Neural Computing and Applications (2021) 33:10129–10139
https://doi.org/10.1007/s00521-021-05776-3

ORIGINAL ARTICLE

Nonlinear quality-relevant process monitoring based on maximizing correlation neural network

Shifu Yan[1] · Xuefeng Yan[1]

Received: 11 May 2020 / Accepted: 28 January 2021 / Published online: 12 March 2021

Abstract
Quality-relevant fault detection aims to reveal whether quality variables are affected when a fault is detected. For current industrial processes, kernel-based methods focus on the nonlinearity within process variables, which is insufficient for obtaining nonlinearities of quality variables. Alternatively, neural network is an option for nonlinear prediction. However, these models are driven by predictive errors on samples. For quality-relevant tasks, the key is to capture the trends of quality variables. Therefore, this study proposes a new model, namely, maximizing correlation neural network (MCNN), to predict the quality-relevant information intuitively. The MCNN is trained to maximize the linear correlation between quality variables and the combinations of nonlinear representations mapped by a multilayer feedforward network. As such, fault detection can be implemented in the quality-relevant and irrelevant subspaces on the basis of the deep most correlated representations of process variables. Considering that different variables have different sensitivities to quality at various locations due to their nonlinear relationship, fault backpropagation is designed in the MCNN to isolate the faulty variables on the basis of real-time faulty information. Finally, numerical example and Tennessee Eastman process are used to evaluate the proposed method, which exhibits a competitive performance.

Keywords Neural network · Quality-relevant · Fault detection · Fault isolation

1 Introduction

The improvement of computer technology has spawned large-scale industrial production. Plant-wide units are difficult to monitor by manpower in real-time because nonlinearities, coupling, and other complex relationships exist in numerous measurements. Therefore, data-driven monitoring methods have been widely studied [1–4]. Quality-relevant tasks that reveal whether quality variables are affected or not have recently been a popular research topic in process monitoring. Thus, faults can be classified as quality relevant or irrelevant [5, 6]. In this way, unnecessary downtime of the units can be avoided, and corporate economic losses are reduced.

Given that quality variables are often difficult or costly to measure with a typical time delay, they are monitored by process variables that can be collected in time. As such, least squares (LS) and partial least squares (PLS) are commonly used to model the relationship between these two sets of variables and extract quality-relevant information in the process space. For the purpose of detecting quality-relevant faults, Zhou et al. [7] and Qin et al. [8] derived four and three subspaces, respectively, on the basis of the standard PLS model to improve the monitoring performance. Wang et al. [9] proposed principal component regression (PCR), which directly performs LS between quality variables and latent features extracted by principal component analysis (PCA). On this basis, various methods, such as mutual information [10] and locally linear embedding [11], were adopted to enhance the division of quality-relevant information.

However, these linear methods may be insufficient for the current industries wherein high nonlinearities exist. Therefore, kernel tricks are introduced. Peng et al. [12] first extended the total PLS in [9] to a kernel version (TKPLS).

✉ Xuefeng Yan
xfyan@ecust.edu.cn

[1] Key Laboratory of Smart Manufacturing in Energy Chemical Process, Ministry of Education, East China University of Science and Technology, 200237 Shanghai, People's Republic of China

Jiao et al. [13] proposed the modified kernel PLS (MKPLS) for the nonlinear process. Wang et al. presented kernel PCR (KPCR) [11] and modified kernel LS [14] to handle the complex relationships within process variables. Given the idea of deep leaning, Deng et al. [15], respectively, proposed deep PCA for complex feature extraction. Jiang et al. [16] learned the deep correlated representations by deep canonical correlation analysis. Although these methods can learn the nonlinearities of variables, they are insufficient for the nonlinear relationship between process and quality variables. For quality-relevant tasks, Yuan et al. [17] applied the variable-wise weighted stack autoencoder (SAE) in the modeling of refinery process. Dong et al. [18] proposed a method based on stacked automatic encoder−canonical correlation analysis and least absolute shrinkage selection operator for fault diagnosis. However, the neural network (NN)-based methods model the quality variables by minimizing the predictive errors of samples that are sensitive to extreme values and local information. While for quality-relevant fault detection, we only need to capture the trend of quality variables.

After fault detection, isolation is often implemented for faulty variables. Traditional methods are commonly based on the variable contribution to Hotelling's T^2 or Q statistics that are used to detect faults, and studies of quality-relevant fault isolation are limited [19]. These methods treat quality variables in the same manner. However, process variables have different sensitivities from quality variables at different locations based on nonlinear relationships. Thus, different variables require various treatments for isolating faulty variables.

On the basis of the above analysis, a new model, namely, maximizing correlation neural network (MCNN), is proposed in this study to learn the deep most correlated representations of process variables to predict quality in an intuitive and interpretive manner. First, representations of process variables are mapped by a feedforward NN. Then, this NN is trained to maximize the linear correlation between quality variables and the combination of such representations. Given a well-trained MCNN, quality-relevant fault detection can be implemented using the deep most correlated representations by T^2 and corresponding thresholds determined by kernel density estimation (KDE). In addition, the idea of fault backpropagation (FBP) is presented to reveal the most sensitive variables to the quality. A new approach for calculating the variable contribution is designed on the basis of such faulty information to isolate faulty variables. Finally, a numerical example and Tennessee Eastman process (TEP) are used to evaluate the proposed method. The contributions of this study are listed as follows: 1) propose a new model to learn the nonlinear quality-relevant representations of process variables intuitively, 2) implement quality-relevant fault detection in an interpretive manner, and 3) find a new approach for calculating the variable contribution based on FBP using real-time faulty information.

2 Related works

Given n normalized observations $\mathbf{X} \in \Re^{n\times m}$ and $\mathbf{Y} \in \Re^{n\times p}$ with m process variables and p quality variables, respectively, take PCR as an example to illustrate quality-relevant task as follows.

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \tilde{\mathbf{X}} \tag{1}$$

where $\mathbf{T} \in \Re^{n\times A}$, $\mathbf{P} \in \Re^{m\times A}$. Quality information is predicted as $\hat{\mathbf{Y}} = \mathbf{T}\mathbf{Q}^T$ and $\mathbf{Q}^T = \left(\mathbf{T}^T\mathbf{T}\right)^{-1}\mathbf{T}^T\mathbf{Y}$. As such, quality-relevant subspace $\mathbf{X}_y = \mathbf{T}_y\mathbf{P}_y^T$ and quality-irrelevant subspace $\mathbf{X}_o = \mathbf{T}_o\mathbf{P}_o^T$ are determined. To detect whether a new sample $\mathbf{x}_{new} \in \Re^m$ is faulty or fault-free, Hotelling's T^2 statistic is often adopted. The quality-relevant and irrelevant statistics of $\mathbf{x}_{new}$ is calculated as follows:

$$\begin{aligned} T_y^2 &= \mathbf{t}_{\text{ynew}}^T\left[\mathbf{T}_y^T\mathbf{T}_y/(n-1)\right]^{-1}\mathbf{t}_{\text{ynew}}, \\ T_o^2 &= \mathbf{t}_{\text{onew}}^T\left[\mathbf{T}_o^T\mathbf{T}_o/(n-1)\right]^{-1}\mathbf{t}_{\text{onew}} \end{aligned} \tag{2}$$

where $\mathbf{t}_{ynew}$, $\mathbf{t}_{ynew}$ are the quality-relevant and irrelevant scores of the new sample. Then the thresholds of the statistics are defined [9].

