Nonlinear dynamic process monitoring using deep dynamic principal component analysis

Simin Li, Shuanghua Yang, Yi Cao and Zuzhen Jia

ABSTRACT
Data-driven method has gained its popularity in fault detection. Conventional methods are associated with one-single-layer process monitoring. Information extracted by such a method may not be sufficient to detect some faults for complicated process systems. Inspired by the deep learning conception, a multi-layer fault detection method, namely Deep Principal Component Analysis (DePCA) was proposed previously in the literature. DePCA has the capability to extract deep features for a process resulting in better fault detection performance. However, it assumes that the value of the variable at each moment is unrelated, which is not suitable for complex nonlinear dynamic systems. To address the concerns, by adopting dynamic PCA to extract dynamic features, a new deep approach, namely Deep Dynamic Principal Component Analysis (DeDPCA), is proposed. In the new approach, both Dynamic feature and nonlinear feature can be extracted in different layers so that more process faults can be detected. A Tennessee Eastman process case study was then employed for application and validation of the DeDPCA, which indicates the proposed method is suitable for monitoring complex dynamic nonlinear processes.

1. Introduction
Since traditional detection systems for chemical process are lack of reliability (Yin et al., 2012), data-driven methods such as principal component analysis (PCA) (Wold et al., 1987) has gained it popularity particularly in the era of industrial digitalization. Using the data-driven methods (Gao et al., 2013), even if the mechanism and knowledge of the system are not known, one can bypass the complex procedure of establishing a mathematical model of the system and discover the faults based on changes in data.

The PCA is to extract several orthogonal principal components from the original multi-dimensional data space, thereby simplifying the analysis model. These principal components contain most variations of the original data (Wold et al., 1987). The PCA was developed based on the presumptions that the system is linear and static, however, most industrial processes are nonlinear and dynamic. Kramer (1991) proposed a nonlinear principal component analysis (NLPCA) method which uses a feedforward neural network to represent the feature mapping process such that the network inputs are reproduced at the output layer. Lee et al. (2004) developed another nonlinear process monitoring method, so called as the kernel principal component analysis (KPCA), which calculates the principal components in a high-dimensional feature space using a kernel function, such as gaussian, polynomial and so on. To extend PCA application to dynamic systems, Ku et al. (1995) applied a 'time lag shift' strategy to contain the dynamic information resulting in the so-called Dynamic PCA (DPCA).

Deep learning (LeCun et al., 2015) is a recently developed machine learning technology, which has been used in many fields like image identification, natural language processing, etc. It consists of multiple layers of nonlinear mapping (Wu & Zhao, 2018) and are aiming at learning feature hierarchies where features from higher levels of the hierarchy are formed by lower level features (Pan et al., 2020). Recently, deep learning has been used in process monitoring and fault detection (Iqbal et al., 2019). Deng et al. (2018) raised a multilayer and multivariate statistical model to extract different categories of data features, such as linear and nonlinear principal components, which is named deep principal component analysis (DePCA). While DePCA is more sensitive to a fault than KPCA and serial PCA (SPCA) (Deng et al., 2016) which are...
commonly used in process monitoring, it cannot extract the dynamic information of a dynamic chemical process.

The DePCA which combined PCA and KPCA assumes that the values of the variable at different moments are unrelated (Russell et al., 2000). Nowadays most industrial production is nonlinear and dynamic. Only considering nonlinearities without addressing auto-correlation and cross-correlation in measurement data is not enough.

In this paper, based on extensive analysis of the pros and cons of different data-driven methods, deep dynamic principal component analysis (DeDPCA) that integrated DPCA and KPCA was developed. Dynamic feature extracting was conducted by the DPCA layer, while nonlinear feature was extracted by the KPCA layer. DPCA explores the temporal autocorrelation of sampled data by augmenting the matrix. Considering that extending time series after kernel mapping may increase computational complexity and lose some dynamic information, DPCA is used to extract dynamic features in the first layer and nonlinear features are extract in the second layer. Bayesian inference was used to fuse information from different layers and conduct the overall decision. Autocorrelation analysis was used for each variable to decide how much time lag should be reserved.

