# Neural Network Soft-sensor Modeling of PVC Polymerization Process Based on Data Dimensionality Reduction Strategy

Cheng Xing, Jie-Sheng Wang *, Lei Zhang, Wei Xie

*Abstract*—**For predicting the conversion rate of vinyl chloride monomer (VCM) in the polyvinyl chloride (PVC) production process, a neural network soft-sensor model based on data dimensionality reduction strategies was proposed. In order to solve the problem of complex neural network topology and long training time caused by the excessive input vector dimension, seven kinds of data dimensionality reduction methods, such as principal component analysis (PCA), locality preserving projection (LPP), kernel principal component analysis (KPCA), expectation max principal component analysis (EMPCA), local tangent space alignment (LTSA), T-distributed stochastic neighbor embedding (TSNE) and neighboring preserving embedding (NPE), are used to reduce the dimension of the high-dimensional input data used in the neural network soft-sensor model. Then the radial basis function (RBF) neural network based on the gradient learning, orthogonal least squares and clustering learning methods and the dynamic fuzzy neural network (D-FNN) were utilized to realize the prediction on the VCM conversion rate. Simulation results show that the proposed neural network soft-sensor models based on seven data dimensionality reduction strategies can effectively predict the key economic and technical indicators of PVC polymerization process and meet the real-time control requirements of PVC production process.**

*Index Terms*—**Polymerization process, Soft-sensor, Data dimensionality reduction, RBF neural network, Dynamic fuzzy neural network**

## I. INTRODUCTION

Polyvinyl chloride (PVC) is a polymer of vinyl chloride monomer (VCM) polymerized by free radical polymerization under the action of light and heat. Vinyl chloride is widely used and can be co-polymerized with vinyl acetate, vinylidene chloride, butadiene, acrylonitrile, acrylates and other monomers to form copolymers, which can also be used as a refrigerant [1]. Usually, the polymerization conversion rate of vinyl chloride is controlled between 85% and 90%, and most of the unconverted monomers are recycled to the vinyl chloride gas cabinet for recycling, but finally about 2% of the residual monomers are difficult to remove, and remain in the residue. In the PVC slurry, a part is lost in the post-treatment process, and a large part remains in the finished product. The conversion rate of VCM has a great influence on the quality of PVC resin products. Different conversion rates have certain effects on the molecular weight, thermal stability, porosity, VCM residue, plasticizer absorption rate, processing fluidity of PVC resin [2]. When VCM is recovered, a low conversion rate results in a resin having a higher porosity, and a high porosity conversion resin can be obtained. The absorption rate of the plasticizer also decreases as the conversion rate increases. Due to the limitations of the field conditions and the lack of mature detection devices, the conversion rate of vinyl chloride in the actual production process is difficult to obtain in real time, so it is difficult to achieve direct quality closed-loop control. The soft-sensor model based on neural network can break through the problem that some variables can't be directly measured. Through the statistical analysis of multiple directly-measurable variables, the direct relationship between these variables and the variables to be measured can be determined by neural network soft-sensor models. However, in practical applications, it has encountered the so-called "Curse of Dimensionality" problem [3], which refers to a phenomenon in which the amount of calculation increases exponentially with the increase of the dimension of the data. Dimensional disasters involve many areas, such as data processing, combination optimization , neural networks, and data mining.

In response to this phenomenon, the data dimensionality reduction is the best solution. Data dimensionality reduction seeks a more compact form of data representation by mapping original data into the lower dimensional space. This compact representation facilitates secondary processing of data [4]. After dimension reduction, the complexity of time and space are greatly reduced, which saves the overhead of unnecessary features, removes the noise mixed in the data set, and the simple model has stronger robustness on low-dimensional data. When there are fewer features, the data can be can explained better, so that the extract on knowledge and the visualization of the data is easily realized. According to the different classification methods, the dimensionality reduction methods can be divided into linear

Manuscript received September 29, 2019; revised December 20, 2019. This work was supported by the the Basic Scientific Research Project of Institution of Higher Learning of Liaoning Province (Grant No. 2017FWDF10), and the Project by Liaoning Provincial Natural Science Foundation of China (Grant No. 20180550700).

Cheng Xing is a Ph.D candidate in the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114044, PR China (e-mail: xingcheng0811@163.com).

Jie-Sheng Wang is with the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China; National Financial Security and System Equipment Engineering Research Center, University of Science and Technology Liaoning. (Corresponding author, phone: 86-0412-2538246; fax: 86-0412-2538244; e-mail: wang_jiesheng@126.com).

Lei Zhang is a postgraduate student in the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114044, PR China (e-mail: zl1764066@163.com).

Wei Xie is a postgraduate student in the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China. (e-mail: xiewei@163.com).

and nonlinear dimensionality reduction. The nonlinear dimensionality reduction methods can be divided into kernel function based methods and the eigenvalue based methods [5]. Typical linear dimensionality reduction methods include PCA, ICA, LDA, and LPP. The nonlinear dimensionality reduction methods based on kernel functions have KPCA and EMPCA. The eigenvalue-based nonlinear dimensionality reduction methods include LTSA, TSNE and NPE. PCA method was used to monitor the variables of the important events, adaptively filter scores and residuals and adjust numerical changes [6]. A two-dimensional linear projection dimension reduction method was proposed [7]. LTSA was used to predict the antenna angular frequency range of the antenna [8]. LDA was effectively used to improve the temporary retrieval and approximates inferences [9]. Because there are many factors affecting temperature, IDA dimension reduction method was proposed [10]. The precise prediction of the acetic acid content in the phthalic acid solvent dehydration column was realized based on the PCA and extreme learning machine [11]. KPCA method was adopted to establish the furnace temperature model and the factors affecting the carbon content of the power plant fly ash was analyzed [12]. The improved EMPCA method was used to classify the operating state of the activated sludge process, which can effectively detect outlines and missing data [13]. The TSNE method is proved to be a very suitable technology for several high-dimensional data visualization [14]. SVM is used as the basic classifier of software defects combined with the NPE algorithm, where the local geometry of the data is kept unchanged during the dimension reduction process.

In this paper, a neural network soft-sensor model for PVC production process based on data dimensionality reduction strategies was proposed to predict the VCM conversion rate. The structure of the paper is described as follows. The technique of the PVC polymerization process is described in Section 2. Section 3 introduces the adopted neural network and the related training algorithms. Section 4 illustrates the seven data dimensionality reduction strategies and the related results. The simulation experiments and result analysis are described in Section 5. Finally the conclusion was drawn.

## II. Polyvinyl Chloride (PVC) Polymerization Process and Soft-sensor Model

### *A. Technique Flowchart of PVC Polymerization Process*

The typical PVC polymerizer polymerization process is shown in Fig. 1 [16].

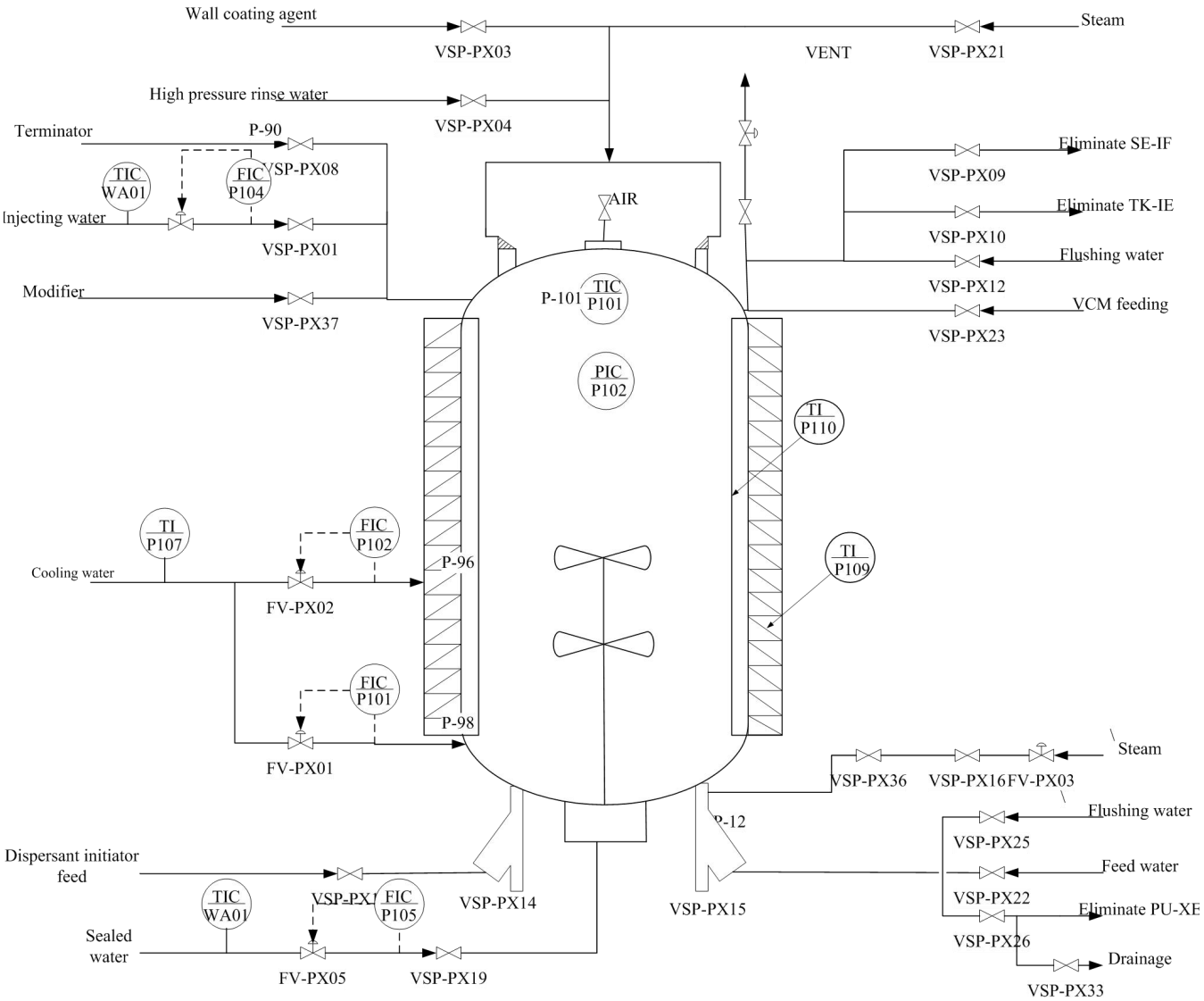


Fig. 1 Technique flowchart of polymerization kettle.

In the polyvinyl chloride (PVC) polymerization process, various raw materials and additives are added to the reaction vessel, which can be fully dispersed uniformly under the action of stirring. These materials form a suspension in the polymerization vessel. Then, an appropriate amount of initiator is added to start the reaction, the polyvinyl chloride monomer is polymerized into PVC particles under the condition of raising temperature and adding an initiator. When the polymerization proceeds to a certain extent, the particles will form a PVC slurry, cooling water is continuously supplied to the jacket and the baffle of the reaction kettle to remove the heat of reaction. When the conversion of vinyl chloride (VCM) reaches a certain level, an appropriate pressure drop occurs, that is to say the reaction is terminated. After the completion of the reaction, the slurry is subjected to stripping to contain VCM, then sent them to a drying process for dehydration and drying [16].

### *B. Structure of Soft-sensor Model*

Ten process variables are selected as the soft-sensor model input, VCM conversion rate is used as the output, seven data reduction strategies are used for the processing on input data, then RBF neural network and D-FNN are used to fit the input and output. The nonlinear relationship is used to establish a soft-sensor model of VCM conversion rate. The structure of soft-sensor model is shown in Fig. 2. According to the characteristics of the polymerization process, 10 process variables related to the VCM conversion rate were identified as auxiliary variables for soft-sensor modeling. The variable names and dimensions are shown in Table 1.