$$\begin{aligned} J_{thy} &= \frac{A_y(n^2-1)}{n(n-A_y)}F_\alpha(A_y, n-A_y), \\ J_{tho} &= \frac{A_o(n^2-1)}{n(n-A_o)}F_\alpha(A_o, n-A_o) \end{aligned} \tag{3}$$

where $F_\alpha(a, b)$ is the F distribution with freedom a, b and significant level α. Thus the decision can be made by

$$\begin{cases} T_y^2 \le J_{thy} \text{ and } T_o^2 \le J_{tho} \Rightarrow x_{new} \text{ is fault-free} \\ T_y^2 > J_{thy} \Rightarrow \text{quality relevant fault occur} \\ T_y^2 \le J_{thy} \text{ and } T_o^2 > J_{tho} \Rightarrow \text{quality irrelevant fault occur} \end{cases} \tag{4}$$

3 Methodology

3.1 Maximizing correlation neural network

Traditional mean square error mainly focuses on the predictive performance in every sample point. While for quality-relevant process monitoring, the key is to predict the variation trend of quality variables. MCNN is proposed to capture the trends of quality variables intuitively through a macro-objective and learn the most correlated representations of process variables in a more interpretive manner than kernels.

Here, $\mathbf{X} \in \Re^{n \times m}$ is considered with single quality variable $\mathbf{y} \in \Re^{n}$. First, $\mathbf{X}$ is mapped by a multilayer feedforward NN with o neurons in the output layer, and the output $\mathbf{F} = [f_1(\mathbf{X};\boldsymbol{\theta}_1), f_2(\mathbf{X};\boldsymbol{\theta}_2), \ldots, f_o(\mathbf{X};\boldsymbol{\theta}_o)] \in \Re^{n \times o}$, where $f(\cdot)$ is the transformation function and $\boldsymbol{\theta}$ is the set of trainable parameters in $f(\cdot)$, including weights or bias. Specifically, for a single neuron in the networks, the output can be calculated as $x_{out} = \sigma(\mathbf{w}^T \mathbf{x}_{in} + b)$, where $\mathbf{w}$, b is trainable and $\sigma(\cdot)$ is the activation function. On the basis of such structure, an arbitrarily complex relationship can be approximated accurately. The objective of MCNN is presented in Eq. (5) on the basis of the Pearson coefficient.

$$\begin{aligned} \arg\max_{\boldsymbol{\theta}} \rho &= \boldsymbol{\gamma}^T \mathbf{H}^T \mathbf{y} \\ s.t.\ \boldsymbol{\gamma}^T \mathbf{H}^T \mathbf{H} \gamma &= 1 \end{aligned} \tag{5}$$

where $\mathbf{H} = \mathbf{F} - (1/n)\mathbf{1}_n \mathbf{1}_n^T \mathbf{F}$ is the centered matrix of $\mathbf{F}$. In order to make this objective computable, we introduce another vector $\boldsymbol{\upsilon}$ that is related to $\boldsymbol{\gamma}$, and let $\mathbf{S}_{xx} = \mathbf{H}^T \mathbf{H}$, $\mathbf{S}_{xy} = \mathbf{H}^T \mathbf{y}$, and $\boldsymbol{\gamma} = \mathbf{S}_{xx}^{-1/2} \boldsymbol{\upsilon}$. Thus, the objective can be rewritten as follows:

$$\begin{aligned} \arg\max_{\boldsymbol{\theta}} \rho &= \boldsymbol{\upsilon}^T \mathbf{S}_{xx}^{-1/2} \mathbf{S}_{xy} \\ s.t.\ \boldsymbol{\upsilon}^T \boldsymbol{\upsilon} &= 1 \end{aligned} \tag{6}$$

Once $\boldsymbol{\theta}$ is initialized, Eq. (6) can be achieved by performing singular value decomposition (SVD) on $\mathbf{K} = \mathbf{S}_{xx}^{-1/2} \mathbf{S}_{xy}$. $\mathbf{v}$ is the vector that corresponds to the nonzero singular value. Mini-batch backpropagation is used to train such NN, and the value of this objective can be calculated as follows:

$$\rho = \left(\mathbf{K}^T \mathbf{K}\right)^{-1/2} \tag{7}$$

The overall training procedure of MCNN is summarized in **Algorithm 1**, wherein the gradient of this objective is derived in the Appendix A.

Algorithm 1: Training MCNN with single quality variable.

```
Require: $X \in \Re^{n \times m}$, $y \in \Re^{n}$
Begin
  Initialize the parameters $\theta$ of MCNN.
  For $i = 1$ to the number of epochs do
    For $j = 1$ to the number of batches do
      1) Draw a batch with $N$ observations of pair samples
         $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$;
      2) Forward calculation of the objective by $\rho = \left(K^T K\right)^{-1/2}$;
      3) Update the parameters of MCNN using $\theta \leftarrow \theta - \eta \frac{\partial \rho}{\partial \theta}$.
    End for
  End for
End
```

For multiple quality variables $\mathbf{y} = \left[y_1, y_2, \ldots, y_p\right]^T$, the objective of MCNN is defined as

$$\begin{aligned} &\arg\max_{\boldsymbol{\theta}} \rho = \sum_{i=1}^{p} \boldsymbol{\gamma}_i^T \mathbf{H}^T \mathbf{y}_i \\ &s.t.\ \frac{1}{p} \sum_{i=1}^{p} \boldsymbol{\gamma}_i^T \mathbf{H}^T \mathbf{H} \gamma_i = 1 \end{aligned} \tag{8}$$

By combining the terms, Eq. (8) can be transformed into Eq. (9), which can be treated in the same manner as a single quality variable.

$$\arg\max_{\boldsymbol{\theta}} \rho = \left[\boldsymbol{\gamma}_1^T, \boldsymbol{\gamma}_2^T, \ldots, \boldsymbol{\gamma}_p^T\right] \mathbf{H}^T \left[\mathbf{y}_1^T, \mathbf{y}_2^T, \ldots, \mathbf{y}_p^T\right]^T \tag{9}$$

3.2 Scheme for detecting faults

Fault detection method is designed in this section on the basis of the most correlated representations $\mathbf{F}$ learned by MCNN. Given that $\mathbf{F}$ is linear to the quality variables, the quality-relevant load matrix can be computed as follows

$$\mathbf{Q}^T = \left(\mathbf{F}^T \mathbf{F}\right)^{-1} \mathbf{F}^T \mathbf{Y} \tag{10}$$

Thus, quality-relevant scores can be obtained as $\mathbf{T}_y = \mathbf{F}\mathbf{Q}^T$, and the quality-relevant subspace $\mathbf{X}_y$ can be computed together with $\mathbf{X}_o = \mathbf{X} - \mathbf{X}_y$. Quality-irrelevant score $\mathbf{T}_o$ and corresponding load matrix $\mathbf{P}_o$ are then obtained by PCA.

