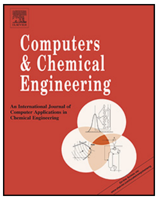
Computers and Chemical Engineering 107 (2017) 395–407

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/compchemeng



A deep belief network based fault diagnosis model for complex chemical processes

Zhanpeng Zhang, Jinsong Zhao*

State Key Laboratory of Chemical Engineering, Department of Chemical Engineering, Tsinghua University, Beijing, China

ARTICLE INFO

Article history:
Received 21 October 2016
Received in revised form 21 February 2017
Accepted 22 February 2017
Available online 24 February 2017

Keywords:
Fault diagnosis
Deep belief network
Feature extraction
Early warning
Alarm management

ABSTRACT

Data-driven methods have been regarded as desirable methods for fault detection and diagnosis (FDD) of practical chemical processes. However, with the big data era coming, how to effectively extract and present fault features is one of the keys to successful industrial applications of FDD technologies. In this paper, an extensible deep belief network (DBN) based fault diagnosis model is proposed. Individual fault features in both spatial and temporal domains are extracted by DBN sub-networks, aided by the mutual information technology. A global two-layer back-propagation network is trained and used for fault classification. In the final part of this paper, the benchmarked Tennessee Eastman process is utilized to illustrate the performance of the DBN based fault diagnosis model.

1. Introduction

Modern chemical processes are highly automated due to the contribution of advanced process control systems. Despite of all the benefits such as lower costs, consistent product quality and improved safety those control systems have to offer, accidents still happen and have caused tremendous casualties, environmental and economic losses. For example, the BP Texas City refinery explosion accident and the PetroChina's Jilin chemical plant explosion accident in 2005 (Shu et al., 2016) occurred in chemical facilities equipped with distributed control systems, alarm systems and safety instrumented systems. Although human factors are usually involved in chemical accidents, abnormal situation management (ASM) is still heavily relying on human operators to handle a large amount of alarms. Therefore, detecting abnormal situations early enough and helping operators make timely and reliable decisions are vital for preventing accidents. As a central component of ASM, process fault detection and diagnosis (FDD) has drawn increasing attention from academia and industry over the last three decades.

The FDD methods can be classified into three parts: quantitative model based, qualitative model based and process history based methods (Venkatasubramanian et al., 2003a, 2003b, 2003c). Among these methods, the quantitative process history based methods or data-driven methods possess larger potential to be applied in the chemical processes. One set of data-driven methods are statistical methods such as principal component analysis (PCA) (Kresta et al., 1991; Russell et al., 2000; Cho et al., 2005; Ge et al., 2009; Fan and Wang, 2014; Rato et al., 2016), partial least squares (PLS) (Piovoso and Kosanovich, 1994), independent component analysis (ICA) (Kano et al., 2003; Lee et al., 2007), fisher discriminant analysis (FDA) (Chiang et al., 2000), subspace aided approach (SAP) (Ding et al., 2009) and correspondence analysis (CA) (Detroja et al., 2007). Yin et al. conducted an interesting comparison study on the above methods and their derivatives through the benchmark Tennessee Eastman (TE) process (Downs and Vogel, 1993). The comparison shows that different FDD methods correspond to distinct fault diagnosis rates and the average fault diagnosis rate of all 21 faults based on TE data sets is about 73.8% to 84.4% (Yin et al., 2012). A framework of Bayesian diagnosis was proposed by Huang (2008). Based on Gaussian mixture model and optimal principal components, a Bayesian diagnosis system was developed for multimode processes (Jiang et al., 2016). Based on Bayesian method, distributed monitor system was effective for large-scale processes (Jiang and Huang, 2016). The other set of data-driven methods is based on pattern classification such as artificial neural network (ANN) (Venkatasubramanian and Chan, 1989; Srinivasan et al., 2005; Eslamloueyan, 2011; Rad and Yazdanpanah, 2015), k-nearest neighbor (K-NN) (He and Wang, 2007), self-organizing map (SOM) (Ng and Srinivasan, 2008a,b; Chen and Yan, 2013), support vector machine (SVM) (Kulkarni et al., 2005; Mahadevan and Shah, 2009) and artificial immune system (AIS) (Dai and Zhao, 2011; Ghosh and Srinivasan, 2011; Shu and Zhao, 2015). Herein, what should be

* Corresponding author.
E-mail address: jinsongzhao@tsinghua.edu.cn (J. Zhao).

http://dx.doi.org/10.1016/j.compchemeng.2017.02.041

mentioned is that data-driven methods are usually combined with the aforementioned distinct methods to achieve desirable performances such as ICA-PCA (Ge and Song, 2007), KICA-SVM (Zhang, 2008), SOM-FDA (Chen and Yan, 2013) and PCA-adaptive Neuro-fuzzy inference system (Lau et al., 2013).

In spite of all the significant contributions to FDD made by a large number of researchers, the implementations of FDD technologies in practical process industry are far from generous due to the undesirable data characteristics including high dimensionality, non-Gaussian distribution, nonlinearity, time-varying and multimode behaviors (Ge et al., 2013). With the big data era coming, information abstracting methodologies such as dimension reduction, variables selection and feature extraction tend to be more indispensable for FDD. For example, latent variable models such as partial least squares or projection to latent structures were used in diagnosis, control and optimization (MacGregor and Cinar, 2012). Mutual information and genetic algorithm (GA) based variables selection was proved to be improving the performance of FDD on the TE process (Verron et al., 2008; Ghosh et al., 2014). However, most of the existing information abstracting methodologies are developed in the spatial domain while the time-varying features in the temporal domain are left poorly studied. The time-varying features are very critical for human experts to differentiate process faults. Dynamic trend analysis therefore was proposed for fault diagnosis (Maurya et al., 2007). However, it is hard for trend analysis to identify faults with complicated time-varying features, such as the random faults in the TE process. In order to extract the features in both the spatial domain and the temporal domain simutaneously, we propose deep learning (DL) for this critical task in this paper.

Since Hinton and Salakhutdinov introduced pre-training and fine-tuning into deep network training, DL has been considered as the most generic and effective methodology for extract information. Using a stack of restricted Boltzmann machines (RBMs) (Smolensky et al., 1986) behaved better compared to using PCA for data dimension reduction (Hinton and Salakhutdinov, 2006a). A deep belief network (DBN) showed excellent classification performance on handwritten digit classification over than K-NN, SVM (Hinton et al., 2006b). Unsupervised pre-training extracts relevant high-level abstract representations from the input data, making DBN effective (Bengio et al., 2007). Hidden layers of deep network are not designed by human engineers beforehand, but automatically determined through unsupervised learning. Deep network can easily take advantage of increases in the amount of computation and data (LeCun et al., 2015). DBN and RBM were used in FDD for some very simple systems with a few variables and faults (Tamilselvan and Wang, 2013; Tran et al., 2014; Sun et al., 2014; Luo et al., 2014). However, since DBN is originally developed for image recognition, it has been applied to systems with binary-value type variables. Applications to complex chemical processes, such that most of the variables are continuous-value type, are not yet reported.

The rest of this paper is organized as follows: Section 2 introduces RBM mathematical model and DBN, the unsupervised learning and supervised training of DBN. The DBN based fault diagnosis model is proposed in Section 3 including details of the model structure and its fault diagnosis procedure. Its application to the benchmarked TE process and comparison with other FDD methods are discussed in Section 4. Finally, conclusions are drawn in Section 5.

2. DBN with continuous-value inputs

The applications of ANNs in process FDD have received much attention since 1980s. Classifiers are trained to diagnose faults based on the pattern recognition ability of ANN. With the back-propagation (BP) training algorithm, data flows from input layer to output layer while error signal is propagated back to optimize the network parameters. Squared error is usually calculated as the error signal to compare the output with target.

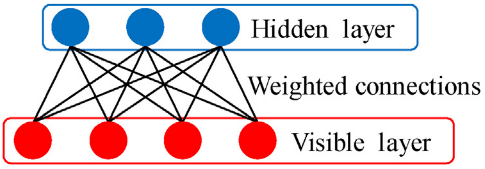


Fig. 1. Schematic of an RBM.

2.1. Restricted Boltzmann machine

RBM is a two-layer neural network. The data input layer or visible layer consists of visible units, and the hidden layer is composed of hidden units. The weighted connections exist between each visible unit and each hidden unit, but it is restricted to set connections between units that belong to the same layer (see Fig. 1).

The joint configuration $(\mathbf{v}, \mathbf{h})$ of the visible units and hidden units are in accordance with the Boltzmann distribution, the probability density of $(\mathbf{v}, \mathbf{h})$ via the energy function $E(\mathbf{v}, \mathbf{h})$ is:

$$p(\mathbf{v}, \mathbf{h}) = \frac{1}{Z} e^{-\mathrm{E}(\mathbf{v},\mathbf{h})}, Z = \sum_{\mathbf{v},\mathbf{h}} e^{-\mathrm{E}(\mathbf{v},\mathbf{h})} \tag{1}$$

And energy function $E(\mathbf{v}, \mathbf{h})$ (Hopfield, 1982) is:

$$E(v, \mathbf{h}) = -\sum_{i \in \text{visible}} a_i v_i - \sum_{j \in \text{hidden}} b_j h_j - \sum_{i,j} v_i h_j w_{ij} \tag{2}$$

where v_i, h_j represent the binary states of visible unit i and hidden unit j, a_i, b_j are their biases and w_{ij} is the weight between unit i and unit j. In an RBM, the value type of the units is binary, which is beneficial for image recognition, especially in the case of handwritten recognition where the image color is either black or white. It should be mentioned that the information contained in the chemical processes data is much more complex and cannot be well-modeled by binary unit based RBMs when it comes to continuous-value data. Most of the data in chemical processes are continuous-values and subjected to Gaussian distribution. Even though Bengio et al. (2007) extended binary RBM to continuous valued inputs, most of the continuous valued RBMs reported are still limited to continuous-values from 0 to 1 (Gao et al., 2014). To further extend the continuous values to an infinite range as from $-\infty$ to $+\infty$, we use Gaussian visible and hidden units in RBMs with the corresponding energy function Eq. (3) (Hinton, 2010),

$$E(\mathbf{v}, \mathbf{h}) = \sum_{i \in \text{visible}} \frac{(a_i - v_i)^2}{2\sigma_i^2} + \sum_{j \in \text{hidden}} \frac{(b_j - h_j)^2}{2\sigma_j^2} - \sum_{i,j} \frac{v_i}{\sigma_i} \frac{h_j}{\sigma_j} w_{ij} \tag{3}$$

which is similar to Eq. (1). In Eq. (3), σ_i, σ_j are standard deviations of the Gaussian noise of the visible unit i and hidden unit j respectively. Via conditional probability of $P(\mathbf{v}|\mathbf{h})$ and $P(\mathbf{h}|\mathbf{v})$, the states of visible unit i and hidden unit j are calculated as Eqs. (4) and (5) (details are shown in Appendix A):

$$h_j = b_j + \sigma_j \sum_{i \in \text{visible}} \frac{v_i}{\sigma_i} w_{ij} \tag{4}$$

$$v_i = a_i + \sigma_i \sum_{j \in \text{hidden}} \frac{h_j}{\sigma_j} w_{ij} \tag{5}$$