The paper has several contributions. Firstly, a deep DPCA model that combining DPCA and KPCA was proposed. It has the potential to improve the performance of process monitoring. Secondly, it is demonstrated through the TE case study that the DeDPCA is able to detect faults for complex processes. Finally, an approach to optimize the time lags for each variable is proposed to reduce computational complexity but still to keep a good performance.

The work is organized as follows. Section 2 illustrates the algorithm and model structure of our proposed method. Section 3 demonstrates a case study on Tennessee Eastman(TE) plant to verify the superiority of the proposed method. Section 4 discussed on the outcomes and limitation. Section 5 draws the final conclusion of the work.

2. DeDPCA model

2.1. DeDPCA algorithm

The DeDPCA method contains two layers of feature mapping as shown in Figure 1, one is the DPCA dynamic linear feature layer and another is the KPCA nonlinear feature layer.

In this method, firstly, the normal dataset $Y \in \mathbb{R}^{n \times m}$ including $n$ samples and $m$ variables is expanded as an augmented matrix $Y(l) \in \mathbb{R}^{(n-l) \times m(l+1)}$ with a lag of $l$, as shown in Equation (1). Then the dynamic linear principal components can be extracted from $Y(l)$ and the first-layer features are extracted as $T^{(1)} \in \mathbb{R}^{(n-l) \times N_{1}}$, where $N_{1} = m(l+1)$. Then the nonlinear principal components can be extracted by applying KPCA mapping method to $T^{(1)}$ to extract the second-layer features $T^{(2)} \in \mathbb{R}^{(n-l) \times N_{2}}$, where $N_{2}$ is the number of nonlinear principal components reserved in the second layer.

The DPCA methods catch the dynamic information by extending each observation vector with the previous $l$ observations and constructing the data matrix as follows

$$Y(l) = \begin{bmatrix} y_{t} & y_{t-1} & \cdots & y_{t-l} \\ y_{t-1} & y_{t-2} & \cdots & y_{t-l-1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{t+l-n} & y_{t+l-n-1} & \cdots & y_{t-n} \end{bmatrix}$$

where $y_{t}$ is the $m$-dimensional observation vector at time $t$. By using PCA on $Y(l)$, a multivariate autoregressive is extracted directly from the data.

The optimization task for the first layer could be wrote as

$$\max_{q^{(1)}} \frac{1}{n-l-1} (q^{(1)})^{T} Y(l) Y(l) q^{(1)}$$

$$\text{s.t. } (q^{(1)})^{T} q^{(1)} = 1$$

from which projection vectors $q_{i}^{(1)} \in \mathbb{R}^{m(l+1)}$ can be obtained. Score vectors can also be calculated as $f_{i}^{(1)} = Y(l) q_{i}^{(1)}$, $1 \leq i \leq m(l+1)$. Therefore, the first layer features are constructed by all these score vectors, $F^{(1)} = [f_{1}^{(1)} f_{2}^{(1)} \ldots f_{m(l+1)}^{(1)}]$.

Similarly, the optimization task for the second layer could be written as

$$\max_{q^{(2)}} \frac{1}{n-l-1} (q^{(2)})^{T} K(F^{(1)}) K(F^{(1)}) q^{(2)}$$

$$\text{s.t. } (q^{(2)})^{T} q^{(2)} = 1$$

from which the nonlinear projection vector $q^{(2)}$ can be derived as follows:

$$q^{(2)} = \sum_{j=1}^{m-l} \Phi^{T} (f_{j}^{(1)}) \alpha = \Phi^{T} (F^{(1)}) \beta$$

where $F_{j}^{(1)} \in \mathbb{R}^{N_{1}}$ is the $j$th row of $F^{(1)}$, $\beta = [\beta_{1} \beta_{2} \ldots \beta_{m-l}]$. Besides, denoting $K = \Phi(F^{(1)}) \Phi^{T}(F^{(1)})$, the Equation (3) can be written as

$$\max_{\alpha} \frac{1}{n-l-1} \beta^{T} (K^{(1)}) (K^{(1)})^{T} \beta$$

$$\text{s.t. } \beta^{T} K^{(1)} \beta = 1$$

from which the projection vectors $\beta_{i} \in \mathbb{R}^{n-l}$ can be obtained. The nonlinear score vectors in the second layer.
can also be calculated as $f^{(2)}_{i,j} = K^{(1)}_i \beta_i \in \mathbb{R}^{n-1}$, $1 \leq i \leq N_2$. Therefore, the second layer features are constructed by all these score vectors, $F^{(2)} = [f^{(2)}_{1,1} f^{(2)}_{1,2} \ldots f^{(2)}_{N_2}] \in \mathbb{R}^{(n-1) \times N_2}$.

