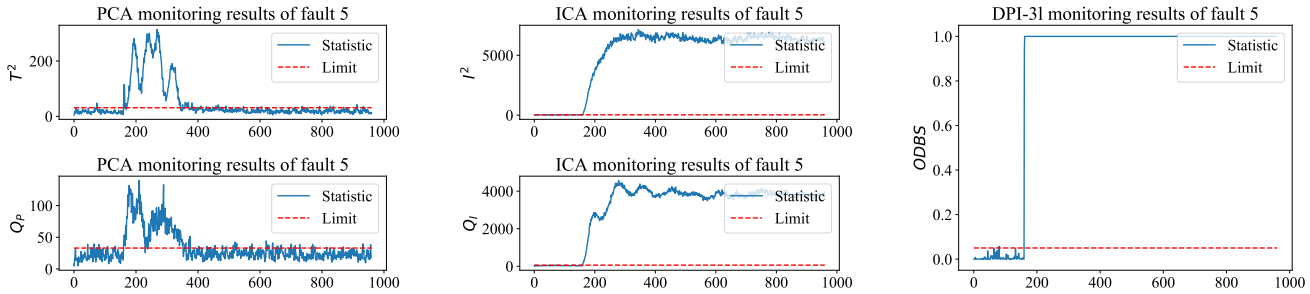
5) The monitoring results of convolutional neural networks (CNN)-based models — CAE, RCAE and ARCAE are all unsatisfactory. Even if the self-attention mechanism and residual connection are used in ARCAE, its monitoring performance is only equivalent to that of ICA. The poor monitoring performance of CNN-based models may be due to the *order of measurement variables* in the dataset [30], [31]. Such a phenomenon is out of this paper's scope, so we do not discuss it in detail.

6) The detection accuracy of ICA is better than PCA, as PCA only involves second-order statistics information, while ICA can extract higher-order statistics information. Moreover, DAE and DePCA perform much better than ICA and PCA. DAE and DePCA are all nonlinear monitoring methods, which are able to extract nonlinear information from process data, thus having great advantages over ICA and PCA. However, even the proposed DPI-2L and DPI-3L are linear models, the MDR achieved by the $ODBS$ statistic of DPI-2L and DPI-3L are still much lower than those nonlinear methods. Compared with DAE and DePCA, the model structure of DPI-2L or DPI-3L only includes linear transformations and does not involve nonlinear operations. As nonlinear transformations will greatly improve the model's complexity and representation ability (which will also wreck the model interpretability), thus leading to more powerful feature extraction. However, despite there is no nonlinear operation in the proposed framework, the performance of linear DPI-2L and DPI-3L is still much better than those nonlinear models, which just confirms the superiority of the proposed framework. The advantage of only linear transformations is good model interpretability, which will be discussed in Section IV-B. The monitoring results of our model are best may be the reason that DPI-2L or DPI-3L is not only based on deep feature extraction, but also integrates the Gaussian information extracted by PCA and the non-Gaussian information extracted by ICA.

The detection delay of fault is also investigated in this paper. The delay is measured by fault detection time (FDT), which is defined as the sample index of the alarming sample from which six successive samples exceed the confidence limit first time. In the following, we will use some monitoring figures to compare the model performance in detail.