2.2. Training of RBM

The weighted connections of RBM are unidirectional so RBM will learn features information from input data without supervision. The Maximum Likelihood theory is used for training an RBM. The aim of training is to increase the probability of input data $P(v)$ by adjusting the weights and biases. To get an idea on the strategy of updating weights, see Eqs. (6) and (7) (details are shown in Appendix A):

$$\Delta w_{ij} = {\frac{v_i}{\sigma_i}\frac{h_j}{\sigma_j}}_{\text{data}} - {\frac{v_i}{\sigma_i}\frac{h_j}{\sigma_j}}_{\text{recon}} \tag{6}$$

$$w_{ij}^{\text{epoch}+1} = w_{ij}^{\text{epoch}} + m\Delta w_{ij}^{\text{epoch}-1} + r\Delta w_{ij}^{\text{epoch}} - dw_{ij}^{\text{epoch}} \tag{7}$$

$$\Delta a_i = {\frac{v_i}{\sigma_i}}_{\text{data}} - {\frac{v_i}{\sigma_i}}_{\text{recon}} \tag{8}$$

$$a_i^{\text{epoch}+1} = a_i^{\text{epoch}} + m\Delta a_i^{\text{epoch}-1} + r\Delta a_i^{\text{epoch}} \tag{9}$$

$$\Delta b_j = {\frac{h_j}{\sigma_j}}_{\text{data}} - {\frac{h_j}{\sigma_j}}_{\text{recon}} \tag{10}$$

$$b_j^{\text{epoch}+1} = b_j^{\text{epoch}} + m\Delta b_j^{\text{epoch}-1} + r\Delta b_j^{\text{epoch}} \tag{11}$$

In Eq. (7), (9) and (11), m is the momentum which can increase the learning speed, r, the learning rate, affects the reconstruction error and noise removing, while d represents weight-decay that penalizes large weights, aiming to provide better performance on testing data. The adjusting procedure of biases a_i and b_j are shown in Eqs. (8)–(11).

The reconstruction error which is the squared error between original visible data and reconstructed ones can be calculated easily, but it is not a suitable quantity for monitoring the training process. Because the aim of RBM training is maximizing the probability of input data, instead of minimizing the reconstruction error, even though they are related. The increase of the reconstruction error is not necessarily indicating the model is going worse. Once the probability of the validating data starts to decrease, the training process should be aborted. Another quantity called free energy is introduced to be monitored in the training process. The free energy $F(\mathbf{v}_{\text{data}})$ is defined by Eq. (12) and (13) (see Appendix A):

$$e^{-F(\mathbf{v}_{\text{data}})} = \int_{\mathbf{h}} e^{-E(\mathbf{v}_{\text{data}},\mathbf{h})} \tag{12}$$

$$F(\mathbf{v}_{\text{data}}) = \sum_{i\in\text{visible}} \frac{(a_i - v_i)^2}{2\sigma_i^2} + \sum_{j\in\text{hidden}} \frac{{b_j}^2 - \left(b_j + \sigma_j \sum_{i\in\text{visible}} \frac{v_i}{\sigma_i} w_{ij}\right)^2}{2\sigma_j^2} - \log\left(\sqrt{2\pi}\sigma_j\right) \tag{13}$$

In each epoch of the training process, we compute average free energy of the training data set and the validating data set respectively. The average free energy decreases with the model performance getting improved. As the model starts to overfit, the average free energy of the validating data will rise relative to the average free energy of the training data. The gap of average free energy related to the validating/training data represents the degree of overfitting (Hinton, 2010).

2.3. Deep belief network

DBN is a generative model with many hidden layers of neurons and Fig. 2 shows the schematic of an example DBN with 5 layers. The first layer, Layer-1, is the input layer and the fourth layer, Layer-4, is the feature layer. From the input layer to the feature layer, each two adjacent layers form an RBM. The weights from one lower layer to its adjacent upper layer, that are represented by W1, W2 and W3 shown in Fig. 2, are referred to as detection weights. Correspondingly, the weights from one upper layer to its adjacent lower layer including $W_1{}^T$, $W_2{}^T$, $W_3{}^T$ are called generative weights, which are transposes of the detection weights. The top layer, Layer-5, is the output layer that executes the classification. Label data represent the class information of the input data and are binary values. In the classification, label data, as targets, are compared with the outputs of Layer-5.

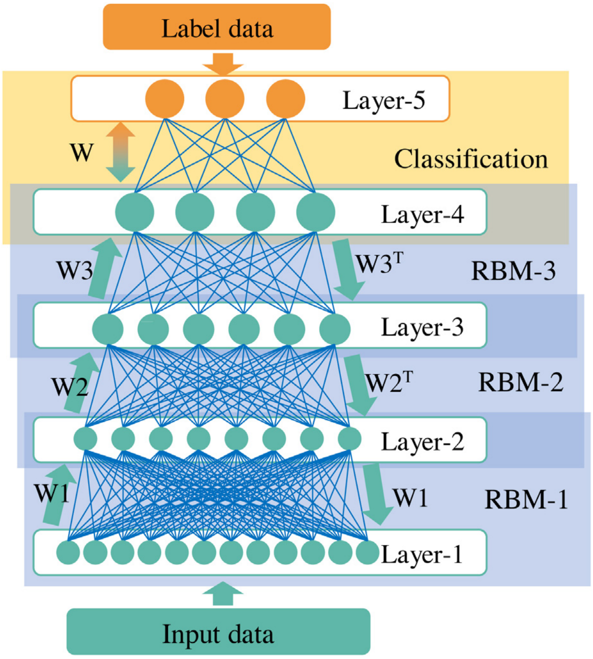


Fig. 2. Schematic of a deep belief network with 5 layers.

2.4. Unsupervised learning

DBN has excellent performance on account of its capability of features extracting. The each hidden layer in DBN extracts feature information from the outputs of itsprevious layer. To motivate the different hidden layers to learn varying level features, neuron numbers of each layer decrease, therefore, the dimension of data can also be reduced without any significant information losses.

The feature extraction is an unsupervised learning process and the reconstruction error is calculated meanwhile to improve performance of network. The BP algorithm is typical for training of neural network, but if used to train the deep network, the training process will fall into Gradient Diffusion (GD) because the error is attenuated when it propagates into deep layers. Pre-training and fine-tuning are introduced to train the deep network and avoid GD problem. In pre-training stage, the hidden layers of DBN are trained layer by layer. Each layer with its previous layer is considered an RBM and each RBM is trained following the process in Section 2.2. After pre-training, the hidden layers constitute a feature extraction model with detection weights, and a data reconstruction model with generative weights that are transpose of detection weights. The BP algorithm is used to train the model for features extraction and data reconstruction. The reconstruction error which is squared error between visible data and reconstructed visible data is calculated as the error signal. Such process is fine-tuning. The error signal is propagated from data reconstruction model to feature extraction model. Feature extraction model is imposed in deep layer so fine-tuning has minor effect on it because of GD. To avoid

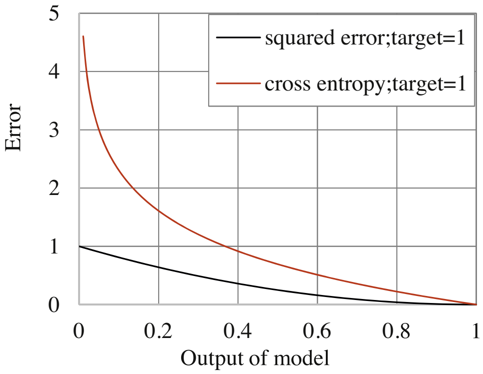


Fig. 3. The comparison of cross-entropy error to squared error.

this, the Wake-Sleep algorithm (Hinton et al., 1995) is introduced in fine-tuning stage. In wake state, only the generative weights are adjusted by the error between input data and reconstruction data. In sleeping state, only the detection weights are adjusted by the error of the features which is detected from input data and reconstruction data. Overall the Wake-Sleep algorithm makes it possible accelerate the convergence of feature extraction.

2.5. Supervised training

Following the unsupervised learning, the feature extraction model can be obtained. With label data, the DBN is supervised to receive training for fault classification. Label data are binary-value and each dimension represents the state of specific fault or normal state. The BP algorithm is used to train the whole DBN, not only in the classification that from feature layer to label data layer, but also in the feature extraction, to get better classification performance. The cross-entropy is calculated as the error between the label data and the classification output data in Eq. (14). As Fig. 3 shows, it's more sensitive of the deviation of model output from the target than squared error.

$$\text{cross entropy} = -t\log\tilde{t} - (1-t) \times \log\left(1-\tilde{t}\right) \tag{14}$$

Where t is the target and $\tilde{t}$ is the output of the model.

3. DBN based fault diagnosis model

Fault is a systematic state that deviates from the normal condition. The root causes may refer to only several factors, but the effects may get the whole system. The DBN based fault diagnosis model extracts feature from a period of process data and classifies the fault state. The input of the model is a one-dimensional vector with $m \times n$ elements, here m is the count of tags and n is the length of a certain period of time. The hidden neurons are halved per layer and the output is a one-dimensional binary value vector with consistent count of faults.

3.1. Variable selection

Variable selection is different from dimension reduction. It is closely related to the fault diagnosis model performance. Whether a variable is valuable or noise is determined by the model performance that if it is improved or on the contrary. A variable selection algorithm based on mutual information was proposed by Sylvain Verron (Verron et al., 2008). This algorithm sorts variables by mutual information values of variable groups for the classification task. Mutual information is the quantitative measurement of correlation among one variable and others. The mutual information of binary-value label data with tag variable groups is calculated and compared so that tags are sorted in order from the most valuable variables to the noisiest variables. Then select the variables into the input data by sequence order and test classification performance to locate optimized variables.