Given a new normalized sample $\mathbf{x}$, the deep correlated representation is $f(\mathbf{x})$, and the quality-relevant and irrelevant scores are calculated.

$$\mathbf{t}_y = \mathbf{Q} f(\mathbf{x}) \tag{11}$$

$$\mathbf{t}_o = \mathbf{P}_o\left(\mathbf{x} - \mathbf{P}_y \mathbf{t}_y\right) \tag{12}$$

Therefore, the statistics T_y^2 and T_o^2 can be obtained using Eq. (2). As the data in real industries do not follow the accurate normal distribution, KDE is used to estimate the distribution of the T^2 statistics and determine the thresholds [20]. $\left\{T_1^2, T_2^2, \ldots, T_n^2\right\}$ of the training n samples are calculated, which follows the probabilistic density function $\phi(T^2)$. This function can be estimated on the basis of the Gaussian kernel function in Eq. (13) or other available functions.

$$\hat{\phi}(T^2) = \frac{1}{nb} \sum_{i=1}^{n} \exp\left(-\frac{1}{2}\left(\frac{T^2 - T_i^2}{b}\right)^2\right) \tag{13}$$

where b is the bandwidth parameter. Given a significant level α, threshold can be determined by

$$prob\left(T^2 \leq J_{th}\right) = \int_0^{J_{th}} \hat{\phi}(T^2) d(T^2) = \alpha \tag{14}$$

Therefore, fault detection can be implemented on the basis of the logic proposed in Eq. (4).

3.3 Scheme for isolating variables

For quality-relevant fault isolation, each quality variable is considered separately. The faulty variables that affect the quality variables can be specified in accordance with the contribution of variables to the quality-relevant or irrelevant statistics. On the basis of the well-trained MCNN, one of the quality variables can be predicted between the representations $\mathbf{F}$ by LS and the overall function is indicated as $\hat{y} = \Upsilon(\mathbf{x})$. Assume that $\Upsilon(\mathbf{x})$ is smooth which can be satisfied due to the network construction and it can be expanded at $\mathbf{x}_{new}$.

$$\Upsilon(\mathbf{x}) = \Upsilon(\mathbf{x}_{new}) + \boldsymbol{\Gamma}_{\Upsilon}(\mathbf{x}_{new})(\mathbf{x} - \mathbf{x}_{new}) + \mathrm{O}\left(\|\mathbf{x} - \mathbf{x}_{new}\|^2\right) \tag{15}$$

where $\boldsymbol{\Gamma}_{\Upsilon}(\mathbf{x}_{new}) \in \Re^m$ is the tangent direction, which can reveal the sensitivities of $\mathbf{x}$ to $\Upsilon(\mathbf{x}_{new})$. Therefore, quality-relevant and irrelevant subspaces indicated as $\mathbf{x}_{ynew} \equiv span\{\boldsymbol{\Gamma}_{\Upsilon}(\mathbf{x}_{new})\}$ and $\mathbf{x}_{onew} \equiv span\{\boldsymbol{\Gamma}_{\Upsilon}(\mathbf{x}_{new})\}^{\perp}$ can be determined on the basis of the information of sensitivities.

Perform SVD on $\boldsymbol{\Gamma}_{\Upsilon}(\mathbf{x}_{new})\boldsymbol{\Gamma}_{\Upsilon}^T(\mathbf{x}_{new})$ as

$$\boldsymbol{\Gamma}_{\Upsilon}(\mathbf{x}_{new})\boldsymbol{\Gamma}_{\Upsilon}^T(\mathbf{x}_{new}) = \begin{bmatrix} \mathbf{u}_1 & \tilde{\mathbf{U}} \end{bmatrix} \begin{bmatrix} \lambda & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^T \\ \tilde{\mathbf{U}}^T \end{bmatrix} = \lambda \mathbf{u}_1 \mathbf{u}_1^T \tag{16}$$

where $\tilde{\mathbf{U}} = [\mathbf{u}_2, \mathbf{u}_3, \ldots, \mathbf{u}_m]$. Then, $\mathbf{x}_{ynew}^T = \mathbf{x}_{new}^T \mathbf{u}_1 \mathbf{u}_1^T$ and $\mathbf{x}_{onew}^T = \mathbf{x}_{new}^T \tilde{\mathbf{U}}\tilde{\mathbf{U}}^T$. Given that the faulty samples deviate from the normal values, training samples $\mathbf{x}_i, i = 1, 2, \ldots, n$ without disturbance are used to calculate the baseline as follows:

$$\bar{\mathbf{x}}_y = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_{y,i},\ \bar{\mathbf{x}}_o = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_{o,i} \tag{17}$$

Thus, the contribution of the kth variable to the faults can be defined as follows:

$$\begin{aligned} Cont(\mathbf{x}_{ynew}, k) &= \mathbf{x}_{ynew,k} - \bar{\mathbf{x}}_y,\ Cont(\mathbf{x}_{onew}, k) \\ &= \mathbf{x}_{onew,k} - \bar{\mathbf{x}}_o \end{aligned} \tag{18}$$

3.4 Implementation details

On the basis of the above description, the overall scheme of quality-relevant detection and isolation based on MCNN is summarized as follows.

3.4.1 Offline modeling:

(1) Collect n samples $\mathbf{X} \in \Re^{n \times m}$ and $\mathbf{Y} \in \Re^{n \times p}$, and normalize them to zero mean and unit variance;
(2) Train an MCNN using $\boldsymbol{X}$ and $\boldsymbol{Y}$ to obtain $\mathbf{F} = f(\mathbf{X})$;
(3) Calculate offline scores $\mathbf{T}_y = \mathbf{F}\left(\mathbf{F}^T\mathbf{F}\right)^{-1}\mathbf{F}^T\mathbf{Y}$, $\mathbf{T}_o = \left(\mathbf{X} - \mathbf{T}_y\mathbf{P}_y^T\right)\mathbf{P}_o$;
(4) Determine thresholds J_{thy} and J_{tho} that, respectively, satisfy $\int_0^{J_{thy}} \hat{\phi}\left(T_y^2\right) d\left(T_y^2\right) = \alpha$ and $\int_0^{J_{tho}} \hat{\phi}\left(T_o^2\right) d\left(T_o^2\right) = \alpha$ given a significant level (99% is set in the following experiments);
(5) Calculate the baseline of variable contributions $\overline{\mathbf{x}}_y = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_{y,i}$ and $\overline{\mathbf{x}}_o = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_{o,i}$.

3.4.2 Online monitoring:

(1) Collect a new normalized sample $\mathbf{x}_{new}$;
(2) Calculate statistics T_y^2 and T_o^2 using $\mathbf{t}_{ynew} = \mathbf{Q}f(\mathbf{x}_{new})$ and $\mathbf{t}_{onew} = \mathbf{P}_o\left(\mathbf{x}_{new} - \mathbf{P}_y\mathbf{t}_y\right)$;
(3) Make decisions of fault detection following the logic in Eq. (4);
(4) Isolate the faulty variables by $Cont(\mathbf{x}_{ynew}, k) = \mathbf{x}_{ynew,k} - \overline{\mathbf{x}}_y$, $Cont(\mathbf{x}_{onew}, k) = \mathbf{x}_{onew,k} - \overline{\mathbf{x}}_o$.