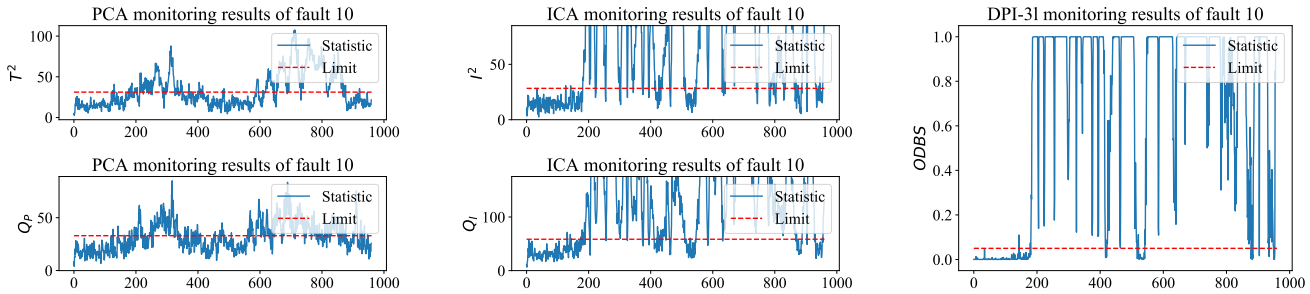
The detection results of fault 5 achieved by several latent variable models are illustrated in Fig. 6 (limited to the article length, this paper will focus on the performance comparison of some representative latent variable models). On the monitoring of fault 5, PCA has detected this fault at about 160th sample, but the monitoring statistics are below the confidence limit since the 350th sample, which means that PCA cannot detect this fault after sample 350. Observe the ICA monitoring results, as ICA can extract higher-order statistics information than PCA, the values of its two statistics are much larger than the confidence limit during the entire fault period. As for the monitoring results achieved by DPI-3L, since the $ODBS$ index can combine the advantages of different models, its MDR is 0% during the whole fault period just like ICA.

Next, we use fault 10 to compare the monitoring performance of different models. Fig. 7 summarizes the detection results of fault 10 for different models. The PCA T^2 statistic finds this fault at 214th sample, while the FDT of Q_P statistic is 207, and the corresponding MDRs of these two statistics are 50.875% and 47.875%, respectively. While for ICA, the FDTs of ICA I^2 and Q_I statistics are both 184, and I^2 has a MDR value 20.750%, while the MDR of Q_I is 16.250%. It has no doubt that the performance of ICA is much better than



(a) PCA (b) ICA (c) DPI-3L

Fig. 6. Different monitoring results on fault 5. The horizontal axis represents the sample point, and the vertical axis is the monitoring statistic.



(a) PCA (b) ICA (c) DPI-3L

Fig. 7. Different monitoring results on fault 10. The horizontal axis represents the sample point, and the vertical axis is the monitoring statistic.

PCA, no matter in FDT or in MDR. Moreover, the DPI-3L model alarms this fault at the 181th sample, with the MDR of 7.250%. Generally considering, the $ODBS$ index of DPI-3L model performs even better than ICA.

Finally, the fault 19 is considered. The monitoring results of fault 19 are shown in Fig. 8. The T^2 statistic of PCA does not detect this fault during the whole fault period, with an extremely high MDR of 91.750%. While the Q_P statistic of PCA alarms this fault at the 344th sample, with a corresponding MDR of 72.125%, a little better than T^2. The ICA I^2 and Q_I statistics detect this fault at the 170th sample and the 219th sample, respectively, with the MDRs of 26.250% and 41.875%. The monitoring performance of ICA is much better than PCA, but the results still have intolerable flaws. On the other hand, the FDT of the $ODBS$ statistic in DPI-3L is 161 (exactly the time when the fault occurred), and the MDR of $ODBS$ is significantly reduced to 3.125%.

The above results have demonstrated the superiority of the proposed framework, as it not only detects the fault information more accurately (minimum MDR), but also gives the fault alarm more quickly (shortest detection delays). On the other hand, the average FPR on 21 TE process faults obtained by some representative latent variable models are listed in Table III. The average FPR of these four methods are all less than 5%. It is worth noting that, the FPR and the MDR are contradictory to each other, some models can indeed reduce the MDR, but the corresponding FPR will become unacceptably high. However, the proposed method doesn't have such a problem, our models can significantly decrease the MDR, while maintaining the FPR at a normal level.