## *III.* NEURAL NETWORKS BASED SOFT-SENSOR MODELS

### *A. RBF Neural Network*

Radial basis function (RBF) neural network is a three-layer network with only one hidden layer except the input and output layers. The transfer function in the hidden layer is a Gaussian function with local response, but for other forward-oriented networks, the transfer function is generally global response function [17]. The structure of the general RBF neural network consisting of input layer, hidden layer and output layer is shown in the Fig. 3. In the RBF neural network, the input layer only acts as a transmission signal, and the input layer and the hidden layer can be regarded as a connection with a connection weight of 1.

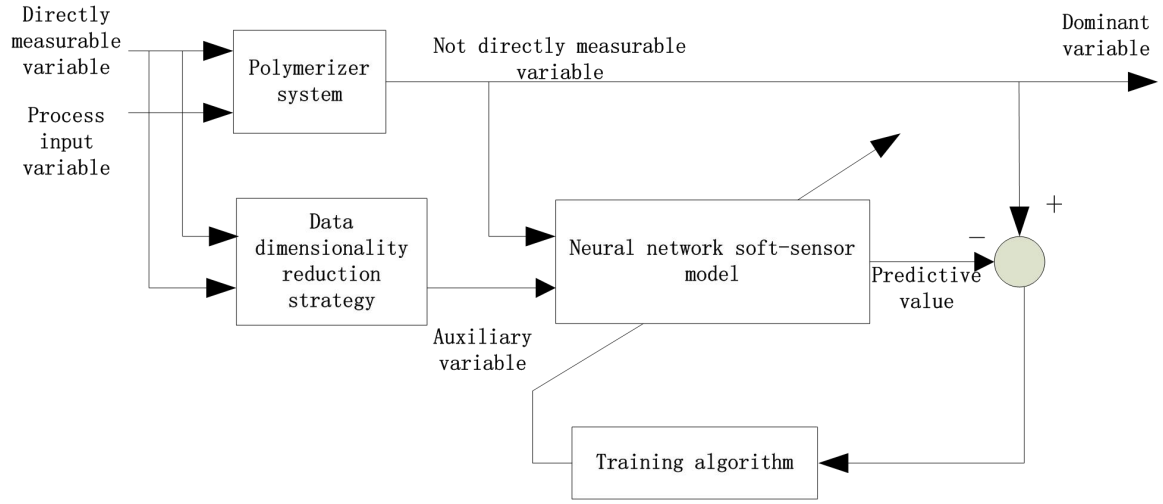


Fig. 2 Structure of the soft-sensor model.

TABLE 1. PROCESS VARIABLES OF THE DISCUSSED POLYMERIZATION

| Process variable | Variable symbol | Unit | Measurement range |
|---|---|---|---|
| Temperature inside kettle | TIC-P101 | ℃ | 0-100 |
| Pressure inside kettle | PIC-P102 | MPa | 0-1.2 |
| Water flow rate of baffle | FIC-P101 | $m^3/h$ | 0-500 |
| Water flow rate of clip set | FIC-P102 | $m^3/h$ | 0-500 |
| Water feed flow rate | FIC-P104 | $m^3/h$ | 0-500 |
| Seal water flow rate | FIC-P105 | $m^3/h$ | 0-500 |
| Water inlet temperature of cooling | TI-P107 | ℃ | 0-100 |
| Water outlet temperature of clip set | TI-P109 | ℃ | 0-100 |
| Water outlet temperature of damper | TI-P110 | ℃ | 0-100 |
| Outlet temperature of cold water tank | TIC-WA01 | ℃ | 0-100 |

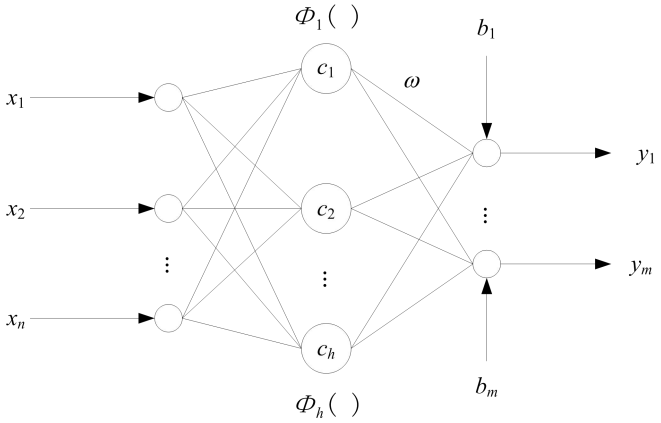

Fig. 3 Structure of RBF neural network.

The output layer and the hidden layer complete different tasks, so their learning strategies are different. The output layer adjusts the linear weight and uses a linear optimization strategy. Therefore, the learning speed is faster. The hidden layer is to adjust the parameters of the activation function by adopting the nonlinear optimization strategy, so the learning speed is slow [18]. The working principle of the RBF neural network can be divided into two parts.

Part I: Nonlinear mapping $x \to h_i(x)$ from the input layer to the hidden layer in the RBF network. The output of the $i$ -th node in the hidden layer can be expressed by Eq. (1).

$$h_i(x) = \varphi\left(\left\|x - c_i\right\|\right), \quad 1 \le i \le h \tag{1}$$

where, $h$ represents the number of hidden layer nodes.

Part II: Linear mapping $h_i(x) \to y_j$ from the hidden layer to the output layer, which is defined in Eq. (2).

$$y_j = \sum_{i=1}^{h} \omega_{ij} h_i(x) + b_j, \quad 1 \le j \le m \tag{2}$$

where, $y_j$ is the output of the $j$ th node of the output layer, $\omega_{ij}$ is the connection weight between the $i$ th hidden layer unit and the $j$ th output layer unit, $h_i(x)$ is the output of the $i$ th node in the hidden layer, and $b_j$ is the output offset.

If the network is viewed as an approximation to an unknown function, any function can be represented as a weighted sum of a set of basis functions. In the RBF neural network, it is equivalent to select the transfer function of each hidden layer neuron, so that it constitutes a set of basis functions to approximate the unknown function [19-20]. The learning algorithms of RBFNN needs to determine three parameters: the center of the basis function, the variance and the weight of the hidden layer to the output layer [21]. According to the different methods of selecting the center of the radial basis function, the RBF neural network has various learning methods, such as gradient learning algorithm, orthogonal least squares algorithm and cluster learning algorithm.

(1) Gradient Learning Algorithm (GLA)

The gradient learning algorithm (GLA) in the RBF neural network is similar to the principle of BP algorithm for training the muti-layer perceptron. It also adjusts the data center, the extended constant and the output weights of each hidden nodes by minimizing the objective function [22]. A single-output RBF neural network learning method with a forgetting factor is described as follows. The objective function is defined as:

$$E = \frac{1}{2}\sum_{j=1}^{N} \beta_j e_j^2 \tag{3}$$

where, $\beta_j$ is a forgetting factor. The error signal $e_j$ can be defined as:

$$e_j = y_j - F(X_j) = y_j - \sum_{i=1}^{h} \omega_i \varphi_i(\left\|X_j - c_i\right\|) \tag{4}$$

When the Gaussian function is selected as the RBF function, the gradients of the function $F(X)$ to $c_i$ , $\sigma_i$ and $\omega_i$ are described as follows.

$$\nabla_{c_i} F(X) = \frac{2\omega_i}{\sigma_i^2} \varphi_i(\left\|X_j - c_i\right\|) \left\|X_j - c_i\right\| \tag{5}$$

$$\nabla_{\sigma_i} F(X) = \frac{2\omega_i}{\sigma_i^3} \varphi_i(\left\|X_j - c_i\right\|) \left\|X_j - c_i\right\|^2 \tag{6}$$

$$\nabla_{\omega_i} F(X) = \varphi_i(\left\|X_j - c_i\right\|) \tag{7}$$

Considering the effects of all training samples and forgetting factors, the adjustments for $c_i$ , $\sigma_i$ and $\omega_i$ are defined in Eq. (8)-(10).

$$\nabla c_i = \frac{2\omega_i}{\sigma_i^2} \sum_{j=1}^{N} \beta_j e_j \varphi_i(\left\|X_j - c_i\right\|) \left\|X_j - c_i\right\| \tag{8}$$

$$\nabla \sigma_i = \frac{2\omega_i}{\sigma_i^3} \sum_{j=1}^{N} \beta_j e_j \varphi_i(\left\|X_j - c_i\right\|) \left\|X_j - c_i\right\|^2 \tag{9}$$

$$\nabla \omega_i = \sum_{j=1}^{N} \beta_j e_j \varphi_i(\left\|X_j - c_i\right\|) \tag{10}$$

where, $\varphi_i(\left\|X_j - c_i\right\|)$ is the output of $i$ th hidden node to $X_j$ .

(2) Orthogonal Least Squares (OLS)

Another method of selecting an RBF neural network data center is the orthogonal least squares (OLS). This method selects the data center from the sample inputs. If all sample inputs are used as data centers, and the extension constants are taken to the same value. According to Micchelli's theorem, the hidden layer output matrix $H \in R^{N \times N}$ is reversible. Thus, the target output $y$ can be linearly represented by $N$ column vectors of $H$ [23]. However, the energy contribution of the $N$ column vectors of $H$ to $y$ is obviously different. According to this feature, $M \le N$ vectors can be selected from the $N$ column vectors in $H$ according to the energy contribution of the vector to form $\hat{H} \in R^{N \times M}$ . Eventually the error need meet the given requirements.

$$\left\|y - \hat{H}\omega_0\right\| < \varepsilon \tag{11}$$

where, $\omega_0$ is the optimal weight vector, which can minimize

the error. The choice of $\hat{H}$ not only affects the effect of the error approximation in Eq. (11), but also influences the performance of the RBF neural network. When $\hat{H}$ is determined, the data center of the RBF neural network is also obtained [24]. A brief introduction on how to calculate the energy contribution is described as follows. Assume that $N$ mutually orthogonal vectors of $H$ (not necessarily unit vectors) $x_1$, $x_2$, …, $x_N$ can linearly represent the target output, that is to say:

$$y = \sum_{i=1}^{N} a_i x_i \tag{12}$$

Multiply $x_i^T$ on the right side of Eq. (12) to obtain:

$$y x_i^T = a_i \|x_i\|^2, \quad i = 1,2,\cdots,N \tag{13}$$

Then:

$$y^T y = \sum_{i=1}^{N} a_i \|x_i\|^2 \tag{14}$$

That is to say:

$$I = \sum_{i=1}^{N} \frac{a_i \|x_i\|^2}{y^T y} \tag{15}$$

Therefore, the total energy contribution of $M$ base vector is:

$$g_A = \sum_{i=1}^{M} g_i = \sum_{i=1}^{M} \frac{a_i \|x_i\|^2}{y^T y} \tag{16}$$

where, $0 \le g_A \le 1$. The larger $M$, the larger $g_A$, which indicates that the approximation effect is better. When all the base vectors are selected, $M = N$, the approximation works best, at this time $g_A = 1$.

The columns of $H$ are not orthogonal, but $x_i$ is orthogonal to each other. By performing Gram-Schmidt orthogonalization on $H$, the orthogonal least squares method implements the selection of the columns of $H$. Selecting a data center by Gram-Schmidt orthogonalization can be achieved by the following steps:

(1) Calculate the hidden node output matrix $H$. If matrix $H$ consists of $P_1^1$, $P_1^2$, …, $P_1^N$, then these vectors form the vector space $E_N^H$ with $N$ dimension.