For the testing vector $\hat{y}_t = [y_t, y_{t-1}, \ldots y_{t-l}]$, the feature vector for the first layer $f^{(1)}_{t,j} = [f^{(1)}_{t,1} f^{(1)}_{t,2} \ldots f^{(1)}_{t,N}]$ can be calculated as

$$f^{(1)}_{t,j} = \hat{y}_t \beta_j$$

Then the second layer feature vectors $f^{(2)}_{t,j} = [f^{(2)}_{t,1} f^{(2)}_{t,2} \ldots f^{(2)}_{t,N}]$ can be obtained as

$$f^{(2)}_{t,j} = k^{(1)}_j \beta_j$$

where $k^{(1)}_j$ is the kernel vector and can be calculated by $k^{(1)}_j = \Phi(F^{(1)}_j) \Phi(f^{(1)}_t) = [k^{(1)}_{1,1} k^{(1)}_{1,2} \ldots k^{(1)}_{N,N}]^T \in \mathbb{R}^{n-1}$.

From above, feature extraction from layer to layer does not need complex network optimization, which is much convenient to train a deep model when compare with other deep learning methods such as convolutional neural networks (CNN) and long short-term memory (LSTM).

After getting the features of each layer, their monitoring metrics can be computed. In first layer, $T^{(1)}$ and $Q^{(1)}$ metrics can be calculated as

$$T^{(1)} = f^{(1)}_t^T \Lambda_1^{-1} f^{(1)}_t$$

$$Q^{(1)} = \|\hat{y}_t - \hat{y}_t\|^2 = (\hat{y}_t - Qf^{(1)}_t)^T (\hat{y}_t - Qf^{(1)}_t)$$

where $\Lambda_1 \in \mathbb{R}^{N_1 \times N_1}$ is a diagonal matrix whose diagonal elements are the eigenvalues $\lambda_{1,i}$, $1 \leq i \leq N_1$ obtained through eigenvalue decomposition. $\hat{y}_t = Qf^{(1)}_t$ is the refactor of $\hat{y}_t$.

In second layer, $T^{(2)}$ and $Q^{(2)}$ metrics is calculated as

$$T^{(2)} = f^{(2)}_t^T \Lambda_2^{-1} f^{(2)}_t$$

$$Q^{(2)} = \|\Phi(f^{(2)}_t) - \Phi(f^{(1)}_t)\|^2 = \sum_{j=1}^{m} \sum_{l=1}^{N_2} (f^{(2)}_j - f^{(1)}_j)^2$$

where $\Lambda_2 \in \mathbb{R}^{N_2 \times N_2}$ is a diagonal matrix whose diagonal elements are the eigenvalues $\lambda_{2,i}$, $1 \leq i \leq N_2$. $\Phi(f^{(1)}_t)$ is the refactor of $\Phi(f^{(1)}_t)$.

### 2.2. Time lag selection

The time series expanding takes the dynamic information into consideration, but it would also increase the computation complexity. The autocorrelation for each variable is different. Thus expanding all the variables to the same time lag is unnecessary. To eliminate the autocorrelation of variables, separate analysis for each variable can be used to choose an appropriate lag $l$ for each variable.

The autocorrelation of a variable is denoted as $\gamma$ and a strong autocorrelation standard is denoted as $\gamma_{\text{min}}$. If $\gamma \geq \gamma_{\text{min}}$, it has strong autocorrelation, otherwise it has weak autocorrelation. The $\gamma_{\text{min}}$ is set to 0.5 empirically.
When deciding the time lag of variable $y_1$, $l$ is first set to 0. If $y'(l) \geq y_{\min}$ and $y'(l+1) \leq y_{\min}$, $[x_{l-1-1} x_{l-1-2} \ldots x_{l-2}]$ would together form a new sample vector at time $t$, where $l \leq l_{\text{max}}$, else $l+1$ and repeat the above process. The final $l$ is the current time lag for variable $y_1$ and $l_{\text{max}}$ is the maximum time lag because timing expansion is finite dimensional.