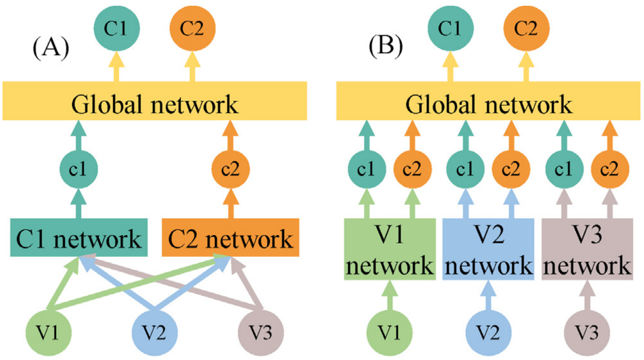


Fig. 4. Structures of two example neural networks with 3 variables and 2 classes. (A) is OCON architecture. (B) is OVON architecture. V represents variable and C represents Class.

3.2. The one-class-one-network architecture

To overcome the curse of dimensionality and complexities in temporal pattern classification, improve the accuracy of classification and minimize the training complexity of neural networks, two new neural network structures, one-class-one-network (OCON) and one-variable-one-network (OVON) architecture were proposed and compared in multi-dimensional temporal pattern classification (Srinivasan et al., 2005). The new neural network structure is composed of several sub-networks and one global network that coordinates the outputs of all sub-networks. The OCON architecture uses a problem decomposition in the class dimension and sub-networks for each pattern class to be identified (as Fig. 4(A)) while the OVON architecture decomposes the problem in the variable dimension and sub-network are corresponding to each variable (as Fig. 4(B)). The OCON architecture exhibits advantages when the count of variables is large and the sub-networks do not necessary have to be retained when a new class is added. In this paper OCON architecture is adopted to improve the performance and extensibility of the fault diagnosis model.

3.3. Fault diagnosis model

The fault diagnosis model is OCON architecture as an example of Fig. 4(A) and the sub-networks are DBNs, the global network is a two-layer BP network. Building and implementation of fault diagnosis model are represented by Fig. 5. Each sub-network of fault or normal state is trained as part (D) to classify if the samples belong to this state or not. The most challenging part of this step is to find suitable parameters. Stochastic Gradient Descent (SGD) method is used at both unsupervised learning and supervised training stage, because the count of the data set is too large to be all calculated in each epoch. SGD method spits the data set into smaller mini-batches stochastically, and calculates each batch in each epoch with a gradient descent algorithm to reduce error. The motivation of using SGD is that after the computation of each batch the parameters of the network are updated, the updating is inconsistent with the optimization direction as the result of the difference between each batch. Training network with SGD method avoids local optimum to some extent and it is fast to obtain optimum parameters. Using Gaussian visible and hidden units requires the learning rate to be about one or two orders of magnitude smaller than using binary-value visible and hidden units because of increasing instability. Part

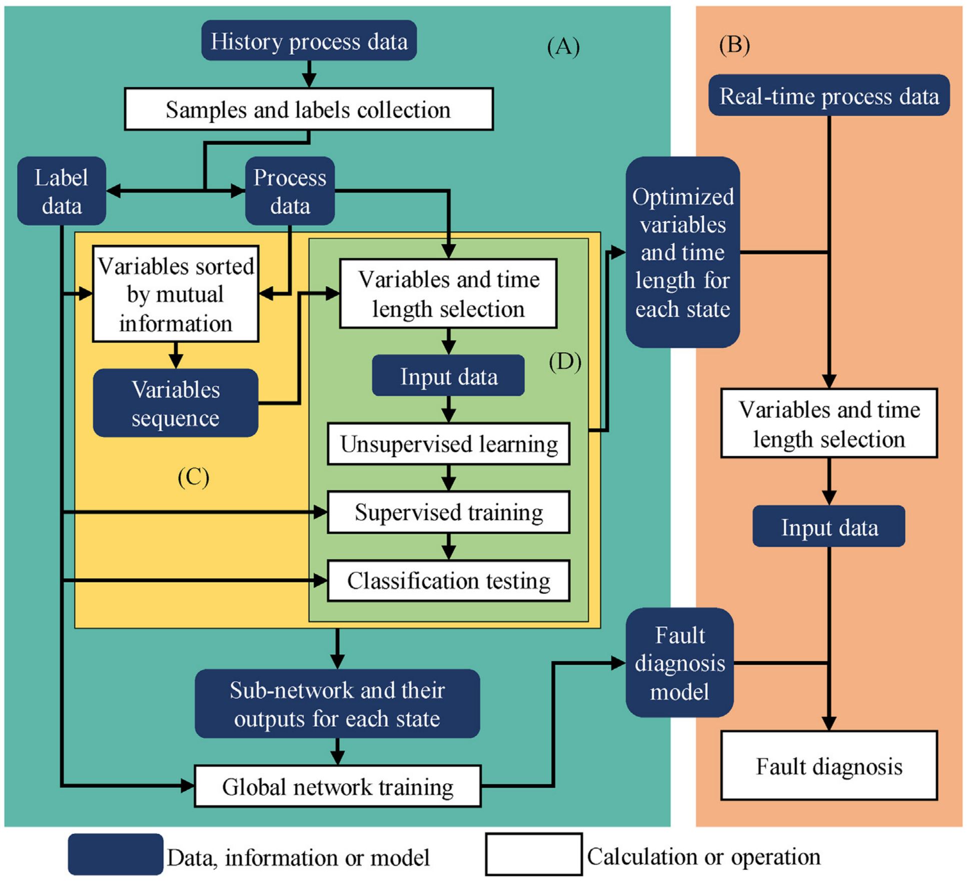


Fig. 5. Framework of fault diagnosis model building and implementation.
Part (A) is the offline stage that the fault diagnosis model is built; Part (B) is the online stage that the fault diagnosis model is applied to real-time data; Part (C) is the optimization of variables and time length selected, and it repeats for every fault or normal state; Part (D) is sub-network training and repeats for every condition of variables and time length.

(D) repeats for every different variables and time length. The optimized variables and time length are selected based on classification performance. Part (C) including variables sorting and sub-network training repeats for each state and finally the global network takes the outputs of sub-networks as input and is supervised training by label data. Because of the extensibility of OCON architecture, the model updating only involves new fault sub-network building and global network re-training for addition of new fault state.

4. Application in the TE process

In this section the proposed DBN based fault diagnosis model is applied to TE process. Moreover, the results for the proposed methods are compared to the results described in existing contributions.

4.1. Tennessee Eastman process

The TE process is the benchmark of fault detection and diagnosis. A revised model was proposed (Bathelt et al., 2015) and more variables and more types of faults were exploded. The revised model with the version of January 23, 2015 is available at http://depts.washington.edu/control/LARRY/TE/download.html. In this paper, the process data are simulated by this revised model (Fig. 6).

To compare with other literatures, the fault diagnosis is on the same basis of variables that include 22 process measurements, 19 component analysis variables and 12 manipulated variables. In the simulation, the sampling period is 3 min, the simulating time is 48 h with fault disturbance added after the first 8 h. With the different random initial status, the simulation of each fault type repeats 10 times, and the normal type repeats 100 times. The XMV(5), XMV(9) and XMV(12) those three manipulated variables are constant values, and the simulation of fault06 is shut down after 15 h, so the process data are 7670 h faults state and 4000 h normal state of 50 variables. The samples are randomly selected from process data, total samples set has 20000 samples and 80% of each faults and normal samples are training data set, others as testing data set (see Tables 1 and 2).

Table 1
The variables and their tag index.

Tag index	Variable number in Downs and Vogel (1993)
Tag01–Tag41	XMEAS(1)–XMEAS(41)
Tag42–Tag45	XMV(1)–XMV(4)
Tag46–Tag48	XMV(6)–XMV(8)
Tag49, Tag50	XMV(10), XMV(11)

Table 2
The status of process data.

Status index	Type	Time length/hours	Count of samples
Fault01–Fault05, Fault06, Fault07	Step	400 × 5 + 70 + 400	500 × 7
Fault08–Fault12	Random	400 × 5	500 × 5
Fault13	Slow drift	400	500
Fault14, Fault15	Sticking	400 × 2	500 × 2
Fault16–Fault20	Unknown	400 × 5	500 × 5
Normal		4000	10000

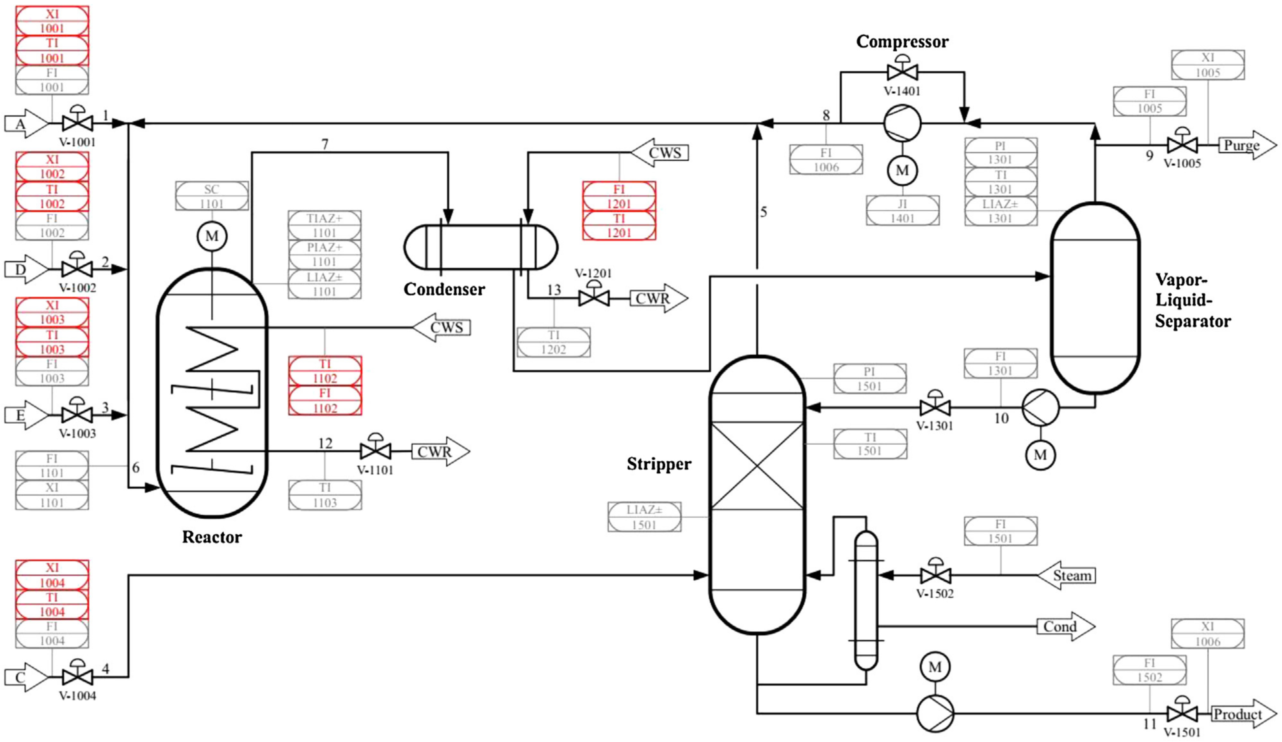


Fig. 6. P&ID of revised TE process model.