(2) At this time, the absolute value of $(y^T P_1^k)/(\|y\|\|P_1^k\|)$ is used to represent the contribution of $P_1^k$ to $y$. The output vector $y$ is in correspond to $P_1^k$ to $P_1^1$, $P_1^2$, …, $P_1^N$. If $y$ and $P_1^k$ make the absolute value of $(y^T P_1^k)/(\|y\|\|P_1^k\|)$ the largest, then the first data center contains the sample input corresponding to $P_1^k$, that is to say $P_1^k$ constitutes the one-dimensional Euclidean space $E_1$.

(3) The weights of the network output (including the offset) and the training error of the network to the samples can be obtained by the pseudo-inverse (generalized inverse) method. If the calculated error is less than the expected error, the operation is stopped. Otherwise, Gram-Schmidt orthogonalizes the remaining $N-1$ vectors in the previous operation and makes these vectors orthogonal to $E_1$ so as to obtain $P_2^1$, $P_2^2$, …, $P_2^{N-1}$.

(4) Finding $P_2^j$ that can have the maximum contribution value to the output vector $y$. The input sample corresponding to the vector is placed in the second data center, calculate the output weight and training error in order to determine whether to terminate the algorithm.

(5) Repeat (1) ~ (4) until finding $M$ data centers so that the training error of the network is less than the expected value.

(3) Cluster Learning Algorithm (CLA)

The cluster learning algorithm (CLA) is one of the most classic RBF neural network learning algorithms proposed by Moody and Darken in 1989. Its basic principle is described below.

Suppose $k$ is the number of iterations, the cluster centers at the $k$ th iteration are $c_1(k)$, $c_2(k)$, …, $c_h(k)$, and the corresponding clustering domains are $\theta_1(k)$, $\theta_2(k)$, …, $\theta_h(k)$. The steps of the K-means clustering algorithm to determine $c_i$ and $\sigma_i$ are described as follows.

(1) Initialization. Select $h$ different initial cluster centers, and let $k = 1$. These $h$ initial cluster centers can be randomly selected from the samples, or can be fixedly selected, as long as the guaranteed values are not the same.

(2) Calculate the distance between all sample inputs and the cluster center $\|X_j - c_i(k)\|$, $i = 1,2,\cdots,h$, $j = 1,2,\cdots,N$.

(3) Classify $X_j$ according to the principle of minimum distance. When $i(X_j) = \min_i \|X_j - c_i(k)\|$ ($i = 1,2,\cdots,h$), $X_j$ is classified as class $i$, that is to say $X_j \in \theta_i(k)$.

(4) The initial cluster center may change after classification, so it is necessary to recalculate the new cluster centers by Eq. (17).

$$c_i(k+1) = \frac{1}{N_i} \sum_{X \in \theta_i(k)} X, \quad i = 1,2,\cdots,h \tag{17}$$

where, $N_i$ is the number of samples in the $i$ th cluster domain $\theta_i(k)$.

(5) If $c_i(k+1) \ne c_i(k)$, go to Step (2), otherwise the clustering process ends and go to Step (6).

(6) Finally, the extension constant $\sigma_i$ of the hidden node is determined according to $\|X_j - c_i(k)\|$. Let $\sigma_i = \kappa d_i$ ($\kappa$ a is an overlap coefficient), where $d_i$ is the distance between the $i$ th data center and the other closest data center, that is to say $d_i = \min_i \|c_i - c_i(k)\|$.

After determining $c_i$ and $\sigma_i$, the output weight vector $\omega = [\omega_1, \omega_2, \cdots, \omega_h]^T$ is directly calculated by least squares method or the output weight vector is obtained by using gradient method training. When the input is $X_j$ ($j = 1,2,\cdots,N$), the output of the $i$ th hidden node is:

$$h_{ji} = \varphi_i(\|X_j - c_i\|) \tag{18}$$

Then the output matrix in the hidden layer can be expressed as:

$$\hat{H} = [h_{ji}] \tag{19}$$

That is to say $\hat{H} \in R^{N \times h}$. If the current weight of the RBF

neural network is $\omega=[\omega_1,\omega_2,\cdots,\omega_h]^T$ , the network output vector is represented as:

$$\hat{y}=\hat{H}\omega \tag{20}$$

Approximation error is $\varepsilon=\|y-\hat{y}\|$ , the given signal is $y=[y_1,y_2,\cdots,y_N]^T$ . When $\hat{H}$ is determined, the output weight of the network can be calculated by:

$$\varepsilon=\|y-\hat{y}\|=\|y-\hat{H}\omega\| \tag{21}$$

$$\omega=\hat{H}^{+}y \tag{22}$$

where, $\hat{H}^{+}$ is the pseudo inverse of $\hat{H}$ , that is to say:

$$\hat{H}^{+}=(\hat{H}^T\hat{H})^{-1}\hat{H}^T \tag{23}$$

*B. Dynamic Fuzzy Neural Network (D-FNN)*

The structure of dynamic fuzzy neural network (D-FNN) is based on extended radial basis neural network. It is functionally equivalent to the TSK fuzzy system. Not considering the feedback, the structure of the D-FNN is shown in Fig. 4, whose essence is to represent a fuzzy system based on the TSK model [25]. The dynamic fuzzy neural network has five layers, which are described in detail as follows.

(1) Input layer. $x_1$ , $x_2$ , … , $x_r$ represent the input variables, and $r$ is the number of input variables.

(2) Membership function layer. $MF_{ij}$ represents the $j$ th membership function of the $i$ -th input variable. The membership function can be in many forms. This paper uses the Gaussian function as the membership function of the dynamic fuzzy neural network.

$$\mu_{ij}(x_i)=\exp[-\frac{(x_i-c_{ij})^2}{\sigma_j^2}] \tag{24}$$

where, $\mu_{ij}$ is the $j$ th membership function of $x_i$ , $u$ is the number of membership function, $c_{ij}$ is the center of the $j$ th membership function of $x_i$ , and $\sigma_j$ is the width of the $j$ th membership function of $x_i$ .

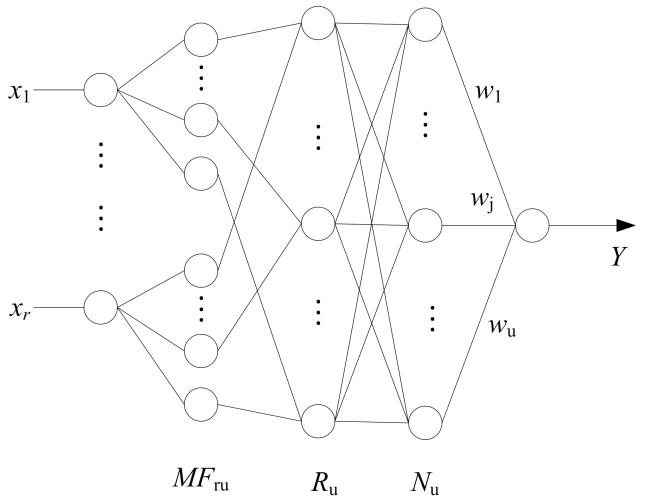


Fig. 4 Structure of dynamic fuzzy neural network.

(3) T-norm layer. $R_j$ represents the $j$ th fuzzy rule. The node in this layer represents the IF part of the fuzzy rule. The output of the $j$ th rule $R_j$ can be expressed as:

$$\varphi_j=\exp[-\frac{\sum_{i=1}^{r}(x_i-c_{ij})^2}{\sigma_j^2}]=\exp[-\frac{\|X-C_j\|^2}{\sigma_j^2}] \tag{25}$$

where, $X=(x_1,x_2,\cdots,x_r)\in\Re^r$ , and $C_j=(c_{1j},c_{2j},\cdots,c_{rj})\in\Re^r$ is the center of the $j$ th RBF neuron.

(4) Normalized layer. $N_j$ represents the $j$ th normalized node. Each node in this layer represents a fuzzy rule. The output of the $j$ th node $N_j$ can be expressed as:

$$\phi_j=\frac{\varphi_j}{\sum_{k=1}^{u}\varphi_k}\quad j=1,2,\cdots,u \tag{26}$$

(5) Output layer. $Y$ is the output of the system. All input signals are superimposed to obtain the output in this layer.

$$y(X)=\sum_{k=1}^{u}w_k\phi_k \tag{27}$$

where, $y$ is output variable, $w_k$ is the consequent parameter or connection weight of the $k$ th rule, and $u$ is the number of total fuzzy rules. For TSK model:

$$w_k=\alpha_0^k+\alpha_1^k x_1+\cdots+\alpha_r^k x_r\ ,\quad k=1,2,\cdots,u \tag{28}$$

When the result parameter is a real number, obtain:

$$w_k=\alpha_k\ ,\quad k=1,2,\cdots,u \tag{29}$$

Substituting Eq. (24)-(27) into Eq. (28) to obtain the following models.

TSK model：

$$y(X)=\frac{\sum_{i=1}^{u}[(\alpha_0^i+\alpha_1^i x_1+\cdots+\alpha_r^i x_r)\exp(-\frac{\|X-C_i\|^2}{\sigma_i^2})]}{\sum_{i=1}^{u}\exp(-\frac{\|X-C_i\|^2}{\sigma_i^2})} \tag{30}$$

S model：

$$y(X)=\frac{\sum_{i=1}^{u}\left[\alpha_0^i\exp(-\frac{\|X-C_i\|^2}{\sigma_i^2})\right]}{\sum_{i=1}^{u}\exp(-\frac{\|X-C_i\|^2}{\sigma_i^2})} \tag{31}$$

By comparing Eq. (30) with Eq. (31), obtain the following principle.

(1) The S model of D-FNN is equivalent to the RBF neural network.

(2) The fuzzy rule number in D-FNN is equivalent to the number of RBF nodes in the RBF neural network.

(3) The structure identification of D-FNN is equivalent to the determination of the number of fuzzy rules.

It can be seen that the D-FNN is indeed developed on the basis of the RBF network. For the TSK model of D-FNN, it

can be considered as an extension of the RBF network [26].

## IV. Data Dimensionality Reduction Methods

For the soft-sensor model, too large input vector dimension will make the network topology bigger and the training more complicated. With the development of computer control technology, the production site records a large amount of production information. On the other hand, because the variables in the chemical process are mutual associated, the collected data may contain a large amount of redundant information [27]. When the dimension of the data is too high, the available data becomes sparse. Due to factors such as the range of values, if the dimension is high, a large number of samples are needed for training, but this is not allowed by objective conditions. Therefore, the dimension reduction methods can not only retain most of the information on the basis of the original data, but also eliminate the redundant information, streamline the model structure and decrease the model calculation time [28]. In this paper, when establishing the soft-sensor models of VCM conversion rate in PVC polymerization process, different dimensionality reduction methods are used to reduce the dimension of complex data. In order to facilitate visualization, the data is reduced to three-dimensional data.

### *A. Principal Component Analysis (PCA)*

Principal component analysis (PCA) is the most commonly used data analysis method, which transforms the original data into a set of linearly independent data by linear transformation so that the low-dimensional data after dimensionality reduction contains most of the information in the original data. In the signal processing, the signal is considered to have a large variance and the noise has a small variance. So the larger the signal-to-noise ratio, the better. The PCA follows the principle of maximum variance between sample points after projection. The specific steps of principal component analysis are described as follows.

(1) Calculate the covariance matrix.

The covariance matrix of the data is $\sum = (s_{ij})_{p\times p}$, where

$$s_{ij} = \frac{1}{n-1}\sum_{k=1}^{n}(x_{ki}-\bar{x}_i)(x_{kj}-\bar{x}_j)\ \ i,j=1,2,3,...,p \qquad (32)$$

(2) Find the eigenvalue $\lambda_i$ and the orthogonal unit eigenvector $a_i$ of $\sum$.

