International Journal of Computational Intelligence Systems
Vol. 12(2), 2019, *pp.* 1062–1074
DOI: https://doi.org/10.2991/ijcis.d.190826.001; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Soft Sensor Modeling Method by Maximizing Output-Related Variable Characteristics Based on a Stacked Autoencoder and Maximal Information Coefficients

Yanzhen Wang, Xuefeng Yan*

Key Laboratory of Advanced Control and Optimization for Chemical Processes of Ministry of Education, East China University of Science and Technology, 130 Meilong Road, Shanghai, 200237, P. R. China

ARTICLE INFO

Article History

Received 11 Apr 2019
Accepted 21 Aug 2019

Keywords

Soft sensor
Deep learning
Stacked autoencoder (SAE)
Maximal information coefficient (MIC)
Modeling method

ABSTRACT

The key factors required to establish a precise soft sensor model for industrial processes include selection of variables affecting vital indicators from a large number of online measurement variables and elimination of the effects of unrelated disturbance variables. How to compress redundant information and retain the unique characteristic information contained by the selected variables is worthy of in-depth research. A novel soft sensor modeling method based on weighted maximal information coefficients (MICs) and a stacked autoencoder (SAE), hereinafter referred to as MICW-SAE, is proposed in this work. In our model, the MICs between each input and output variable are calculated and compared with the threshold before training each network in SAE. Then, input variables with low MICs are selected, and the average MIC index is calculated using other input variables. If the index is higher than the second threshold, the MIC of this specific variable is set to 0. Finally, the weights of all input variables are determined in accordance with the scale and placed into the loss function for training. The Boston house-price and naphtha dry point temperature datasets are used to prove the prediction ability of our model. Results demonstrate that MICW-SAE can enhance the output-related features of the input variables. Moreover, redundant information that can also be represented by other input variables are identified and excluded.

1. INTRODUCTION

Demands for industrial process control are increasing with the flourishing development of modern industry [1]. Since process variables are often highly related to the quality of the final product, controlling these variables in a timely and precise manner is important. In a typical process, some variables are key indicators of process performance. However, most variables are difficult to measure due to several issues, including economic problems, environmental harshness, technical constraints, and severe time-delay. Soft sensor technology is often used to solve such problems [2,3]. Soft sensor technology allows accurate and real-time online measurement of variables dominant to a process, thereby rendering stable process control and ensuring the quality and economic benefits of industrial products. Thus, soft sensors are an important research field in modern control.

Soft sensor modeling methods are of three types, namely, first-principles models (FPMs), data-driven models, and mixed models (combination of FPMs with data-driven models). FPMs require prior knowledge of the mechanism of process objects and usually need to idealize and simplify some of the processes. Thus, implementation of FPMs to complex industrial processes is difficult. By contrast, data-driven models do not need detailed mechanistic information of the process during modeling as they only utilize the sample data from an industrial process to describe its state. Therefore, data-driven models are widely used in industrial processes [4]. A number of data-driven modeling methods are available, and these methods are mainly divided into two categories. One is based on multivariate statistical algorithms, including principal component regression (PCR) [5] and partial least-squares regression (PLS) [6], and the other is based on statistical machine learning algorithms, such as support vector regression (SVR) [7], genetic algorithm (GA) [8], and artificial neural network (ANN) [9]. Although these algorithms can be applied to various fields, some problems, such as those related to robustness and accuracy, still exist in the soft sensor modeling process. ANN has become an important modeling tool for soft sensors, but it only allows shallow learning, which presents a single hidden layer, and easily falls into a local optimum or undergoes gradient diffusion. Establishing an accurate and effective estimation model is a key endeavor in soft sensor modeling processes. Bengio *et al.* [10] proved that deep learning, a branch of machine learning, can overcome these complex problems well.

Deep learning, which was first proposed by Hinton and Salakhutdinov [11], is a data analysis and processing method. In recent years, it has received increased attention and gradually become a key research area of artificial intelligence [12]. Convolutional

*Corresponding author. Email: xfyan@ecust.edu.cn neural network (CNN), recurrent neural network (RNN), deep autoencoder (DAE), long short-term memory network, and deep belief network (DBN) are considered deep learning algorithms. Initially, deep learning was applied to the fields of image identification [13–16] and natural language processing [17–19]. Today, it is also gradually applied on process monitoring and modeling prediction [20–23]. Zhu *et al.* [24] issued a novel prediction method based on DBN that used linear regression as the final layer of its general structure. This model could examine the nonlinear and unstable features of internal valve leakage signals. Yan *et al.* [25] proposed a novel soft sensor modeling method that combines denoising autoencoders with a neural network (DAE-NN). In contrast to shallow learning methods, DAE-NN could remarkably improve the robustness and performance of predictions of the oxygen content in flue gasses in 1000 MW ultrasuperficial units. Despite their many benefits, however, these existing methods place all input variables into the network for training and barely consider the correlation between variables. Thus, even unimportant variables, which contribute to prediction errors, are treated equally when training the model.

Some works considering weighted features in deep neural networks have been reported. Yu and Yan [26] monitored and enhanced the layer-wise abnormal fluctuation information extracted by DBN to prevent this information from vanishing during the training process. The enhancement degree was determined by the fluctuation degree. The enhanced features showed the current working status by integrating support vector data, moving average filter, and kernel density estimation techniques, and the resulting method improved the detectability and performance of the Tennessee Eastman benchmark process. Yuan *et al.* [27] developed a soft sensor modeling with a variable-wise weighted stacked autoencoder (VW-SAE). This model considered the linear correlation between independent and dependent variables measured by the Pearson correlation coefficient before training. Application to estimations of butane content showed that this model achieves higher accuracy than three other methods. The method proposed by Yuan *et al.* [27] only calculated the linear correlation between independent and dependent variables. In fact, however, input and output variables are often highly nonlinearly correlated in the practical industrial process. The Pearson correlation coefficient cannot describe the nonlinear correlation between variables well. Although some variables are given relatively small weights, these variables and their effects on the output still exist during training. Furthermore, the number of variables collected at each sampling point may be massive during the industrial process, but not all input variables are useful or have a significant influence on the output variables due to differences in physical or chemical mechanisms. Thus, filtering redundant information while retaining unique feature information contained by other variables relevant to the output is particularly important.

Our work aims to uniformly select input variables and measure the relationships between them before training the network. We propose a novel SAE soft sensor modeling method based on maximal information coefficients (MICs) weighted between variable, hereinafter referred to as MICW-SAE. In this new model, the MIC between each input and output variables is initially calculated. If the MIC of one independent variable is lower than the threshold, the average MIC between this and the other variables is calculated. Whether this MIC must be changed to 0 is determined by comparing this average MIC and the second threshold. Thereafter, weights are proportionally assigned according to the adjusted MIC and added as a multiplier to the loss function of each autoencoder in SAE. The regression portion of this model is a simple ANN. Finally, the pre-training and fine-tuning processes are carried out as usual.

The rest of this study is organized as follows. First, a brief review of SAE and MIC is introduced in Section 2. Section 3 provides the detailed descriptions of our proposed method. Then, the Boston house-price dataset is used as a benchmark dataset to verify the effectiveness of our proposed method in Section 4. Following this step, our method is validated on the predicted output of naphtha dry points during the practical industrial process in Section 5. Finally, Section 6 draws our conclusions.

2. RELATED WORK

The concepts and basic algorithms of autoencoder (AE), SAE, and MIC are briefly reviewed in this section to introduce our proposed method in detail.

2.1. AE and SAE

AE, which was first proposed by Rumelhart in 1986, is a typical unsupervised machine learning method [28]. Hinton and Salakhutdinov [11] improved the structure of the original AE and introduced the concept of DAE. In DAE, the unsupervised layer-by-layer greedy training algorithm is first applied to complete the pre-training of the hidden layer. Then, the back-propagation (BP) algorithm is used to optimize the system parameters of the whole network, which can greatly improve the learning ability of the network and effectively solve the problem of local optimum observed in neural networks. This training mechanism has been widely used in deep learning.