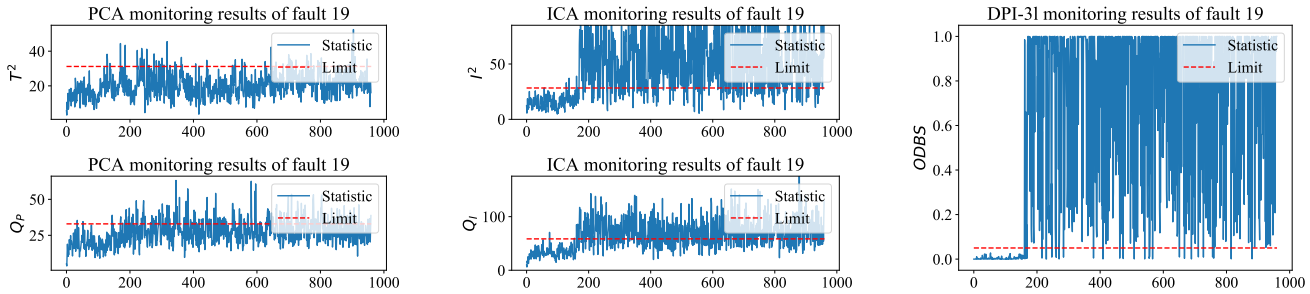
TABLE III
AVERAGE FPRS (%) OF THE 21 TE PROCESS FAULTS OBTAINED BY PCA, ICA, DPI-2L, AND DPI-3L

Statistics	PCA		ICA		DPI-2L	DPI-3L
	T^2	Q_P	I^2	Q_I	$ODBS$	$ODBS$
Average FPR	3.04	4.20	2.50	4.40	3.45	3.99

B. Further Discussions and Analyses

The difference between PCA and ICA has been analyzed in Section III. PCA is looking for Gaussian information in data, while ICA is designed to extract non-Gaussian information. To provide insight on how the proposed deep model improves monitoring performance, we compare the features at different layers of DPI-3L from several aspects.

The first principal component is very important in PCA, as it not only determines the direction of the subsequent principal components, but also contains the most data variation information. Here, we investigate the PDF of the first principal component at three different layers. The PDF estimated by KDE of the fist principal component at three different PCA layers in DPI-3L is shown in Supplementary Material, Fig. S2. Fig. S2 gives the PDF at different layers with their corresponding negentropy values. As described in Supplementary Material, Section I, the negentropy and kurtosis are used to measure the non-Gaussianity of random variables. The larger the value of negentropy or kurtosis, the stronger the non-Gaussianity. The random variable which follows Gaussian



(a) PCA (b) ICA (c) DPI-3L

Fig. 8. Different monitoring results on fault 19. The horizontal axis represents the sample point, and the vertical axis is the monitoring statistic.

TABLE IV
THE SUM OF KURTOSIS VALUES OF THE TOP 20 PRINCIPAL COMPONENTS AT DIFFERENT LAYERS

Layer Number	1	2	3
Kurtosis	5.320	4.330	3.501

TABLE V
THE SUM OF KURTOSIS VALUES OF THE TOP 10 COMPONENTS EXTRACTED BY ICA AT DIFFERENT LAYERS

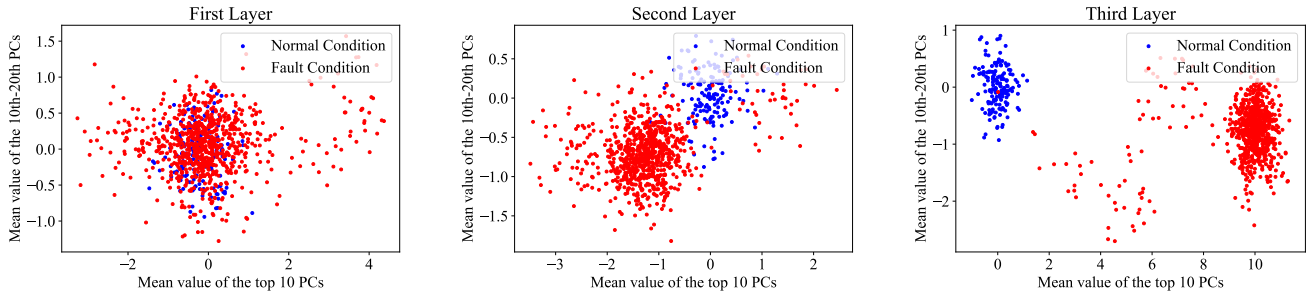
Layer Number	1	2	3
Kurtosis	10.593	10.997	14.642