Using this method, the time lags for variables with strong autocorrelation are large and with weak autocorrelation are small. Hence the dynamic information of the process can be kept as much as possible while reducing the number of expanded variables, thus ultimately reducing the computational complexity.

### 2.3. Control limit calculation

Since Gaussian assumption is not applicable to the most industrial (Wang et al., 2016), the control limits can be calculated using actual probability density functions. Kernel density estimation (KDE) (Odiowei & Cao, 2009) has been wildly used for estimating probability density function (PDF), particularly for univariate stochastic events. Hence it can be used to estimate the PDF of $T^2$ and $Q$ metrics.

The PDF $\hat{p}(x)$ of a variable at point $x$ can be estimated using the kernel function $K(\cdot)$ that can be wrote as

$$\hat{p}(x) = \frac{1}{Mh} \sum_{k=1}^{M} K\left(\frac{x-x_k}{h}\right)$$

where $x_k$ is the sample of $x$, $k$ is the sample number and $h$ is a smoothing parameter called bandwidth.

$$\int_{-\infty}^{T_{\text{lim}}^{2}} \hat{p}(T^2) d(T^2) = \alpha \quad \text{and} \quad \int_{-\infty}^{Q_{\text{lim}}} \hat{p}(Q) d(Q) = \alpha$$

(11)

Given a confidence interval $\alpha$, the control limits $T_{\text{lim}}^{2}$ and $Q_{\text{lim}}$ can be calculated using the PDF by Equation (10).

### 2.4. Fault information fusion

In DeDPCA model, there are two monitoring statistics $T^{2(s)}$ and $Q^{(s)}$ in each layer, where $s = 1, 2$. For the sake of fusing all the information of different layers together, Bayesian inference was adopted to merge informations of all layers. Firstly, Bayesian inference (Woolrich et al., 2004) is used to convert each monitoring statistic into posterior fault probability. Then, these probabilities are weighted and summed together. Finally, the probability-based combined monitoring statistics (Cai et al., 2017) $P(T^2)$ and $P(Q)$ is determined.

For monitoring statistics $T^{2(s)}$ and $Q^{(s)}$, the fault probability of sample $y_t$ in fault condition $C_f$, $P^{(s)}(y_t | C_f)$ and

$$P^{(s)}(y_t | C_f), \text{can be respectively defined as}$$

$$P_{T_2}^{(s)}(y_t | C_f) = \exp(-\kappa T_{\text{lim}}^{2(s)}/T_2^{(s)})$$

(12)

$$P_{Q}^{(s)}(y_t | C_f) = \exp(-\kappa Q_{\text{lim}}^{(s)}/Q^{(s)})$$

(13)

Similarly, the fault probability of sample $y_t$ at the normal condition $C_n$, $P^{(s)}(y_t | C_n)$ and $P^{(s)}(y_t | C_n)$ are defined as

$$P_{T_2}^{(s)}(y_t | C_n) = \exp(-\kappa T_{\text{lim}}^{2(s)}/T_2^{(s)})$$

(14)

$$P_{Q}^{(s)}(y_t | C_n) = \exp(-\kappa Q_{\text{lim}}^{(s)}/Q^{(s)})$$

(15)

where $\kappa$ is a parameter used to reduce sensitivity to data that is abnormal. It was equal to 0.1.

Using Bayesian inference theory, $T^{2(s)}$ and $Q^{(s)}$ can be changed to the posterior probability $P^{(s)}(C_f | y_t)$ and $P^{(s)}(C_n | y_t)$ as

$$P^{(s)}(C_f | y_t) = \frac{P^{(s)}(y_t | C_f)P^{(s)}(C_f)}{P^{(s)}(y_t)}$$

(16)

$$P^{(s)}(C_n | y_t) = \frac{P^{(s)}(y_t | C_n)P^{(s)}(C_n)}{P^{(s)}(y_t)}$$

(17)

where $P^{(s)}(C_f)$ and $P^{(s)}(C_n)$ are the prior fault probability. They are set to the significance level $\delta$.