In general, a three-layer neural network constitutes a common AE. In contrast to ANN, AE aims to reproduce input signals. Figure 1 shows the schematic structure of AE. The unlabeled sample dataset is assumed to be $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n\}$, where $\boldsymbol{x}_i \in R_m$. m denotes the number of variables and n is the number of samples. The input and hidden layers form the encoding portion to extract the features of input signals. The extracted feature is denoted as $\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \dots, \boldsymbol{h}_n\}$, where $\boldsymbol{h}_i \in R_d$ and d indicates the number of nodes in the hidden layer. The encoding portion can be calculated as follows:

$$\boldsymbol{h} = s_1\left(\boldsymbol{W}_1\boldsymbol{x} + \boldsymbol{b}_1\right), \tag{1}$$

where $\boldsymbol{W}_1$ and $\boldsymbol{b}_1$ denote the weight matrix and bias vector of the encoding process, respectively, and. $s_1(\cdot)$ is the activation function of the network. In this study, the sigmoid function, which is determined with below equation, is selected as the activation function of our model:

$$f(t) = \frac{1}{(1 + e^{-t})}. \tag{2}$$

The transformation process from the hidden layer to the output layer is called the decoding portion and used to reconstruct the input signal. The reconstructed signal can be expressed as

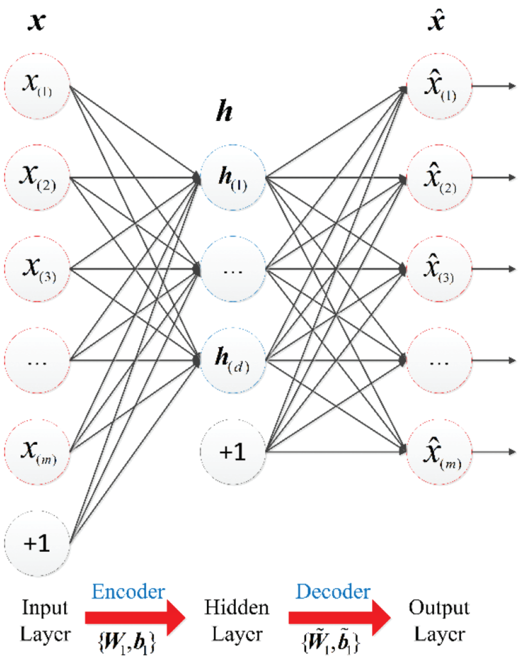

Figure 1 Schematic structure of autoencoder (AE).

$\hat{\boldsymbol{X}} = \{\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \ldots, \hat{\boldsymbol{x}}_n\}$, where $\hat{\boldsymbol{x}}_i \in R_m$. The decoding portion can be calculated as follows:

$$\hat{\boldsymbol{x}} = s_2\left(\tilde{\boldsymbol{W}}_1 \boldsymbol{h} + \tilde{\boldsymbol{b}}_1\right), \tag{3}$$

where $\tilde{\boldsymbol{W}}_1$ and $\tilde{\boldsymbol{b}}_1$ denote the weight matrix and bias vector of the decoding layer, respectively.

To make the input and output as equal as possible, AE uses the reconstructed error as a loss function, which can be described as:

$$L(x, \hat{x}) = \frac{1}{2n} \sum_{i=1}^{n} \sum_{j=1}^{m} \| x_{i(j)} - \hat{x}_{i(j)} \|^2. \tag{4}$$

SAE, which is an improved model of AE, was first developed by Bengio *et al.* [10] The training mechanism of SAE is divided into two steps: Layer-wise greedy unsupervised pre-training and supervised fine-tuning. First, SAE abandons the decoding portion of AE and only extracts the feature information obtained by AE. This feature information can represent the original information within the tolerance of the error, and, thus, the reconstructed process can be removed. Moreover, the extracted feature can be compressed once more by another AE. The AE in SAE is connected layer by layer. In other words, the feature achieved by the previous AE is the input signal of the next AE, which can be expressed as:

$$\boldsymbol{h}_i = f_{\theta_i}(\boldsymbol{h}_{i-1}), 2 \le i \le n_s, \tag{5}$$

where n_s denotes the number of AEs in the whole network and $f_{\theta_i}(\cdot)$ is the mapping function. The structure of SAE is shown in Figure 2. After pre-training, a multi-layer deep neural network with the same number of nodes as in the hidden layers in AEs is constructed. The weights and bias of each hidden layer $\{\boldsymbol{W}_i, \boldsymbol{b}_i\}$ are placed into the network as initial values, and the output of this network is set as labeled data and denoted as $\boldsymbol{Y} = \{\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n\}$, where $\boldsymbol{y}_i \in R_k$ and k is the number of output variables. Afterwards, the BP algorithm is run to fine-tune the parameters of this network and minimize the error between the output of the model and the labeled data. Figure 3 illustrates the training procedure of fine-tuning.

2.2. Maximal Information Coefficient

MIC, which was proposed by Reshef *et al.* [29], measures the dependence between variables. The calculation process of MIC aims to find the maximal grid resolution of all grids by meshing the scatterplot of two sets of variables. A labeled dataset is denoted as $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \in R_2$, where n is the number of samples. According to the dataset D, a scatterplot can be drawn on the coordinate system and divided into a grid G of $i \times j$. The corresponding probability distribution in each grid is $D|_G$. Then, the maximal mutual information (MI) value is calculated on all possible meshes G on the dataset D, which divides the X-axis into i grids and the Y-axis into j grids. The definition of maximal MI value can be expressed as follows:

$$I*(D, i, j) = \max I\left(D|_G\right), \tag{6}$$

where $I\left(D|_G\right)$ denotes the MI under the probability distribution $D|_G$. The elements of the ith row and jth column in the eigen matrix $M(D)$ on dataset D are denoted as:

$$M(D)_{i,j} = \frac{I*(D, i, j)}{\log \min(i, j)}. \tag{7}$$

Suppose the number of grids is less than $B(n)$; then, the MIC of dataset D can be defined as:

$$MIC(D) = \max\left\{M(D)_{i,j}\right\} \\ s.t \begin{cases} ij < B(n) \\ B(n) = n^{0.6} \end{cases}. \tag{8}$$

MIC highlights an excellent characteristic, that is, as the number of samples increases, the MIC score approaches 1 with probability 1, regardless whether the relationship between variables is a never-constant noiseless functional relationship or a larger class of noiseless relationships. Moreover, the MIC approaches 0 for statistically independent variables. The use of MIC to mathematically measure the association between variables causes the MIC score to be concentrated at both ends of the [0, 1] range, which improves the ability of the model to distinguish the relationship of data because the association between variables is a binary relationship [30]. Consequently, compared with other traditional correlation coefficients, such as Pearson, Spearman, and MI, MIC can better identify different relationship types [29].

3. METHODOLOGY

The layer-wise greedy pre-training and fine-tuning process can effectively solve the problems of neural networks easily falling into local optimum and gradient diffusion. Despite the powerful ability of SAE to extract the features of the original input data, irrelevant information in the output is still added to the network during

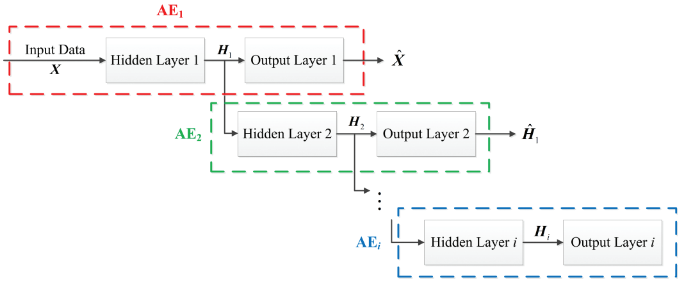

Figure 2 Structure of stacked autoencoder (SAE).