distribution will have a small negentropy or kurtosis value. The first principal component at the 1st layer (the red line) hardly follows Gaussian distribution, while the PDF of the component at the 2nd layer (the blue line) shows strong Gaussianity. However, the distribution of the first component at the 3rd layer (the yellow line) is more concentrated around the mean, which means its distribution is even more Gaussian than the one at 2nd layer. Check the negentropy value listed in Fig. S2 we can also find that the value of negentropy decreases layer by layer. In each PCA layer, there are about 20 PCs retained to construct the principal component subspace. Table IV lists the sum of kurtosis values for the top 20 principal components at three layers. Table IV shows that the sum of kurtosis value also decreases layer by layer, which is consistent with the change of negentropy value. According to the analyses, we can conclude that the PCA features extracted by DPI-3L become more and more Gaussian with the increase of layers. This may account for why the monitoring performance of T^2 and Q_P statistics becomes better and better with the increase of layers, as the underlying assumption of PCA is that the principal components should follow Gaussian distribution. The more Gaussian the features are, the more successful the PCA algorithm is performed, and the features may reveal more inherent information of process data.

We also investigate the distribution of principal components in the two-dimensional space. The top 20 PCs retained by average eigenvalue method are divided into two groups, the first group contains the top 10 PCs, while the second one involves the 10th-20th PCs. Use the mean value of the PCs in group 1 as the horizontal axis, while the mean value of the PCs in group 2 are treated as the vertical axis. Taking fault 5 for example, the first 160 samples are under normal condition, and the rest 800 samples are under fault condition. After establishing the above coordinate system, the scatter diagrams of all 960 samples at three PCA layers are plotted in Fig. 9. The blue dots represent the normal condition, while the red dots represent the fault condition. At the first layer, the normal samples and the fault samples are completely mixed together, causing the PCA method not able to successfully distinguish fault samples from normal samples. At the second layer, fault samples and normal samples have been basically separated, but there are still a few points mixed together. While at the third layer, fault samples and normal samples are almost all split, with the normal samples located in the upper left position of the two-dimensional space, and the fault samples occupy the right part in the two-dimensional space.

Different from PCA, ICA is interested in the non-Gaussian information. Actually, the independent components to be estimated must strictly follow the non-Gaussian distribution, otherwise the ICA algorithm may not work well. The more non-Gaussian the components are, the more successful the ICA algorithm is performed. However, in practice, limited by the implementation of ICA algorithm, this method may converge to a local optimum. Due to the hierarchical structure in the proposed deep model, at each layer the ICA algorithm will be given a new initial state, which can effectively reduce the possibility of falling into local optimums. To prove this merit, the non-Gaussianity of the components extracted by ICA at different layers are analysed here. The sum of the kurtosis of the top 10 independent components extracted by all three ICA layers is listed in Table V. The kurtosis value in Table V increases layer by layer, which demonstrates the non-Gaussianity of the extracted independent components becomes stronger layer by layer.

The objective of ICA is to make the extracted components as independent of each other as possible. Here, we introduce the concept of mutual information. Mutual information is a natural measure of the dependence between random variables, it is always non-negative, and zero if and only if the variables are statistically independent. A smaller mutual information between two random variables indicates that they are more independent with each other. In the TE process case, the ICA sources matrix has 52 features at each layer. For each layer, the first feature is taken as a reference, and the mutual information



(a) First Layer (b) Second Layer (c) Third Layer

Fig. 9. Scatter diagram of the top 20 PCs at the first, second, and third layer.

between the remaining 51 features and the first feature is calculated in turn. The results are shown in Supplementary Material, Fig. S3. Since the mutual information equals to zero represents that these two random variables are independent, according to Fig. S3, there are 22 features that are independent to the first feature at the 1st layer. While for the 2nd and 3rd layer, the numbers are both 30. However, the average mutual information of the 3rd layer (0.007) is smaller than the 2nd layer (0.009), which means that the features in the 3rd layer are more independent with each other than in the 2nd layer.