4 Experimental results

4.1 Experimental setup

The following experiments run on a computer with Intel Xeon Silver 4110 2.1 GHz, 64GB RAM. The hyper-parameters of MCNN are specified as follows: (1) rectified linear unit is adopted as activation; (2) optimizer, Adam with learning rate = 0.001, (3) batch size = 100, (4) epoch = 500, and (5) dropout rate = 0.1. Monitoring a new sample can be finished in 2ms, thus meeting the real-time requirement. For evaluation, false alarm rate (FAR) and fault detection rate (FDR) are considered to evaluate the monitoring results.

$$\begin{aligned} FAR &= prob\left\{T^2 > J_{th} \middle| f = 0\right\}, \\ FDR &= prob\left\{T^2 > J_{th} \middle| f \neq 0\right\} \end{aligned} \tag{19}$$

Table 1. Possible results of one-class classification

-	Predictive positive	Predictive negative
Real positive	True positive (TP)	False negative (FN)
Real negative	False positive (FP)	True negative (TN)

In this way, fault-free and faulty samples can be regarded as two classes that can be indicated as positive and negative classes, respectively. Table 1 presents four results in this type of tasks. To provide an overall result, receiver operating characteristic (ROC) curve is commonly adopted. This curve uses the 1-specificity defined by true positive rate (TPR) as the horizontal axis and the specificity defined by false positive rate (FPR) as the vertical axis. The calculations of TPR and FPR are expressed as follows:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}},\ \mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}} \tag{20}$$

Generally, the ROC curve starts at (0, 0) and the optimal parameters is the point nearest to (0, 1). Efficient performance mainly relies on a large area (AROC) under the ROC curve. The quantitative indicator accuracy (ACC) can be calculated as

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{21}$$

4.2 Numerical example

4.2.1 Data generation

A numerical example containing seven process variables $\mathbf{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7]^T$ and a quality variable y is designed in this section.

$$\begin{cases} x_1 = t_1 + e_1 \\ x_2 = t_1 + t_2^2 + e_2 \\ x_3 = t_3 + t_4 + e_3 \\ x_4 = t_3^2 - 2t_5 + e_4 \\ x_5 = t_5 - t_5^3 + e_5 \\ x_6 = t_1 + \sin(t_2) + e_6 \\ x_7 = t_2 - t_3 + e_7 \\ y = x_2x_4 + x_3 + e_8 \end{cases} \tag{22}$$

where t_i, $i = 1, 2, ..., 5$ are the latent variables following the uniform distribution $U(0, 1)$. Moreover, Gaussian noises e_k, $k = 1, 2, ..., 8$ are added to simulate further realistic data. x_2, x_3, x_4 are relevant to the quality, and the remaining variables are irrelevant. Hence, faults that occur on x_2, x_3, x_4 will affect the values of y.

In this case, a total of 800 samples without disturbance are generated for offline modeling by an MCNN with 7-7-7-3 neurons. In addition, two types of faults are simulated to evaluate the proposed method as follows, and 800 samples are involved in each faulty dataset.

(1) *Fault 1*: quality-relevant fault with the faulty magnitude $f = 2, 2.5, 3, 3.5, 4$ interferes x_3 in the last 600 samples;

Table 2 FDRs (%) of MCNN, KPCR, KPLS, MKPLS, TKPLS and TSSAE for numerical example

Fault No.	Faulty magnitude	MCNN		KPCR		KPLS		MKPLS			TKPLS		TSSAE		
		T_y^2	T_o^2	T_y^2	T_o^2	T^2	Q	T_y^2	T_o^2	T_y^2	T_o^2	T_r^2	Q_r	D_y	D_o	D_x
1	2	50.5	**98.7**	41.8	71.8	36.2	92.5	40.6	93.5	44.8	72.5	83.2	78.7	45.7	97.6	97.7
	2.5	71.7	**100**	57.5	95.2	55.6	99.8	55.2	96.2	62.7	92.8	97.3	96.2	61.3	97.2	98.2
	3	90.9	**100**	82.3	**100**	74.8	**100**	79.5	**100**	84.8	**100**	**100**	**100**	84.5	**100**	**100**
	3.5	98.5	**100**	93.8	**100**	91.7	**100**	91.5	**100**	95.8	**100**	**100**	**100**	94.2	**100**	**100**
	4	99.9	**100**	96.7	**100**	98.3	**100**	95.6	**100**	98.3	**100**	**100**	**100**	97.5	**100**	**100**
2	2	2.1	**100**	1.7	99.8	6.2	98.8	1.4	99.1	0.5	**100**	90.3	79.3	0.7	99.7	99.9
	2.5	1.1	**100**	2.7	**100**	8.2	**100**	1.9	**100**	0.1	**100**	99.2	97.5	1.2	**100**	**100**
	3	1.1	**100**	5.2	**100**	16	**100**	3.2	**100**	0.5	**100**	**100**	**100**	2.1	**100**	**100**
	3.5	0.8	**100**	6.8	**100**	22.7	**100**	3.2	**100**	0.5	**100**	**100**	**100**	1.8	**100**	**100**
	4	1.5	**100**	9.5	**100**	33.2	**100**	10.2	**100**	0.9	**100**	**100**	**100**	2.2	**100**	**100**

The bold values indicate the highest detection rates

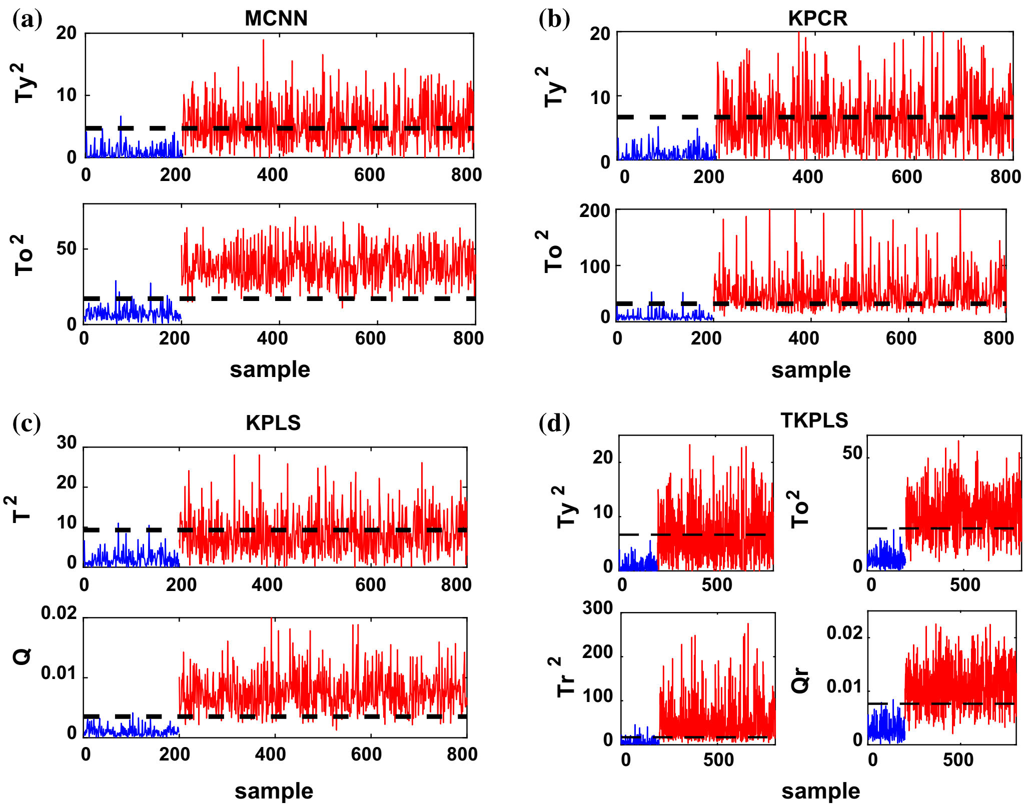

Fig. 1 Monitoring results of fault 1: **a** MCNN, **b** KPCR, **c** KPLS, **d** TKPLS. The black dash lines are the thresholds for corresponding statistics. Statistics for fault-free and faulty samples are plotted in blue and red, respectively.

(2) *Fault 2:* quality-irrelevant fault with faulty magnitude $f = 2, 2.5, 3, 3.5, 4$ interferes x_7 in the last 600 samples.

