IOP Publishing

Measurement Science and Technology

Meas. Sci. Technol. 33 (2022) 105117 (18pp)

https://doi.org/10.1088/1361-6501/ac7b6c

A denoising and multiscale residual deep network for soft sensor modeling of industrial processes

Renzhi Gao[1], **Hegong Zhu**[1], **Gang Wang**[1,2,3,*] and **Zhangjun Wu**[1,2,3]

[1] School of Management, Hefei University of Technology, Hefei, Anhui, People's Republic of China
[2] Key Laboratory of Process Optimization and Intelligent Decision-Making (Hefei University of Technology), Ministry of Education, Hefei, Anhui, People's Republic of China
[3] Ministry of Education Engineering Research Center for Intelligent Decision-Making & Information System Technologies, Hefei 230009, People's Republic of China

E-mail: wgedison@gmail.com

Received 19 April 2022, revised 10 June 2022
Accepted for publication 22 June 2022
Published 22 July 2022

Abstract

Deep learning plays an important role in soft sensors of industrial processes for the timely measurement of key quality variables. However, since sensors are often operated under noisy and nonstationary industrial conditions, the collected industrial process data exhibit extreme complexity, which severely restricts the learning capacity and measurement accuracy of deep learning methods. In this paper, a novel denoising and multiscale residual deep network (DMRDN) is proposed for soft sensor modeling. Firstly, a stacked denoising autoencoder with level-aware attention is developed to denoise the process data, in which denoised features on different levels are learned and fused. Secondly, the denoised features are fed into multiscale residual convolutional neural network with scale-aware attention, which is designed to capture and fuse deep dynamic features from different scales. Finally, experiments were conducted on an industrial debutanizer column. The experimental results demonstrate that the proposed DMRDN greatly strengthens the learning ability and achieves better prediction performance compared with other methods.

Keywords: soft sensor, deep learning, stacked denoising autoencoder, convolutional neural network, denoising and multiscale residual deep network

(Some figures may appear in colour only in the online journal)

1. Introduction

In modern industrial processes, timely measurement of key quality variables is of great significance for process monitoring, energy conservation, and environmental protection [1–3]. However, due to high technical requirements and expensive analyzer costs, it is difficult to measure most of these quality variables directly. Therefore, soft sensors have been developed to estimate the quality variables in real-time by building predictive mathematical models, which establish certain relationships between these target variables and some easily measurable process variables [4–6]. Up to now, many soft sensors have been extensively researched and successfully applied in diverse practical industrial processes [7–9].

Nowadays, with the development of the Internet of Things and data acquisition technologies, it is convenient to collect and store a huge amount of data from modern industries, which provides enormous data for data-driven modeling [10]. Hence, data-driven soft sensors that are built based on massive available historical data have attracted growing attention for quality prediction. During the past few decades, a great many

[*] Author to whom any correspondence should be addressed.

statistical methods and machine learning methods have been employed to establish data-driven soft sensors. Principal component analysis (PCA) and partial least squares (PLS) are the most popular multivariate linear statistical methods for soft sensors. PCA is a feature extraction technique that decreases the dimensionality of high-dimensional data by building linear combinations of input variables. For example, Wang *et al* proposed a sliding window PCA to eliminate data correlations and redundancy for steady-state detection. The experiments verified the effectiveness and adaptability of the proposed method [11]. Tang *et al* developed a new active learning strategy based on PCA. The application cases confirmed that the proposed strategy showed a great capability to select the most representative dimension [12]. Li *et al* employed PCA to reduce data dimension and extract critical features from the thermal images for raceway depth estimation. The experimental results proved that the proposed method can effectively capture the potential relationship between the thermal images and the raceway depth [13]. Moreover, PLS is a mathematical optimization algorithm that has been widely used to deal with data collinearity in industrial processes. For instance, Xie *et al* improved the PLS framework to handle the outliers of the process data. The experimental results verified that the framework enhanced the efficiency and accuracy of the key quality indicators prediction [14]. Galicia *et al* proposed a reduced-order dynamic PLS soft sensor approach, which reduced the model size appropriately. The experiments on two types of digester showed the excellent performance of the developed approach [15]. Zheng and Song incorporated semi-supervised learning with PLS to improve the performance of soft sensors. The experiments demonstrated the validity and superiority of the designed method [16]. Although PCA and PLS mentioned above show excellent performance in linear data modeling, they have unavoidable defects in processing nonlinear data. Due to the complex process mechanism in large-scale industrial systems, the industrial process data are highly nonlinear. For this reason, many machine learning methods that are suitable for dealing with nonlinear data have been widely exploited for soft sensors, such as artificial neural networks (ANNs) and support vector regression (SVR). ANN is composed of a number of interconnected neuron nodes, which can analyze the nonlinear processes owing to the nonlinear activation function in the nodes. For example, Gonzaga *et al* developed an ANN-based soft sensor to provide online measurement in the polyethylene terephthalate production process. The application showed that the proposed method can adequately estimate the polymer viscosity [17]. Pani *et al* developed a soft sensor based on a feed-forward ANN for product quality monitoring. The experiments showed that the designed method achieved high accuracy [18]. Bispo *et al* utilized a ANN to estimate the apparent viscosity of water-based drilling fluids. The results demonstrated that the method provided good estimations of the apparent viscosity [19]. SVR is a classical supervised machine learning algorithm, which handles nonlinear modeling problems based on the kernel method. For example, Lian *et al* utilized SVR to predict the rotor thermal deformation value. The experiments proved that the designed method dramatically improved the prediction performance [20]. Desai *et al* employed SVR for soft sensor applications in fed-batch processes. The applications indicated that SVR is attractive for the development of soft sensors in bioprocesses [21]. Zhang and Liu developed a new soft sensor based on SVR for industrial melt index prediction. The experimental results demonstrated the exceptional performance of the proposed method [22]. However, the actual industrial process data usually have the characteristics of complicated nonlinearity, high dimension, and redundancy, which make it difficult for the aforementioned methods with shallow architectures to extract complex features from industrial process data.