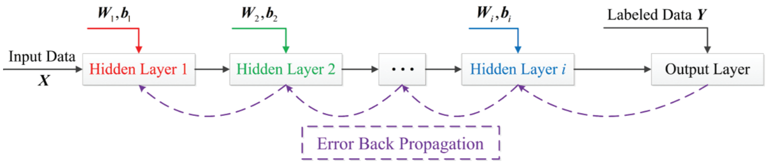

Figure 3 Training procedure of the fine-tuning.

the pre-training process if the input data are not judged by importance. In addition, the model developed by Yuan *et al.* [27] only considered the linear relationship between variables. In practical industrial processes, the mathematical relationship between variables is highly nonlinear and cannot be identified using Pearson or other traditional statistical correlation coefficients. MIC presents the advantage of describing both linear and nonlinear relationships. Simultaneously, although most independent variables have a strong mathematical functional relationship to the dependent variables, still there are some relatively minor independent variables that have a slight impact on the output signals. Those variables should be filtered to eliminate their impact on reducing the prediction accuracy.

As a result, we combined MIC with SAE to build a novel soft sensor modeling method. Assume the existence of a labeled dataset $\boldsymbol{T} = \{\boldsymbol{X}, \boldsymbol{y}\}$, where $\boldsymbol{X} \in \boldsymbol{R}^{n\times m}$, $\boldsymbol{y} \in \boldsymbol{R}^{n\times 1}$, n is the number of samples, and m expresses the number of measurable variables. Before pretraining, the MIC index, denoted as $MIC_{(i)}$ between $\boldsymbol{x}_{(i)}$ and $\boldsymbol{y}$ is first calculated, where $i \in [1, m]$. That is to say, the correlation between the ith independent variable and the labeled data is determined by the absolute value of $MIC_{(i)}$. Second, if $MIC_{(i)}$ is greater than a relatively large threshold denoted as T_1, then ith independent variable exerts a great influence on the dependent variable. Conversely, the sum of MICs between the ith input variable and each of the other input variables is obtained, and the average MIC is expressed as:

$$avgMIC_{(i)} = \sum_{j=1}^{m} xMIC_{(ij)} \Big/ n-1, j \neq i, \tag{9}$$

where $xMIC_{(ij)}$ expresses the MIC between $\boldsymbol{x}_{(i)}$ and $\boldsymbol{x}_{(j)}$. Next, $avgMIC_{(i)}$ and a relatively large threshold denoted as T_2 are compared. MIC_i is set to 0 when $avgMIC_{(i)}$ is greater than T_2, indicating that the ith input variable has a slight impact on the output. The information contained by $\boldsymbol{x}_{(i)}$ can also be replaced by other input variables. Hence, the weight of this variable is set to 0 during the training process. By contrast, if $avgMIC_{(i)}$ is less that T_2, $MIC_{(i)}$ remains unchanged, thereby implying that, although the ith input variable exerts a minor effect on the output, the information carried by $\boldsymbol{x}_{(i)}$ is unique, and the variable is irreplaceable. Then, the weight of each input variable is set as:

$$\omega_{(i)} = MIC_{(i)} \Big/ \sum_{i=1}^{m} MIC_{(i)}. \tag{10}$$

Finally, the weights are added to the loss function as a multiplier to train the network. Each AE in SAE will conduct this mechanism during pre-training and fine-tuning. The loss function is modified from Eq. (4) to:

$$L(x, \hat{x}) = \frac{1}{2n} \sum_{i=1}^{n} \sum_{j=1}^{m} \omega_{(j)} \|x_{i(j)} - \hat{x}_{i(j)}\|^2. \tag{11}$$

Figure 4 presents the MICW-SAE structure. MICs naturally represent the correlation between the feature extracted from the previous AE and the labeled data since the second AE when pre-training, which means this mechanism of weighted MICs only targets the input signals of the current AE and the labeled data. By executing this step before pre-training, the primary variables directly affecting the output are gradually strengthened by the training progress due to the presence of large weights; secondary variables are gradually lost. Ultimately, the features abstracted by SAE for regression contain rich and valuable information representing the original input signal. Similarly, the variables must be weighted before fine-tuning.

To conclude, the MICW-SAE algorithm is conducted in a stepwise manner:

(a) The thresholds are denoted as T_1, T_2, and i is set to 1.

(b) The MIC of the ith independent and dependent variables is calculated and denoted as $MIC_{(i)}$.

(c) If $MIC_{(i)} > T_1$, then go to step (e); otherwise, the average MIC of the ith independent variable and other independent variables is calculated and denoted as $avgMIC_{(i)}$.

(d) If $avgMIC_{(i)} > T_2$, then $MIC_{(i)} = 0$.

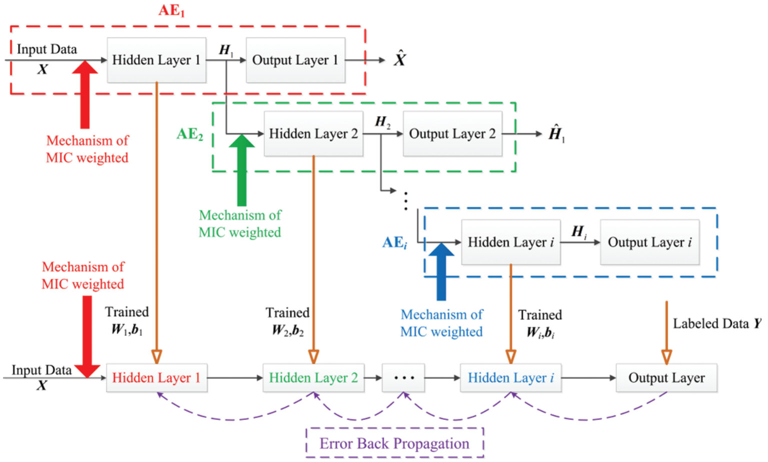

Figure 4 Structure of maximal information coefficient-stacked autoencoder (MICW-SAE).

(e) $i = i + 1$. If $i \leq m$, then go to step (b); otherwise, the weights of all variables are calculated according to Eq. (10).

(f) The labeled dataset is normalized to $[0, 1]$. The standard dataset is divided into training $\boldsymbol{T}_{train}\left\{\boldsymbol{X}_{train}, \boldsymbol{y}_{train}\right\}$, valid $\boldsymbol{T}_{valid} = \left\{\boldsymbol{X}_{valid}, \boldsymbol{y}_{valid}\right\}$, and test $\boldsymbol{T}_{test} = \left\{\boldsymbol{X}_{test}, \boldsymbol{y}_{test}\right\}$ datasets. Early stopping is applied to stop the training process on AE when the error of the valid dataset no longer decreases.

(g) The aforementioned steps are carried out during layer-wise greedy pre-training and fine-tuning. Eq. (11) is used as the new loss function of all AEs in SAE and the whole fine-tuning network. Thereafter, the performance of soft sensor model is tested.

The flow chart in Figure 5 demonstrates our proposed modeling algorithm intuitively. The regression portion of MICW-SAE is a two-layer simple neural network. Two indicators, namely, the root mean squared error (RMSE) and the regression index R^2, are applied to evaluate the model. The model with the lowest RMSE and the R^2 closest to 1 is considered to provide the best effect on regression estimation. RMSE can be calculated as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}\left(\hat{y}_i - y_i\right)}{n}}, \tag{12}$$

where $\hat{y}_i$ is the predicted value of the model. R^2 is calculated as:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i, \tag{13}$$

$$R^2 = \frac{SSR}{SST} = \frac{\sum_{i=1}^{n}\left(\hat{y}_i - \bar{y}\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2}. \tag{14}$$