4.2.2 Monitoring results and discussion

To monitor this example, statistics involving T_y^2 and T_o^2 based on representations learned by MCNN are calculated. Table 2 lists the monitoring results through different models. The number of principal components of these models are determined by a fivefold cross-validation:

Table 3 ACCs of MCNN, KPCR, KPLS and TKPLS for fault 1

Faulty magnitude	MCNN		KPCR		KPLS		MKPLS			TKPLS			TSSAE		
	T_y^2	T_o^2	T_y^2	T_o^2	T^2	Q	T_y^2	T_o^2	T_y^2	T_o^2	T_r^2	Q_r	D_y	D_o	D_x
2	86.4	**97.8**	83.8	**97.8**	80	96.1	82.1	97.4	84.6	94.3	90.8	91.6	85.1	96.5	97.5
2.5	92.4	**99.9**	92	98.4	87	99.1	91.6	98.6	92.6	99	95.4	97.5	92.1	99.8	98.7
3	94.2	**99.9**	94.3	99.6	92.9	**99.9**	93.4	99.7	94.8	**99.9**	98.6	99.6	93.1	99.8	99.8
3.5	97.4	**99.9**	97.4	**99.9**	96.6	**99.9**	97	**99.9**	97.6	**99.9**	**99.9**	**99.9**	97.4	**99.9**	**99.9**
4	99.5	**99.9**	99	**99.9**	99.1	**99.9**	99.2	**99.9**	99.1	**99.9**	**99.9**	**99.9**	99.2	**99.9**	**99.9**
Average	94	**99.5**	93.3	99.1	91.1	99	92.7	99.1	93.7	98.6	96.9	97.7	93.4	99.2	99.2

The bold values indicate the highest detection rates

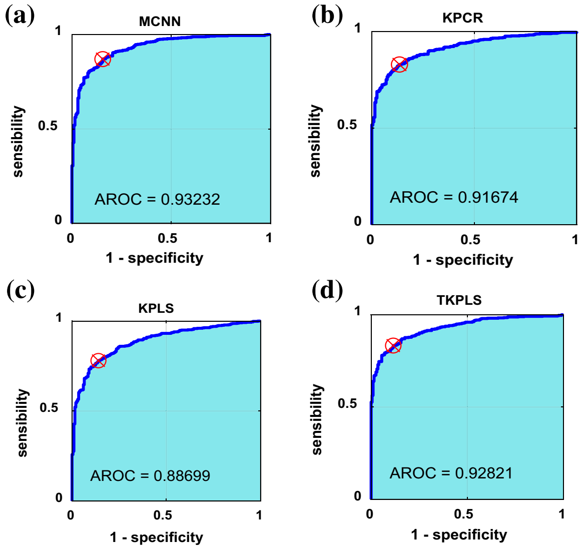

Fig. 2 ROC curves of T_y^2 for fault 1 with $f = 2$. **a** MCNN, **b** KPCR, **c** KPLS, **d** TKPLS. The values of AROC indicate the size of the shadow areas and the red remark is the optimal threshold where the values of ACCs are calculated.

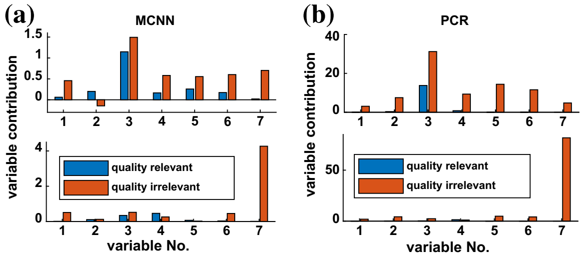

Fig. 3 Results (average values of five samples) of fault isolation. **a** MCNN, **b** PCR.

KPCR ($A = 6$), KPLS ($A = 4$), TKPLS ($A_o = 5$, $A_r = 4$). Detail results of fault 1 are presented in Fig. 1. Only the FDRs of the statistics are presented, because the FARs of these methods are at the same level as our expectations. For quality-irrelevant statistic T_o^2, these two faults are almost successfully recognized by these methods and MCNN can achieve competitive results. However, MCNN captures the nonlinearity of quality variables and detects the quality-relevant faults to be more sensitive than others for quality-relevant statistics T_y^2.

On the basis of the above-mentioned findings, these methods provide reliable results for the two faults because the quality-relevant or irrelevant faults can be specified. For more comparison, the ACCs of monitoring these faults by these methods are calculated and summarized in Table 3. Furthermore, Fig. 2 presents the ROC curve apart from the results of FARs and FDRs to evaluate the proposed method for quality-relevant subspace in fault 1. As such, the first 200 and last 600 samples are regarded as two classes, namely, fault-free and faulty. MCNN with large values of AROC outperforms other methods for quality-relevant faults based on the results of ROC curves.

After the fault is detected, fault isolation must be implemented to find the faulty variables for stable operations. Based on the well-trained MCNN, the FBP method is performed to calculate the variable contribution to quality in Fig. 3a and b where results in the above subgraph represent for fault 1 and the following is fault 2. Obviously, the faulty variables have been successfully discovered by MCNN and PCR (x_3 for fault 1 and x_7 for fault 2) that are consistent with the faulty types. Since fault 2 only relates to quality-irrelevant subspace, the results are significant. However for fault 1, the contribution of variables in the quality-irrelevant subspace by PCR (x_3) is nearly equal to that (x_5,x_6) in the quality-relevant subspace for fault 1. Therefore, MCNN achieved a better isolated result in quality relevant part. Specifically, FBP can weigh the contributions through the backpropagation of faults and thus exhibits a further significant result, which is efficient for operators to isolate these variables and restore the production.

Therefore, the quality-relevant information is accurately modeled by MCNN and improved performance is exhibited, especially for quality-relevant fault because it can sensitively detect the anomaly situation.

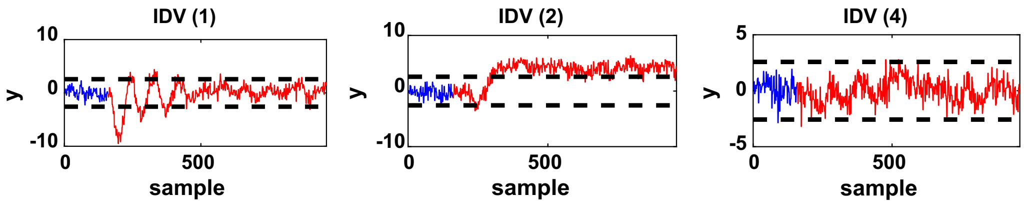

Fig. 4 Real values of IDV (1), (2) and (4). The black dash lines represent the upper and lower quantiles of normal distribution where $\Phi(2.5878) = 0.995$ and $\Phi(-2.5878) = 0.005$. The blue and red values of quality variable are before and after faults occur.

4.3 Application for the Tennessee Eastman process

4.3.1 Brief introduction to TEP

The TEP provided by Downs and Vogel is a simulated platform for real industrial process, which has been used as a benchmark problem, especially for process monitoring [21]. TEP comprises five units, namely, a reactor, condenser, separator, compressor, and stripper, wherein four reactants (A, C, D, and E) together with an inert B generate the products (G, H) and byproduct (F). As such, control strategies and monitoring schemes can be implemented by relying on 52 measurements with 41 measured variables, i.e., XMEAS (1)–(41) and 11 manipulated variables, i.e., XMV (1)–(11) [22]. Two types of datasets have been collected under 24 and 48 h, and the sampling interval is 3 min. Both datasets contain a fault-free case, IDV (0), and 21 faulty cases, IDV (1)–(21), which are available at http://web.mit.edu/braatzgroup/links.html.