In recent years, deep learning has received great attention and recognition due to its powerful feature representation capability. Compared with traditional machine learning methods, deep neural networks with numerous hidden layers allow the model to directly learn abstract latent feature representations from raw data and show great advantages in extracting complex nonlinear features. Therefore, many deep learning methods have been introduced into industrial processes to construct data-driven soft sensors for quality prediction, including the stacked autoencoder (SAE), recurrence neural network (RNN), and convolutional neural network (CNN). Among these, the SAE is a classic unsupervised deep learning method that is composed of multiple basic autoencoders with the pretraining and fine-tuning strategy. The SAE aims to learn the deep high-level feature representations by hierarchically reconstructing the data samples. For instance, Yuan *et al* proposed a nonlinear variable-wise weighted SAE for quality-relevant feature extraction. The experiments confirmed that the method can learn hierarchical quality-related features [23]. Yan *et al* designed an SAE-based deep relevant representation learning method. The experimental results demonstrated the effectiveness of the developed approach [24]. Liu *et al* developed a novel stacked neighborhood-preserving autoencoder to obtain layer-wise neighborhood-preserving representations. The experiments on the hydrocracking process showed that the proposed method attained the best results [25]. Moreover, as one of the variants of the SAE, the stacked denoising autoencoder (SDAE) has been introduced to capture more critical information by recovering the original data from corrupted data, which can learn more robust feature representations in the case of input fluctuation. Hence, many efforts have been made to apply SDAE in industrial processes. For example, Yan *et al* introduced SDAE for the measurement of oxygen content in flue gasses. The experiments demonstrated that the developed method achieved excellent performance as a result of capturing the essential information [26]. Ba-Alawi *et al* employed SDAE to monitor sustainable influent quality in wastewater treatment plants. The experimental results verified that the SDAE showed better learning capability than the ordinary SAE [27]. Zhang *et al* designed a nonlinear process monitoring method based on the SDAE. The experiments verified that the SDAE can capture crucial features [28]. Meanwhile, considering time-delayed feedback control and the retention of various materials and products in the plants, most industrial processes are naturally dynamic and there are strong temporal relationships between consecutive data samples [29, 30]. The RNN is a typical deep learning method that pays attention to temporal relationships, which can memorize historical information due to its sequence structure [31]. Moreover, to overcome the gradient disappearance problem, the modified variants of the RNN like the long short-term memory (LSTM) and the gated recurrent unit (GRU) have been widely applied in the industrial field [32–34]. For example, Guo and Liu developed a dynamic soft sensor based on GRUs. The experiments verified that the method can catch the dynamic characteristics of process data and spread them over time [35]. Han *et al* utilized LSTM to develop a production capacity analysis and energy-saving method. The experiments proved the validity of the proposed method in coping with time-series data [36]. Zhang and Ge designed an encoder-decoder model based on the GRU for dynamic feature extraction. The proposed method was successfully implemented on a cloud computing platform to analyze industrial big data [37]. However, to capture the dynamic relevance between far temporal sampling instants, the RNN and its variants need a large number of memory units to store long-term dependency information, resulting in highly complex model structures. Meanwhile, the recursive relationships make parallel computing impossible for the RNN, which takes a lot of computing time when handling massive process data. Alternatively, theCNN is another effective deep learning model that can capture dynamic features from industrial data. The raw one-dimensional signal data are usually augmented into the two-dimensional matrix data as the input of a CNN [38, 39]. Besides, since the CNN supports parallel and distributed computing, the calculation speed of the CNN is much faster than that of other sequence models, which is important to satisfy the real-time requirement of soft sensors. Recently, many industrial soft sensor applications based on CNNs have been exploited. For instance, Geng *et al* developed a novel transformer based on a gated CNN for dynamic soft sensor modeling. The experimental results showed the efficient dynamic feature extraction capability of the proposed method [29]. Wang *et al* combined the finite impulse response with CNN to build a dynamic soft sensor. The experiments demonstrated that the method can provide the most interpretable trend of the quality variable [40]. Yuan *et al* established a multichannel CNN for the representation of various local dynamic features. The experiments on the debutanizer column and hydrocracking process verified the feasibility and effectiveness of the approach [41].

To sum up, recent studies have shown the advantages of soft sensors based on deep learning. However, there are still several disadvantages to the existing methods. On the one hand, in the process of data acquisition and transmission in practical industrial systems, various factors such as internal aging of industrial equipment and external electromagnetic radiation disturbance result in a large amount of noise mixed into the industrial process data [5, 42]. The noisy data tremendously affect the accuracy and robustness of soft sensors, and may even submerge the actual signal data. Though the SDAE has been applied to eliminate the effect of noise, the existing SDAE only used the denoised features of the last layer for the final prediction [43]. Since different layers of the SDAE represent the denoising of input data on different levels, the denoised features of shallow layers also have their modeling value. On the other hand, due to changes in raw material composition, catalyst concentration, and internal equipment components, most industrial processes are always operating under complex nonstationary conditions [44–46]. As a result, the industrial process data exhibit time-varying characteristics and contain complex dynamic feature information distributed on different timescales, which raises notable difficulties for feature extraction. Besides, since the industrial process data are collected from complex and nonstationary industrial systems, deeper and more complex network structures are expected to extract more complicated features from industrial process data. Nevertheless, experiments have found that with the network depth increasing, serious degradation problems may emerge and decrease the accuracy of prediction [47, 48]. Therefore, it is worth considering how to develop soft sensors based on deep learning with high performance and robustness to deal with noisy and nonstationary industrial processes.

To address these intractable problems, a novel denoising and multiscale residual deep network (DMRDN) is proposed for soft sensor modeling. The network mainly consists of two parts, i.e. a SDAE with level-aware attention (SDAE-LA) and a multiscale residual CNN with scale-aware attention (MRCNN-SA). Firstly, the SDAE-LA is developed to denoise the raw process data. Specially, the SDAE is used to hierarchically learn denoised features from the corrupted process data, in which denoised features of hidden layers are the representations of input on different levels and all contribute to building the prediction model. Then, level-aware attention is employed to quantify the contributions of different hidden layers and integrate the denoised features on different levels. The denoised features obtained in this way are steady and robust for subsequent soft sensor modeling. Secondly, the MRCNN-SA is designed to further extract the multiscale deep feature representations from the denoised features. In the MRCNN-SA, a CNN with multiple convolutional kernels on different scales is employed to simultaneously capture the dynamic characteristics on different time scales. Meanwhile, the residual connection is utilized to make the networks applicable for learning deep features and concurrently overcome the performance degradation problem. Furthermore, scale-aware attention is employed to reveal the importance of different scales and fuse the complementary features of different scales. Thus, the complex multiscale feature representations can be adaptively captured from time-varying industrial processes, which improves the robustness and generalization ability even under nonstationary industrial conditions. Finally, the fused multiscale deep features are fed to the fully connected network to obtain the final quality prediction results. To verify the performance of the proposed method, the experiments are conducted on an actual industrial data set of the debutanizer column. The experimental results illustrated that the prediction performance of the proposed method is superior to other commonly used traditional machine learning methods and deep learning methods. Additionally, the effect of each component in the proposed DMRDN is also investigated. The results confirm that the DMRDN is an effective soft sensor modeling method with good denoising performance and powerful feature extraction capability.

The main contributions of this paper are summarized as follows.

(a) An end-to-end soft sensor modeling framework is proposed for quality prediction in industrial processes in consideration of the noisy and nonstationary industrial conditions. The proposed framework conduces to the improvement of product quality and optimization of process control.
(b) A novel deep learning method named DMRDN that combines a SDAE-LA and a MRCNN-SA is designed. The SDAE-LA is utilized to extract and integrate denoised features on different levels, which can eliminate the effect of noise. In addition, the MRCNN-SA is developed to learn and fuse deep multiscale feature representations, which contain short-term and long-term dynamic features.
(c) The proposed DMRDN is evaluated through experiments on the industrial debutanizer column. The results and comprehensive analysis validate that the proposed method can achieve the most accurate and stable prediction compared with other traditional machine learning methods and deep learning methods.

The remaining parts of this paper are organized as follows. Section 2 describes the proposed method for soft sensors. In section 3, the details of the experimental data set from a real industrial process are given. In section 4, the experimental results are presented and analyzed . Finally, section 5 summarizes the conclusions and future directions of the work.