The former $m$ larger eigenvalues of $\sum$ is $\lambda_1 \ge \lambda_2 \ge ... \ge \lambda_m \ge 0$. The unit eigenvector $a_i$ of $\lambda_i$ is the coefficient $\lambda_1 \ge \lambda_2 \ge ... \ge \lambda_m \ge 0$ of the principal component $F_i$ about original variable. So the principal component $F_i$ of the original variable can be represented as:

$$F_i = a_i X \qquad (33)$$

The variance contribution rate $\alpha_i$ of the principal component is used to reflect the amount of information.

$$\alpha_i = \lambda_i / \sum_{i=1}^{m}\lambda_i \qquad (34)$$

(3) Select principal components.

The final number of principal components is determined by the variance cumulative contribution rate $G(m)$.

$$G(m) = \sum_{i=1}^{m}\lambda_i / \sum_{k=1}^{p}\lambda_k \qquad (35)$$

When the cumulative contribution rate is greater than 85%, it is considered to be sufficient to reflect the information of the original data. The data dimensionality reduction effect based on PCA method is shown in Fig. 5.

### *B. Locality Preserving Projection (LPP)*

Locality preserving projection (LPP) is a dimensionality reduction method with supervised learning, whose essence is to achieve good low-dimensional visualization through linear projection and make initial data keep as much local structure as possible in high-dimensional space. LPP algorithm inherits the idea of Laplacian feature mapping algorithm, which reduces dimensionality of high-dimensional data and effectively preserves the nonlinear structure inside the data [29]. In the reconstruction process, the source points of adjacent probe points are the same, each column of the subsystem matrix is regarded as a multivariate statistical variable, then each variable is linearly related, that is to say there is redundant information. By modeling the manifold structure to obtain a matrix of low-dimensional subsystems. These dimension reduced subsystem matrices can represent most of the information of the original data.

Construct a neighbor graph G. In the data set X, the Euclidean distance between each sample point $x_i$ and the remaining sample points is calculated. The $k$ sample points closest to the Euclidean distance with respect to each sample point $x_i$ are determined as the neighbor points of the requested point. If the data points $x_i$ and $x_j$ are contiguous, there is an edge between the points $x_i$ and $x_j$ in the figure.

Select weights and construct weights matrix $W$. In the neighbor graph, select a weight $w_{i,j}$ for each edge. There are two ways to choose the weights.

1) If points $x_i$ and $x_j$ are contiguous, set the weight of the edge as $w_{i,j} = \exp(-\| x_i - x_j \|^2 / t)$, otherwise $w_{i,j} = 0$, where $t$ is a proportional parameter.

2) If points $x_i$ and $x_j$ are contiguous, $d$ set the weight of the edge as $w_{i,j} = 1$, otherwise $w_{i,j} = 0$.

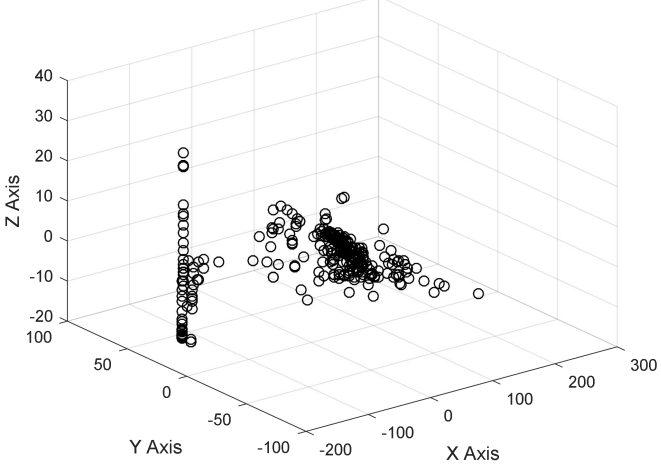


Fig. 5 Data dimensionality reduction effect based on PCA.

Perform feature mapping, calculate $d$ -dimensional embedding, and minimize the objective function of low-dimensional coordinates corresponding to neighbor reconstruction.

$$\begin{cases} \underset{Y}{Min}\dfrac{1}{2}\sum_{i,j}(y_i-y_j)^2W_{ij} \\ S.t.:YY^T=I \end{cases} \tag{36}$$

Substitute the linear transformation $y=\alpha^T x$ to obtain:

$$\begin{cases} \underset{Y}{Min}\dfrac{1}{2}\sum_{i,j}(\alpha^T x_i-\alpha^T x_j)^2W_{ij} \\ S.t.:\alpha^T XX^T\alpha=I \end{cases} \tag{37}$$

where, $L=D-W$ . The diagonal elements of diagonal matrix $D$ is $D_{ii}=\sum_j W_{ij}$ .

The eigenvector corresponding to the $d$ minimum eigenvalue of the generalized eigenvalue equation is taken as a $d$ projection vector. The eigenvector of the generalized characteristic equation is taken as the projection vector.

$$XLX^T\alpha=\lambda XX^T\alpha \tag{38}$$

The corresponding eigenvectors $\alpha_1,\alpha_2,\cdots,\alpha_{\mathrm{d}}$ of the $d$ minimum eigenvalues of the above characteristic equations form a linear transformation matrix that preserves the characteristics of neighborhood reconstruction. The data dimensionality reduction effect based on the LPP method is shown in Fig. 6.

### *C.Kernel Principal Component Analysis (KPCA)*

Kernel principal component analysis (KPCA) has a good effect on solving the nonlinear mode problem, whose core idea is to map the data into the appropriate space through some linear mapping, and then analyze and process the data in a new space through a linear learner. Compared with PCA, KPCA has a good ability to extract nonlinear features in images and the extracted principal components contain nonlinear features of the original data. Thus it can represent the original data more completely so as to reduce the training difficulty of the network, make the network convergence speed faster and improve the accuracy of the network.

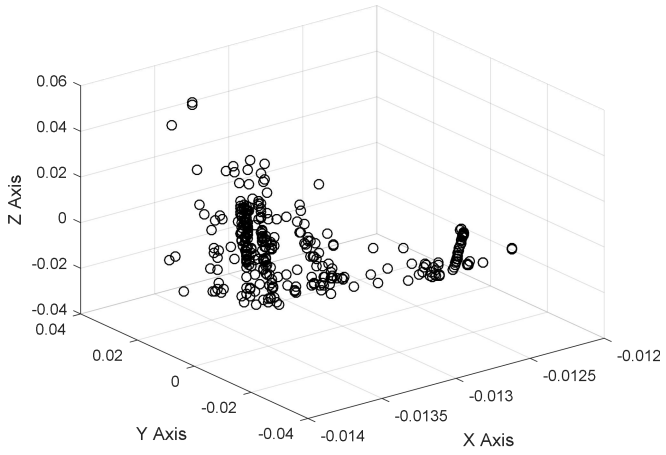


Fig. 6 Data dimensionality reduction effect based on LPP.

Sample set is $x_i$ （ $i=1,2,\cdots,M$ ）, $x_i\in R^N$ . So the nonlinear mapping relationship can be described as:

$$\varphi: R^N\to F \qquad x\mapsto\varphi(x) \tag{39}$$

Map sample point $x_i$ to $\varphi(x_i)$ by Eq. (39) and calculate the covariance matrix of the new sample space by Eq. (40).

$$R=\frac{1}{M}\sum_{i=1}^{M}\varphi(x_i)(x_i)^T \tag{40}$$

Decompose it with eigenvalues to obtain:

$$\lambda Q=RQ \tag{41}$$

where, $\lambda$ （ $\lambda>0$ ） is the eigenvalue of $R$ , and $Q$ is the corresponding eigenvector. Multiply $\varphi(x_i)$ by both sides of Eq. (41) to obtain:

$$\lambda\left(\varphi(x_i)\cdot Q\right)=\left(\varphi(x_i)\cdot RQ\right),\ i=1,2,\cdots,M \tag{42}$$

There is the coefficient $\alpha_i(i=1,2,\cdots,M)$ to make:

$$Q=\sum_{i=1}^{M}\alpha_i\varphi(x_i) \tag{43}$$

Combine above two formulas and define the matrix $K$ of $M\times M$ , where:

$$\begin{aligned}&\lambda\sum_{i=1}^{M}\alpha_i\left(\varphi(x_k),\varphi(x_i)\right)=\\&\frac{1}{M}\sum_{i=1}^{M}\alpha_i\left(\varphi(x_k),\sum_{j=1}^{M}\varphi(x_j)\right)\left(\varphi(x_j),\varphi(x_i)\right)\end{aligned} \tag{44}$$

$$K_{i,j}=\left(\varphi(x_i)\varphi(x_j)\right)=K(x_i,x_j) \tag{45}$$

Let $\alpha$ be the corresponding feature vector of the kernel matrix $K$ , then:

$$K\alpha=M\lambda\alpha \tag{46}$$

where， $\alpha=(\alpha_1,\alpha_2,\cdots,\alpha_M)^T$ .

Assume that the solution of Eq. (46) is $\lambda_1\ge\lambda_2\ge\cdots\ge\lambda_P\ge\cdots\ge\lambda_M$ , where $\lambda_P$ represents the last non-zero eigenvalue and the corresponding feature vector is $\alpha_1^k,\cdots,\alpha_p^k,\cdots\alpha_M^k$ . Then normalize the feature vector in $F$ based on Eq. (47).

$$\left(Q^k\cdot Q^k\right)=I,\quad k=1,2,\cdots,p \tag{47}$$

Plug $Q=\sum_{i=1}^{M}\alpha\varphi(x_i)$ and $K_{ij}=\left(\varphi(x_i)\varphi(x_j)\right)$ into Eq. (47) to get:

$$\begin{aligned}I&=\sum_{i,j=1}^{M}\alpha_i^k\alpha_j^k\left(\varphi(x_i)\varphi(x_j)\right)\\&=\sum_{i.j=1}^{M}\alpha_i^k\alpha_j^kK_{ij} \qquad k=1,2,\cdots,p\\&=\alpha^kK\alpha^k=\lambda_k\left(\alpha^k\cdot\alpha^k\right)\end{aligned} \tag{48}$$

Thus, for a new sample $x_i$ , extract its main integral quantity and simple project the corresponding mapping

sample $\varphi(x)$ in $F$ to $Q^k$.

$$\begin{aligned}&\left(Q^k\cdot\varphi(x)\right)\\&=\sum_{j=1}^{M}\alpha_j^k\left(\varphi(x_i)\varphi(x)\right)\quad k=1,2,\cdots,p\\&=\sum_{j=1}^{M}\alpha_j^k K\left(x_j,x\right)\end{aligned}\tag{49}$$

For simplicity's sake, replace the kernel matrix of all mapped samples with $\hat{K}=K-I_M K-KI_M+I_M KI_M$, where $(I_M)_{ij}=1/M$. Gaussian function shown in Eq. (50) is selected as the kernel function of KPCA.

$$K\left(x_j,x\right)=\exp\left\{-\frac{\left|x_j-x\right|^2}{\sigma^2}\right\}\tag{50}$$

The data dimensionality reduction effect based on the KPCA method is shown in Fig. 7.

### *D. Expectation Max Principal Component Analysis (EMPCA)*

Expectation max principal component analysis (EMPCA) plays an important role in processing incomplete data. In chemical production, due to equipment failure and other reasons, the lack of measurement data often occurs. For the research and analysis of chemical processes with missing data, the general processing method is to delete the data points or supplement the missing data with the mean, but such processing will bring bias to the network model and affect the predictive accuracy of soft-sensor model. When EMPCA estimating the incomplete data, if the dimension of the data is high or the data volume is large, the method can quickly calculate the feature values and the feature vectors, so it can effectively process the lost data and predict the related indicators of the chemical process.

Suppose sample matrix $Y_{m\times n}=\begin{pmatrix}a_{11}&\cdots&a_{1n}\\\vdots&\ddots&\vdots\\a_{m1}&\cdots&a_{mn}\end{pmatrix}$, and there are $m$ measurement samples and $n$ measurement variables.