The above analyses reveal why the performances of deep PCA and deep ICA are better than the corresponding single layer models, thus proving the advantages of the proposed deep model structure. To demonstrate the effectiveness of Bayesian inference and the weighting strategy, we studied the value of the standardized weighting coefficient $\bar{w}_S^l$ in (8). The standardized weighting coefficients of T^2 statistic for fault 5 and fault 10 are illustrated in Supplementary Material, Fig. S4. In the initial state of fault 5, for all three layers, almost all weighting coefficients are assigned with the same value 0.33. When the impact of the fault has accumulated to a certain extent, as the 1st layer cannot detect fault 5, the weights in the 1st layer nearly all decrease to zero. On the other hand, after a certain time, the weighting coefficients in the 2nd and 3rd layer all increase to larger values, causing the T^2 statistic in these two layers become capable of sounding fault alarms. As for fault 10, before the fault occurs, all weighting coefficients still have the same value 0.33. After the fault is introduced, nearly half of the weights in the 1st layer drop to 0, while many of the weights in the 2nd layer increase to a slightly larger value of 0.50. Moreover, the detection performance of the 3rd layer is further improved because the weighting coefficients of most fault samples in this layer are given with large values, and some weights even increase to the largest value 1.00. Consistent with changes in the weighting coefficients, the MDRs of the T^2 statistic for fault 10 achieved by the single layer PCA method, DPI-2L method and DPI-3L method are 50.875%, 24.250% and 11.625%, respectively. Apparently, through the investigation of the weighting coefficients, the effectiveness of Bayesian inference and the weighting strategy have been demonstrated.

V. Conclusion

A novel framework for latent variable models-based process monitoring has been proposed in this paper. This framework is based on deep feature extraction and information fusion. We make use of hierarchical structure to extract deep features, then utilize Bayesian inference and appropriate weighting strategy to achieve information fusion. There are three main contributions of the present paper:

1) We have built a generalized deep latent variable monitoring framework that can be used for almost all latent variable models through the idea of deep learning. Compared with traditional latent variable models, our model can extract more inherent features. Compared with deep neural network, the proposed model does not require backpropagation and computation-costly nonlinear optimizations, which means its training is much more faster than deep neural network. Meanwhile, deep neural network is like a "black box" and has no gratifying interpretability, while the analyses in Section IV-B demonstrate our proposed model has good interpretability.
2) Using the above framework, we have established a deep PCA-ICA model. Based on the different characteristics of PCA and ICA, this model realizes deep Gaussian information extraction and deep non-Gaussian information extraction simultaneously, successfully separating the Gaussian parts and non-Gaussian parts of process data.
3) We have constructed a new probability-based monitoring statistic $ODBS$ which can combine the advantages of different latent variable models. Since the proposed deep learning framework can be used for most latent variable models, by suitably assuring the diversities of models, the $ODBS$ index plays an important role of making a more comprehensive judgement about the process status.

Experimental results of the TE benchmark process have validated the effectiveness of the proposed framework.

There are many open questions deserving further study. One important issue is to determine the depth of the proposed deep model. Further study is needed to investigate how many layers is most appropriate for a specific problem. At present, the proposed methods can only be used for structured data. In the field of fault diagnosis, there are some methods that focus on unstructured data [32]–[35]. In the future, we will consider extending the application of our methods to unstructured data.

Other avenues include exploring a suitable way to use the proposed methods for other problems, for example industrial soft sensor [36]. How to efficiently deploy the proposed framework in an adaptive and distributed way is also worth investigating.

REFERENCES

[1] X. Dai and Z. Gao, "From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2226–2238, 2013.

[2] Z. Ge, "Review on data-driven modeling and monitoring for plant-wide industrial processes," *Chemometrics and Intelligent Laboratory Systems*, vol. 171, pp. 16–25, 2017.

[3] H. Luo, H. Zhao, and S. Yin, "Data-driven design of fog-computing-aided process monitoring system for large-scale industrial processes," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4631–4641, 2018.