2. The proposed soft sensor

2.1. Overall framework

Soft sensors are playing an increasingly important role in estimating the production quality in industrial processes, which is conducive to improving the production processes. As an effective tool, deep learning has stimulated many studies that proposed various soft sensors for the measurement of quality variables. However, existing methods are still faced with some nonnegligible problems under noisy and nonstationary industrial conditions. On the one hand, noise often exists in industrial process data, which seriously affects the performance of soft sensors. Although some methods have been developed to alleviate noisy data, few of them pay attention to the denoised features of different layers. On the other hand, the nonstationary industrial environment leads to complex and changeable feature information in process data, which cannot be effectively captured in current deep learning methods. In this paper, we propose the DMRDN to handle the noisy and nonstationary industrial conditions for soft sensor modeling. The overall framework of the proposed method is illustrated in figure 1, which can be summarized as follows.

(a) Data acquisition: the industrial process data are collected over a continuous period for soft sensor modeling. The process variable data are acquired by physical sensors, while the quality variable data are acquired by offline laboratory analysis.
(b) SDAE-LA: SDAE is utilized to hierarchically extract critical features and filter irrelevant noise information, where denoised features on different levels are obtained. Then, level-aware attention is developed to quantify the contributions of the multilevel denoised features and integrate them with different weights.
(c) MRCNN-SA: the integrated denoised features are augmented and fed into the MRCNN to further extract deep dynamic feature representations from multiple timescales. Then, scale-aware attention is employed to reveal the importance of each scale and fuse the multiscale features with different weights. Finally, the fused features are input into the fully connected network to obtain the prediction results.

2.2. Data acquisition

To establish the soft sensor, it is necessary to collect a certain amount of data about process variables and quality variables. The process variables, like temperature, pressure, and flow are determined according to theoretical analysis and operator experience. In industrial plants, distributed measuring systems are installed to monitor these process variables, whose detailed values are recorded at a certain sampling frequency from the analog signals of sensor devices. Then, the recorded data are transmitted to the server through communication protocols and stored in industrial databases. Meanwhile, since the quality variables are difficult to obtain directly by sensor devices, the values of quality variables are measured and recorded through offline laboratory analysis, which usually consumes more time and is more costly.

2.3. SDAE-LA

In industrial systems, many devices are installed to collect the values of some process variables that are easy to measure for the estimation of important quality variables. Due to random disturbances of the internal and external environment in industrial processes, the collected industrial process data are often mixed with noise. Since the SDAE learns hidden feature representations from corrupted data, it can eliminate the interference of noisy data to a certain extent [49]. However, most methods have just utilized the last hidden layer for subsequent quality prediction, which ignores the modeling value of other layers. To quantify the contributions of each

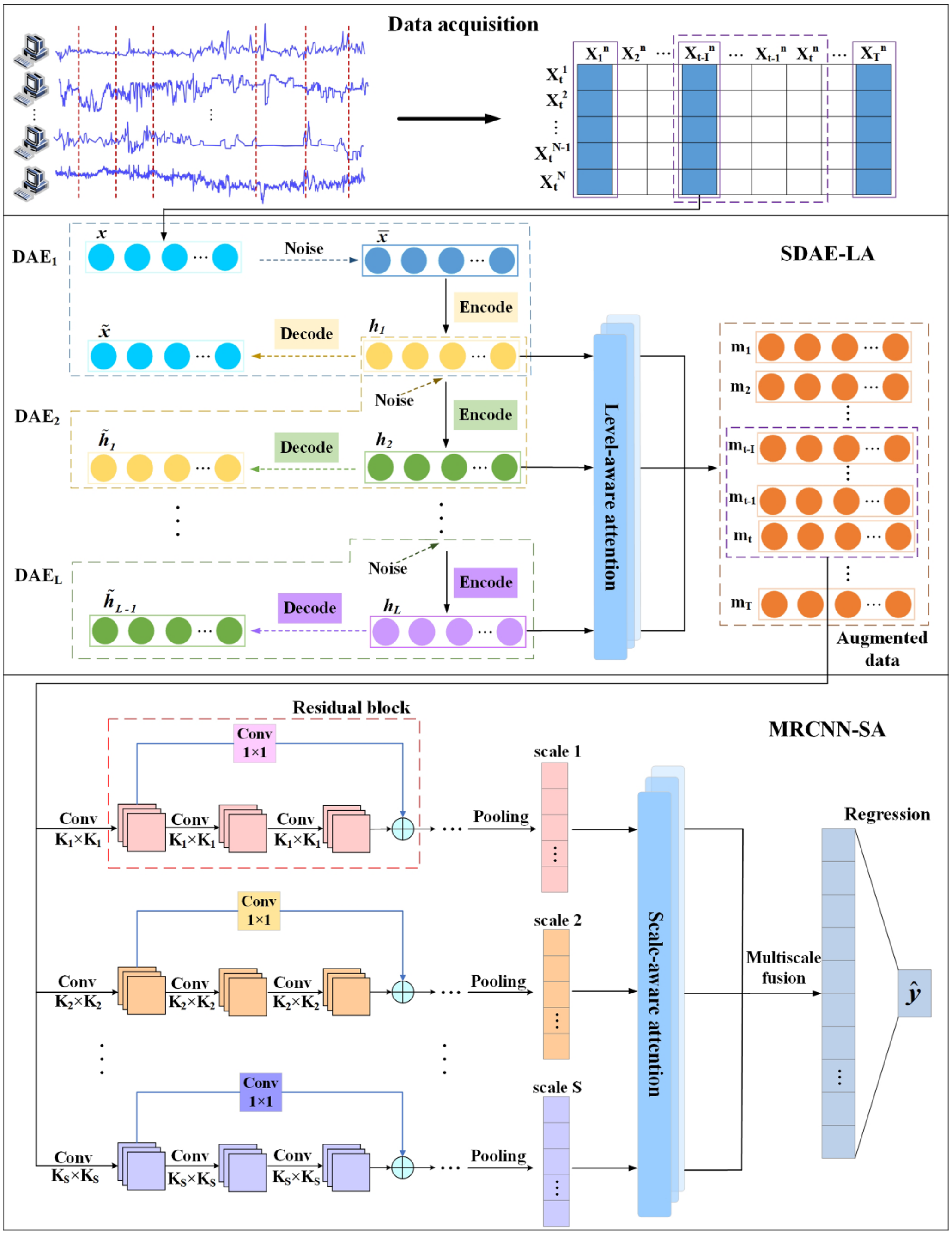

Figure 1. The framework of the proposed DMRDN.

hidden layer and take full advantage of the denoised features, level-aware attention is embedded into the SDAE to construct the SDAE-LA, which can learn multilevel denoised feature representations from the original data.