4.2. Variables sorting

The mutual information based variable sequences are shown in Table 3. The variable position in a sequence from 1 to 50 represents the degree of its correlation to the fault state, the variable in position 1 is most relative. The mutual information values are the measurements of correlation between samples data and each state of Fault01–Fault20 and Normal state label data. The mutual information values of Fault03, Fault05, Fault09, Fault15 and Fault16 are much small as Fig. 7 shows and the diagnosis of those faults are difficult, according to the literatures, so the main impact is that their process data are similar to Normal state.

4.3. Sub-network of fault classification

Each sub-networks of Fault01–Fault20 are built on only a specific fault and normal state sample data set to get better diagnosis performance. The sub-network of Normal is built on all sample data that detects the state is normal or not. Via the guidance of training RBM (Hinton, 2010), the batch sizes at pre-training step and at fine-tuning/supervised training step are set as 100, 1200 respectively. The learning rate is around 0.00001–0.0001. To enable the network to learn various features, the count of neurons decreases approximately by half by layer. Fault06 data is presented as a detail example. With first 10 tags of sequence and 20 data points of each tag, the input data layer as Layer-1 has 200 neurons. The neuron counts of layers in feature extraction model as Layer-2, Layer-3 and Layer-4 are 93, 45, and 20. The 20% of total 20000 samples data set are testing data and others are training data. After unsupervised learning, the data flows of 100 Fault06 samples show at Fig. 8. From the output of each layer we are convinced, the difference between 100 samples of input data is at Tag05 to Tag10, as the Layer-1 neurons from 81 to 200. However, for other layers, the difference is dispersed to almost every neuron. The difference between each fault state is transferred to different values of the neurons.

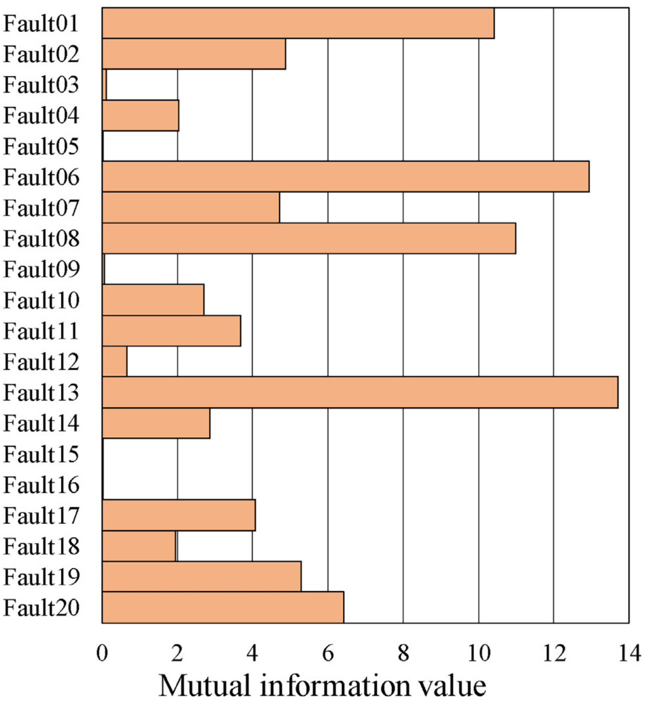


Fig. 7. The mutual information values between 50 variables samples data and all fault and normal state label data.

After unsupervised learning the classification layer is added to feature extraction model and trained with label data. For each state of faults and normal, the fault diagnosis rate (FDR), false positive rate (FPR), accurate classification rate (ACR) are defined as below,

Table 3
The variable sequences based on mutual information.

States	Type	First 10 Tag index of sequence									
Fault01	Step	44	18	47	20	48	46	34	07	21	15
Fault02	Step	34	44	48	28	47	46	42	43	30	15
Fault03	Step	21	16	11	49	42	46	45	18	44	07
Fault04	Step	49	21	09	13	11	45	42	10	44	22
Fault05	Step	50	22	43	11	18	13	37	20	42	34
Fault06	Step	44	18	20	01	38	30	46	10	47	16
Fault07	Step	45	31	16	44	46	18	48	21	11	20
Fault08	Random	44	20	47	48	18	10	07	21	34	30
Fault09	Random	21	16	11	46	49	45	42	18	44	31
Fault10	Random	18	07	11	47	48	22	44	46	21	43
Fault11	Random	49	09	21	16	18	47	46	11	48	22
Fault12	Random	11	18	16	47	44	46	22	20	13	21
Fault13	Slow drift	44	07	47	18	30	21	48	46	34	20
Fault14	Sticking	21	49	09	16	11	42	45	18	46	44
Fault15	Sticking	22	11	18	43	37	16	38	45	34	20
Fault16	Unknown	43	44	38	18	32	42	40	28	35	34
Fault17	Unknown	21	16	18	09	11	47	46	42	44	22
Fault18	Unknown	22	20	11	18	16	47	46	44	13	21
Fault19	Unknown	48	47	18	43	42	46	44	16	45	31
Fault20	Unknown	44	47	20	13	07	18	46	21	11	16
Normal		44	01	21	45	18	16	47	34	48	49

p is the count of this state samples that are classified to this state, q is the count of other state samples that are classified to another state.

$$\mathrm{FDR} = \frac{p}{\text{total count of this state samples}} \tag{15}$$

$$\mathrm{FPR} = 1 - \frac{q}{\text{total count of other state samples}} \tag{16}$$

$$\mathrm{ACR} = \frac{p+q}{\text{total count of samples}} \tag{17}$$

The Sigmoid function and Gaussian function (see Fig. 9) is adopted as the activation function from the feature layer to classification layer. The Sigmoid function is appropriate for step change type faults as Fault01–Fault07 and the FPRs are lower than Gaussian function (see Tables 4 and 5). But Sigmoid function is unable to deal with the faults that their deviations are random. Gaussian function as Eq. (19) is adopted to solve those faults diagnosis and exhibits great performance at random type faults than Sigmoid function (see Fig. 10). So each state of Fault01–Fault20 and Normal will has

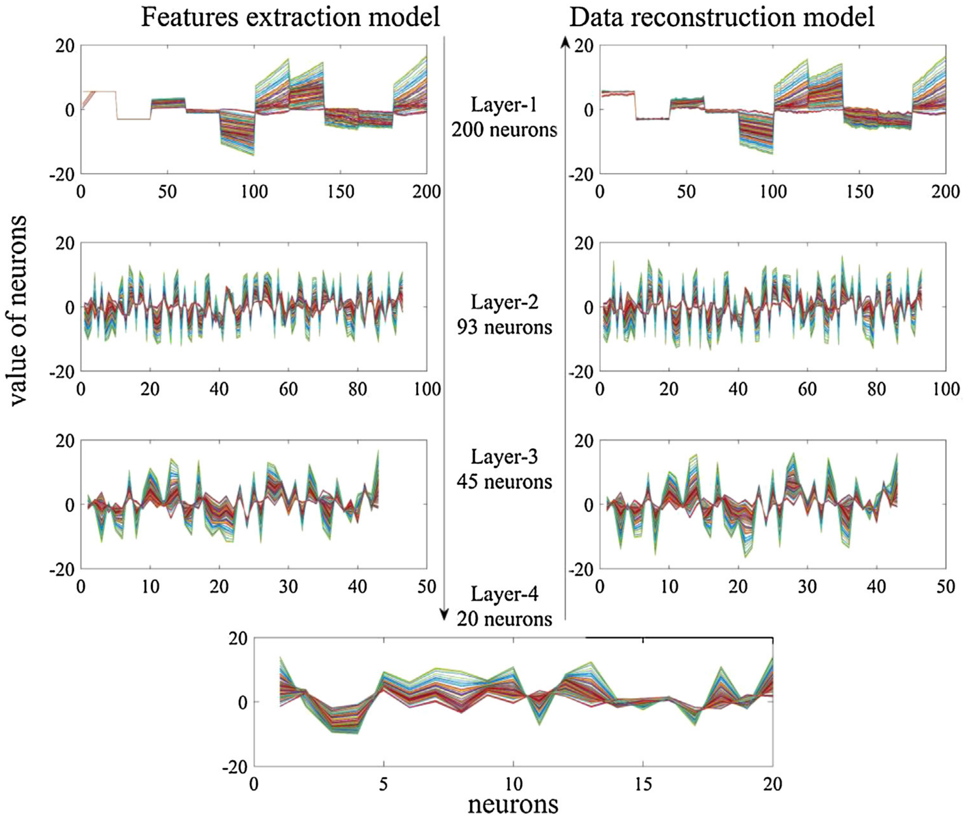


Fig. 8. The unsupervised learning test result of Fault06 sample samples set. The different color line represents each one of 100 samples.

Table 4
The diagnosis results of sub-network using Sigmoid function.

Status	Count of variables	Time length (minutes)	FDR (%)		FPR (%)	
			Training	Testing	Training	Testing
Fault01	5	30	100	100	6.07	6.15
Fault02	15	3	98	97	4.69	4.64
Fault03	25	48	100	98	13.51	13.67
Fault04	5	3	100	100	2.43	3
Fault05	10	60	91.50	87	8.39	8.44
Fault06	5	3	100	100	0	0
Fault07	5	3	100	100	1.84	1.64
Fault08	30	39	85	77	14.39	13.77
Fault09	35	24	0	0	11.94	11.87
Fault10	20	24	0	0	10.01	9.67
Fault11	35	39	11.25	12	12.68	12.49
Fault12	30	48	1	1	15.49	15.59
Fault13	15	39	63	60	4.17	3.95
Fault14	40	24	3.5	5	9.01	8.74
Fault15	15	30	0	0	9.51	9.59
Fault16	5	6	0	0	2.88	2.92
Fault17	5	30	100	100	3.83	3.69
Fault18	5	60	100	100	4.73	4.62
Fault19	5	39	15.75	13	6.60	6.69
Fault20	25	24	98	93	13.46	13.59
Normal	40	39	95.05	93.2	41.83	44

Table 5
The diagnosis results of sub-network using Gaussian function.