(1) For the sample matrix $Y$ with missing data, supplement the missing data with a certain initial value and carry out the sample standardization process.

(2) Select the main element $k$ and the number of iterations $N$, and randomly initialize the matrix.

(3) $C$ is obtained by $N$ iteration of E step and M step.

$$\text{E step:}\quad X=(C^TC)^{-1}C^TY\tag{51}$$

$$\text{M step:}\quad C=YX^T(XX^T)^{-1}\tag{52}$$

(4) Calculate new estimates $Y^{m\times n}=CX$. Carry out the inverse normalization operation on $Y^{m\times n}$.

(5) The missing value in sample Y is supplemented with with $Y^{m\times n}$ in the corresponding position .

The data dimensionality reduction effect based on the EMPCA method is shown in Fig. 8.

### *E. Local Tangent Space Alignment (LTSA)*

Local tangent space alignment (LTSA), as a kind of unsupervised manifold learning dimension reduction method, can effectively extract the essential features of samples and has high robustness.

By constructing the minimum objective function, the high-dimensional samples are embedded in low-dimensional coordinates, and the data is classified according to the principle of similar clustering and heterogeneous separation. For the real-time addition of data in the chemical process, LTSA adds new samples to the original data every time, and all samples are dimension reduced again, instead of using the results of dimensionality reduction from the original data [30]. By updating the global coordinate matrix through real-time processing, the low-dimensional coordinates of training data and low-dimensional coordinates of new data are obtained. Then it is used as the initial values to perform eigenvalue iteration and realize the update of all data global coordinates, which will improve the fault state recognition rate and have incremental learning ability at the same time. In the process of production, the original data contains unlabeled information and label information that helps to identify and classify. LTSA reduces dimensionality of all data as unlabeled data.

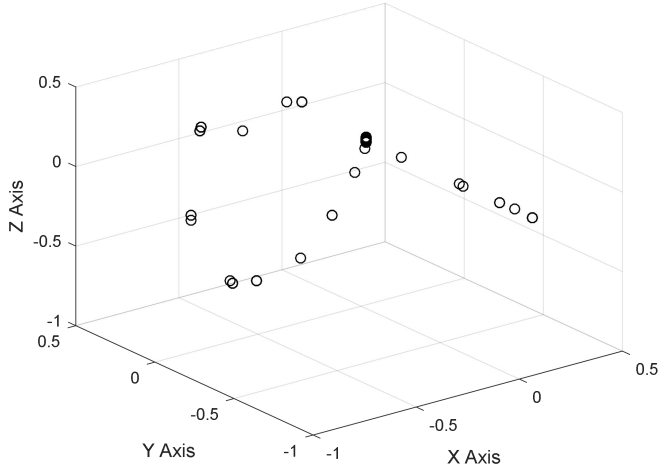


Fig. 7 Data dimensionality reduction effect based on KPCA.

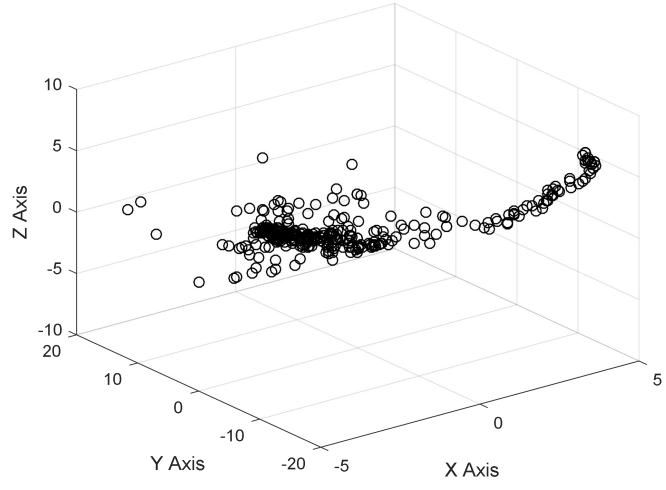


Fig. 8 Data dimensionality reduction effect based on EMPCA.

(1) Construct a neighborhood. For each data point $x_i$, find its $k$ neighborhood points including itself to form a matrix $X_i = [x_{i_1}, \cdots, x_{i_k}]$.

(2) Local linear projection. For the neighborhood $X_i$ of each sample point, calculate the left singular vector of the centralization matrix $X_i(I - ee^T / k)$ and construct the matrix $Q_i$ with the $d$ left singular vectors.

(3) Based on Eq. (53), calculate the left singular vector $U$, and take the first $d$ column of $U$ to form the matrix $Q_i$ ($Q_i$ is the approximation of the tangent space of the point $x_i$).

$$X_i(I - ee^T / k) = U\Sigma V \tag{53}$$

(4) Calculate the projection of each neighborhood point in the tangent space $Q_i$.

$$\Phi_i = Q_i^T X_i(I - ee^T / k) = [\theta_1^{(i)}, \cdots, \theta_k^{(i)}] \tag{54}$$

$$\theta_j^{(i)} = Q_i^T(x_{i_j} - \overline{x}_i) \tag{55}$$

(5) Arrangement of local coordinate system.

For the local tangent space coordinate $\theta_i = (\theta_1^{(i)}, \theta_2^{(i)}, \cdots, \theta_k^{(i)})(i = 1, 2, \cdots, N)$ of each neighborhood, construct a transformation matrix $L_i = \theta_i^+ \in R^{d \times d}$. Obtain the solution by minimize $\sum \left\| T_i(I - ee^T / k) - L_i\theta_i \right\|^2$, that is to calculate the eigenvector corresponding to the eigenvalues from the second smallest to the $d+1$ smallest. $\theta_i^+$ is the generalized Moor-Penrose inverse of $\theta$. The data dimensionality reduction effect based on the LTSA method is shown in Fig. 9.

### *F. T-distributed Stochastic Neighbor Embedding (TSNE)*

T-distributed stochastic neighbor embedding (TSNE) is an algorithm derived from SNE. SNE changed the idea of distance-invariant in MDN and ISOMAP algorithms. When mapping high-dimensional data to low-dimensional data, the distribution probability among data is guaranteed to be unchanged. SNE regards the distribution of data in high-dimensional and low-dimensional as Gaussian distribution, while TSNE treats data in low-dimensional as T distribution. This is to solve the problem of data congestion by making the distance between clusters bigger.

Its advantage over PCA is that PCA is a linear algorithm that cannot explain the complex relationship between features, but TSNE can do. When calculating the corresponding conditional probability in the dimensionality reduction algorithm and minimizing the sum of the probability differences of the higher and lower dimensions, this involves a large number of calculations and requires high system resources. The complexity of TSNE has a rule of increasing quadratic time and space with the amount of data.

(1) Construct the local neighborhoods. For data set $X = \{x_1, x_2, \cdots, x_n\}$, calculate the Euclidean distance $d_x(x_i, x_j)$ between any two sample vectors $x_i$ and $x_j$ and compare each point with all points. When the distance between two points is less than the fixed radius $\varepsilon$, they are considered to be adjacent and connected. The length of this edge is $d_x(x_i, x_j)$, then the neighborhood graph G is obtained.

(2) Calculate the shortest distance. In figure G, the shortest distance between any two sample vectors $x_i$ and $x_j$ is $d_G(x_i, x_j)$. If there is a connection between $x_i$ and $x_j$, the initial value of $d_G(x_i, x_j)$ is $d_x(x_i, x_j)$, otherwise $d_G(x_i, x_j) = \infty$. For $i, j = 1, 2, 3, ..., n$, calculate:

$$\begin{aligned} &d_G(x_i, x_j) = \\ &\min\{d_G(x_i, x_j), d_G(x_i, x_k) + d_G(x_k, x_j)\} \end{aligned} \tag{56}$$

Thus, get matrix $D_G = \{d_G(x_i, x_j)\}$, which is composed of the shortest path of all point pairs in figure G.

(3) Construct $d$ dimension vector that maintains the intrinsic geometry embedded in space $Y$.

$$\tau(D_G) = -\frac{H * (D_G)^2 * H}{2} \tag{57}$$

$H$ is the unit matrix having same order with $D_G$. The feature decomposition is performed on $\tau(D_G)$. Take the largest $d$ previous eigenvalues are $\lambda_1, \lambda_2, \cdots, \lambda_d$ and the corresponding eigenvectors are $V_1, V_2, \cdots, V_d$. Set $V_p^i$ is the $i$ component of the $p$ feature vector, so the corresponding low dimension data can be represented as $y_i = \lambda_p^{1/2} V_p^i$. The data dimensionality reduction effect based on TSNE method is shown in Fig. 10.

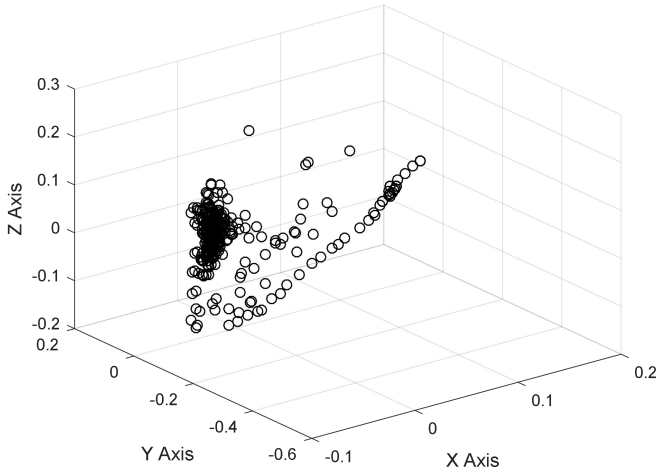


Fig. 9 Data dimensionality reduction effect based on LTSA.

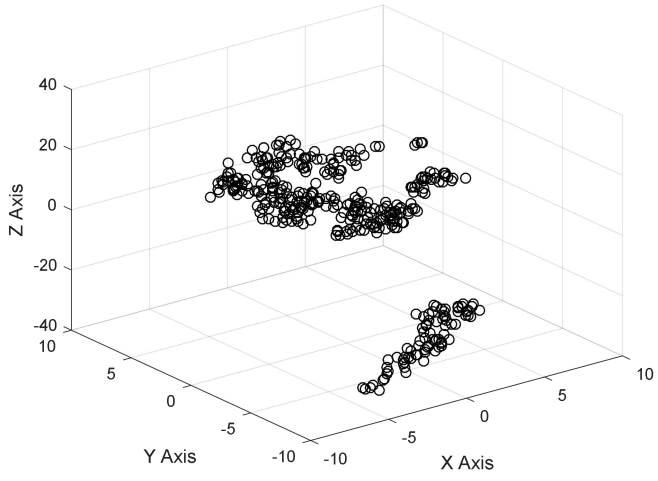


Fig. 10 Data dimensionality reduction effect based on TSNE.

### G. Neighborhood Preserving Embedding (NPE)

Neighborhood preserving embedding (NPE) using the tag information of the data to obtain a low-dimensional feature with higher recognition so as to maximize the center point distance of non-similar data while keeping the data neighborhood structure unchanged in the selected projection direction. NPE algorithm can be regarded as an improvement of the LLE algorithm. It can keep the neighborhood structure unchanged and obtain the projection matrix from high dimension to low dimension while carrying out the data dimensionality reduction operation so that it greatly facilitates the processing of new data. NPE algorithm is to minimize the objective function for the low-dimensional coordinate reconstruction, that is to say:

$$\begin{cases} \underset{Y}{Min} \sum_{i=1}^{n} \left\| y_i - \sum_{j=1}^{k} W_{ij} y_{ij} \right\|^2 \\ S.t.: YY^T = I \end{cases} \tag{58}$$

Substitute the linear transformation $y = \alpha^T x$ to get:

$$\begin{cases} \underset{\alpha}{Min} \sum_{i=1}^{n} \left\| \mathrm{x}_i^T - \sum_{j=1}^{k} W_{ij} x_{ij}^T \alpha \right\|^2 \\ S.t.: \alpha^T XX^T \alpha = I \end{cases} \tag{59}$$

$$\begin{cases} \underset{\alpha}{Min}\, \alpha^T XMX^T \alpha \\ S.t.: \alpha^T XX^T \alpha = I \end{cases} \tag{60}$$

where, $M = (I - W)^T (I - W)$ .