The first step of the SDAE-LA is to obtain denoised features through multiple stacked DAEs, which hierarchically recover the original process data from the corrupted data in an unsupervised manner. Suppose that we are given the original industrial process data $D = \{(x_t, y_t)\}_{t=1}^{T}$, where $x_t = [x_t^1, x_t^2, x_t^3, \ldots, x_t^N] \in R^{N\times T}$ is collected at tth time with N process variables, and T is the total number of process samples. Thus, the corrupted industrial process data can be denoted as $\bar{D} = \{(\bar{x}_t, y_t)\}_{t=1}^{T}$, where $\bar{x}_t = [\bar{x}_t^1, \bar{x}_t^2, \bar{x}_t^3, \ldots, \bar{x}_t^N] \in R^{N\times T}$ is added with some Gaussian noise. As shown in figure 1, firstly, the noise-added process variable data are input into the first DAE, which are encoded to be transferred into hidden feature representations via nonlinear mapping. The feature representations in the hidden layer of the first DAE can be denoted as $h_{t,1} = [h_{t,1}^1, h_{t,1}^2, h_{t,1}^3, \ldots, h_{t,1}^{d_\mathrm{h}}] \in R^{d_\mathrm{h}\times T}$, where d_h is the dimension of the hidden feature vectors. The calculation process of encoding is as follows:

$$h_{t,1} = f_\mathrm{e}\left(W_{\mathrm{e},1}\bar{x}_t + b_{\mathrm{e},1}\right) \tag{1}$$

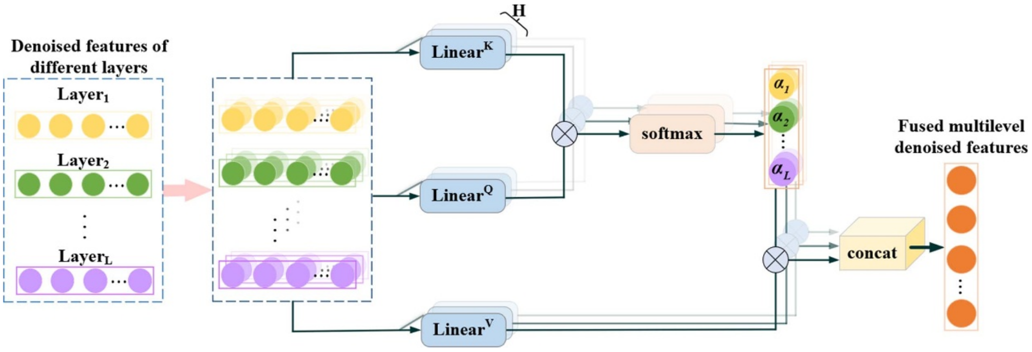
Denoised features of different layers
Layer1
Layer2
LayerL
H
LinearK
LinearQ
LinearV
softmax
α1
α2
αL
concat
Fused multilevel denoised features

Figure 2. The network structure of level-aware attention.

where f_e is the nonlinear activation function such as a sigmoid function and a tanh function. $W_{\mathrm{e},1}$ and $b_{\mathrm{e},1}$ are the parameters of the weight matrix and bias for the encoder phase of the first hidden layer, respectively. Then, the feature vector $h_{t,1}$ is decoded to map back to the original process variable data, which are calculated as:

$$\tilde{x}_t = f_\mathrm{d}\left(W_{\mathrm{d},1}h_{t,1} + b_{\mathrm{d},1}\right) \tag{2}$$

where $\tilde{x}_t = [\tilde{x}_t^1, \tilde{x}_t^2, \tilde{x}_t^3, \ldots, \tilde{x}_t^N] \in R^{N\times T}$ represents the reconstructed input vectors, and f_d, $W_{\mathrm{d},1}$, and $b_{\mathrm{d},1}$ are the activation function, weight matrix, and bias for the decoder phase of the first hidden layer, respectively. To optimize the parameters and obtain the denoised hidden features, the reconstruction error between the original process data and the reconstructed process data is minimized. Thus, the following loss function is minimized by the backpropagation algorithm:

$$J\left(W_{\mathrm{e},1}, b_{\mathrm{e},1}, W_{\mathrm{d},1}, b_{\mathrm{d},1}\right) = \frac{1}{2T}\sum_{t=1}^{T}\|x_t - \tilde{x}_t\|^2. \tag{3}$$

In this way, the first DAE is established, and the corresponding first-level denoised feature vector $h_{t,1}$ from process variable data can be obtained. Then, some noise is added to $h_{t,1}$ and used as the input of the second DAE, and the above encoding, decoding, and loss computation processes are repeated to obtain the denoised feature of the second hidden layer $h_{t,2}$. In this successive way, the process data are denoised hierarchically, and the denoised features on different levels are learned progressively from the original observed process variable data by layer-wise training. For the SDAE-LA with L DAEs, the denoised feature sequence from different layers $h_t = [h_{t,1}, h_{t,2}, \ldots, h_{t,L}] \in R^{L\times d_\mathrm{h}}$ is learned.

Since each DAE in different layers of the SDAE is trained in sequential order, the denoised features obtained from different layers reflect the denoising of process variable data on distinct levels. It means that the denoised features from each hidden layer of the SDAE contain useful information and have different contributions to the final quality prediction, and the denoised features of shallow layers should also be utilized [50]. Hence, it is necessary to appropriately take advantage of all denoising layers for improving the performance of industrial data denoising. To achieve this, level-aware attention is introduced to measure the importance of different layers. The core of level-aware attention is to obtain information that needs to be focused on and suppressed from the denoised features. It calculates the weight for denoised features of each layer, and then the sequence features are merged according to their weights. Significantly, the advantage of the attention mechanism over conventional methods is that it is executed on a sample level, which means a different importance can be distinguished for each prediction [51]. Besides, level-aware attention is based on multi-head attention, which is an improvement of the self-attention mechanism [52]. Compared with single-head attention, multi-head attention learns the correlative representation information from different subspaces at different positions, and can learn more abundant feature representations. Moreover, the multiple independent heads can be regarded as having the ensemble function, which can prevent overfitting and improve the generalization ability. Meanwhile, although the features are mapped into multiple heads to calculate attention values, the number of weight parameters in multi-head attention is consistent with that of single-head attention with full dimensionality since the dimension of each head is reduced [52]. Therefore, multi-head attention has a more powerful representation ability and simultaneously maintains a similar computational cost similar to single-head attention.

As illustrated in figure 2, the denoised features of different layers h_t are projected to $h_t = [h_t^1, h_t^2, \ldots, h_t^h \ldots, h_t^H] \in R^{L\times d_\mathrm{h}/H}$ with H heads for linear transformation. In each projection part, the attention calculation is conducted to assign the weight for denoised features of each layer. Then, the attention weights in all heads are concatenated together. The mathematical forms of the calculation are shown as follows:

$$\alpha_t^h = \mathrm{softmax}\left(W_Q^h h_t^h \left(h_t^h W_K^h\right)^\mathrm{T}\right) \tag{4}$$

$$\mathrm{DA}_t^h = \alpha_t^h h_t^h W_V^h \tag{5}$$

$$m_t = \mathrm{Concat}\left(\mathrm{DA}_t^1, \mathrm{DA}_t^2, \ldots, \mathrm{DA}_t^H\right) \tag{6}$$

where W_Q^h, W_K^h, and W_V^h are the projection parameters in the hth head. $\alpha_t^h \in R^{1\times L}$ is the attention weights in the hth head for each layer, and m_t is the obtained multilevel denoised features. In this way, denoised features on different levels are fully integrated and utilized to strengthen the robustness in the face of noisy data for subsequent soft sensor modeling.