[4] L. Yao and Z. Ge, "Industrial big data modeling and monitoring framework for plant-wide processes," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 6399–6408, 2020.

[5] Z. Yang and Z. Ge, "Monitoring and prediction of big process data with deep latent variable models and parallel computing," *Journal of Process Control*, vol. 92, pp. 19–34, 2020.

[6] R. Raveendran, H. Kodamana, and B. Huang, "Process monitoring using a generalized probabilistic linear latent variable model," *Automatica*, vol. 96, pp. 73–83, 2018.

[7] J. V. Kresta, J. F. MacGregor, and T. E. Marlin, "Multivariate statistical monitoring of process operating performance," *The Canadian journal of chemical engineering*, vol. 69, no. 1, pp. 35–47, 1991.

[8] Z. Chen, Y. Cao, S. X. Ding, K. Zhang, T. Koenings, T. Peng, C. Yang, and W. Gui, "A distributed canonical correlation analysis-based fault detection method for plant-wide process monitoring," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2710–2720, 2019.

[9] G. Wang and S. Yin, "Quality-related fault detection approach based on orthogonal signal correction and modified pls," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 2, pp. 398–405, 2015.

[10] J.-M. Lee, C. Yoo, and I.-B. Lee, "Statistical process monitoring with independent component analysis," *Journal of process control*, vol. 14, no. 5, pp. 467–485, 2004.

[11] Z. Ge and Z. Song, "Process monitoring based on independent component analysis- principal component analysis (ica- pca) and similarity factors," *Industrial & Engineering Chemistry Research*, vol. 46, no. 7, pp. 2054–2063, 2007.

[12] J. Huang and X. Yan, "Gaussian and non-gaussian double subspace statistical process monitoring based on principal component analysis and independent component analysis," *Industrial & Engineering Chemistry Research*, vol. 54, no. 3, pp. 1015–1027, 2015.

[13] Y. Xu, S.-Q. Shen, Y.-L. He, and Q.-X. Zhu, "A novel hybrid method integrating ica-pca with relevant vector machine for multivariate process monitoring," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 4, pp. 1780–1787, 2019.

[14] Y. Dong and S. J. Qin, "A novel dynamic pca algorithm for dynamic data modeling and process monitoring," *Journal of Process Control*, vol. 67, pp. 1–11, 2018.

[15] J. Yu, L. Ye, L. Zhou, Z. Yang, F. Shen, and Z. Song, "Dynamic process monitoring based on variational bayesian canonical variate analysis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–11, 2021.

[16] D. Zheng, L. Zhou, and Z. Song, "Kernel generalization of multi-rate probabilistic principal component analysis for fault detection in nonlinear process," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 8, pp. 1465–1476, 2021.

[17] Y. Zhang, H. Zhou, S. J. Qin, and T. Chai, "Decentralized fault diagnosis of large-scale processes using multiblock kernel partial least squares," *IEEE Transactions on Industrial Informatics*, vol. 6, no. 1, pp. 3–10, 2010.

[18] C. Lu, Z.-Y. Wang, W.-L. Qin, and J. Ma, "Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification," *Signal Processing*, vol. 130, pp. 377–388, 2017.

[19] J. Yu and X. Yan, "Whole process monitoring based on unstable neuron output information in hidden layers of deep belief network," *IEEE Transactions on Cybernetics*, vol. 50, no. 9, pp. 3998–4007, 2020.

[20] K. Wang, M. G. Forbes, B. Gopaluni, J. Chen, and Z. Song, "Systematic development of a new variational autoencoder model based on uncertain data for monitoring nonlinear processes," *IEEE Access*, vol. 7, pp. 22 554–22 565, 2019.

[21] V. E. Liong, J. Lu, and G. Wang, "Face recognition using deep pca," in *2013 9th International Conference on Information, Communications & Signal Processing*. IEEE, 2013, pp. 1–5.