The eigenvector corresponding to the $d$ minimum eigenvalues of the generalized eigenvalue equation is taken as a $d$ projection vector.

$$XMX^T \alpha = \lambda XX^T \alpha \tag{61}$$

Therefore, a linear transformation matrix is constructed by the eigenvectors $\alpha_1, \alpha_2, \cdots, \alpha_d$ corresponding to the $d$ minimum eigenvalues $\lambda_1, \lambda_2, \cdots, \lambda_d$ of the above characteristic equation. NPE algorithm realizes linear transformation from data to new space by $x_i \to y_i = A^T x_i$ . Obviously, the data transformation realized by this linear transformation is an explicit form, which solves the problem that the LLE algorithm does not have an explicit mapping function. The data dimensionality reduction effect based on the NPE method is shown in Fig. 11.

It can be seen that although using different dimensionality reduction methods, the classification results of the data are roughly the similar. The similar data are distributed around the main features, and the heterogeneous data is dispersed. This also fully illustrates the complexity and continuity of chemical data. The rules for dimensionality reduction are artificially formulated, but the main data characteristics are objective , it do not shift from human will. The data after the operation under seven dimensionality reduction methods are compactly gathered in the three-dimensional space, so it facilitates more intuitive differentiation of data. However, in the actual production process, the three-dimensional space is not the optimal expression dimension of the data.

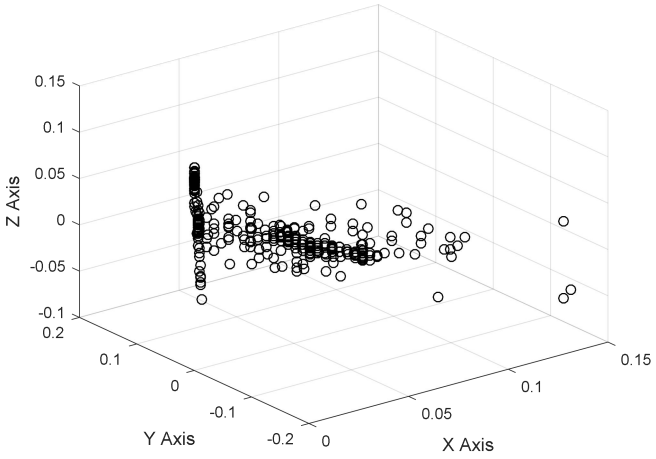


Fig. 11 Data dimensionality reduction effect based on NPE.

In this paper, combined with the polyvinyl chloride production process, ten dimensional data on the site was reduced to five dimension having good results. Therefore, under the premise of satisfying the accuracy of certain production requirements, the five dimensional variables after dimension reduction operation are selected as the input variables of the proposed soft-sensor model to predict the VCM conversion rate.

## V. Simulation Experiments and Results Analysis

In this paper, the polymerization process of a chemical factory with 40,000 tons/year polyvinyl chloride (PVC) production device is taken as background, whose technology is introduced by the Goodrich (B·F·G) company in the United States, which takes vinyl chloride monomer (VCM) as a raw material, and uses a suspension polymerization technology to produce polyvinyl chloride (PVC) resin. For predicting the VCM conversion rate in the PVC polymerization process, a neural network soft-sensor modelling method based on seven data dimensionality reduction strategies was proposed. Before carrying out the soft-sensor modeling, in order to measure the performance of the prediction models, several performance indicators are defined in Table 2, where $\hat{y}$ is the estimated value and $y$ is the actual value. The historical production data of the PVC polymerization process are collected. 600 sets historical data in Two kettles with uniformity and representativeness are selected. These preprocessed data is divided into two parts, the first 500 sets data are used as training data, and the remaining 100 sets data are used to verify the performance of the soft-sensor models. Seven data dimensionality reduction methods (PCA, LPP, KPCA, EMPCA, LTSA, TSNE and NPE) are adopted to carry out dimension reduction. Then RBF neural network based on gradient learning, orthogonal least squares and cluster learning algorithms and D-FNN are used to fit the input and output data. The prediction results and prediction errors are shown in Fig. 12-25. Table 3 is performance comparison results of different soft-sensor models. According to the above simulation results, the accurate prediction on the VCM conversion rate are realized and the time complexity requirement are satisfied. In view of the adopted four performance indicators (MPE, MNE, SSE

and RMSE), in the neural network soft-sensor models based on GLA-RBF, the model with LTSA dimensionality reduction has the highest prediction accuracy. In the soft-sensor models based on CLA-RBF, the model with TSNE dimensionality reduction has the highest prediction accuracy. In the soft-sensor model based on OLS-RBF, the model with PCA dimensionality reduction has the highest prediction accuracy; in the soft-sensor model based on DFNN, the model with EMPCA dimensionality reduction has the highest prediction accuracy. Therefore, the dimensionality reduction methods can simplify the neural network topology and reduce the training time of soft-sensor models.

TABLE 2. DEFINITION OF MODEL PERFORMANCE INDEX

| | |
|---|---|
| Maximum positive error | $MPE = \max\{(\hat{y} - y), 0\}$ |
| Maximum negative error | $MNE = \min\{(\hat{y} - y), 0\}$ |
| Root mean square error | $RMSE = \left[\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2\right]^{\frac{1}{2}}$ |
| Sum of squared error | $SSE = \sum_{i=1}^{n}(\hat{y}_i - y_i)^2$ |

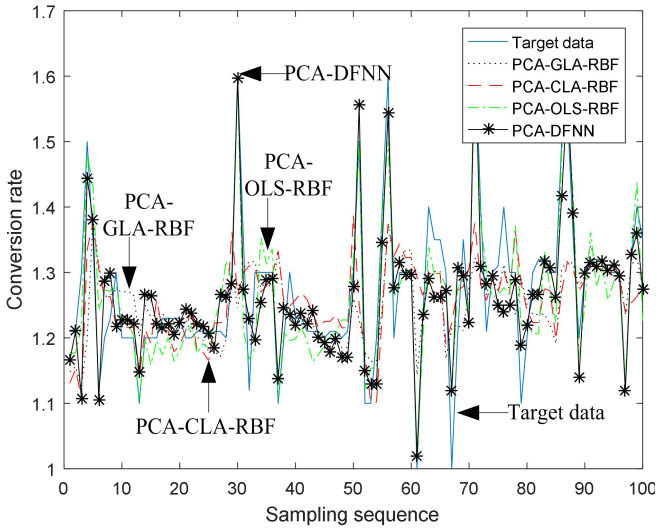


Fig. 12 Predictive results of conversion rate by PCA based soft-sensor models.

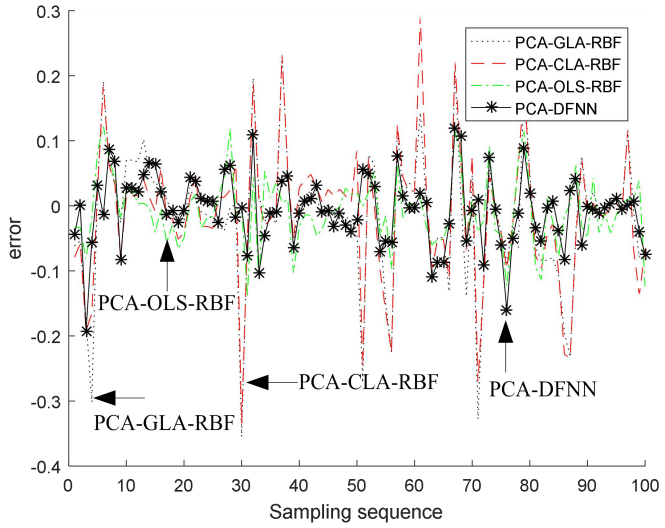


Fig. 13 Predictive error of conversion rate by PCA based soft-sensor models.

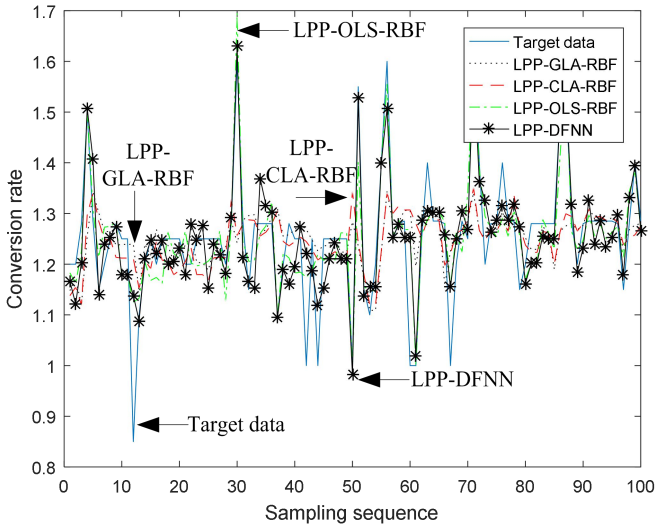


Fig. 14 Predictive results of conversion rate by LPP based soft-sensor models.

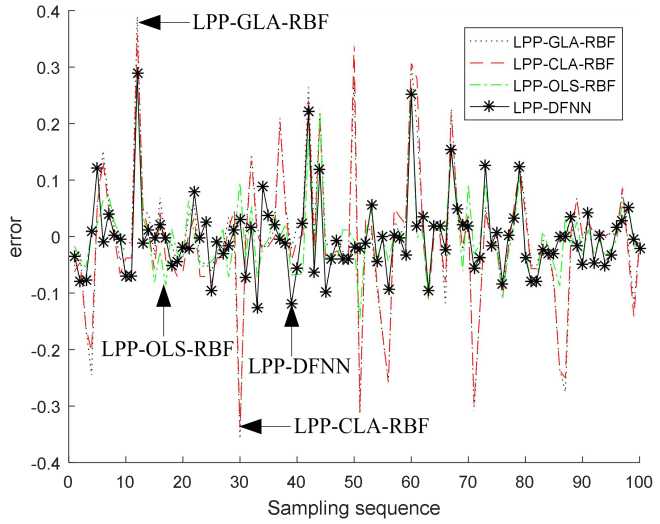


Fig. 15 Predictive error of conversion rate by LPP based soft-sensor models.

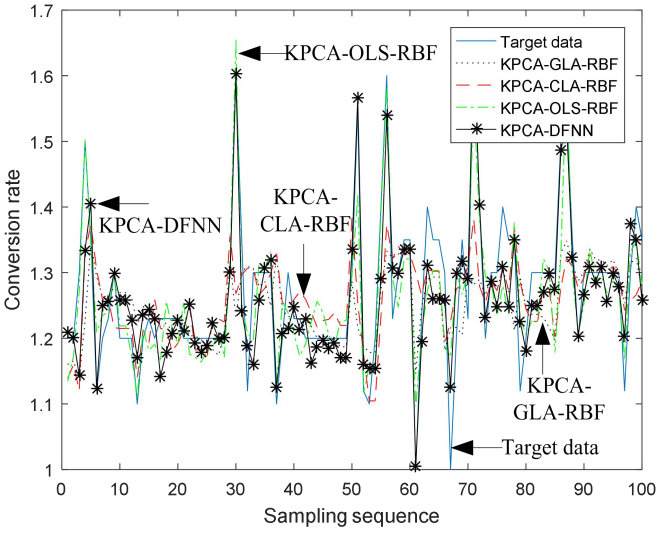


Fig. 16 Predictive results of conversion rate by KPCA based soft-sensor models.