In this study, the datasets collected under 48h are considered, the component G of purge gas XMEAS (35) is selected as the quality variable, and 22 measured variables XMEAS (1)–(22) and manipulated variables XMV (1)–(11) are used for monitoring. For convenience, $[x_1, x_2, \ldots, x_{33}]^T$ and y are used in the following descriptions. The fault-free samples in IDV (0) including 960 samples are used for training an MCNN with 33-33-33-20 neurons for process modeling. The faulty samples in IDV (1)–(21) with 960 samples per case are used for testing and faults are introduced in the last 800 samples. Quality-relevant and irrelevant faults must be specified before applying the monitoring methods. Assuming that the variables are normally distributed, the influence of faults on the quality variables can be specified. Therefore, quality-irrelevant faults involve IDV (3), (4), (9)–(11), and (14)–(17) in which y is unaffected by these disturbances. The rest are quality-relevant faults that can be further divided into two types. One includes IDV (1), (5), and (7), which indicate the faults that can recover due to the close-loop control strategies. The other type involves IDV (2), (6), (8), (12), (13), (18), and (21), which cannot recover after the faults occur. Figure 4 presents the normalized samples of IDV (1), (2), and (4). To reveal the real situations of y, the proportion of samples outside the normal area N_t is calculated, and the normal area is defined between the black dash lines in Fig. 4.

4.3.2 Monitoring results and discussion

Table 4 summarizes the overall FDRs (%) of different methods, including KPCR, KPLS, TKPLS, MKPLS, and MCNN. The number of principal components of these models is determined as: KPCR ($A = 9$), KPLS ($A = 6$), TKPLS ($A_o = 7$, $A_r = 6$). The FAR of the monitoring results are limited to the expected levels and average values of different methods are present as: MCNN (0.9%), KPCR (1.2%), KPLS (0.7%), MKPLS (1.2%), TKPLS (0.7%), and TSSAE (1.2%). For quality-irrelevant statistics, the MCNN outperforms other methods in detecting most faults. Given that the real numbers of faulty samples of y is absent, the monitoring results of quality-relevant statistic T_y^2 are evaluated based on the values of N_t. In this way, MCNN exhibits a more reliable result for quality-relevant faults than others based on the average value of FDRs. The excellent and reliable performance mainly relies on the accurate information of process and quality variables captured by the MCNN and is efficient in helping operators master the production of industries. IDV (5), (19) are discussed in detail.

IDV (5) is a step fault in the inlet temperature of the condenser cooling water which is indicated as x_{33} in this study. As described previously, y in this case is affected at the beginning when the fault introduced and recovers with in short period (around 10 hours) because of the application of close-loop strategy. Figure 5 presents the monitoring results by MCNN, KPCR, and TKPLS. The quality-relevant statistics of these methods are close to the real values, whereas the MCNN obtains better results, because FDR reaches 100% and the quality-irrelevant statistics of KPCR and TKPLS only detect part of the samples. After that, FBP is implemented for the MCNN in Fig. 6 to isolate the faulty variables of IDV (5). Obviously, the faulty variable x_{33} can be easily isolated based on the most significant contribution in this figure. The description of IDV (19) is unknown and y is unaffected based on its real values in this case. The N_t

Table 4 FDRs (%) of MCNN, KPCR, KPLS, TKPLS, MKPLS and TSSAE for TEP

Fault No.	Real	KPCR		KPLS		MKPLS			TKPLS			TSSAE			MCNN	
	N_t	T_y^2	T_o^2	T^2	Q	T_y^2	T_o^2	T_y^2	T_o^2	T_r^2	Q_r	D_y	D_o	D_x	T_y^2	T_o^2
1	14.6	38.1	99.5	99.6	*99.8*	96.9	*99.8*	96.9	99.6	98.8	99.3	25.8	99.5	*99.8*	26.6	**100**
2	65.2	87.8	98.3	98	98.0	89.5	98.5	89.5	98.0	98.0	97.0	75.5	**98.6**	*98.5*	95.5	98.4
3	2.7	**13.8**	1.5	2.5	0.3	1.5	0.6	1.5	1.6	0.1	0.0	2.1	1.9	*6.8*	1.6	5.0
4	1.3	8.5	95.3	66.6	60.9	20.4	99.8	20.4	48.8	0.3	14.0	0.4	99.8	**100**	6.3	**100**
5	9.6	30.8	23.9	24.9	21.6	10.8	**100**	10.8	24.6	9.6	22.0	17.4	*31.4*	**100**	6.5	**100**
6	79.6	10.0	*99.5*	*99.5*	**100**	97.5	**100**	97.5	98.8	97.8	**100**	98.6	**100**	**100**	96.5	**100**
7	13.1	40.9	**100**	*99.0*	**100**	55.8	**100**	55.8	97.3	91.0	95.9	58.1	**100**	**100**	17.8	**100**
8	44.8	89.4	*97.6*	96.9	97.1	64.9	*97.6*	64.9	94.4	81.6	95.0	85.1	96.0	96.2	51.4	**98.0**
9	1.5	**11.1**	3.6	1.1	0.1	2.0	0.4	2.0	0.9	0.4	0.0	4.1	4.5	*9.8*	2.0	4.5
10	5.0	36.9	52.1	82.4	35.6	62.6	69.4	62.6	79.4	2.4	32.9	33.0	63.5	*84.4*	14.8	**89.6**
11	1.3	13.8	*86.1*	54.8	48.3	19.4	67.1	19.4	52.3	2.1	36.6	3.4	83.3	**89.1**	8.6	81.3
12	49.8	87.3	*99.8*	98.8	97.9	68.0	99.4	68.0	97.3	74.4	97.0	68.9	99.3	*99.8*	40.9	**99.9**
13	52.3	84.8	95.0	94.6	93.8	86.8	*95.1*	86.8	94.5	70.8	94.3	88.1	92.3	92.7	59.4	**95.3**
14	0.6	2.5	**100**	98.5	99.9	70.5	**100**	70.5	98.6	33.1	*99.9*	20.1	**100**	**100**	35.3	**100**
15	1.3	**15.5**	4.4	6.3	0.1	4.9	2.5	4.9	3.8	0.0	1.0	4.1	0.9	*8.2*	3.5	7.4
16	2.7	28.6	30.3	63.4	15.0	38.6	65.3	38.6	51.0	0.3	16.4	28.4	33.5	*86.5*	6.1	**92.9**
17	4.6	11.3	*96.8*	82.3	84.8	48.8	94.3	48.8	79.8	37.0	80.5	16.0	87.5	91.8	56.3	**97.4**
18	71.5	19.8	*89.6*	89.0	89.4	87.6	*89.8*	87.6	88.6	87.4	89.1	86.9	87.0	89.4	82.6	**90.6**
19	1.3	3.3	17.9	2.8	2.5	1.4	30.5	1.4	1.6	1.3	0.0	0.6	29.3	*89.5*	0.1	**94.5**
20	3.8	35.6	59.9	37.1	38.5	16.1	57.9	16.1	33.3	2.8	36.0	11.8	75.3	*91.1*	6.9	**91.4**
21	20.4	50.0	*55.9*	51.9	34.0	54.5	45.4	54.5	37.6	6.1	32.3	55.4	43.8	49.1	11.4	**57.0**
Avg*	21.3	34.3	67.0	64.3	58.0	47.5	72.1	47.5	61.0	37.9	54.2	37.32	68.0	*80.2*	30.0	**81.1**