$P^{(s)}(y_t | C_f)$ and $P^{(s)}(y_t | C_n)$ are the occurrence probability of sample $y_t$:

$$P_{T_2}^{(s)}(y_t | C_f) = P_{T_2}^{(s)}(y_t | C_f)P_{T_2}^{(s)}(C_f) + P_{Q}^{(s)}(y_t | C_n)P_{T_2}^{(s)}(C_n)$$

(18)

$$P_{Q}^{(s)}(y_t | C_f) = P_{Q}^{(s)}(y_t | C_f)P_{Q}^{(s)}(C_f) + P_{Q}^{(s)}(y_t | C_n)P_{Q}^{(s)}(C_n)$$

(19)

where $P_{T_2}^{(s)}(C_f)$ and $P_{Q}^{(s)}(C_f)$ are the prior normal probability. They are set to 1 $-$ $\delta$.

Giving weights for $P_{T_2}^{(s)}(C_f | y_t)$ and $P_{Q}^{(s)}(C_f | y_t)$, two overall monitoring statistics metrics $P(T^2)$ and $P(Q)$ are constructed

$$P(T^2) = \frac{\sum_{s=1}^{2} \omega_{T_2}^{(s)} P_{T_2}^{(s)}(C_f | y_t)}{\sum_{s=1}^{2} \omega_{T_2}^{(s)}}$$

(20)

$$P(Q) = \frac{\sum_{s=1}^{2} \omega_{Q}^{(s)} P_{Q}^{(s)}(C_f | y_t)}{\sum_{s=1}^{2} \omega_{Q}^{(s)}}$$

(21)

where $\omega_{T_2}^{(s)}$ and $\omega_{Q}^{(s)}$ are two weighting factors. $\omega_{T_2}^{(s)} = \omega_{T_2}^{(s)} / \sum_{s=1}^{2} \omega_{T_2}^{(s)}$, $\omega_{Q}^{(s)} = \omega_{Q}^{(s)} / \sum_{s=1}^{2} \omega_{Q}^{(s)}$ are the normalized weighting factors. To emphasize the fault alarm information, if the statistical information means there is a fault, a larger weighting factor should be assigned. Thus
3. Case study

The proposed DeDPCA method was then compared with the KPCA, DPCA and DePCA methods. The comparison is via a case study by applying them to the Tennessee Eastman process. DePCA combines PCA and KPCA, while DeDPCA combines DPCA and KPCA to construct the deep model. To confirm the excellence of the proposed DeDPCA method, firstly it was compared with DePCA to examine the importance of deep mining. Then we changed the feature extraction methods in different layers to determine which structure is better. Finally, the improved DeDPCA was compared with DeDPCA to show the benefits of extending time series based on each variable.

3.1. TE process

The Tennessee Eastman process (e.g. Lyman & Georgaklis, 1995 and Lau et al., 2013) is a benchmark process whose flowchart is illustrated in Figure 3. The TE process is constructed by five main units, hence reactor, condenser, compressor, stripper and separator and eight material components that represented from A to H. In process modelling, 52 variables are selected as monitored variables, which include 12 manipulated variables, 22 measured variables, and 19 composition measurements.

Training and testing datasets were collected from the TE simulation for 48 h of operation. The data of each variable was collected in every 3 min. In the simulation, the fault is introduced at 8 h. Each data set contains 960 samples and faults occurs at the 161th sample. The relative simulation results can be obtained from Braatz (2009).

3.2. Implementation details

The Gaussian kernel function was chosen with a kernel width of $\sigma = 100n$ for nonlinear mapping based on tests. For DeDPCA, the lag $l$ is set to 2 according to Ku’s theory (Ku et al., 1995). The numbers of the reserved principal components are decided by over 90% of the total sum of eigenvalues to keep most of the information. Hotelling’s $T^2$ and SPE(Q) statistics were used together to represent the current system status. Kernel density estimation was adopted to calculate the control limits and determining the boundaries of normal and faulty for each layer and statistic. Bayesian inference based fusion method was used to combine all layers of statistical information and to draw the final decision. We denote the monitoring statistics of six consecutive samples exceeding the control limit as a fault in the deep methods to avoid the influence of noise empirically.