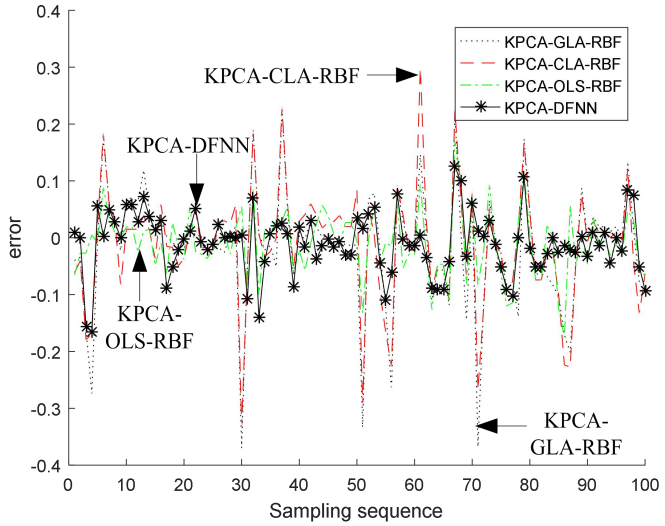


Fig. 17 Predictive error of conversion rate by KPCA based soft-sensor models.

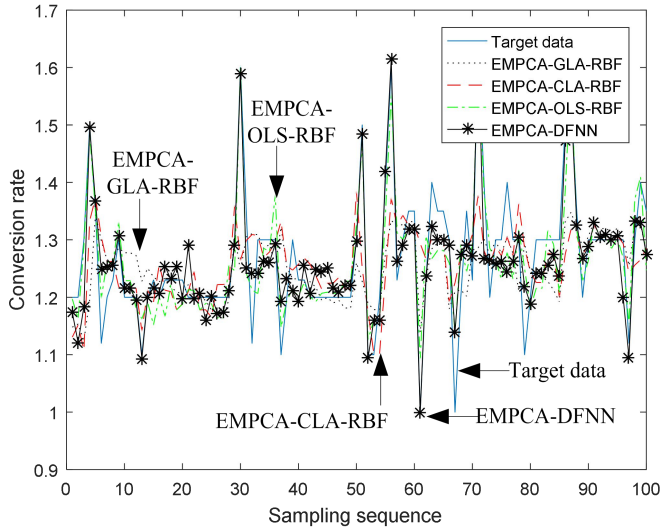


Fig. 18 Predictive results of conversion rate by EMPCA based soft-sensor models.

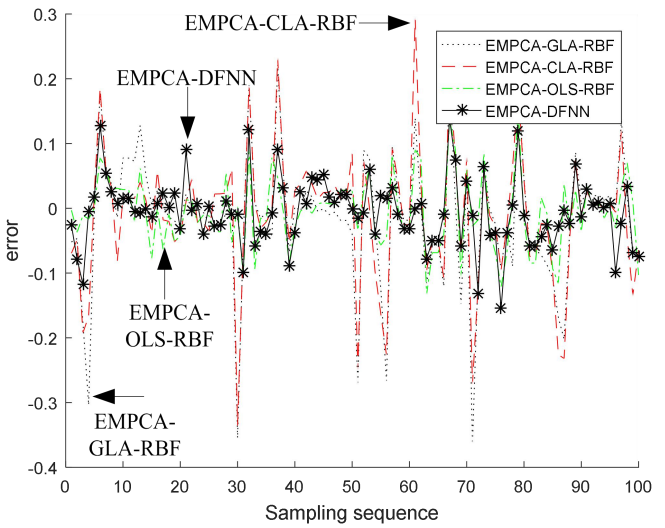


Fig. 19 Predictive error of conversion rate by EMPCA based soft-sensor models.

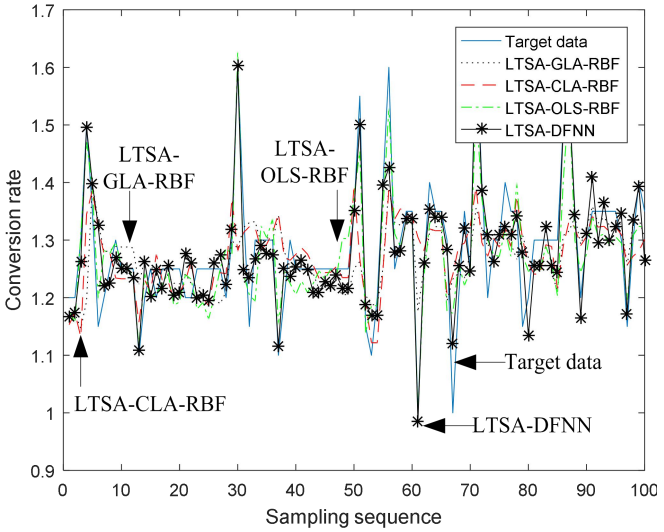


Fig. 20 Predictive results of conversion rate by LTSA based soft-sensor models.

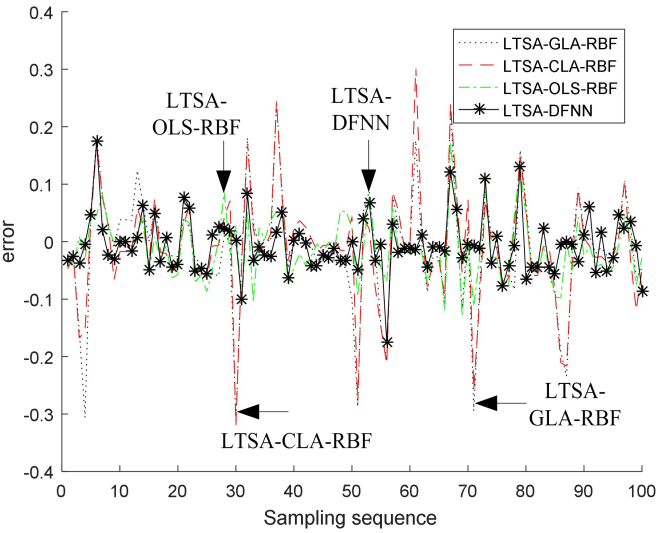


Fig. 21 Predictive error of conversion rate by LTSA based soft-sensor models.

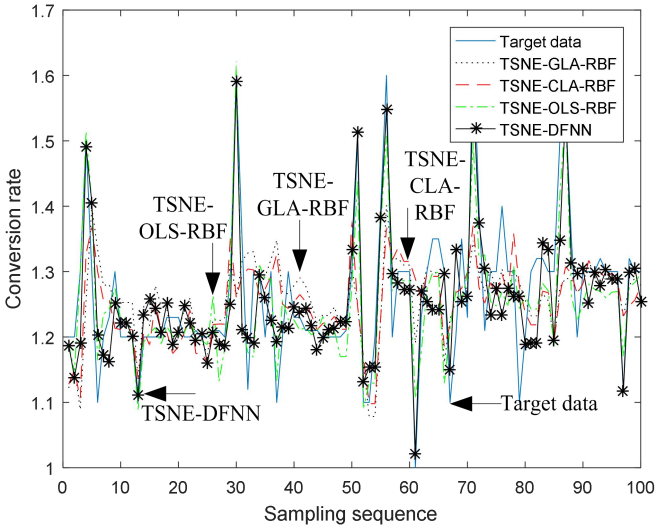


Fig. 22 Predictive results of conversion rate by TSNE based soft-sensor models.

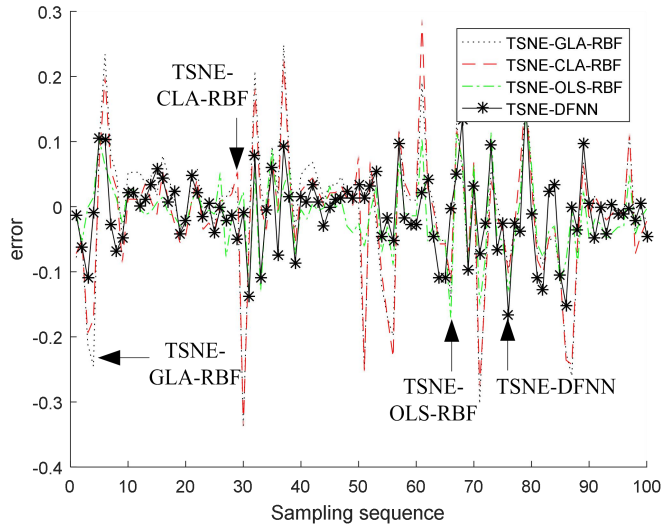


Fig. 23 Predictive error of conversion rate by TSNE based soft-sensor models.

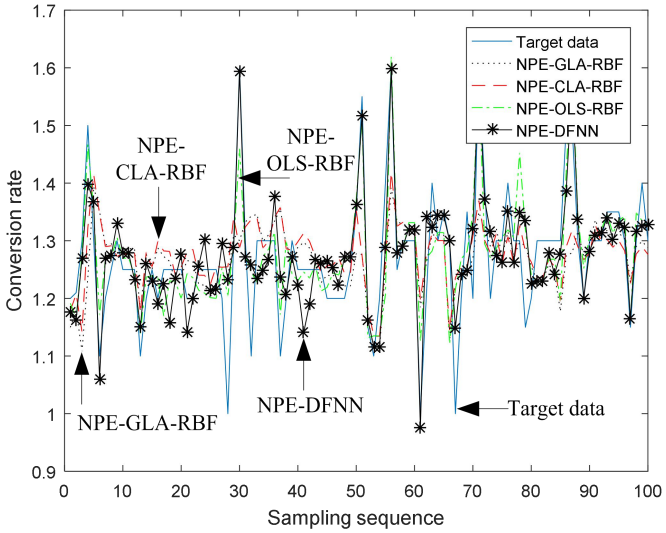


Fig. 24 Predictive results of conversion rate by NPE based soft-sensor models.

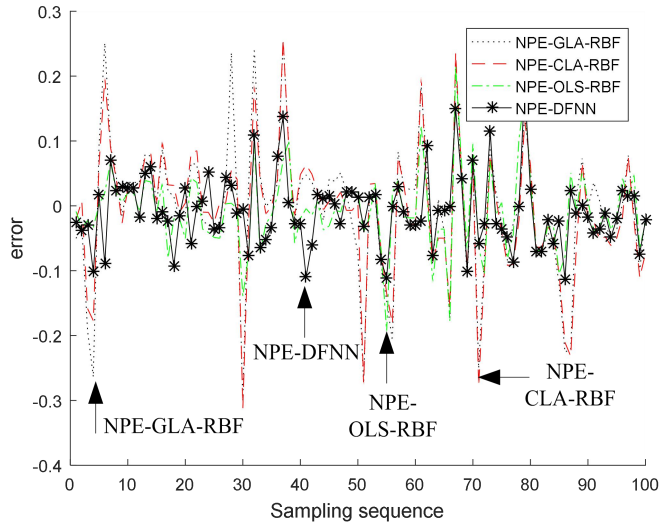


Fig. 25 Predictive error of conversion rate by NPE based soft-sensor models.