* Avg is the average value of IDV (1)-(21)

The highest detection rates are in bold and the second highest detection rates are in italic

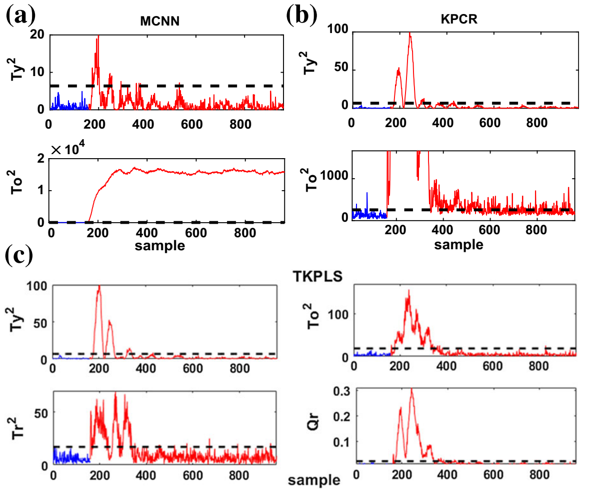

Fig. 5 Monitoring results of IDV (5). **a** MCNN, **b** KPCR, **c** TKPLS.

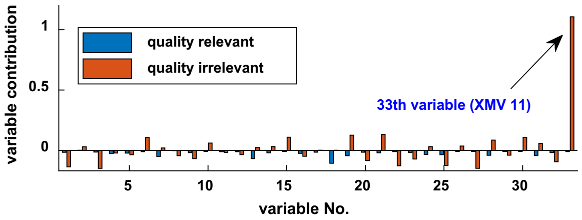

Fig. 6 Results (average values of five samples) of fault isolation for IDV (5) by MCNN.

calculated for $[x_1, x_2, \ldots, x_{33}]^T$ shows that y is caused by the change in recycle flow which is indicated as x_5 in this study. Figure 7 displays the monitoring results, and the effect of fault on y is successfully identified by these methods. However, MCNN exhibits a significantly better performance than other methods for quality-irrelevant statistics. Furthermore, isolation is implemented to isolate the faulty variable in Fig. 8. On the basis of the above results, we can reasonably infer that the main cause of IDV (19) is the fluctuation of x_5.

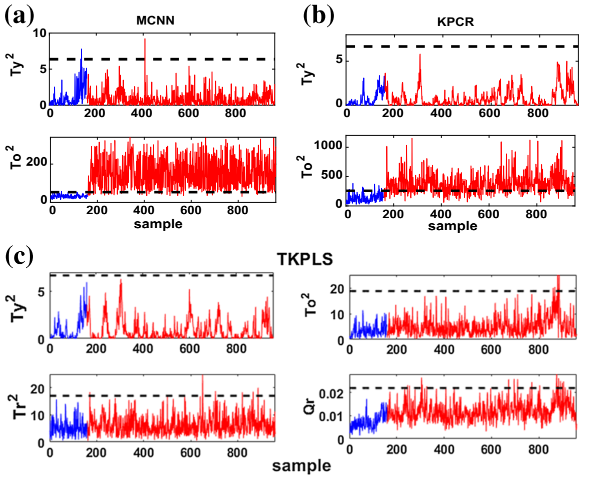

Fig. 7 Monitoring results of IDV (19). **a** MCNN, **b** KPCR, **c** TKPLS.

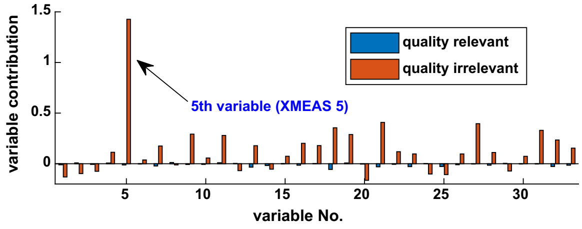

Fig. 8 Results (average values of five samples) of fault isolation for IDV (19) by MCNN.

5 Conclusion

This paper focuses on modeling the quality-relevant information intuitively and learning the deep most correlated representations of process variables. By designing the new objective, MCNN is less sensitive to high noise than the traditional mean square error and can extract the nonlinear representations of quality variables from process variables in a more interpretive way than kernel-based methods. Thus, the trend of quality variables is efficiently captured, and the monitoring performance is enhanced. After that, the FBP reveals the sensitivity of process variables to quality, which can be treated as a weighted regularization for calculating the variable contribution. Compared with traditional statistical methods, the superiority of proposed method mainly relies in the extraction of nonlinear information by innovatively combining the nonlinearly mapping and linear regression. At the same time, the framework is flexible and easy to construct, especially for the large volume of datasets.

For quality-relevant tasks, fault detection and isolation are just the fundamental problem. Apart from the nonlinearities, the changes also lie in the process with strong coupling, dynamic properties or multimode. At the same time, the design of recovering strategies based on the faulty information can be meaningful for industries.

Appendix 1: Derivation of MCNN gradient

In order to perform the mini-batch back propagation algorithm to train MCNN, gradient of the objective $\frac{\partial\rho}{\partial\boldsymbol{\theta}_k}$ is needed, where $\boldsymbol{\theta}_k$ is one of the trainable parameters of MCNN. Based on the chain rule, we can get that

$$\frac{\partial\rho}{\partial\boldsymbol{\theta}_k}=\sum_{ij}\frac{\partial\rho}{\partial\mathbf{F}_{ij}}\frac{\partial\mathbf{F}_{ij}}{\partial\boldsymbol{\theta}_k}\tag{23}$$

Since $\rho=\left(\mathbf{K}^T\mathbf{K}\right)^{-1/2}=\left[\left(\mathbf{S}_{xx}^{-1/2}\mathbf{S}_{xy}\right)^T\left(\mathbf{S}_{xx}^{-1/2}\mathbf{S}_{xy}\right)\right]^{-1/2}$, we firstly show that

$$\frac{\partial\rho}{\partial\left(\mathbf{S}_{xy}\right)_a}=\left(\mathbf{K}^T\mathbf{K}\right)^{-1/2}\left(\mathbf{K}^T\mathbf{S}_{xx}^{-1/2}\right)_a\tag{24}$$

$$\frac{\partial\rho}{\partial(\mathbf{S}_{xx})_{ab}}=-\frac{1}{2}\left(\mathbf{K}^T\mathbf{K}\right)^{-1/2}\left(\mathbf{S}_{xy}\mathbf{S}_{xx}^{-1}\right)_a\left(\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}\right)_b\tag{25}$$

Proof of (32) and (33) is illustrated as follows.