2.4. MRCNN-SA

Through the above data-denoising by the SDAE-LA, the denoised features $m = [m_1, m_2, \ldots, m_t, \ldots, m_T] \in R^{T\times d_\mathrm{h}}$ can be obtained. In consideration of the dynamic characteristics of industrial processes, the CNN has been extensively applied in soft sensors, and has shown excellent performance in automatically capturing the dynamic features from augmented two-dimensional matrix samples [53]. With the particular advantages of local connection and weight sharing, CNN can extract deep abstract feature representations with fewer model parameters. However, industrial systems are usually operating under non-ideal conditions, and the process behaviors such as materials variation, catalyst degradation, and mechanical abrasion change frequently, The nonstationary industrial conditions lead to variable process characteristics, which means the dynamic relevance of process data may not always be on the same scale. The traditional CNN usually used one fixed convolution kernel size to analyze the process data, which is not applicable for the variable industrial processes. In addition, due to the complicated and nonstationary industrial conditions, deeper networks are essential to extract more complex and deeper representations from industrial process data. Nevertheless, it has been proven that when the number of network layers increases gradually, the accuracy tends to a maximum and then degrades, and adding more network layers leads to worse performance, which is known as the network degradation problem [47, 48]. For solving the above problems, the MRCNN-SA is designed to capture the deep dynamic representation features with various scales and residual connection.

Firstly, the primary one-dimensional feature vectors obtained from the SDAE-LA are augmented into two-dimensional feature matrices using a sliding window based on a time lag, which means that the features at the previous sampling times are utilized to predict the quality variable at the current time. In this way, the augmented two-dimensional feature matrices contain information from previous samples, from which the dynamic characteristics of industrial process data can be captured. Given the quality variable y_t at the tth sampling moment, the corresponding feature vector is $m_t = [m^1, m^2, m^3, \ldots, m^{d_\mathrm{h}}]$. Then, the augmented two-dimensional matrix features at the tth time based on time series can be organized as:

$$M_t = \begin{bmatrix} m_{t-I} \\ \vdots \\ m_{t-1} \\ m_t \end{bmatrix} = \begin{bmatrix} m^1_{t-I}, m^2_{t-I}, m^3_{t-I}, \ldots, m^{d_\mathrm{h}}_{t-I} \\ \vdots \\ m^1_{t-1}, m^2_{t-1}, m^3_{t-1}, \ldots, m^{d_\mathrm{h}}_{t-1} \\ m^1_{t-0}, m^2_{t-0}, m^3_{t-0}, \ldots, m^{d_\mathrm{h}}_{t-0} \end{bmatrix} \tag{7}$$

where I is the selected time step size of the used sliding window, which represents the number of utilized previous samples.

Then, the augmented features are fed to the multiscale residual CNN to further extract the multiscale deep features of the process data. As shown in figure 1, the multiscale residual CNN in the proposed method consists of multiple parallel CNN branches. The different branches are constructed with different sizes of convolution kernels, which enable convolution layers to learn short-term and long-term dynamic features from different time scales. For each branch, the augmented sample M_t is input to the convolution layers to obtain the feature maps, which are expressed as:

$$y_t = f_\mathrm{c}(W_s \otimes M_t + b_s) \tag{8}$$

where $\otimes$ represents the convolution operation, which can filter useful information from process data. W_s and b_s are the convolution kernel and bias of the sth branch, respectively. f_c is the nonlinear activation function, which refers to the LeakyReLU function used in this paper to retain relevant features and filter uncorrelated features. Generally, the shallow convolution layers only capture low-level features while the deep convolution layers can capture more complex and high-level features. To extract more complicated and deeper features, more convolution layers are required to form deep CNN networks. Thus, the residual connection is embedded in every two convolution layers to construct residual CNN blocks, which can form deep network structures while avoiding the degradation problem [48]. In each branch, there are several sequentially connected residual CNN blocks, which are replaced by an ellipsis in figure 1. The basic residual CNN block is formalized as follows:

$$r_{s,j} = W_{s,j} \otimes f_c(W_{s,j-1} \otimes x + b_{s,j-1}) + b_{s,j} \tag{9}$$

$$c_{s,j} = f_c(r_{s,j} + W_s^{1\times 1} \otimes x + b_s) \tag{10}$$

where j is the index of the convolution layer and $c_{s,j}$ is the output of the residual CNN block to which the jth convolution layer of the sth branch belongs. $W^{1\times 1}_{s,j}$ represents the 1 × 1 convolution kernel that is performed to match the input and output dimensions in the residual block and ensure the implementation of the connection operation. In this way, when the residual value of a certain layer is 0, the layer is equivalent to identity mapping, which can avoid the gradient vanishing problem. Meanwhile, the residual learning structure makes it easier than the traditional structures for the networks to learn features , which simplifies the training process of complex deep networks. Besides, at the end of each CNN branch, a pooling layer is added to reduce the dimension of feature space and prevent overfitting. The average pooling operation, which uses average value in the local region as the output, is chosen in this paper to simplify the feature representations. Then, the output feature maps of each branch are flattened to vectors as the obtained features on each scale. For different CNN branches with different convolution kernel sizes, the feature representations of process data from different views with multiscale can be learned, and are denoted as $c_t = [c_1, c_2, \ldots, c_S] \in R^{S\times d_\mathrm{c}}$, where S is the number of scales and d_c is the dimension of the obtained features on each scale.

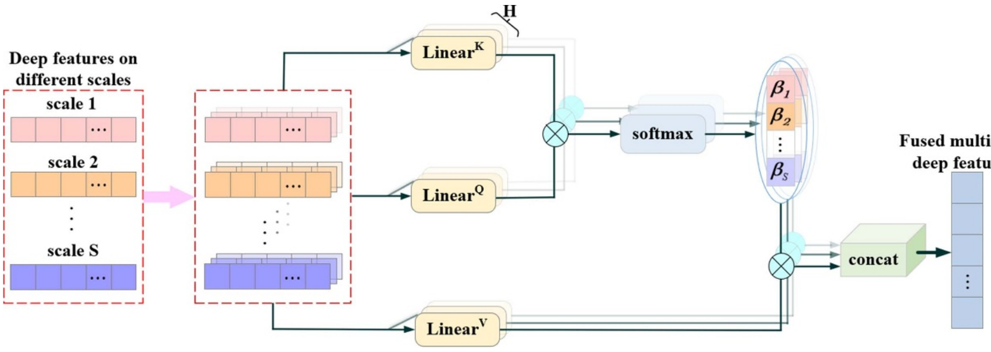

Figure 3. The network structure of scale-aware attention.