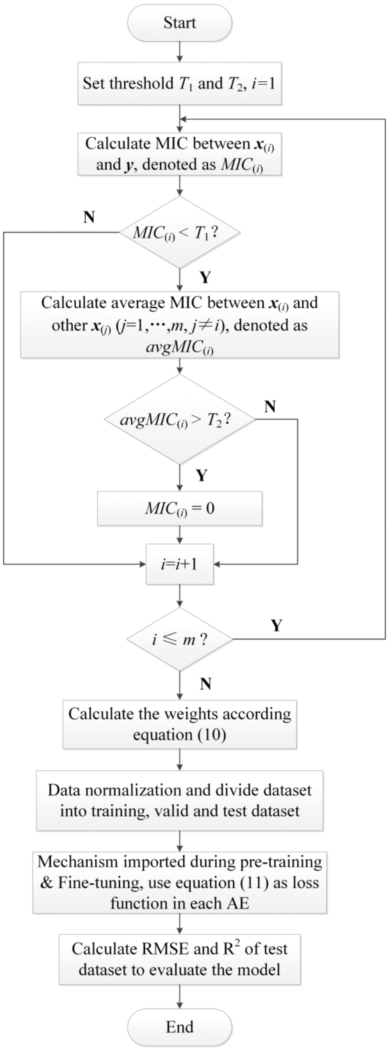
Start
Set threshold T_1 and T_2, i=1
Calculate MIC between $\boldsymbol{x}_{(i)}$ and $\boldsymbol{y}$, denoted as $MIC_{(i)}$
N
$MIC_{(i)} < T_1$?
Y
Calculate average MIC between $\boldsymbol{x}_{(i)}$ and other $\boldsymbol{x}_{(j)}$ (j=1,⋯,m, j≠i), denoted as $avgMIC_{(i)}$
$avgMIC_{(i)} > T_2$?
N
Y
$MIC_{(i)} = 0$
i=i+1
$i \leqslant m$?
Y
N
Calculate the weights according equation (10)
Data normalization and divide dataset into training, valid and test dataset
Mechanism imported during pre-training & Fine-tuning, use equation (11) as loss function in each AE
Calculate RMSE and R^2 of test dataset to evaluate the model
End

Figure 5 Flow chart of maximal information coefficient-stacked autoencoder (MICW-SAE).

4. EXPERIMENT AND RESULTS OF THE BOSTON HOUSE-PRICE DATASET

In this example, we take the Boston house-price dataset, which is available online in the sklearn library, as a benchmark dataset to verify the effectiveness of MICW-SAE. Benchmark data sets are generally used to illustrate the performance of the algorithm. The Boston house-price dataset contains 13 variables affecting the output, namely, average house-prices in Boston [31]. A total of 506 samples are included in this dataset. After random disruption, the numbers of training, valid, and test datasets are 323, 81, and 102, respectively.

The program is written in Python and run on a computer with an Intel Core i5-4590 (3.3 GHz) processor and 4 GB storage. MICW-SAE consists of three AEs, the hidden nodes of which number 7, 5, 3. The thresholds T_1, T_2 are set as 0.3 and 0.2 by trial-and-error method. Since the variables in the open machine learning database are all considerably relevant to the output, these two thresholds are relatively conservative. The weighted MIC mechanism forces the weight of one input variable to 0. The output of MICW-SAE is depicted in Figure 6, which shows that the predicted value of our model is very close to the true value. After the original input is enhanced and eliminated by weights calculated by MIC, nearly all sample points regress well to the true values. The number of sample points with large errors is also extremely small. This conclusion can be clearly drawn from Figure 7, which shows the absolute error of prediction by MICW-SAE. Despite the existence of one point with relatively great error, the predicted outputs of MICW-SAE on Boston house-price dataset are mostly fit well to their true values. Figure 8 proves whether the prediction error follows a Gaussian distribution. As shown in Figure 8, there are some error distribution intervals that do not strictly follow the red diagonal line indicates a normal distribution. However, most testing prediction errors lie roughly around the red diagonal line, which represents that our model prediction results basically match the local house-price pricing distribution. Thus, Figures 7 and 8 both verify the effectiveness and rationality of our proposed method.

For comparison, four modeling methods based on machine learning, including partial least square (PLS), SVR, multi-layer ANN (M-ANN), and eXtreme Gradient Boosting (XGBoost) proposed by Chen and Guestrin [32] are applied to the same dataset. Moreover, three deep learning methods based on SAE (without weighted MICs), VW-SAE [27] as mentioned above and SAE-MIPCA-NN proposed by Wang and Yan [33] are also imported as comparative experiments. XGBoost, which is an extension of the Gradient Boosting Decision Tree (GDBT), has shown its good regression ability in various data mining competitions. The parameters are set as follows in this experiment: the max depth is 5, the learning rate is 0.1 and the number of trees is 20. SAE-MIPCA-NN uses principle component analysis (PCA) to extract the raw data and features weighted with MI above the threshold from each AE, and then regressed by ANN. The parameter setting method of this parallel experiment is the same as that of Wang and Yan [33]. As a result, the number of extracted components is eight. Additionally, because SVR does not require a valid dataset, the valid and training datasets are merged to train the SVR model in the experiments. The kernel function of SVR is the Gaussian radial basis function ($\sigma = 1$). The optimal component number chosen in PLS is 10 after 10-fold cross validation. The traditional SAE has the same hidden structure as MICW-SAE. The neuron numbers of M-ANN are also set to 13, 7, 5, 3, and 1. The same structures are also used in SAE, VW-SAE, and SAE-MIPCA-NN. The selection of batch size is consistent with that of MICW-SAE due to the random batch gradient descent algorithm in the neural network and AE on the computer.

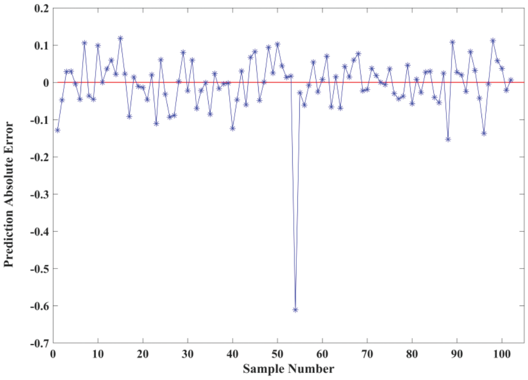

Figure 7 Plot of the prediction absolute errors by maximal information coefficient-stacked autoencoder (MICW-SAE) (Boston Dataset).

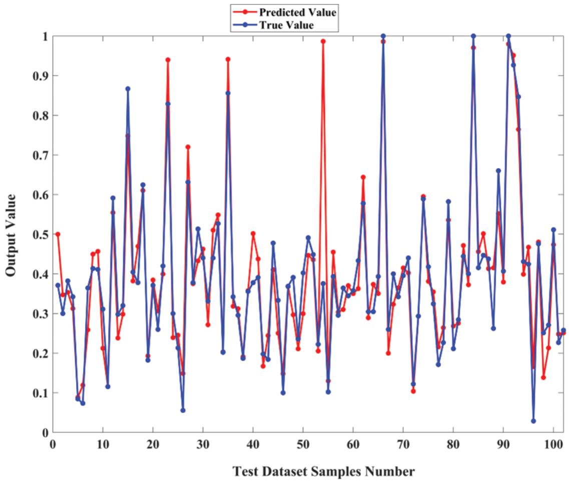

Figure 6 Results of estimation of the Boston house-price dataset by maximal information coefficient-stacked autoencoder (MICW-SAE).

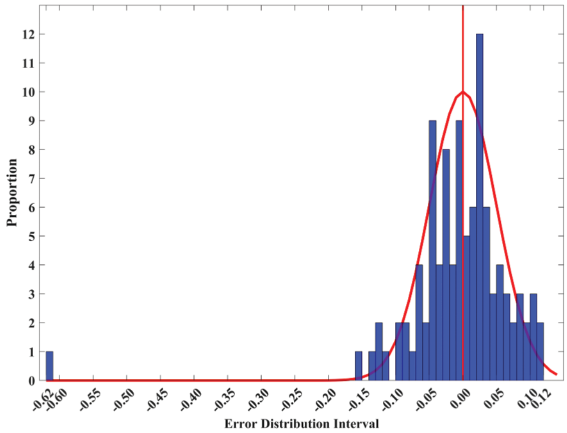

Figure 8 Histogram of the prediction absolute errors by maximal information coefficient-stacked autoencoder (MICW-SAE).

Table 1 shows the RMSE and R^2 results of these comparative modeling methods and the best results are shown in **bold**. In comparison with the other models, MICW-SAE can predict housing prices in Boston more accurately with the lowest prediction error and the highest R^2. These findings prove the effectiveness of our proposed method.