Status	Count of variables	Time length (minutes)	FDR (%)		FPR (%)	
			Training	Testing	Training	Testing
Fault01	5	3	98	98	8.03	7.95
Fault02	10	6	96.25	95	9.06	8.79
Fault03	10	30	99.75	100	28.71	28.92
Fault04	5	6	100	100	7.571	7.72
Fault05	10	48	85.75	79	13.29	13.26
Fault06	5	3	100	100	3.47	3.44
Fault07	5	3	100	100	5.46	5.69
Fault08	15	12	88	89	13.75	13.9
Fault09	30	39	76.25	66	32.41	31.95
Fault10	5	60	96.75	98	16.69	16.59
Fault11	35	39	98.25	91	9.37	9.79
Fault12	5	30	73.25	72	14.53	14.77
Fault13	30	30	95.50	91	13.62	13.67
Fault14	5	30	95.75	91	4.55	4.41
Fault15	20	30	0	0	0	0
Fault16	20	30	0	0	0	0
Fault17	10	39	100	100	6.64	6.87
Fault18	45	48	81.25	78	8.90	8.92
Fault19	25	24	98.50	98	9.60	9.61
Fault20	20	39	96.75	93	9.67	9.95
Normal	30	39	81.35	80.70	3.69	3.55

two sub-networks that based on same feature extraction model but different activation function classification layer.

$$\mathrm{sigmoid}\,(x) = \frac{1}{1+e^{-x}} \tag{18}$$

$$\mathrm{Gaussian}\,(x) = 1 - e^{-x^2} \tag{19}$$

4.4. Variable and time length selection

In Section 4.2, variables are sorted by mutual information for each fault and normal state. To select the valuable variables for better performance of the sub-network, the variables are added one by one to the input of each sub-network until a satisfactory fault diagnosis rate is achieved. For diverse faults the time length of input data also needs to be optimized. Since the sampling period in the TE process is 3 min in this paper, the time length of input data can be a whole number times of 3 min. Here in this paper, the maximum time length is limited to one hour. For each different count of the selected variables and each time length, the sub-networks are trained to get the FDR and ACR. The FDR contour maps Figs. 11 and 12 show that the FDR generally increases with the time length augmented and the count of valuable variables for some of the 20 faults is around 20–30 while the count of valuable variables for some of the 20 faults is only about 5. If an improper number of variables are selected as the input of the sub-networks, noisy information may be brought in so that the FDR may decrease. This could be observed in the FDR map of the Fault08. The suitable variables and time length are selected based on better ACR with shorter time length (see Tables 4 and 5).

4.5. Diagnosis results of the fault diagnosis model with OCON architecture

With reasonable variables and time length, the sub-network of each fault or normal state diagnoses all samples. The output of state sub-network is between 0 and 1 that represents the likelihood of

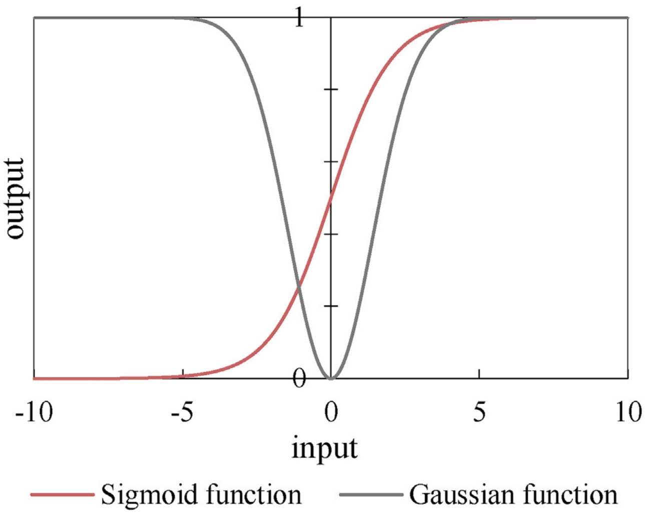


Fig. 9. The Sigmoid function and Gaussian function that used in classification layer.

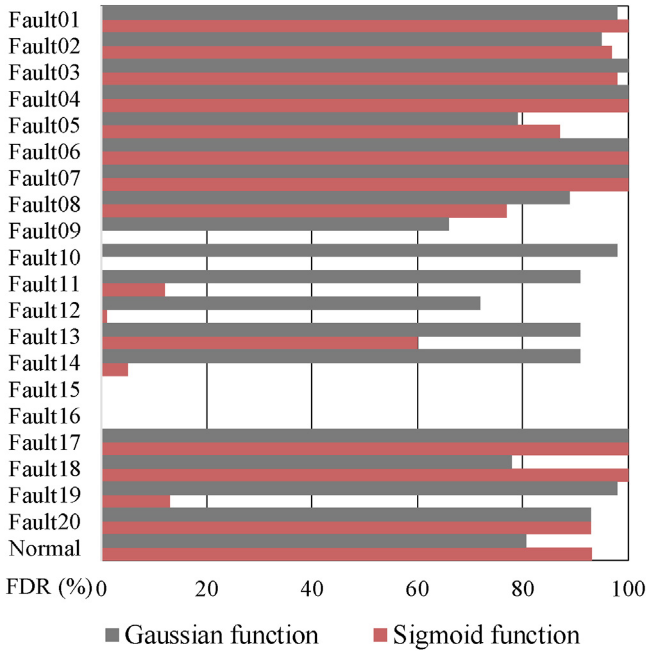


Fig. 10. The FDR of each sub-network with Sigmoid and Gaussian activation function.

sample's belonging to this state. If the output value exceeds 0.5, the sample belongs to this state. All the outputs of the 42 sub-networks are taken as the input of the global network trained with the BP algorithm. If the output of the global network shows that the sample does not belong to any fault or normal state, this sample is classified to New-fault.

As for fault detection, among the 2000 testing fault samples, the testing faults detection rate is 88.05%, that means 11.95% of the testing fault samples are missed. What should bu noted is that the false detection rate is as low as 1.3% for the 2000 normal samples. As for fault diagnosis, the fault diagnosis results of the global network can be found in Table 6 and Fig. 13. It is found out from Tables 4–6 that the FPRs of global network decrease significantly compared with those of the DBN sub-networks.

The area of each blue dot represents the correct diagnosis percentage of the corresponding fault while the area of each red dot represents the percentage of false classification of the corresponding fault into other faults, the normal state or new fault.

Table 7
Performance comparison of different faults diagnosis methods. (a) basic PCA (Yin et al., 2012); (b) optimized variable selection based PCA (Ghosh et al., 2014); (c) supervised local multilayer perceptron (Rad and Yazdanpanah, 2015); (d) Bayesian method (Jiang and Huang, 2016); (e) DBN based model (proposed in this paper).

FDR(%)	(a)	(b)	(c)	(d)	(e)
Fault01	100	99.87	96.37	100	100
Fault02	99.38	97.87	97.62	99	99
Fault03	10.25	2.37	20.62	6	95
Fault04	100	100	82.75	100	98
Fault05	34.75	99.87	96	100	86
Fault06	100	99.5	100	100	100
Fault07	100	100	100	100	100
Fault08	98.63	96.62	96.87	99	78
Fault09	9.88	3.37	12.12	3	57
Fault10	71	82.25	88.25	84	98
Fault11	83	64.75	73.5	82	87
Fault12	99	99	93.62	100	85
Fault13	95.75	95	72.25	95	88
Fault14	100	100	95.87	100	87
Fault15	17.25	9.75	21.12	17	0
Fault16	65.75	81.62	78.12	89	0
Fault17	96.88	84.87	80.25	96	100
Fault18	91.13	89.5	86.37	90	98
Fault19	47.38	76.12	96.12	52	93
Fault20	71.50	66.37	86.75	88	93
Average	74.58	77.44	78.73	80	82.1

The fault diagnosis results of all 20 faults are shown in Table 6 and the average testing FDR is 82.1%. Compared with fault diagnosis model in literatures (see Table 7), the performance of our fault diagnosis model is improved significantly, especially for Fault03 and Fault09. Per the improved results, the FDR of Fault03 reaches 95% and the FDR of Fault09 reaches 57%. In fact, diagnosing those two faults is challenging because their tag deviations from the normal state are much slighter than other faults. Although the FPRs of the sub-networks are unfavorably high, the global network can satisfactorily reduce them. It needs to be noted that, at the present, DBN is not a competitive method for diagnosing Fault15 and Fault16. The reason needs further investigation.

5. Conclusions and outlooks

To extract the fault features from continuous valued process data in both spatial and temporal domains, an improved DBN with the OCON architecture is proposed in this paper. Two types of activation functions, Gaussian function and Sigmoid function, are studied and compared on their classification performances. Final results show that, DBN sub-networks with the Gaussian activation function showed better performance than those with the Sigmoid function in diagnosis of random type faults. With the outputs of the DBN sub-networks of process faults as the inputs, a three layer BP neural network is constructed as the global classification network to generate a comprehensive fault diagnosis result based on the fault diagnosis results from each individual DBN.

The application of the improved DBN based fault diagnosis model on TE process shows outstanding performance, where the average fault diagnosis rate of all 20 faults reaches 82.1%, which is favorably comparable to the average fault diagnosis rates reported in other papers with other fault diagnosis methods. Within the 20 widely studied faults, fault 3 has been known as one of the most difficult faults to diagnose because its fault diagnosis rate reported in former literatures is less than 40%. While in this paper, with the proposed DBN based fault diagnosis model, the diagnosis rate of fault 3 can be achieved to 95%.

Table 6
The diagnosis results of the global network.