All data were standardized at first to ensure the contribution of each variable is equal when reducing dimensionality. Standardization involves two steps. The first is to subtract the sample mean of each variable. The next is to divide the variable after subtracting the sample mean by its standard deviation, and scale each variable to unit variance.
Three indices were used to represent the performance of the fault detection methods. The first is called fault detection rate (FDR). The ratio of a method can detect a fault in faulty condition. The second is called fault detection time (FDT). It is how much time a method need to first detect the fault in faulty condition. The last is false alarming rate (FAR), its the ratio of a method can wrongly detect a fault in normal condition. Fault detection rate shows the accurateness of detection method. Fault detection time shows the sensitivity and promptness. False alarming rate shows the robustness of a fault detection method. Therefore, a fault detection method is more reliable than other methods if the FDR of this method is numerically higher and the FDT and FAR is numerically lower than those of others.

### 3.3. Results and analysis

Table 1 shows the fault detection results of 21 TE faults using DPCA, KPCA, DePCA and DeDPCA methods. To facilitate comparison, we divide the 21 TE faults in to three parts. The first part including F1, F2, F4, F6, F7, F8, F12, F13, F14, F17 and F18 as they can be easily detected. The FDRs of these faults using KPCA and DPCA all greater than
The second part including F5, F10, F11, F16, F19, F20 and F21. These faults can be detected most of the time using KPCA and DPCA. The third part including F3, F9 and F15. F3, F9 and F15 are the faults most difficult to detect in TE process because there are basically no different between the F3, F9 and F15 faulty samples and normal samples.

To appreciate the superior of multi-layer feature extraction, single-layer methods DPCA and KPCA are first compared with multi-layer methods DePCA and DeDPCA. In the first part faults, all the four fault detection methods can easily detect these faults as their FDRs are close to 100%. But DePCA and DeDPCA can detect the faults earlier than DPCA and KPCA. Besides, DePCA and DeDPCA have lower false alarming rates than DPCA and KPCA. As the confidence level was set to 95%, the ideal FAR is less than 5%. But the average FAR of DPCA is over 5%. In the first part faults, some faults that are not easily be detected by DPCA or KPCA but can easily be detected by DePCA and DeDPCA especially for F5, F10, F16, F19 and F20. In the second part faults, all these four methods can hardly detect these faults. Overall, multi-layer methods can detect more faults, find faults earlier and cause less false alarming rates.

The largest fault detection rate of each fault is in bold. For DePCA and DeDPCA, DeDPCA gets the larger FDRs for most faults except fault F21, F3 and F9, while the FDRs of F21, F3 and F9 for DePCA and DeDPCA are very similar. It can be concluded that DeDPCA is more sensitive to faults in terms of FDRs. For fault detection time, most faults can be detected using less than 10 samples time in first part and the mean FDT for DePCA and DeDPCA are only one sample point difference. In the second part, DeDPCA can detect faults 4 samples earlier than DePCA for average. The most obvious difference is F20, which need 65 samples time to detect using DePCA but 17 samples time using DeDPCA. In the third part, it is the same for DePCA and DeDPCA to detect fault F3 and F9. However, DeDPCA can greatly reduce the time to detect f5. For false alarming rate, they are similar and both less than 5% for average. Thus DeDPCA gets larger average FDR and smaller average FDT, indicating that DeDPCA is a more efficient method for fault detection. These results validate the benefits of extending the time series for fault detection, rather than not consider timing dependencies.

With the aim of verifying the current deep model gets more excellent performance, we designed another deep model which also used DPCA for dynamic feature extraction.
The effectiveness of proposed model has been verified above. However, because we use a time lag shift strategy, the number of variables increases which costs more time in calculation. To solve this, we changed the way of time lag selection, and we called the new method improved DeDPCA. Table 3 demonstrates the average fault detection results and operation time of 21 TE faults using original DeDPCA and improved DeDPCA. The computation time for a sample was relatively recorded 10 times and find the average. The computer is equipped with Intel Core i5-8500U CPU(3.0 GHz) and 8G RAM memory. Original DeDPCA and improved DeDPCA have not much different as for FDRs, FDTs and FARs, while improved DeDPCA can reduce the computation time by 34% than before which means that many redundant calculations are reduced.

By comparison with FDRs, FDTs and FARs of DeDPCA and other methods, DeDPCA was found to achieve higher FDR, earlier FDT and lower FAR. That is, the proposed DeDPCA method is likely to detect faults earlier and more precisely than other methods. Besides, extending time series by each variable can reduce the computation complexity and keep good performance at the same time. Hence, the DeDPCA method is an efficient technique that suitable for nonlinear dynamic process monitoring.