Table 1. Pseudo-code of the proposed algorithm.

```
Input: Original dataset $D = \{(x_t, y_t)\}_{t=1}^{T}$, $t = 1, 2, \cdots, T$;
         Hyperparameter set $P$;
         Number of denoising autoencoders $L$;
         Sliding window size $I$
         Number of convolution kernel scales $S$;
Output: Predicted quality variable $\hat{y}_t$
Processing:
for $t \in \{1, 2, \ldots, T\}$ do
   for $l \in \{1, 2, \ldots, L\}$ do
      if $l == 1$ then
         Denoised features $h_{t,l} \leftarrow \mathrm{DAE}^{l}\ (x_t)$;
      else then
         $h_{t,l} \leftarrow \mathrm{DAE}^{l}\ (h_{t,l\text{-}1})$;
      end if
   end for
   Multilevel denoised features $m_t \leftarrow$ Level-aware attention
   $(h_{t,1}, h_{t,2}, \ldots, h_{t,L})$;
   for $s \in \{1, 2, \ldots, S\}$ do
      $M_t \leftarrow$ concat $(h_{t\text{-}I}, h_{t\text{-}I+1}, \ldots, h_t)$
      $c_{t,s} \leftarrow$ Residual convolution layers $(M_t)$;
      $c_{t,s}$, sesidual convolution lay$c_{t,s}$);
   end for
   Multiscale deep features $o_{(t)} \leftarrow$ Scale-aware attention
   $(c_{t,1}, c_{t,2}, \ldots, c_{t,S})$;
   Predicted quality variable $\hat{y}_t \leftarrow$ Linear layer $(o_t)$;
end for
```

Since the variations of industrial processes are usually disorderly and elusive, the effect of features on each scale for the quality prediction is inconsistent. To fuse the extracted complementary multiscale features, it is not advisable to simply add the features of different scales directly, which may ignore important information. Therefore, to fuse the features of different scales appropriately, scale-aware attention is introduced, which is also based on the multi-head attention mechanism. The feature representations of various scales are input into the scale-aware attention network to calculate the weight of each scale and add them according to the weights. Since the weight parameters reflect the importance of each scale, the network can pay more attention to information on more important scales.

As illustrated in figure 3, the deep multiscale features c_t are projected to $c_t = [c_t^1, c_t^2, \ldots, c_t^h \ldots, c_t^H]$ with H heads for linear transformation, where $c_t^h \in R^{S \times d_c/H}$. The attention values of multiscale features on each scale are calculated parallelly in each projection part, then, the attention values in all heads are concatenated together. The calculation formulas are as follows:

$$\beta_t^h = \mathrm{softmax}\left(W_Q^h c_t^h \left(c_t^h W_K^h\right)^{\mathrm{T}}\right) \tag{11}$$

$$\mathrm{SA}_t^h = \beta_t^h c_t^h W_V^h \tag{12}$$

$$o_t = \mathrm{Concat}\left(\mathrm{SA}_t^1, \mathrm{SA}_t^2, \ldots, \mathrm{SA}_t^H\right) \tag{13}$$

where W_Q^h, W_K^h, and W_V^h are the projection parameters in the hth head. $\beta_t^h \in R^{1 \times S}$ is the attention weights in the hth head for each scale, and o_t is the fused multiscale feature representations. The scale-aware attention enables the proposed method to adaptively distinguish whether features on a certain scale are more or less important for each sample, and the fused multiscale features can well reflect the current industrial process states.

Finally, the fused multiscale features are used as the input of the fully connected regression network. To reduce the parameters and avoid the overfitting problem, only one fully connected layer is used to obtain the ultimate prediction results.

For model training, the loss function with the mean square error is used as the objective function. Meanwhile, L2 regularization is added to smooth the network and avoid overfitting caused by the excessive complexity of the soft sensor model. Therefore, the eventual form of the objective function is formalized as:

$$\mathrm{Loss}(\theta) = \frac{1}{T_{\text{train}}} \sum_{t=1}^{T_{\text{train}}} \left(\hat{y}_t^{\text{train}} - y_t^{\text{train}}\right)^2 + \lambda \|\theta\|^2 \tag{14}$$

where $\hat{y}_t^{\text{train}}$ and y_t^{train} are the predicted value and target value of the training set, respectively. λ is the penalty coefficient of the regularization term, and θ is the parameter set to be learned. Besides, the Adam optimizer is applied in the backpropagation algorithm, which is conducive to accelerating the convergence of the model.

Table 1 shows the pseudo-code of the proposed algorithm.

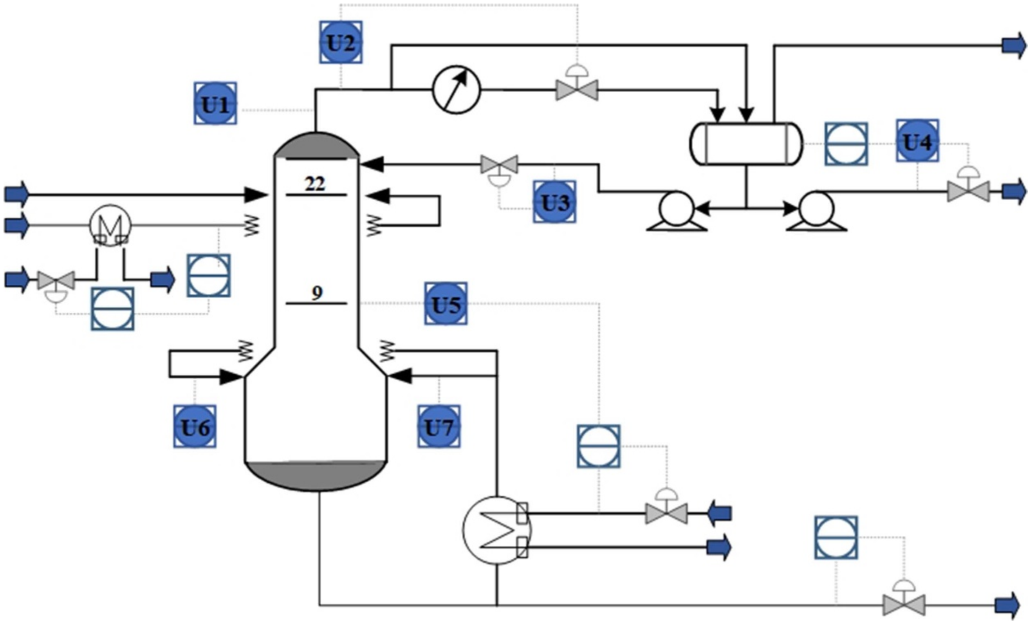

Figure 4. Schematic of the debutanizer column.

3. Experiments

3.1. Experimental data set

To verify the effectiveness of the proposed soft sensor, relevant experiments were carried out on actual industrial process data from a debutanizer column. The debutanizer column is an important component of the petroleum refining process that is designed for desulphurization and naphtha cracking. Figure 4 shows the schematic flowchart of the debutanizer column, which mainly consists of six devices: heat exchanger, bottom reboiler, overhead condenser, feed pump to the splitter, head reflux pump, and reflux accumulator. Meanwhile, several hardware sensors indicated with circles in figure 4, are physical sensors installed on the plant to measure some regular variables, such as temperature, pressure, and flow. The main purpose of the debutanizer column is to remove propane (C3) and butane (C4) from the naphtha stream. The minimizing of C4 content in the debutanizer bottom is significant to ensure product quality and increase economic benefits [54]. Therefore, it is essential to estimate the concentration of C4 in real-time for effective control and optimization. However, the C4 concentration is not measured directly at the debutanizer bottom, but is usually measured by gas chromatographs at the overhead of the debutanizer column. The distant analyzer location leads to a long time delay. To cope with the timely measurement problems, soft sensors have been adopted to implement online estimation of the C4 concentration for real-time process control. For this purpose, seven typical process variables measured by hardware sensors are selected as the inputs for soft sensors, which are given in table 2 with their corresponding descriptions. Moreover, in order to obtain more accurate estimation values, the process dynamics are further considered in this paper. In other words, on the basis of the input variables and previous input and output samples, new variables are added as the raw input variables. The ultimate input variables for the proposed method are designed as:

Table 2. Process variable description for soft sensors in debutanizer column.