Status	FDR (%)		FPR (%)	
	Training (400 samples)	Testing (100 samples)	Training (15600 samples)	Testing (3900 samples)
Fault01	100	100	0	0.077
Fault02	100	99	0	0.077
Fault03	99	95	0.397	0.487
Fault04	98	98	0.064	0.103
Fault05	90	86	0.237	0.308
Fault06	100	100	0	0
Fault07	100	100	0	0
Fault08	96	78	0.019	0.205
Fault09	65.5	57	0.122	0.615
Fault10	97.5	98	0.083	0.128
Fault11	97.5	87	0.128	0.256
Fault12	85.5	85	0.147	0.333
Fault13	96.5	88	0.013	0.179
Fault14	96	87	0	0.128
Fault15	0	0	0	0.051
Fault16	0	0	0	0
Fault17	100	100	0	0
Fault18	100	98	0	0.026
Fault19	97	93	0.006	0.103
Fault20	98.75	93	0.019	0.026

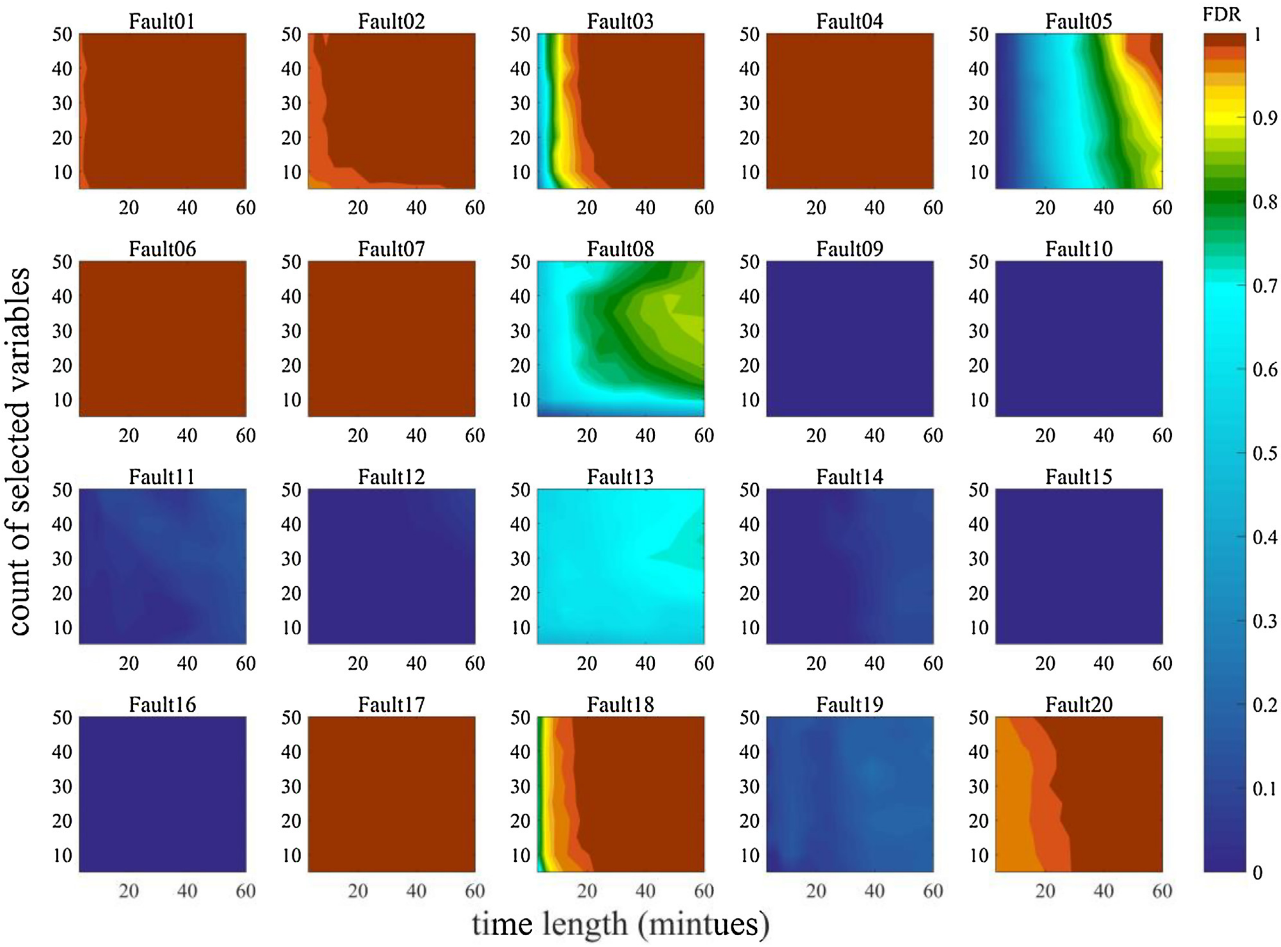


Fig. 11. FDR maps of DBN sub-networks using Sigmoid function.

The data driven FDD methods depend on the collection of a large amount of various process malfunction samples. Inevitably, our DBN based fault diagnosis model suffers from the same drawback. In the near future, research will be focused on fault diagnosis with limited number of fault samples available.

Acknowledgement

The authors gratefully acknowledge support from the National Natural Science Foundation of China (No. 61433001).

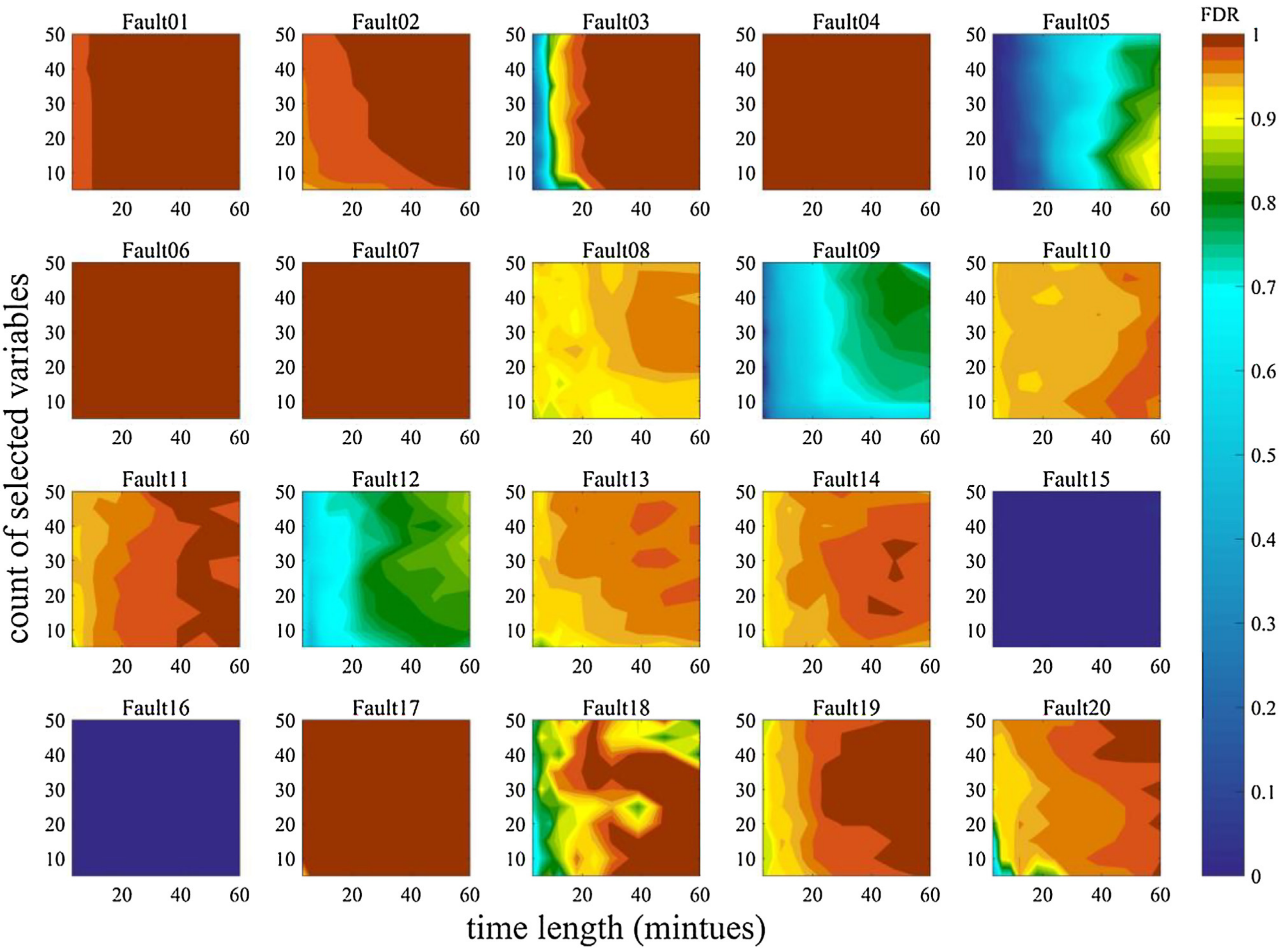


Fig. 12. FDR maps of DBN sub-networks using Gaussian function.

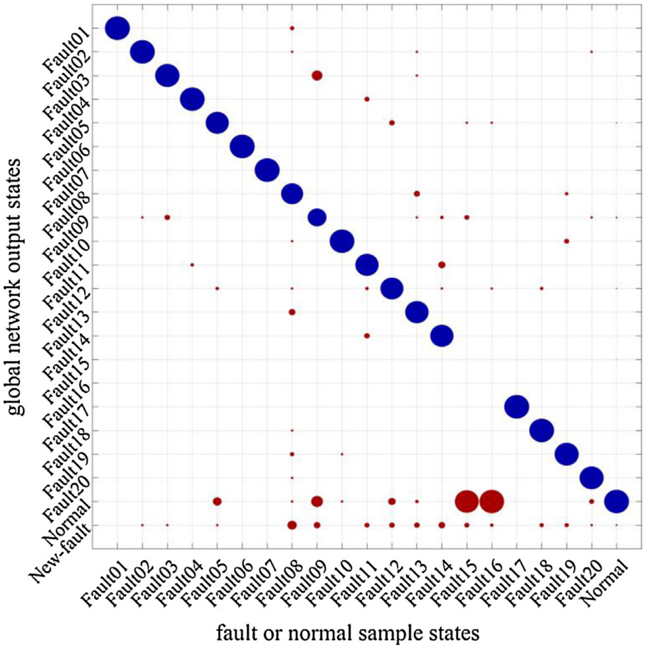


Fig. 13. The faults diagnosis testing results of faults and normal states.

Appendix A. about training Gaussian units RBM

This Appendix presents the demonstration of Eqs. (6) and (13) which is about adjusting the weights and biases of Gaussian units RBM and monitoring the free energy in training process to avoid over fitting.