[22] H. Chen, B. Jiang, N. Lu, and Z. Mao, "Deep pca based real-time incipient fault detection and diagnosis methodology for electrical drive in high-speed trains," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 4819–4830, 2018.

[23] X. Deng, X. Tian, S. Chen, and C. J. Harris, "Deep principal component analysis based on layerwise feature extraction and its application to nonlinear process monitoring," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 6, pp. 2526–2540, 2019.

[24] Z. Ge and Z. Song, "Multimode process monitoring based on bayesian method," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 23, no. 12, pp. 636–650, 2009.

[25] Q. Chen, U. Kruger, and A. T. Leung, "Regularised kernel density estimation for clustered process data," *Control Engineering Practice*, vol. 12, no. 3, pp. 267–274, 2004.

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

[27] J. Qian, L. Jiang, and Z. Song, "Locally linear back-propagation based contribution for nonlinear process fault diagnosis," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 3, pp. 764–775, 2020.

[28] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, no. 12, pp. 3371–3408, 2010.

[29] S. Li, J. Luo, and Y. Hu, "Nonlinear process modeling via unidimensional convolutional neural networks with self-attention on global and local inter-variable structures and its application to process monitoring," *ISA Transactions*, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019057821002111

[30] X. Jiang and Z. Ge, "Augmented multidimensional convolutional neural network for industrial soft sensing," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–10, 2021.

[31] X. Yuan, S. Qi, Y. A. Shardt, Y. Wang, C. Yang, and W. Gui, "Soft sensor model for dynamic processes based on multichannel convolutional neural network," *Chemometrics and Intelligent Laboratory Systems*, vol. 203, p. 104050, 2020.

[32] A. Glowacz, "Ventilation diagnosis of angle grinder using thermal imaging," *Sensors*, vol. 21, no. 8, p. 2853, 2021.

[33] A. Kumar, G. Vashishtha, C. Gandhi, Y. Zhou, A. Glowacz, and J. Xiang, "Novel convolutional neural network (ncnn) for the diagnosis of bearing defects in rotary machinery," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–10, 2021.

[34] A. Glowacz, "Fault diagnosis of electric impact drills using thermal imaging," *Measurement*, vol. 171, p. 108815, 2021.

[35] A. Glowacz, R. Tadeusiewicz, S. Legutko, W. Caesarendra, M. Irfan, H. Liu, F. Brumercik, M. Gutten, M. Sulowicz, J. A. A. Daviu *et al.*, "Fault diagnosis of angle grinders and electric impact drills using acoustic signals," *Applied Acoustics*, vol. 179, p. 108070, 2021.

[36] Q. Sun and Z. Ge, "A survey on deep learning for data-driven soft sensors," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 5853–5866, 2021.

Xiangyin Kong received the B. Eng. degree in marine engine engineering from the School of Naval Architecture and Ocean Engineering, Huazhong University of Science and Technology, Wuhan, China, in 2019.

He is currently pursuing the Ph.D. degree in control science and engineering with the State Key Laboratory of Industrial Control Technology, College of Control Science and Engineering, Zhejiang University, Hangzhou, China. His research interests include data-driven industrial modeling, industrial security, machine learning and deep learning.

Zhiqiang Ge (M'13-SM'17) received the B.Eng. and Ph.D. degrees in Automation from the Department of Control Science and Engineering, Zhejiang University, Hangzhou, China, in 2004 and 2009, respectively. He was a Research Associate with the Department of Chemical and Biomolecular Engineering, Hong Kong University of Science Technology from Jul. 2010 to Dec. 2011 and a visiting Professor with the Department of Chemical and Materials Engineering, University of Alberta from Jan. 2013 to May 2013.

Dr. Ge was an Alexander von Humboldt research fellow with University of Duisburg-Essen during Nov. 2014 to Jan. 2017, and also a JSPS invitation Fellow with Kyoto University during Jun. 2018 to Aug. 2018. He is currently a Full Professor with the College of Control Science and Engineering, Zhejiang University. His research interests include industrial big data, process monitoring, soft sensor, data-driven modeling, machine intelligence, and knowledge automation.