Variable	Detailed description
X_1	Top temperature
X_2	Top pressure
X_3	Reflux flow
X_4	Flow out of the process
X_5	6th tray temperature
X_6	Bottom temperature A
X_7	Bottom temperature B

$$\begin{bmatrix} x_1(t), x_2(t), x_3(t), x_4(t), x_5(t), x_5(t-1), \\ x_5(t-2), x_5(t-3), (x_6(t)+x_7(t))/2, \\ y(t-1), y(t-2), y(t-3), y(t-4) \end{bmatrix}. \tag{15}$$

In [55], the detailed procedures to obtain these additional variables have been described. In this paper, a total of 2368 data samples were collected from the debutanizer column, of which 70% served as the training set and the remaining served as the testing set.

3.2. Evaluation metrics

To evaluate the performance of the proposed soft sensor, two common evaluation metrics of regression analysis were

Table 3. Details about the parameter settings.

Method	Parameters
SVR	Kernel: rbf; C: 8; gamma: 0.05.
ANN	Number of hidden layer neurons: 7.
SAE	Number of hidden layer neurons: 12-7-5.
SDAE	Number of hidden layer neurons: 12-7-5; mean/standard deviation of added noise: 0/0.1.
RNN	Number of hidden layer neurons: 7.
CNN	Depth of convolutional layers: 5; number/size/stride of kernels: 2/5/1, 2/5/1, 2/5/1, 4/5/2, 4/5/1; pooling size/stride: 2/2.
SDAE-CNN	SDAE: number of hidden layer neurons: 12-12-12; mean/standard deviation of added noise: 0/0.1. CNN: depth of convolutional layers: 5; number/size/stride of kernels: 2/5/1, 2/5/1, 2/5/1, 4/5/2, 4/5/1; pooling size/stride: 2/2.
DMRDN	SDAE: number of hidden layer neurons: 12-12-12; mean/standard deviation of added noise: 0/0.1. CNN: depth of convolutional layers: 5; number/size/stride of kernels: 2/3/1, 2/3/1, 2/3/1, 4/3/2, 4/3/1; 2/4/1, 2/4/1, 2/4/1, 4/4/2, 4/4/1; 2/5/1, 2/5/1, 2/5/1, 4/5/2, 4/5/1; 2/6/1, 2/6/1, 2/6/1, 4/6/2, 4/6/1; 2/7/1, 2/7/1, 2/7/1, 4/7/2, 4/7/1; pooling size/stride: 2/2.

adopted in this paper. One of them is the root mean squared error (RMSE), which is defined as:

$$\mathrm{RMSE} = \sqrt{\sum_{t=1}^{T_{\mathrm{test}}} \frac{\left(y_t^{\mathrm{test}} - \hat{y}_t^{\mathrm{test}}\right)^2}{T_{\mathrm{test}}}} \tag{16}$$

where $\hat{y}_t^{\mathrm{test}}$ and y_t^{test} are the predicted and target output value of the tth testing sample, respectively, and T is the total number of the testing set. The RMSE was used to judge the error between the target value and the predicted value, and a smaller RMSE value generally indicates better predictive performance. Another index used in this paper is the determination coefficient R^2, which is defined as:

$$R^2 = 1 - \frac{\sum_{t=1}^{T_{\mathrm{test}}} \left(y_t^{\mathrm{test}} - \hat{y}_t^{\mathrm{test}}\right)^2}{\sum_{t=1}^{T_{\mathrm{test}}} \left(y_t^{\mathrm{test}} - \bar{y}_t^{\mathrm{test}}\right)^2} \tag{17}$$

where $\bar{y}_t^{\mathrm{test}}$ is the mean value of the output variable in the testing set. The R^2 reflects the fitting degree of the predicted values for the true values, and a better method provides a larger R^2.

3.3. Experiment procedure

In the experiments, we contrasted the proposed DMRDN with the competitors including the ANN and SVR in the field of traditional machine learning, as well as the SAE, SDAE, RNN, CNN, and SDAE with CNN (SDAE-CNN) in deep learning. The detailed parameters of each method are shown in table 3. To further verify the denoising ability of the proposed method, Gaussian noise was added to the testing dataset to stimulate the complex noise in actual industrial processes. The mean value of the added Gaussian noise is 0 and the standard deviation is 0.2. For training the abovementioned models, the early stopping criteria and dropout, set as 0.5, were applied to alleviate overfitting. Moreover, to reduce the impact of randomness, ten repeated experiments were carried out for each model with different random seeds. The average values of the prediction performance were calculated to obtain the final results. In this paper, all the experiments were implemented by Python 3.8 with a Xeon-E5-2620 CPU and NVIDIA-Tesla-K80 GPU.

Table 4. Prediction results of different methods.

	RMSE		R^2	
Methods	Mean	SD	Mean	SD
ANN	0.0694	0.0051	0.8520	0.0224
SVR	0.0743	0.0021	0.8311	0.0090
SAE	0.0662	0.0029	0.8616	0.0142
SDAE	0.0630	0.0025	0.8786	0.0096
RNN	0.0640	0.0044	0.8744	0.0174
CNN	0.0542	0.0082	0.9084	0.0304
SDAE-CNN	0.0532	0.0062	0.9124	0.0190
DMRDN	**0.0379**	0.0013	**0.9561**	0.0032

4. Results and discussion

4.1. Results

The performance comparisons of the above methods are summarized in table 4, in which the mean and standard deviation (SD) of the eight independent experimental results are recorded.

It can be seen from the results in table 4 that the proposed method outperforms other compared methods. Firstly, in comparison with the conventional machine learning methods including ANN and SVR, the deep learning methods attain better prediction accuracy results. Secondly, the RMSE and R^2 of the SDAE-CNN that integrates SDAE and CNN are 0.0532 and 0.9124, respectively, which provides a more accurate prediction performance compared with the commonly used deep learning methods (SAE, SDAE, RNN, and CNN). Thirdly, the proposed method with RMSE of 0.0379 and R^2 of 0.9561 shows better prediction accuracy than other methods, and the standard deviations of the DMRDN are smaller than other methods, which indicates it has the best performance and stability. This is because the DMRDN takes full advantage of the denoising and multiscale residual network, and is more stable and robust under noisy and nonstationary industrial conditions. The detailed prediction results on the testing data set of the proposed method and other compared methods are shown in figure 5, where the red curve represents the true values of the quality variable and the blue curves represent the predicted values of different methods.