5. APPLICATION ON ATMOSPHERIC COLUMN

In this section, MICW-SAE is validated on a practical atmospheric column process. A description of this industrial process is first provided in Section 5.1. Next, an experiment to predict the naphtha dry point temperature is conducted, and seven regression modeling methods are used for comparison. Finally, an analysis and a discussion of the experiment results on the changing in the thresholds, the loss function value and reconstruction error are given.

5.1. Description of Atmospheric Column

Figure 9 shows a simple flow chart of an atmospheric column unit. The atmospheric column has 60 column plates. After heating by an atmospheric pressure kiln, the crude oil enters the fractionation distillation tower, and the top of the tower is driven into a cold reflux, which controls the temperature to approximately 120 °C. The temperature gradually increases from the top of the tower to the feeding section. Given different boiling point ranges, the gasoline vapor and steam are distilled from the top of the tower. Kerosene, light diesel oil, and heavy diesel oil are separately distilled from the No. 1, No. 2, and No. 3 side-streams, respectively [34]. The product at the top of the tower enters the reflux accumulator through the heat exchanger and condenser. Then, the product is pumped out by an atmospheric overhead pump. Part of the product is refluxed at the top of the tower, and another part is collected as the final product, naphtha. Naphtha is the distillate product obtained after the first step of crude oil distillation. The quality index of naphtha, as a raw material of the subsequent product, is an important consideration. However, the measurement accuracy of the naphtha dry point temperature is not ideal due to the large number of side-streams and complicated structure of the system, the diverse crude oil composition, and the high nonlinearity between variables. Thus, building a prediction model that can reliably estimate the naphtha dry point temperature and quality is essential. MICW-SAE is tested on this complex industrial process.

5.2. Experiment on the Naphtha Dry Point Dataset

A total of 150 samples in the naphtha dry point dataset are collected from practical industrial processes. Table 2 provides a detailed description of each variable in this dataset. All samples are randomly disrupted, similar to the previous experiment. After normalization, the test dataset is composed of 30 samples; 120 samples are used as the training dataset, from which 24 samples are extracted as the valid dataset to monitor the training process. The batch size is set as 24 samples through the trial-and-error method. The thresholds T_1, T_2 are set as 0.2 and 0.26, respectively. Figure 10 shows that the MICs of input variables 3, 4, 7, 8, 9, 10, and 14 are below the threshold T_1, which means these variables contain relatively less useful information than other variables. Figure 11 illustrates that, under the intervention of the threshold T_2, the temperature and pressure at the top of the atmospheric column hardly affect the current dry point temperature and can be represented by other inputs. The final weights of all input variables are given in Table 3. The numbers of the three nodes in the hidden layer of AEs in SAE

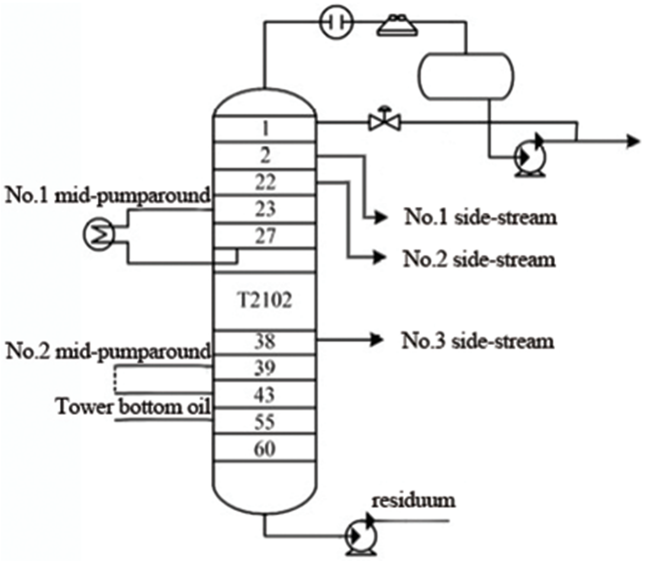

Figure 9 Flow chart of the atmospheric column.

Table 1 RMSE and R^2 results of the Boston house-price test dataset.

Model	MICW-SAE	PLS	SVR	XGBoost	M-ANN	SAE	VW-SAE	SAE-MIPCA-NN
RMSE	**0.083223**	0.129535	0.094930	0.095231	0.114164	0.098563	0.098805	0.095944
R^2	**0.917145**	0.768137	0.885209	0.889115	0.824841	0.871915	0.895120	0.890171

Table 2 Description of each variable in the naphtha dry point dataset.

No.	Variable Description	No.	Variable Description
1	Total exit temperature	9	No.2 side-stream
2	Total flow	10	No.3 side-stream
3	Temperature at tower top	11	Energy brought by circulation at top tower
4	Pressure at tower top	12	Energy of No.1 mid-pumparound
5	Pumparound Energy at tower top	13	Energy of No.2 mid-pumparound
6	Product flow at tower top	14	Vaporization temperature
7	Flow of light diesel oil	15	Stripping steam flow
8	No.1 side-stream	16	Previous value of naphtha dry point

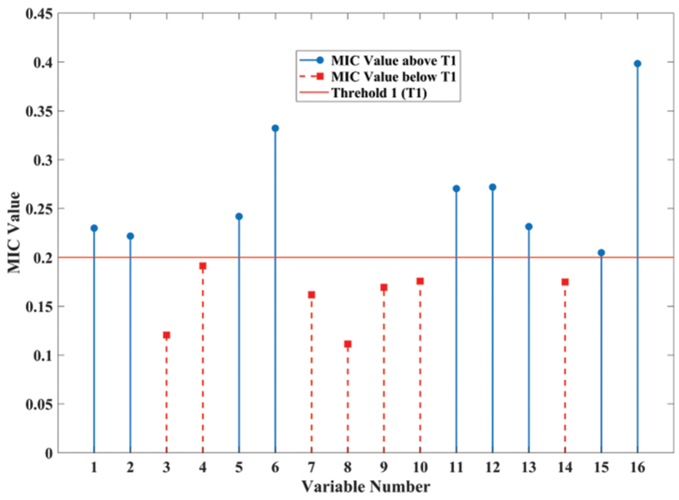

Figure 10 Maximal information coefficients (MICs) between input variables and output.

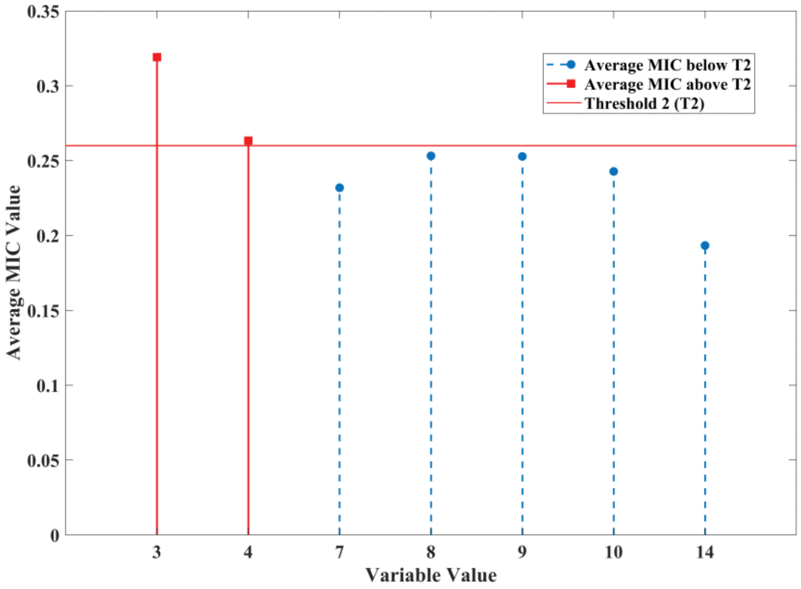

Figure 11 Average maximal information coefficients (MICs) of $\boldsymbol{x}_{(i)}$ below T_1.