TABLE 3. PERFORMANCE COMPARISON RESULTS UNDER DIFFERENT SOFT-SENSOR MODELS

| Soft-sensor model | | MPE | MNE | SSE | RMSE |
|---|---|---|---|---|---|
| PCA | PCA-GLA-RBF | 0.2337 | -0.3547 | 1.0727 | 0.0641 |
| | PCA-CLA-RBF | 0.5093 | -0.5828 | 3.1760 | 0.1104 |
| | PCA-OLS-RBF | 1.2459 | -0.1382 | 0.2884 | 0.0332 |
| | PCA-DFNN | 0.1198 | -0.1924 | 0.2923 | 0.0325 |
| LPP | LPP-GLA-RBF | 0.3896 | -0.3581 | 1.4419 | 0.0743 |
| | LPP-CLA-RBF | 0.6890 | -0.6550 | 5.2565 | 0.1419 |
| | LPP-OLS-RBF | 1.6488 | -0.1484 | 0.4897 | 0.0433 |
| | LPP-DFNN | 0.2884 | -0.1263 | 0.4717 | 0.0425 |
| KPCA | KPCA-GLA-RBF | 0.2254 | -0.3698 | 1.1170 | 0.0654 |
| | KPCA-CLA-RBF | 0.5460 | -0.6056 | 3.4455 | 0.1149 |
| | KPCA-OLS-RBF | 1.3532 | -0.1755 | 0.3525 | 0.0367 |
| | KPCA-DFNN | 0.1496 | -0.1243 | 0.2929 | 0.0335 |
| EMPCA | EMPCA-GLA-RBF | 0.2334 | -0.3573 | 1.0483 | 0.0634 |
| | EMPCA-CLA-RBF | 0.5107 | -0.5839 | 3.1314 | 0.1095 |
| | EMPCA-OLS-RBF | 1.2682 | -0.1329 | 0.3162 | 0.0348 |
| | EMPCA-DFNN | 0.1198 | -0.1447 | 0.2451 | 0.0306 |
| LTSA | LTSA-GLA-RBF | 0.2430 | -0.3158 | 0.9938 | 0.0617 |
| | LTSA-CLA-RBF | 0.5546 | -0.5807 | 3.1783 | 0.1104 |
| | LTSA-OLS-RBF | 1.3550 | -0.1286 | 0.3128 | 0.0346 |
| | LTSA-DFNN | 0.1747 | -0.1747 | 0.2420 | 0.0305 |
| TSNE | TSNE-GLA-RBF | 0.2481 | 0.3393 | 1.0577 | 0.0637 |
| | TSNE-CLA-RBF | 0.5023 | -0.5906 | 3.0550 | 0.1082 |
| | TSNE-OLS-RBF | 1.2361 | -0.1724 | 0.3476 | 0.0365 |
| | TSNE-DFNN | 0.1627 | -0.1656 | 0.3668 | 0.0375 |
| NPE | NPE-GLA-RBF | 0.2797 | -0.3046 | 1.0131 | 0.0623 |
| | NPE-CLA-RBF | 0.4693 | -0.5684 | 3.1909 | 0.1106 |
| | NPE-OLS-RBF | 1.3973 | -0.1938 | 0.3922 | 0.0388 |
| | NPE-DFNN | 0.1841 | -0.1315 | 0.3373 | 0.0360 |

## VI. CONCLUSIONS

Aiming at prediction on VCM conversion rate in the PVC production process, the neural network soft-sensor models based on data dimensionality reduction strategies was proposed. Seven data dimensionality reduction methods (PCA, LPP, KPCA, EMPCA, LTSA, TSNE and NPE) are adopted to carry out the dimension reduction on the high-dimensional input data of the soft sensor models. The RBF neural network based on three learning algorithms (gradient learning, orthogonal least squares and cluster learning) and D-FNN are adopted to realize the match between the input and output data. The simulation results verify that the established soft-sensor models can meet the requirements on the accurate prediction and time complexity on the VCM conversion rate. By utilizing the multiple dimensionality reduction methods on the modeling data, on the other hand, most of the information in the raw data is preserved and the prediction accuracy of the VCM conversion rate is ensured. On the other hand, the simplification of the neural network topology and the reduction the training time of soft-sensor model are realized.

## REFERENCES


[1] W. Shi, J. Zhang, X. M. Shi, and G. D. Jiang, "Different Photodegradation Processes of PVC with Different Average Degrees of Polymerization," *Journal of Applied Polymer Science*, vol. 107, no. 1, pp. 528-540, 2008.

[2] Liu, and Jing, "The Application and Simulation Research of the Fuzzy PID Control Used in PVC Polymerization Temperature Control System," *Advanced Materials Research*, vol. 466-467, pp. 47-51, 2012.

[3] H. Kim, P. Howland, H. Park, and N. Christianini, "Dimension Reduction in Text Classification with Support Vector Machines," *Journal of Machine Learning Research*, vol. 6, no. 1, pp. 37-53, 2005.

[4] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, "Dimensionality Reduction Using Genetic Algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 2, pp. 164-171, 2002.

[5] T. Zhang, D. Tao, X. Li, and J. Yang, "Patch Alignment for Dimensionality Reduction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1299-1313, 2009.

[6] S. Wen, J. S. Wang, and J. Gao, "Fault Diagnosis Strategy of Polymerization Kettle Equipment Based on Support Vector Machine and Cuckoo Search Algorithm," *Engineering Letters*, vol. 25, no. 4, pp. 474-482, 2017.

[7] S. Chen, H. Zhao, M. Kong, and B. Luo, "2D-LPP: a Two-dimensional Extension of Locality Preserving Projections," *Neurocomputing*, vol. 70, no. 4-6, pp. 912-921, 2007.

[8] Y. J. Cheng, W. Hong, and K. Wu, "Design of a Monopulse Antenna Using a Dual V-type Linearly Tapered Slot Antenna (DVLTSA)," *IEEE Transactions on Antennas and Propagation*, vol. 56, no. 9, pp. 2903-2909, 2008.

[9] T. Naiki, K. Hayashi, and S. Takemura, "An LDA and Flow Visualization Study of Pulsatile Flow in an Aortic Bifurcation Model," *Biorheology*, vol. 32, no. 1, pp. 43-59, 2017.

[10] S. Emori, K. Takahashi, Y. Yamagata, S. Kanae, S. Mori, and Y. Fujigaki, "Risk Implications of Long-term Global Climate Goals: Overall Conclusions of the Ica-rus Project," *Sustainability Science*, vol. 13, no. 2, pp. 279-289, 2018.

[11] A. A. Mohammed, R. Minhas, Q. M. J. Wu, and M. A. Sid-Ahmed, "Human Face Recognition Based on Multidimensional PCA and Extreme Learning Machine," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2588-2597, 2011.

[12] L. J. Cao, K. S. Chua, W. K. Chong, H. P. Lee, and Q. M. Gu, "A Comparison of PCA, KPCA and ICA for Dimensionality Reduction in Support Vector Machine," *Neurocomputing*, vol. 55, no. 1-2, pp. 321-336, 2003.

[13] F. S. Tsai, "Dimensionality Reduction for Computer Facial Animation," *Expert Systems with Applications*, vol. 39, no. 5, pp. 4965-4971, 2012.

[14] N. Pezzotti, B. P. F. Lelieveldt, L. V. D. Maaten, H. Thomas, and A. Vilanova, "Approximated and User Steerable TSNE for Progressive Visual Analytics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 7, pp. 1739-1752, 2016.

[15] L. Nanni, A. Lumini, and S. Brahnam, "Survey on LBP Based Texture Descriptors for Image Classification," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3634-3641, 2012.

[16] L. Lu, Z. Yi-Ju, J. Qing, and C. Qing-Ling, "Recognizing Human Actions by Two-level Beta Process Hidden Markov Model," *Multimedia Systems*, vol. 23, no. 2, pp. 183-194, 2017.

[17] M. J. Er, S. Wu, J. Lu, and H. L. Toh, "Face Recognition with Radial Basis Function (RBF) Neural Networks," *IEEE Transactions on Neural Networks*, vol. 13, no. 3, pp. 697-710, 2002.

[18] P. Lin, S. Chang, H. Wang, Q. J. Huang, and J. He, "Spikecd: a Parameter-insensitive Spiking Neural Network with Clustering Degeneracy Strategy," *Neural Computing & Applications*, no. 5786, pp. 1-13, 2018.

[19] Z. Yun, Z. Quan, S. Caixin, L. Shaolan, L. Yuming, and S. Yang, "RBF Neural Network and ANFIS-based Short-term Load Forecasting Approach in Real-time Price Environment," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 853-858, 2008.

[20] H. T. Zhang, M. M. Du, G. F. Wu, and W. S. Bu, "PD Control with RBF Neural Network Gravity Compensation for Manipulator," *Engineering Letters*, vol. 26, no. 2, pp. 236-244, 2018.

[21] G. B. Huang, Q. Y. Zhu, and C. K. Siew, "Extreme Learning Machine: Theory and Applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489-501, 2006.

[22] J. S. Wang, S. Han, and Q. P. Guo, "Echo State Networks Based Predictive Model of Vinylchloride Monomer Convention Velocity Optimized by Artificial Fish Swarm Algorithm," *Soft Computing*, vol. 18, no. 3, pp. 457-468, 2014.

[23] A. F. Hayes, and J. Matthes , "Computational Procedures for Probing Interactions in OLS and Logistic Regression: SPSS and SAS Implementations," *Behavior Research Methods*, vol. 41, no. 3, pp. 924-936, 2009.

[24] G. B. Huang, P. Saratchandran, and N. Sundararajan," A Generalized Growing and Pruning RBF (GGAP-RBF) Neural Network for Function Approximation," *IEEE Transactions on Neural Networks*, vol. 16, no. 1, pp. 57-67, 2005.

[25] J. S. Wang, and Q. P. Guo, "D-FNN Based Soft-sensor Modeling and Migration Reconfiguration of Polymerizing Process," *Applied Soft Computing*, vol. 13, no. 4, pp. 1892-1901, 2013.

[26] M. Torabi, "Likelihood Inference in Generalized Linear Mixed Measurement Error Models," *Computational Statistics & Data Analysis*, vol. 57, no. 1, pp. 549-557, 2013.

[27] M. D. Ritchie, L. W. Hahn, N. Roodi, L. R. Bailey, W. D. Dupont, and F. F. Parl, et al., "Multifactor-dimensionality Reduction Reveals High-order Interactions among Estrogen-metabolism Genes in Sporadic Breast Cancer," *American Journal of Human Genetics*, vol. 69, no. 1, pp. 138-147, 2001.

[28] S. Yan, D. Xu, B. Zhang, H. Zhang, and S. Lin, "Graph Embedding and Extensions: a General Framework for Dimensionality Reduction," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 29, no. 1, pp. 40-51, 2007.

[29] Y. Xu, A. Zhong, J. Yang, and D. Zhang, "LPP Solution Schemes for Use with Face Recognition," *Pattern Recognition*, vol. 43, no. 12, pp. 4165-4176, 2010.

[30] W. Jiang, X. Wang, and Z. Wang, " LTSA and Combined Index Based MICA and PCA Process Monitoring and Application," *CIESC Journal*, vol. 66, no. 12, pp. 4895-4903, 2015.



**Cheng Xing** received her B. Sc. degree in automation from University of Science and Technology Liaoning, China in 2011, and her M. Sc. degree in computer science and engineering from Hanyang University, Korea in 2013. She is a Ph.D candidate in the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Her main research interest is modeling of complex industry process.

**Jie-Sheng Wang** received his B. Sc. And M. Sc. degrees in control science from University of Science and Technology Liaoning, China in 1999 and 2002, respectively, and his Ph. D. degree in control science from Dalian University of Technology, China in 2006. He is currently a professor and Master's Supervisor in School of Electronic and Information Engineering, University of Science and Technology Liaoning. His main research interest is modeling of complex industry process, intelligent control and Computer integrated manufacturing.

**Lei Zhang** is a postgraduate student in the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114044, PR China.His main research interest is mathematical modeling of complex system and intelligent optimization algorithms.

**Wei Xie** is a postgraduate student in the School of Electronic and Information Engineering, University of Science and Technology Liaoning, Anshan, 114051, PR China. His main research interest is mathematical modeling of complex system and intelligent optimization algorithms.