### 4. Discussion

#### 4.1. Summary of outcomes

A new hierarchical statistical model structure namely DeDPCA was designed based on DPCA and KPCA algorithms. The proposed method contains two layers, hence extraction. The original model called DeDPCA1 firstly extracts dynamic features using DPCA and then extracts nonlinear features using KPCA. The another model called DeDPCA2 firstly extracts nonlinear features using KPCA and then extract dynamic features using DPCA. The relative monitoring results are shown in Table 2. It is apparent to see that the average FDR of DeDPCA1 is larger than that of DeDPCA2 over 4%. Although the average FDT of DeDPCA1 is later than that of DeDPCA2, DeDPCA1 causes lower false alarming. Therefore extracting the dynamic features at first layer seems to be more reasonable. Otherwise extracting the dynamic features at second layer may lose some dynamic information due to the nonlinear transformation and independent assumption of the first KPCA layer.

The effectiveness of proposed model has been verified above. However, because we use a time lag shift strategy, the number of variables increases which costs more time in calculation. To solve this, we changed the way of time lag selection, and we called the new method improved DeDPCA. Table 3 demonstrates the average fault detection results and operation time of 21 TE faults using former DeDPCA and improved DeDPCA.
DPCA and KPCA. The DPCA extracts the linear dynamic features while the KPCA accounts for the nonlinearities of the process. Bayesian inference was adopted to combine features of different layers and conduct the overall decision. The proposed method has been employed to the benchmark TE process for its application and validation. The monitoring performance of the proposed DeDPCA was compared with that of KPCA, DPCA and DePCA. FDR, FDT and FAR were used to evaluate the monitoring performance of the proposed method with that of all other methods considered in this study. The result is that the DeDPCA attained higher FDR and earlier FDT and smaller FAR than DePCA. Besides, the necessity of extracting dynamic features in first layer was verified and a new timing extension technology was used to reduce the calculation complexity.

4.2. Limitations of the work

A limitation of this work may be that it is difficult to compute the contributions of each variable for deep-layer monitoring statistics. They are calculated after several mappings from original variables.

Another limitation is that DPCA is the most simple and straightforward method for dynamic process monitoring. Although DPCA can detect faults, because lagged variables are considered, diagnosis of abnormal behaviour becomes more complicated (Treasure et al., 2004). Besides, Negiz and Çinar (1998) noted that principal components extracted by this method are not exactly the minimal dynamic representations. It would also influence the nonlinear feature extraction in the second layer.

4.3. Further research

Future work could associate with determining the number of feature layer for data mining and how to arranging the relation between different layers. Moreover, determining the variables that causing the fault using contribution map may be one of the potential work.

5. Conclusions

Conventional fault detection methods are normally with one-singlelayer of feature extraction, which cannot extract sufficient information. Although the deep fault detection methods was proposed recently, the methods simply extract deep features and linear features. They assumed unwisely that the variables are time independent. However, most industrial production is nonlinear and dynamic, which is different from conventional methods. The paper here proposes a solid integration between deep fault detection and dynamic methods. It provides a novel method to monitor the safety of complex nonlinear dynamic system.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the Institute of Zhejiang University-Quzhou Science and Technology Project (I-ZQ2019-KJ-021).