On the basis of Eq. (1), we use the logarithm probability of the data $\log P(\mathbf{v}=\mathbf{v}_{\text{data}})$ and the derivative of the logarithm probability to weight w_{ij} is:

$$\frac{\partial \log P(\mathbf{v}=\mathbf{v}_{\text{data}})}{\partial w_{ij}} = \frac{\partial \log \int_{\mathbf{h}} p(\mathbf{v}_{\text{data}},\mathbf{h})}{\partial w_{ij}} \tag{A.1a}$$

$$= \frac{\partial \left[\log \int_{\mathbf{h}} e^{-E(\mathbf{v}_{\text{data}},\mathbf{h})} - \log \iint_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}\right]}{\partial w_{ij}} \tag{A.1b}$$

$$= \int_{\mathbf{h}} \frac{e^{-E(\mathbf{v}_{\text{data}},\mathbf{h})}}{\int_{\mathbf{h}} e^{-E(\mathbf{v}_{\text{data}},\mathbf{h})}} \frac{\partial - E(\mathbf{v}_{\text{data}},\mathbf{h})}{\partial w_{ij}} - \iint_{\mathbf{v},\mathbf{h}} \frac{e^{-E(\mathbf{v},\mathbf{h})}}{\iint_{\mathbf{v},\mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}} \frac{\partial - E(\mathbf{v},\mathbf{h})}{\partial w_{ij}} \tag{A.1c}$$

$$= \int_{\mathbf{h}} P(\mathbf{h}|\mathbf{v}=\mathbf{v}_{\text{data}}) \frac{\partial - E(\mathbf{v}_{\text{data}},\mathbf{h})}{\partial w_{ij}} - \iint_{\mathbf{v},\mathbf{h}} p(\mathbf{v},\mathbf{h}) \frac{\partial - E(\mathbf{v},\mathbf{h})}{\partial w_{ij}} \tag{A.1d}$$

Via Eq. (3):

$$\frac{\partial - E(\mathbf{v},\mathbf{h})}{\partial w_{ij}} = \frac{v_i}{\sigma_i}\frac{h_j}{\sigma_j} \tag{A.2}$$

so we get:

$$\frac{\partial \log P(\mathbf{v}=\mathbf{v}_{\text{data}})}{\partial w_{ij}} = \frac{v_{i\text{data}}}{\sigma_i}\int_{\mathbf{h}} P(\mathbf{h}|\mathbf{v}=\mathbf{v}_{\text{data}})\frac{h_j}{\sigma_j} - \iint_{\mathbf{v},\mathbf{h}} p(\mathbf{v},\mathbf{h})\frac{v_i}{\sigma_i}\frac{h_j}{\sigma_j} \tag{A.3a}$$

$$= < \frac{v_i}{\sigma_i}\frac{h_j}{\sigma_j} >_{\text{data}} - < \frac{v_i}{\sigma_i}\frac{h_j}{\sigma_j} >_{\text{model}} \tag{A.3b}$$

The < > represents the expected value of joint configuration $(\mathbf{v},\mathbf{h})$. Based on the weight connections between visible unit and hidden unit, the Markov Chain of $\mathbf{v}$ and $\mathbf{h}$ status is constructed by $\mathbf{h}^k = f\left(\mathbf{v}^k\right)$ and $\mathbf{v}^{k+1} = g\left(\mathbf{h}^k\right)$. When the $k=\infty$, the status of $(\mathbf{v},\mathbf{h})$ will be constant, so the subscript “data” represents expected value of $(\mathbf{v},\mathbf{h})$ when $k=0$, and “model” as $k=\infty$.

The connection function between visible units and hidden units is deduced by Eqs. (1) and (3). Because the hidden units are independent of each other, we can consider the h_j and $\mathbf{v}$ as a subsystem.

$$p\left(h_j|\mathbf{v}\right) = \frac{p\left(\mathbf{v},h_j\right)}{\int_{-\infty}^{+\infty} p\left(\mathbf{v},h_j\right)dh_j} \tag{A.4a}$$

$$= e^{-\frac{1}{2\sigma_j^2}\left[h_j-\left(b_j+\sigma_j\sum\limits_{i\in\text{visible}}\frac{v_i}{\sigma_i}w_{ij}\right)\right]^2} \Bigg/ \int_{-\infty}^{+\infty} e^{-\frac{1}{2\sigma_j^2}\left[h_j-\left(b_j+\sigma_j\sum\limits_{i\in\text{visible}}\frac{v_i}{\sigma_i}w_{ij}\right)\right]^2} dh_j \tag{A.4b}$$

h_j conform to a Gaussian distribution with the mean of $b_j + \sigma_j \sum\limits_{i\in\text{visible}}\frac{v_i}{\sigma_i}w_{ij}$ and standard deviation of σ_j, and we regard the expected value of h_j as the value of h_j. So we get:

$$h_j = b_j + \sigma_j\sum_{i\in\text{visible}}\frac{v_i}{\sigma_i}w_{ij} = \text{line}(\mathbf{v}) \tag{A.5}$$

$$v_i = a_i + \sigma_i\sum_{j\in\text{hidden}}\frac{h_j}{\sigma_j}w_{ij} = \text{line}(\mathbf{h}) \tag{A.6}$$

$$< \frac{v_i}{\sigma_i}\frac{h_j}{\sigma_j} >= \frac{v_i}{\sigma_i}\frac{h_j}{\sigma_j} \tag{A.7}$$

The Contrastive Divergence is introduced to solve the calculation of $< \frac{v_i}{\sigma_i}\frac{h_j}{\sigma_j} >_{\text{model}}$, and it works well enough to achieve success in many significant applications (Hinton, 2002). In the Markov Chain (Fig. A1) when $k=0$, $\mathbf{v}^{k=0}$ is the input data, and $\mathbf{h}^{k=0}$ calculated via line $\left(\mathbf{v}^{k=0}\right)$, we mark this status as subscript “data”. When $k=0$, $\mathbf{v}^{k=1}$ is reconstructed by line $\left(\mathbf{h}^{k=0}\right)$, so we mark this status as subscript “recon”. Then we use $< \frac{v_i}{\sigma_i}\frac{h_j}{\sigma_j} >_{\text{data}} - < \frac{v_i}{\sigma_i}\frac{h_j}{\sigma_j} >_{\text{recon}}$ instead of $< \frac{v_i}{\sigma_i}\frac{h_j}{\sigma_j} >_{\text{data}} - < \frac{v_i}{\sigma_i}\frac{h_j}{\sigma_j} >_{\text{model}}$. In practical applications, the data are normalized to make sure that standard deviations σ_i is equal to 1, so σ_j is supposed to be 1 too. In training process σ_i is calculated from data and σ_j is calculated as Eq. (A.8):

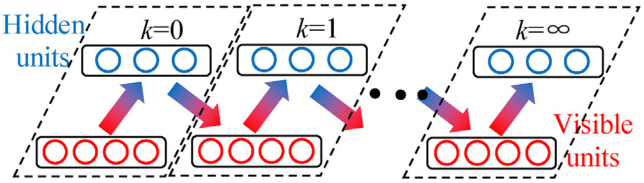


Fig. A1. The Markov Chain of visible units and hidden units.

$$\boldsymbol{\sigma} = \text{diag}\left(\mathbf{W}'\text{cov}(\mathbf{V})\mathbf{W}\right)^{1/2} \tag{A.8}$$

Finally we get:

$$\Delta w_{ij} = \frac{v_i}{\sigma_i}\frac{h_j}{\sigma_j}_{\text{data}} - \frac{v_i}{\sigma_i}\frac{h_j}{\sigma_j}_{\text{recon}} \tag{A.9}$$

$$\Delta a_i = \frac{v_i}{\sigma_i}_{\text{data}} - \frac{v_i}{\sigma_i}_{\text{recon}} \tag{A.10}$$

$$\Delta b_j = \frac{h_j}{\sigma_j}_{\text{data}} - \frac{h_j}{\sigma_j}_{\text{recon}} \tag{A.11}$$

The free energy, $F(\mathbf{v}_{\text{data}})$ is defined based on Eqs. (3) and (12):

$$e^{-F(\mathbf{v}_{\text{data}})} = \int_{\mathbf{h}} e^{-E(\mathbf{v}_{\text{data}},\mathbf{h})} \tag{A.12a}$$

$$= \int_{\mathbf{h}} e^{-\sum\limits_{i\in\text{visible}}\frac{\left(a_i-v_i\right)^2}{2\sigma_i^2} - \sum\limits_{j\in\text{hidden}}\frac{\left(b_j-h_j\right)^2}{2\sigma_j^2} + \sum\limits_{i,j}\frac{v_i}{\sigma_i}\frac{h_j}{\sigma_j}w_{ij}} \tag{A.12b}$$

$$= e^{-\sum\limits_{i\in\text{visible}}\frac{\left(a_i-v_i\right)^2}{2\sigma_i^2} - \sum\limits_{j\in\text{hidden}}\frac{b_j{}^2-\left(b_j+\sigma_j\sum_{i\in\text{visible}}\frac{v_i}{\sigma_i}w_{ij}\right)^2}{2\sigma_j^2}} \int_{\mathbf{h}} e^{-\sum\limits_{j\in\text{hidden}}\frac{\left[h_j-\left(b_j+\sigma_j\sum_{i\in\text{visible}}\frac{v_i}{\sigma_i}w_{ij}\right)\right]^2}{2\sigma_j^2}} \tag{A.12c}$$

$$= e^{-\sum\limits_{i\in\text{visible}}\frac{\left(a_i-v_i\right)^2}{2\sigma_i^2} - \sum\limits_{j\in\text{hidden}}\frac{b_j{}^2-\left(b_j+\sigma_j\sum_{i\in\text{visible}}\frac{v_i}{\sigma_i}w_{ij}\right)^2}{2\sigma_j^2}} \prod_{\mathbf{h}}\sqrt{2\pi}\sigma_j \tag{A.12d}$$

Then we get the $F(\mathbf{v}_{\text{data}})$:

$$F(\mathbf{v}_{\text{data}}) = \sum_{i\in\text{visible}}\frac{\left(a_i-v_i\right)^2}{2\sigma_i^2} + \sum_{j\in\text{hidden}}\frac{b_j{}^2-\left(b_j+\sigma_j\sum_{i\in\text{visible}}\frac{v_i}{\sigma_i}w_{ij}\right)^2}{2\sigma_j^2} - \log\left(\sqrt{2\pi}\sigma_j\right) \tag{A.13}$$

References

Bathelt, A., Ricker, N.L., Jelali, M., 2015. Revision of the tennessee eastman process model. IFAC (International Federation of Automatic Control) Papers Online 48 (8), 309–314.