$$\begin{aligned}&\frac{\partial\rho}{\partial\left(\mathbf{S}_{xy}\right)_a}=\frac{\partial\rho}{\partial\left(\mathbf{K}^T\mathbf{K}\right)}\\&\sum_b\frac{\partial\left(\mathbf{K}^T\mathbf{K}\right)}{\partial(\mathbf{K})_b}\frac{\partial(\mathbf{K})_b}{\partial\left(\mathbf{S}_{xy}\right)_a}=\frac{1}{2}\left(\mathbf{K}^T\mathbf{K}\right)^{-1/2}\\&\sum_b 2(\mathbf{K})_b\left(\mathbf{S}_{xx}^{-1/2}\right)_{ba},\\&=\left(\mathbf{K}^T\mathbf{K}\right)^{-1/2}\left(\mathbf{K}^T\mathbf{S}_{xx}^{-1/2}\right)_a\end{aligned}\tag{26}$$

$$\begin{aligned}\frac{\partial\rho}{\partial(\mathbf{S}_{xx})_{ab}}&=\frac{\partial\rho}{\partial\left(\mathbf{K}^T\mathbf{K}\right)}\frac{\partial\left(\mathbf{K}^T\mathbf{K}\right)}{\partial(\mathbf{S}_{xx})_{ab}}\\&=\frac{1}{2}\left(\mathbf{K}^T\mathbf{K}\right)^{-1/2}\sum_{df}\frac{\partial\left(\mathbf{K}^T\mathbf{K}\right)}{\partial\left(\mathbf{S}_{xx}^{-1}\right)_{df}}\frac{\partial\left(\mathbf{S}_{xx}^{-1}\right)_{df}}{\partial(\mathbf{S}_{xx})_{ab}}\\&=-\frac{1}{2}\left(\mathbf{K}^T\mathbf{K}\right)^{-1/2}\sum_{df}\left(\mathbf{S}_{xy}\mathbf{S}_{xy}^T\right)_{df}\left(\mathbf{S}_{xx}^{-1}\right)_{da}\left(\mathbf{S}_{xx}^{-1}\right)_{bf}\\&=-\frac{1}{2}\left(\mathbf{K}^T\mathbf{K}\right)^{-1/2}\left(\mathbf{S}_{xy}\mathbf{S}_{xx}^{-1}\right)_a\left(\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}\right)_b\end{aligned}\tag{27}$$

Next, $\frac{\partial(S_{xx})_{ab}}{\partial\mathbf{F}_{ij}}$ and $\frac{\partial\left(S_{xy}\right)_a}{\partial\mathbf{F}_{ij}}$ are computed as follows.

$$\frac{\partial(\mathbf{S}_{xx})_{ab}}{\partial\mathbf{F}_{ij}}=\frac{\partial\left(\mathbf{F}^T\mathbf{A}\mathbf{F}\right)_{ab}}{\partial\mathbf{F}_{ij}}=\left(\mathbf{F}^T\mathbf{A}\mathbf{J}^{ij}+\mathbf{J}^{ji}\mathbf{A}\mathbf{F}\right)_{ab}\tag{28}$$

$$\frac{\partial(\mathbf{S}_{xy})_{ab}}{\partial\mathbf{F}_{ij}}=\frac{\partial\left(\mathbf{F}^T\mathbf{B}\right)_{ab}}{\partial\mathbf{F}_{ij}}=\left(\mathbf{J}^{ij}\mathbf{B}\right)_{ab}\tag{29}$$

where $\mathbf{J}^{ij}$ is the single-entry matrix, 1 at (i,j) and zero elsewhere, and $\mathbf{A} = \left[\mathbf{I} - (1/n)\mathbf{1}_n\mathbf{1}_n^T\right]^T\left[\mathbf{I} - (1/n)\mathbf{1}_n\mathbf{1}_n^T\right]$, $\mathbf{B} = \left[\mathbf{I} - (1/n)\mathbf{1}_n\mathbf{1}_n^T\right]^T\mathbf{y}$

Since $\frac{\partial \mathbf{F}_{ij}}{\partial \mathbf{\theta}_i}$ can be derived in a traditional way in traditional NNs, it is not repeated here and thus the gradient of MCNN is complete for the backpropagation training.

Acknowledgements This work was supported by the National key research and development program of China (2020YFA0908303) and National Natural Science Foundation of China (21878081).

Compliance with ethical standard

Conflict of interests The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Qin SJ (2012) Survey on data-driven industrial process monitoring and diagnosis. Annu Rev Control 36(2):220–234
2. Ge Z, Song Z, Gao F (2013) Review of recent research on data-based process monitoring. Ind Eng Chem Res 52(10):3543–3562
3. Wang Y, Si Y, Huang B, Lou Z (2018) Survey on the theoretical research and engineering applications of multivariate statistics process monitoring algorithms: 2008–2017. Can J Chem Eng 96(10):2073–2085
4. Yin S, Li X, Gao H, Kaynak O (2015) Data-based techniques focused on modern industry: an overview. IEEE Trans Ind Electron 62(1):657–667
5. Zhang K, Hao H, Chen Z, Ding SX, Peng K (2015) A comparison and evaluation of key performance indicator-based multivariate statistics process monitoring approaches. J Process Contr 33:112–126
6. Yan S, Huang J, Yan X (2019) Monitoring of quality-relevant and quality-irrelevant blocks with characteristic-similar variables based on self-organizing map and kernel approaches. J Process Contr 73:103–112
7. Zhou D, Li G, Qin SJ (2010) Total projection to latent structures for process monitoring. AIChE J 56(1):168–178
8. Qin SJ, Zheng Y (2013) Quality-relevant and process-relevant fault monitoring with concurrent projection to latent structures. AIChE J 59(2):496–504
9. Wang G, Luo H, Peng K (2016) Quality-related fault detection using linear and nonlinear principal component regression. J Frankl Inst 353(10):2159–2177
10. Huang J, Yan X (2017) Quality relevant and independent two block monitoring based on mutual information and KPCA. IEEE Trans Ind Electron 64(8):6518–6527
11. Zhou J, Ren Y, Wang J (2018) Quality-relevant fault monitoring based on locally linear embedding orthogonal projection to latent structure. Ind Eng Chem Res 58(3):1262–1272
12. Peng K, Zhang K, Li G (2013) Quality-related process monitoring based on total kernel PLS model and its industrial application. Math Prob Eng. https://doi.org/10.1155/2013/707953
13. Jiao J, Zhao N, Wang G, Yin S (2017) A nonlinear quality-related fault detection approach based on modified kernel partial least squares. ISA Trans 66:275–283
14. Wang G, Jiao J (2017) A kernel least squares based approach for nonlinear quality-related fault detection. IEEE Trans Ind Electron 64(4):3195–3204
15. Deng X, Tian X, Chen S, Harris CJ (2019) Deep principal component analysis based on layerwise feature extraction and its application to nonlinear process monitoring. IEEE Trans Contr Syst Tech 27(6):2526–2540
16. Jiang Q, Yan S, Yan X, Chen S, Sun J (2020) Data-driven individual–joint learning framework for nonlinear process monitoring. Contr Eng Pract. https://doi.org/10.1016/j.conengprac
17. Yuan X, Huang B, Wang Y, Yang C, Gui W (2018) Deep Learning-based feature representation and its application for soft sensor modeling with variable-wise weighted SAE. IEEE Trans Ind Informat 14(7):3235–3243
18. Dong J, Sun R, Peng K, Shi Z, Ma L (2019) Quality monitoring and root cause diagnosis for industrial processes based on Lasso-SAE-CCA. IEEE Access 7:90230–90242
19. Li G, Qin SJ, Ji Y, Zhou D (2009) Total PLS based contribution plots for fault diagnosis. Acta Autom Sinica 35(6):759–765
20. Lv Z, Yan X, Jiang Q (2016) Batch process monitoring based on multiple-phase online sorting principal component analysis. ISA Trans 64:342–352
21. Downs JJ, Vogel EF (1993) A plant-wide industrial process control problem. Comput Chem Eng 17(3):245–255
22. Ricker NL, Lee JH (1995) Nonlinear model predictive control of the Tennessee Eastman challenge process. Comput Chem Eng 19(9):961–981