References

Braatz, R. (2009). Tennessee eastman problem simulation data. Retrieved from http://web.mit.edu/braatzgroup/links.html.
Cai, B., Huang, L., & Xie, M. (2017). Bayesian networks in fault diagnosis. IEEE Transactions on Industrial Informatics, 13(5), 2227–2240. https://doi.org/10.1109/TII.2017.2695583
Deng, X., Tian, X., Chen, S., & Harris, C. J. (2016). Nonlinear process fault diagnosis based on serial principal component analysis. IEEE Transactions on Neural Networks and Learning Systems, 29(3), 560–572. https://doi.org/10.1109/TNNLS.2016.2635111
Deng, X., Tian, X., Chen, S., & Harris, C. J. (2018). Deep principal component analysis based on layerwise feature extraction and its application to nonlinear process monitoring. IEEE Transactions on Control Systems Technology, 27(6), 2526–2540. https://doi.org/10.1109/TCST.87
Gao, Z., Saxen, H., & Gao, C. (2013). Guest editorial: Special section on data-driven approaches for complex industrial systems. IEEE Transactions on Industrial Informatics, 9(4), 2210–2212. https://doi.org/10.1109/TII.2013.2281002
Iqbal, R., Maniak, T., Doctor, F., & Karyotis, C. (2019). Fault detection and isolation in industrial processes using deep learning approaches. IEEE Transactions on Industrial Informatics, 15(5), 3077–3084. https://doi.org/10.1109/TII.9424
Kramer, M. A. (1991). Nonlinear principal component analysis using autoassociative neural networks. AIChE Journal. American Institute of Chemical Engineers, 37(2), 233–243. https://doi.org/10.1002/(ISSN)1547-5905
Ku, W., Storer, R. H., & Georgakis, C. (1995). Disturbance detection and isolation by dynamic principal component analysis. Chemometrics and Intelligent Laboratory Systems, 30(1), 179–196. https://doi.org/10.1016/0169-7439(95)00076-3
Lau, C., Ghosh, K., Hussain, M. A., & Hassan, C. C. (2013). Fault diagnosis of tennessee eastman process with multi-scale PCA and ANFIS. Chemometrics and Intelligent Laboratory Systems, 120(1), 1–14. https://doi.org/10.1016/j.chemolab.2012.10.005
LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539
Lee, J.-M., Yoo, C., Choi, S. W., Vanrolleghem, P. A., & Lee, I.-B. (2004). Nonlinear process monitoring using kernel principal component analysis. Chemical Engineering Science, 59(1), 223–234. https://doi.org/10.1016/j.ces.2003.09.012
Lyman, P. R., & Georgakis, C. (1995). Plant-wide control of the Tennessee Eastman problem. *Computers & Chemical Engineering, 19*(3), 321–331. https://doi.org/10.1016/0098-1354(94)00057-U

Negiz, A., & Çinar, A. (1998). Monitoring of multivariable dynamic processes and sensor auditing. *Journal of Process Control, 8*(5-6), 375–380. https://doi.org/10.1016/S0959-1524(98)00006-7

Odiowei, P.-E. P., & Cao, Y. (2009). Nonlinear dynamic process monitoring using canonical variate analysis and kernel density estimations. *IEEE Transactions on Industrial Informatics, 6*(1), 36–45. https://doi.org/10.1109/TII.2009.2032654

Pan, T., Chen, J., Xie, J., Zhou, Z., & He, S. (2020). Deep feature generating network: A new method for intelligent fault detection of mechanical systems under class imbalance. *IEEE Transactions on Industrial Informatics, 17*(9), 6282–6293. https://doi.org/10.1109/TII.2020.3030967

Russell, E. L., Chiang, L. H., & Braatz, R. D. (2000). Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems, 51*(1), 81–93. https://doi.org/10.1016/S0169-7439(00)00058-7

Treasure, R. J., Kruger, U., & Cooper, J. E. (2004). Dynamic multivariate statistical process control using subspace identification. *Journal of Process Control, 14*(3), 279–292. https://doi.org/10.1016/S0959-1524(03)00041-6

Wang, G., Jiao, J., & Yin, S. (2016). A kernel direct decomposition-based monitoring approach for nonlinear quality-related fault detection. *IEEE Transactions on Industrial Informatics, 13*(4), 1565–1574. https://doi.org/10.1109/TII.2016.2633989

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems, 2*(1-3), 37–52. https://doi.org/10.1016/0169-7439(87)80084-9

Woolrich, M. W., Behrens, T. E., Beckmann, C. F., Jenkinson, M., & Smith, S. M. (2004). Multilevel linear modelling for fMRI group analysis using bayesian inference. *NeuroImage, 21*(4), 1732–1747. https://doi.org/10.1016/j.neuroimage.2003.12.023

Wu, H., & Zhao, J. (2018). Deep convolutional neural network model based chemical process fault diagnosis. *Computers & Chemical Engineering, 115*(620), 185–197. https://doi.org/10.1016/j.compchemeng.2018.04.009

Yin, S., Ding, S. X., Haghani, A., Hao, H., & Zhang, P. (2012). A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process. *Journal of Process Control, 22*(9), 1567–1581. https://doi.org/10.1016/j.jprocont.2012.06.009