Table 3 Weights of each variable in the naphtha dry point dataset.

Variable No.	Weight	Variable No.	Weight	Variable No.	Weight	Variable No.	Weight
$x_{(1)}$	0.071943	$x_{(5)}$	0.075694	$x_{(9)}$	0.053006	$x_{(13)}$	0.072433
$x_{(2)}$	0.069409	$x_{(6)}$	0.103963	$x_{(10)}$	0.054964	$x_{(14)}$	0.054684
$x_{(3)}$	0	$x_{(7)}$	0.050637	$x_{(11)}$	0.084609	$x_{(15)}$	0.064075
$x_{(4)}$	0	$x_{(8)}$	0.034854	$x_{(12)}$	0.085078	$x_{(16)}$	0.124650

are 9, 6, and 4. Figure 12 gives the absolute error trend along with the test sample number in the naphtha dry point dataset by our proposed method. Since the dataset is acquired from the industrial processes, which contains large noises, most absolute errors of the prediction results are located between $[-0.1, 0.1]$.

The same training, valid, and test datasets are used with the seven other models, and the rules for setting the training dataset and parameters in these models are identical to those in the previous experiment. The radial basis function is still applied to SVR ($\sigma = 0.1$). The number of components is 7 in PLS. In XGBoost, the max depth is 4 and the number of trees is 30. The learning rate is set to 0.1, which the same as the previous experiment. The structures of SAE, VW-SAE, SAE-MIPCA-NN, and M-ANN are also consistent with MICW-SAE, and the batch size is 24 samples. In SAE-MIPCA-NN, nine eligible components are selected. The results of these models are provided in Table 4 and the optimal results are written in **bold**.

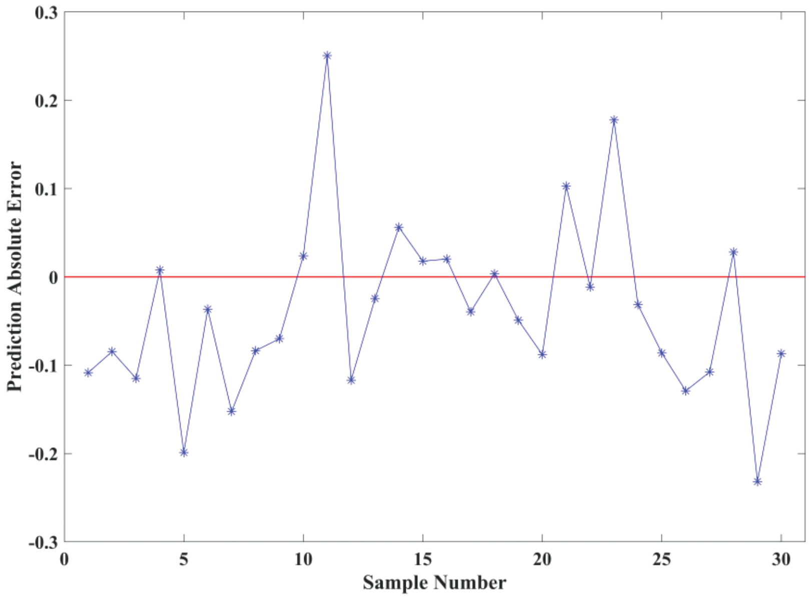

Figure 12 Plot of the prediction absolute errors by maximal information coefficient-stacked autoencoder (MICW-SAE) (Naphtha Dry Point Dataset).

Table 4 RMSE, R^2, and MARE results of the naphtha dry point test dataset.

Model	RMSE	R^2	MARE	Performance Increase (MARE)
MICW-SAE	**0.161319**	**0.560128**	**0.043065**	/
PLS	0.167904	0.490067	0.047862	10.02%
SVR	0.174369	0.466648	0.047105	8.58%
XGBoost	0.171400	0.491696	0.053284	19.18%
M-ANN	0.173193	0.217630	0.052893	18.58%
SAE	0.177296	0.451228	0.052668	18.23%
VW-SAE	0.165601	0.514201	0.045435	5.22%
SAE-MIPCA-NN	0.164790	0.523348	0.046048	6.48%

5.3. Analysis

To compare the performance of models comprehensively, the maximal absolute relative error (MARE) indicator is imported to assess the predicted point with the maximum error of model. MARE is defined as follows:

$$MARE = \max \frac{|\hat{y}_i - y_i|}{y_i}, i = 1, ..., n \tag{15}$$

where $\hat{y}_i$ and y_i demonstrate the predicted output of model and the true value of test dataset, respectively, n is the number of the sample.

According to Table 4, the prediction accuracies of the tested model can be ranked in the order of MICW-SAE > SAE-MIPCA-NN > VW-SAE > XGBoost > PLS > SVR > SAE > M-ANN. In addition, MICW-SAE can increase R^2 by over than 20% compared with the SVR model, which has the best performance among the eight parallel modeling methods. As for MARE of XGBoost, MICW-SAE can improve the worst prediction point error up to 19.18%. Notice that SAE model predicts slightly worse than the SVR. However, the MARE of SAE is greater than that of SVR. This expresses that while the prediction performance of SAE is similar to that of SVR, the error at a certain point regressed by SAE is much larger. The same situation also occurs on PLS and XGBoost. Besides, among the four machine learning algorithms, the PLS performance has a large ranking improvement relative to its performance in the Boston house-price dataset, which reflects that the prediction accuracy varies with different datasets to some extent. On the contrary, MICW-SAE performs stably and exhibits its powerful nonlinear fitting ability with high prediction accuracy in the two datasets.

The original 16-dimensional data are all directed toward the other seven soft sensor models for training, which results in training of negligible information. By contrast, after weight assignment and extraction by MIC, the weights of only two input variables are set to 0 and 14 variables are retained. Useless information is filtered out by the proposed mechanism. During the training process, valuable information is further magnified layer by layer because of the existence of weights. In contrast, these two useless features are trained by VW-SAE, which leads to those invalid information being considered during the training process. Thus, the error of the test dataset by MICW-SAE is relatively lower.

Although SAE-MIPCA-NN performs better than VW-SAE in the second datasets, the former method has a larger prediction error for some sample points than the latter, which reflects on the lower MARE of VW-SAE. The mechanism of SAE-MIPCA-NN is to preserve the original data and the features extracted by each AE. Then, PCA is imported to select the components from all the weighted features calculated by MI above the threshold. If the number of original input variables or AEs in SAE becomes larger, the workload will be greatly increased. The calculation time and calculation cost will be increased as well. To conclude, VW-SAE only enhances the useful information related to the labeled data, but does not consider the interference of the useless one. MICW-SAE and SAE-MIPCA-NN both aim for the extraction of useful information but MICW-SAE proposed in this paper is not only simpler but also have a higher precision.

5.4. Discussion

During the experiment process, the thresholds T_1 and T_2 are chosen by trial-and-error methods. To demonstrate the impact on the accuracy of MICW-SAE when these thresholds vary, we did a series of experiments shown in Tables 5 and 6. The last column in Tables 5 and 6 shows the number of variables will participate in the training under the mechanism we proposed. The best choices of T_1 and T_2 are written in **bold**. We find that different thresholds bring various effects on the final prediction accuracy of the model and the values shown in **bold** are also both optimal in their respective datasets. From Tables 5 and 6, we can conclude that when T_2 remains unchanged, more input variables need to be further calculated and fewer variables are considered to contain the information directly related to the labeled data if T_1 increases. In this case, only six and nine variables can be chosen to be trained as shown in Tables 5 and 6. The significant reduction in the number of input variables leads to a large loss of information when training so that the prediction effect becomes worse. Conversely, the metrics of mathematical relationship between the input and output variables becomes weak, which means that nearly all variables are considered useful and redundant information cannot be identified well in the case of a decrease in T_1. Under this condition, all the input variables in the two datasets are involved in the training process. The prediction accuracy of the MICW-SAE is close to that of the VW-SAE compared Tables 1 and 4 with Tables 5 and 6. Notice that in the naphtha dry point dataset, when $T_1 = 0.1, T_2 = 0.26$, both VW-SAE and MICW-SAE use all input variables for training. However, the prediction effect of MICW-SAE is better than VW-SAE. This is mainly because MIC can measure the high nonlinearity relationship between variables more clearly than Pearson coefficient in industrial processes.