Bengio, Y., Lamnlin, P., Popovici, D., Larochelle, H., 2007. Greedy layer-wise training of deep networks. Advances in Neural Information Processing Systems 19 (NIPS'06), 153–160.

Chen, X.Y., Yan, X.F., 2013. Fault diagnosis in chemical process based on self-organizing map integrated with fisher discriminant analysis. Chin. J. Chem. Eng. 21 (4), 382–387.

Chiang, L., Russell, E., Braatz, R., 2000. Fault diagnosis and fisher discriminant analysis discriminant partial least squares, and principal component analysis. Chemom. Intell. Lab. Syst. 50, 243–252.

Cho, J.H., Lee, J.M., Choi, S.W., Lee, D., Lee, I.B., 2005. Fault identification for process monitoring using kernel principal component analysis. Chem. Eng. Sci. 60, 279–288.

Dai, Y.Y., Zhao, J.S., 2011. Fault diagnosis of batch chemical processes using a dynamic time warping (DTW)-based artificial immune system. Ind. Eng. Chem Res. 50 (8), 4534–4544.

Detroja, K.P., Gudi, R.D., Patwardhan, S.C., 2007. Plant-wide detection and diagnosis using correspondence analysis. Control Eng. Pract. 15 (12), 1468–1483.

Ding, S.X., Zhang, P., Naik, A., Ding, E., Huang, B., 2009. Subspace method aided data-driven design of fault detection and isolation systems. J. Process Control 19, 1496–1510.

Downs, J.J., Vogel, E.F., 1993. A plant-wide industrial process control problem. Comput. Chem. Eng. 17, 245–255.

Eslamloueyan, R., 2011. Designing a hierarchical neural network based on fuzzy clustering for fault diagnosis of the Tennessee-Eastman process. Appl. Soft Comput. 11, 1407–1415.

Fan, J.C., Wang, Y.Q., 2014. Fault detection and diagnosis of non-linear non-Gaussian dynamic processes using kernel dynamic independent component analysis. Inf. Sci. 259, 369–379.

Gao, X.Y., Shang, C., Jiang, Y.H., Huang, D.X., Chen, T., 2014. Refinery scheduling with varying crude: a deep belief network classification and multimodel approach. AIChE J. 60, 2525–2532.

Ge, Z.Q., Song, Z.H., 2007. Process monitoring based on independent component analysis-principal component analysis (ICA-PCA) and similarity factors. Ind. Eng. Chem. Res. 46, 2054–2063.

Ge, Z.Q., Yang, C.J., Song, Z.H., 2009. Improved kernel PCA-based monitoring approach for nonlinear processes. Chem. Eng. Sci. 64, 2245–2255.

Ge, Z.Q., Song, Z.H., Gao, F.R., 2013. Review of recent research on data-based process monitoring. Ind. Eng. Chem. Res. 52, 3543–3562.

Ghosh, K., Srinivasan, R., 2011. Immune-system-inspired approach to process monitoring and fault diagnosis. Ind. Eng. Chem. 50, 1637–1651.

Ghosh, K., Ramteke, M., Srinivasan, R., 2014. Optimal variable selection for effective statistical process monitoring. Comput. Chem. Eng. 60, 260–276.

He, Q.P., Wang, J., 2007. Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes. IEEE Trans. Semicond. Manuf. 20 (4), 345–354.

Hinton, G.E., Salakhutdinov, R.R., 2006a. Reducing the dimensionality of data with neural networks. Science 313, 504–507.

Hinton, G.E., Dayan, P., Frey, B.J., Neal, R., 1995. The wake-sleep algorithm for self-organizing neural networks. Science 268, 1158–1161.

Hinton, G.E., Osindero, S., Teh, Y., 2006b. A fast learning algorithm for deep belief nets. Neural Comput. 18 (7), 1527–1554.

Hinton, G.E., 2002. Training products of experts by minimizing contrastive divergence. Neural Comput. 14 (8), 1711–1800.

Hinton, G.E., 2010. A Practical Guide to Training Restricted Boltzmann Machines (Accessed 12 October 2015) Available at: http://www.cs.toronto.edu/~hinton/absps/guideTR.pdf.

Hopfield, J.J., 1982. Neural networks and physical systems with emergent collective computational abilities. Proc. Natl. Acad. Sci. 79, 2554–2558.

Huang, B., 2008. Bayesian methods for control loop monitoring and diagnosis. J. Process Control 18, 829–838.

Jiang, Q.C., Huang, B., 2016. Distributed monitoring for large-scale processes based on multivariate statistical analysis and Bayesian method. J. Process Control 46, 75–83.

Jiang, Q.C., Huang, B., Yan, X.F., 2016. GMM and optimal principal components-based Bayesian method for multimode fault diagnosis. Comput. Chem. Eng. 84, 338–349.

Kano, M., Tanaka, S., Hasebe, S., Hashimoto, I., Ohno, H., 2003. Monitoring independent components for fault detection. AIChE J. 49, 969–976.

Kresta, J., MacGregor, J., Marlin, T., 1991. Multivariate statistical monitoring of process operating performance. Can. J. Chem. Eng. 69, 35–47.

Kulkarni, A., Jayaraman, V.K., Kulkarni, B.D., 2005. Knowledge incorporated support vector machines to detect faults in Tennessee Eastman Process. Comput. Chem. Eng. 29, 2128–2133.

Lau, C.K., Ghosh, K., Hussain, M.A., Che Hassan, C.R., 2013. Fault diagnosis of Tennessee Eastman process with multi-scale PCA and ANFIS. Chemom. Intell. Lab. Syst. 120, 1–14.

LeCun, Y., Bengio, Y., Hinton, G.E., 2015. Deep learning. Nature 521, 436–444.

Lee, Jong-Min, Joe Qin, S., Lee, In-Beum, 2007. Fault detection of non-linear processes using kernel independent component analysis. Can. J. Chem. Eng. 85, 526–536.

Luo, L., Su, H.Y., Ban, L., 2014. Independent component analysis based sparse auto encoder in the application of fault diagnosis. In: the 11th World Congress on Intelligent Control and Automation, Shenyang, China.

MacGregor, J., Cinar, A., 2012. Monitoring, fault diagnosis, fault-tolerant control and optimization: data driven methods. Comput. Chem. Eng. 47, 111–120.

Mahadevan, S., Shah, S.L., 2009. Fault detection and diagnosis in process data using one-class support vector machines. J. Process Control 19, 1627–1639.

Maurya, M.R., Rengaswamy, R., Venkatasubramanian, V., 2007. Fault diagnosis using dynamic trend analysis: A review and recent developments. Eng. Appl. Artif. Intell. 20, 133–146.

Ng, Y.S., Srinivasan, R., 2008a. Multivariate temporal data analysis using self-organizing maps part I: visual exploration of multi-state operations. Ind. Eng. Chem. Res. 47 (20), 7744–7757.

Ng, Y.S., Srinivasan, R., 2008b. Multivariate temporal data analysis using self-organizing maps part II: monitoring and diagnosis of multi-state operations. Ind. Eng. Chem. Res. 47 (20), 7758–7771.

Piovoso, M.J., Kosanovich, K.A., 1994. Applications of multivariate statistical methods to process monitoring and controller design. Int. J. Control 59, 743–765.

Rad, M.A.A., Yazdanpanah, M.J., 2015. Designing supervised local neural network classifiers based on EM clustering for fault diagnosis of Tennessee Eastman process. Chemom. Intell. Lab. Syst. 146, 149–157.

Rato, T., Schmitt, E., Ketelaere, B.D., Hubert, M., Reis, M., 2016. A systematic comparison of PCA-based statistical process monitoring methods for high-dimensional, time-dependent processes. AIChE J. 62 (5), 1478–1493.

Russell, L., Chiang, L., Braatz, R., 2000. Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. Chemom. Intell. Lab. Syst. 51, 81–93.

Shu, Y.D., Zhao, J.S., 2015. Dynamic artificial immune system with variable selection based on causal inference. In: 12th International Symposium on Process Systems Engineering and 25th European Symposium on Computer Aided Process Engineering, Copenhagen, Denmark.

Shu, Y.D., Ming, L., Cheng, F.F., Zhang, Z.P., Zhao, J.S., 2016. Abnormal situation management: challenges and opportunities in the big data era. Comput. Chem. Eng. 91, 104–113.

Smolensky, P., Rumelhart, D.E., McClelland, J.L., 1986. Information processing in dynamical systems: foundations of harmony theory. Parall. Distrib. Process. 1, 194–281.

Srinivasan, R., Wang, C., Ho, W.K., Lim, K.W., 2005. Neural network systems for multi-dimensional temporal pattern classification. Comput. Chem. Eng. 29, 965–981.

Sun, J.W., Wyss, R., Steinecker, A., Glocker, P., 2014. Automated fault detection using deep belief networks for the quality inspection of electromotor. TM Tech. Mess. 81 (5), 255–263.

Tamilselvan, P., Wang, P.F., 2013. Failure diagnosis using deep belief learning based health state classification. Reliab. Eng. Syst. Saf. 115, 124–135.

Tran, V.T., AlThobiani, F., Ball, A., 2014. An approach to fault diagnosis of reciprocating compressor valves using Teager–Kaiser energy operator and deep belief networks. Expert Syst. Appl. 41, 4113–4122.

Venkatasubramanian, V., Chan, K., 1989. A neural network methodology for process fault diagnosis. AIChE J. 35, 1993–2002.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., Kavuri, S.N., 2003a. A review of process fault detection and diagnosis part I: quantitative model-based methods. Comput. Chem. Eng. 27, 293–311.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., 2003b. A review of process fault detection and diagnosis: part II: qualitative models and search strategies. Comput. Chem. Eng. 27, 313–326.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S.N., Yin, K., 2003c. A review of process fault detection and diagnosis part III: process history based methods. Comput. Chem. Eng. 27, 327–346.

Verron, S., Tiplica, T., Kobi, A., 2008. Fault detection and identification with a new feature selection based on mutual information. J. Process Control 18, 479–490.

Yin, S., Ding, S.X., Haghani, A., Hao, H.Y., Zhang, P., 2012. A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process. J. Process Control 22, 1567–1581.

Zhang, Y.W., 2008. Fault detection and diagnosis of nonlinear processes using improved kernel independent component analysis (KICA) and support vector machine (SVM). Ind. Eng. Chem. Res. 47, 6961–6971.