On the other hand, when T_1 is unchanged, after all the input variables that has little relationship with the output are found, the increase of T_2 enforces almost all the information contained in these selected variables considered to be unique. Thus, some variables that contain useless information are participated in the training resulting in a decrease in the prediction precision. On the contrary, when T_2 is reduced, the information contained in the selected variable is always considered replaceable by other variables. Therefore, part of these minor variables, the information of which may be unique to the output, are also deleted. The elimination of these important information will also lead to a decline in prediction accuracy. Currently, there is no strong justification or index for these parameters setting. Trial-and-error method is the only feasible way to solve this problem at present. Consequently, how to find a suitable indicator to adjust the thresholds will be one of our research priorities in the future.

Furthermore, we investigated the reconstruction error of the different neural network models in the same batch size case. Figure 13 demonstrates the learning error of MICW-SAE, VW-SAE and SAE during fine-tuning within 200 epochs. SAE produces the highest learning errors among three methods surveyed at first because its loss function is without weights. Leading-in of the weighted process allows the variables in the VW-SAE and MICW-SAE models to be multiplied by $[0, 1]$ so that their loss functions decrease. We conducted another experiment to explain why the reconstruction error of SAE is lower than those of MICW-SAE and VW-SAE during later stage. The first AEs in both models are selected and trained to record their respective reconstruction errors and loss functions values. All parameters among the three models are identical. Table 7 illustrates that MICW-SAE has the second lowest loss function but the highest reconstruction error mainly because the weights obtained by MIC range from 0 to 1. Since the weights of the two independent variables are adjusted to 0 in MICW-SAE, the weights of the other variables are increased by contrast to VW-SAE, which contribute to a higher loss function value by MICW-SAE. When these weights are added to the loss function as multipliers, the loss function is bound to decrease and be lower than that of SAE. As for the larger reconstruction error, both the goal of MICW-SAE and VW-SAE network optimization are a new loss function multiplied by weights on the basis of the original reconstruction error.

Table 5 RMSE and R^2 results of the Boston house-price dataset varies with T_1 and T_2.

	RMSE	R^2	Number of Train/Total Variables
$T_1 = 0.4, T_2 = 0.2$	0.115486	0.822850	6 / 13
$T_1 = 0.2, T_2 = 0.2$	0.095972	0.881375	13 / 13
$T_1 = 0.3, T_2 = 0.2$	**0.083223**	**0.917145**	12 / 13
$T_1 = 0.3, T_2 = 0.3$	0.096463	0.879099	13 / 13
$T_1 = 0.3, T_2 = 0.1$	0.088135	0.901740	11 / 13

Table 6 RMSE and R^2 results of the naphtha dry point dataset varies with T_1 and T_2.

	RMSE	R^2	Number of Train/Total Variables
$T_1 = 0.3, T_2 = 0.26$	0.188237	0.442012	9 / 16
$T_1 = 0.1, T_2 = 0.26$	0.165230	0.517732	16 / 16
$T_1 = 0.2, T_2 = 0.26$	**0.161319**	**0.560128**	14 / 16
$T_1 = 0.2, T_2 = 0.27$	0.165026	0.528801	15 / 16
$T_1 = 0.2, T_2 = 0.25$	0.174754	0.455374	12 / 16

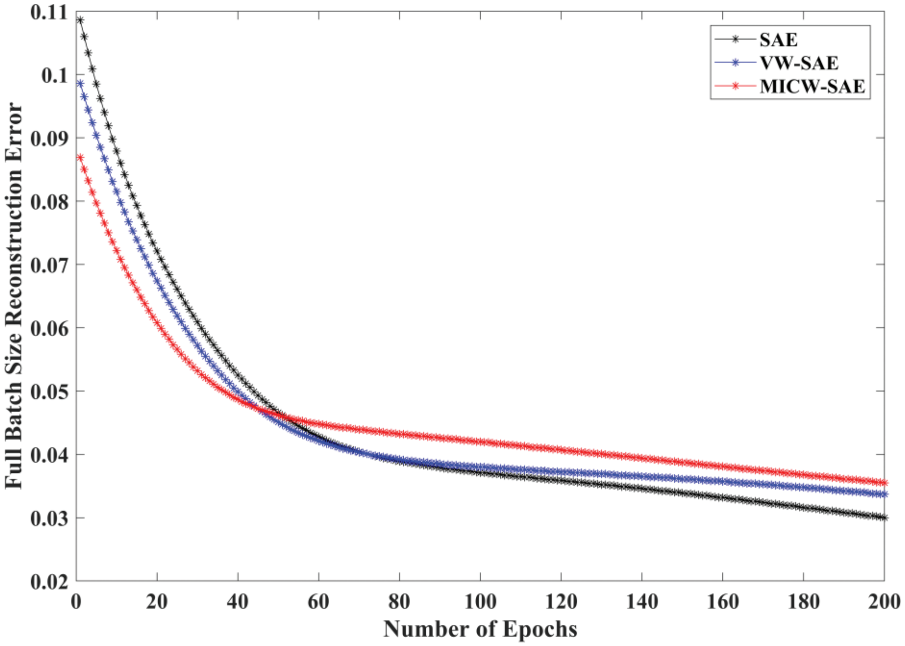

Figure 13 Loss function values of the three models.

Table 7 Reconstruction errors of the first AE in MICW-SAE and SAE.

	SAE	VW-SAE	MICW-SAE
Loss function value	0.004084	0.000171	0.000194
Reconstruction error	0.004084	0.004865	0.010889

Compared with the loss function directly used in SAE as the optimization target, the reconstructed errors of MICW-SAE and VW-SAE are thus slightly higher. However, in the test and prediction process, the reconstruction error is pointless. The trained features and weights of hidden layers by AEs are actually used, and the prediction error is more concerned. Taken together, the test results of the two datasets prove the superiority of MICW-SAE over other modeling algorithms.

6. CONCLUSION

A novel soft sensor modeling method called MICW-SAE is developed in this article. Before pre-training and fine-tuning, the relationship between input and output variables is first measured by MIC and compared with the threshold T_1. Variables with high MICs are restored, whereas those lower than the threshold T_1 are further calculated. The average MICs of other variables are also obtained to determine whether these variables contain unique and irreplaceable information and whether their MIC should be set to 0 compared with the threshold T_2. Then, weights, which are placed into the loss function as multipliers of the input, are assigned according to the scale. Through this mechanism, useful features are gradually amplified and useless features with information already contained by other input variables are discarded along with the training process. Finally, the features most relevant to the output are extracted by the trained model. Redundant information is not trained, and prediction errors decrease. The public machine learning dataset used in this work proves the validity of our proposed model, and predictions of the naphtha dry point temperature in an atmospheric column confirm that MICW-SAE outperforms among four machine learning and three deep learning modeling methods in terms of accuracy. For future work, the proposed MICW-SAE cannot be limited to the industrial soft sensor field but can be extended to the application of pattern recognition or fault detection.

CONFLICT OF INTEREST

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

We confirm that the manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us.

We confirm that we have given due consideration to the protection of intellectual property associated with this work and that there are no impediments to publication, including the timing of publication, with respect to intellectual property. In so doing we confirm that we have followed the regulations of our institutions concerning intellectual property.

We understand that the Corresponding Author is the sole contact for the Editorial process (including Editorial Manager and direct communications with the office). He/she is responsible for communicating with the other authors about progress, submissions of revisions and final approval of proofs. We confirm that we have provided a current, correct email address which is accessible by the Corresponding Author and which has been configured to accept email from xfyan@ecust.edu.cn.

AUTHORS' CONTRIBUTIONS

We promise that all listed authors, named Yanzhen Wang and Xuefeng Yan, made substantial contributions to the conception or design of the work; or the acquisition, analysis, or interpretation of data for the work. And both listed authors drafted the work or revised it critically for important intellectual content and gave final approval of the version to be published. Wang and Yan agreed to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

The specific contribution that each Author made to the article is listed as follows:

Yanzhen Wang: literature search, study design, manuscript writing, data analysis.

Xuefeng Yan: guidance on experimental design, guidance on experimental ideas, data provider, manuscript revision, paper submission.

Funding Statement

The authors are grateful for the support of National Natural Science Foundation of China (21878081), Fundamental Research Funds for the Central Universities under Grant of China (222201717006), and the Program of Introducing Talents of Discipline to Universities (the 111 Project) under Grant B17017.

We also declare that the funding source(s) had no involvement in the work beyond providing a financial contribution.

ACKNOWLEDGMENTS

The authors are grateful for the support of National Natural Science Foundation of China (21878081), Fundamental Research Funds for the Central Universities under Grant of China (222201917006), and the Program of Introducing Talents of Discipline to Universities (the 111 Project) under Grant B17017.

REFERENCES

[1] T. Chai, S. Li, H. Wang, Modeling and control for complex industrial processes in networked information, Acta. Automatica. Sinica. 39 (2013), 469–470.

[2] S. Khatibisepehr, B. Huang, S. Khare, Design of inferential sensors in the process industry: a review of Bayesian methods, J. Process Contr. 23 (2013), 1575–1596.

[3] J. Yu, Soft sensing technology and its application, *Process Automation Instrumentation.* 29 (2008), 1–7.

[4] P. Kadlec, B. Gabrys, S. Strandt, Data-driven soft sensors in the process industry, Comput. Chem. Eng. 33 (2009), 795–814.

[5] A. Wibowo, M.I. Desa, Kernel based regression and genetic algorithms for estimating cutting conditions of surface roughness in end milling machining process, Expert Syst. Appl. 39 (2012), 11634–11641.

[6] A.L. Korzenowski, M.J. Anzanello, M.S. Portugal, C. ten Caten, Predictive models with endogenous variables for quality control in customized scenarios affected by multiple setups, Comput. Ind. Eng. 65 (2013), 729–736.

[7] C.-L. Chan, C.-L. Chen, H.-W. Ting, D.-V. Phan, An agile mortality prediction model: hybrid logarithm least-squares support vector regression with cautious random particle swarm optimization, Int. J. Comput. Int. Sys. 11 (2018), 873–881.

[8] M. Hassan, M. Hamada, Genetic algorithm approaches for improving prediction accuracy of multi-criteria recommender systems, Int. J. Comput. Intell. Syst. 11 (2018), 146–162.

[9] M. Rohani, H. Jazayeri-Rad, R.M. Behbahani, Continuous prediction of the gas dew point temperature for the prevention the foaming phenomenon in acid gas removal units using artificial intelligence models, Int. J. Comput. Intell. Syst. 10 (2017), 165–175.

[10] Y. Bengio, P. Lamblin, D. Popovici, H. Larochelle, Greedy layer-wise training of deep networks, Adv. Neural Inf. Process. 19 (2007), 153–160. http://papers.nips.cc/paper/3048-greedy-layer-wise-training-of-deep-networks.pdf

[11] G.E. Hinton, P.R. Salakhutdinov, Reducing the dimensionality of data with neural network, Science. 313 (2006), 504–507.

[12] R. Zhang, W. Li, T. Mo, Review of deep learning, Inf. Control. 47 (2018), 385–397.

[13] Y. Fu, Y. Zhang, Y. Gao, H. Gao, T. Mao, H. Zhou, D. Li, Machining vibration states monitoring based on image representation using convolutional neural networks, Eng. Appl. Artif. Intell. 65 (2017), 240–251.

[14] A.V. Savchenko, N.S. Belova, Unconstrained face identification using maximum likelihood of distances between deep off-the-shelf features, Expert Syst. Appl. 108 (2018), 170–182.

[15] Z. Zhao, A. Kumar, Improving periocular recognition by explicit attention to critical regions in deep neural network, IEEE Trans. Inf. Forensics Security. 13 (2018), 2937–2952.

[16] M. Sajid, I.A. Taj, U.I. Bajwa, N.I. Ratyal, Facial asymmetry-based age group estimation: role in recognizing age-separated face images, J. Forensic Sci. 63 (2018), 1727–1749.

[17] D.T. Grozdic, S.T. Jovicic, M. Subotic, Whispered speech recognition using deep denoising autoencoder, Eng. Appl. Artif. Intell. 59 (2017), 15–22.

[18] R.K. Vuddagiri, H.K. Vydana, A.K. Vuppala, Curriculum learning based approach for noise robust language identification using DNN with attention, Expert Syst. Appl. 110 (2018), 290–297.

[19] J. Yun, J. Jiang, R. Xia, Global inference for aspect and opinion terms co-extraction based on multi-task neural networks, IEEE-ACM Trans. Audio Speech. 27 (2019), 168–177.

[20] J. Liu, Y. An, R. Dou, H. Ji, Dynamic deep learning algorithm based on incremental compensation for fault diagnosis model, Int. J. Comput. Intell. Syst. 11 (2018), 846–860.

[21] L. Jiang, Z. Ge, Z. Song, Semi-supervised fault classification based on dynamic Sparse Stacked auto-encoders model, Chemom. Intell. Lab. Syst. 168 (2017), 72–83.

[22] J. Sun, C. Yan, J. Wen, Intelligent bearing fault diagnosis method combining compressed data acquisition on deep learning, IEEE Trans. Instrum. Meas. 67 (2018), 185–195.

[23] S. Yan, X. Yan, Using labeled autoencoder to supervise neural network combined with k-Nearest neighbor for visual industrial process monitoring, Ind. Eng. Chem. Res. 58 (2019), 9952–9958.

[24] S. Zhu, Z. Li, S. Zhang, Ying-Yu, H. Zhang, Deep belief network-based internal valve leakage rate prediction approach, Measurement. 133 (2019), 182–192.

[25] W. Yan, D. Tang, Y. Lin, A data-driven soft sensor modeling method based on deep learning and its application, IEEE Trans. Ind. Electron. 64 (2017), 4237–4245.

[26] J. Yu, X. Yan, Layer-by-layer enhancement strategy of favorable features of the deep belief network for industrial process monitoring, Ind. Eng. Chem. Res. 57 (2018), 15479–15490.

[27] X. Yuan, B. Huang, Y. Wang, C. Yang, W. Gui, Deep learning-based feature representation and its application for soft sensor modeling with variable-wise weighted SAE, IEEE Trans. Ind. Inform. 14 (2018), 3235–3243.

[28] W. Fu, T. Sun, J. Liang, B. Yan, F. Fan, Review of principle and application of deep learning, Comput. Sci. 45 (2018), 11–15, 40.

[29] D.N. Reshef, Y.A. Reshef, H.K. Finucane, S.R. Grossman, G. McVean, P.J. Turnbaugh, E.S. Lander, M. Mitzenmacher, P.C. Sabeti, Detecting novel associations in large data sets, Science. 334 (2011), 1518–1524.

[30] W. Qiu, H. Liu, DEG identification of citrus Huanglongbing disease based on maximal information coefficient, J. Gannan Normal Univ. 39 (2018), 72–77.

[31] J. Du, X. Sun, R. Cao, Z. Zhang, Statistical inference for partially linear additive spatial autoregressive models, Spat. Stat. Neth. 25 (2018), 52–67.

[32] T. Chen, C. Guestrin, XGBoost: a scalable tree boosting system, in Knowledge Discovery and Data Mining '16, San Francisco, 2016, pp. 785–794.

[33] J. Wang, X. Yan, Mutual information-weighted principle components identified from the depth features of stacked autoencoders and original variables for oil dry point soft sensor, IEEE Access. 7 (2019), 1981–1990.

[34] S. Jin, Y. Li, M. Xia, Crude column gasoline endpoint soft-sensing of atmospheric and vacuum unit, Chem. Ind. Eng. Prog. 25 (2006), 74–76.