As shown in figure 5, the predicted curve of the proposed DMRDN tracks the true curve better with smaller errors compared to the other methods. Firstly, we can see from

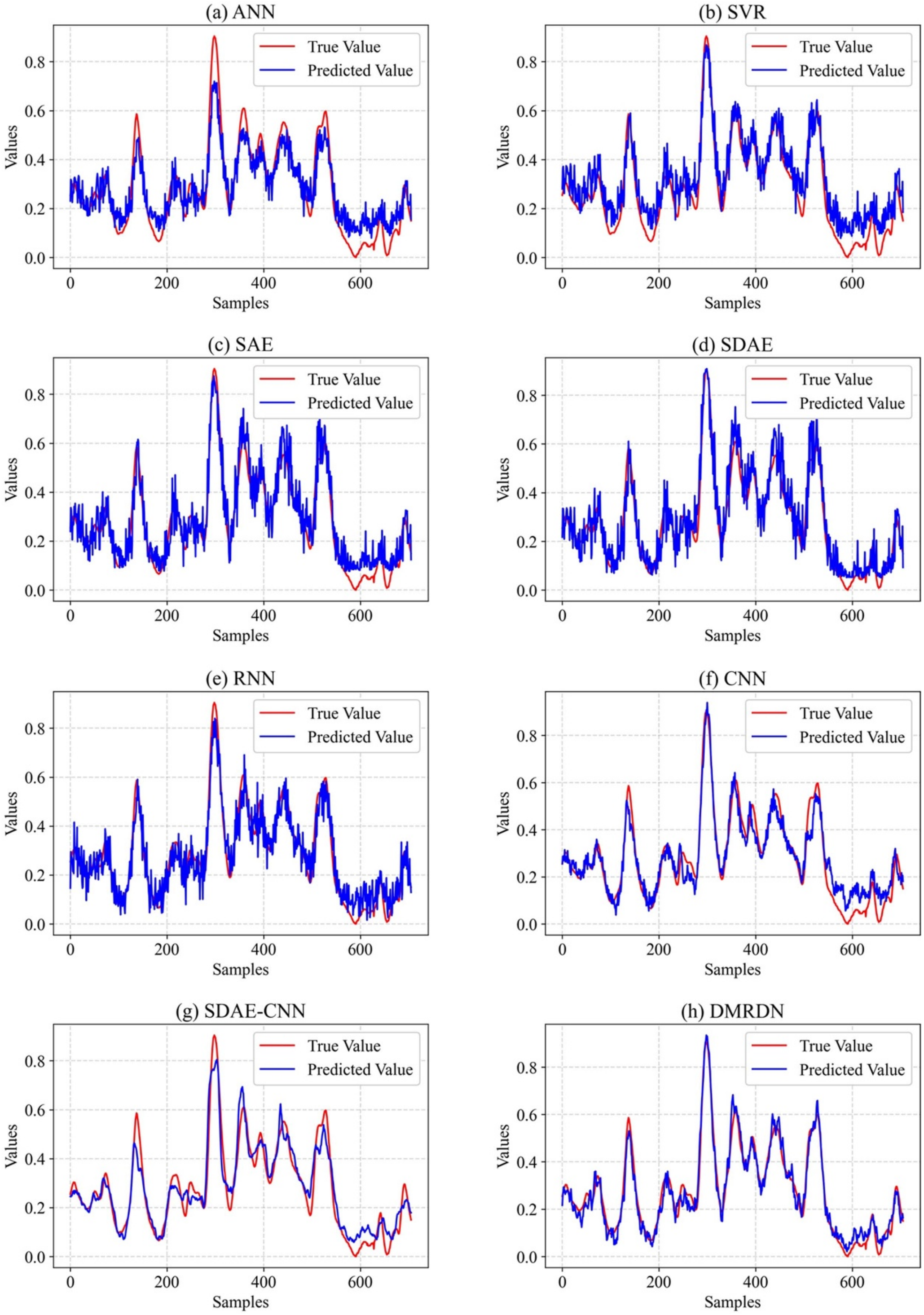

Figure 5. The predicted curves of the methods: (a) ANN, (b) SVR, (c) SAE, (d) SDAE, (e) RNN, (f) CNN, (g) SDAE-CNN, and (h) DMRDN.

figures 5(a) and (b) that the ANN and SVR do not fit the true values well, and there are large gaps between the predicted values and the true values. Secondly, as shown in figures 5(c)–(e), although the prediction curves of the SAE, SDAE, and RNN roughly track the changing trend of the true curve, the prediction curves of the three methods fluctuate greatly, which means that there are large prediction errors between the predicted values and the true values at some sample points. Thirdly, from figures 5(f) and (g), we can find that although the prediction curves of the CNN and the SDAE-CNN are smoother with less fluctuation, which may result from the filtering function of the CNN, the prediction curves do not track the true curve well,

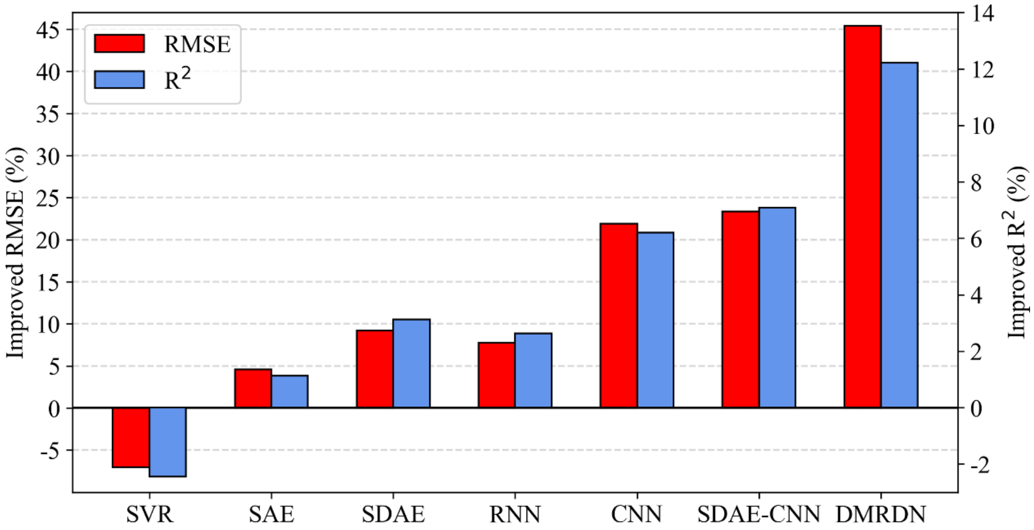

Figure 6. The improvement comparison of different methods in terms of the RMSE and R^2.

Table 5. Prediction results of ablation study.

	RMSE		R^2	
Methods	Mean	SD	Mean	SD
SDAE-MRCNN-SA	0.0442	0.0036	0.9398	0.0100
SDAE-LA-MRCNN	0.0457	0.0017	0.9361	0.0048
SDAE-LA-RCNN-SA	0.0477	0.0038	0.9302	0.0109
SDAE-LA-MCNN-SA	0.0428	0.0039	0.9437	0.0105
DMRDN	**0.0379**	0.0013	**0.9561**	0.0032

especially at some extreme points. Finally, figure 5(h) shows the prediction curve of the proposed method where it can be seen that the proposed DMRDN shows excellent performance in tracking the true trend of quality variables, and simultaneously the prediction curve has little fluctuation, which means the overall prediction error of the DMRDN is smaller compared with other methods. Therefore, it can be concluded that the proposed DMRDN has the best fitting ability and prediction accuracy.

4.2. Discussion

4.2.1. Evaluation of different benchmark methods. In order to quantitatively evaluate the superiority of the proposed DMRDN, we calculate the performance improvement percentage of the above experimental methods with the ANN as a benchmark. The improvement percentage of the RMSE and R^2 are given in figure 6.

Firstly, as shown in figure 6, SVR as a machine learning method shows poorer performance than the benchmark method, while the deep learning methods show better performance than the benchmark method. The main reason is that deep learning methods with deep network structures are more powerful in feature extraction and can learn more abstract feature representations than those with shallow structures. Secondly, comparing the improvement of the SAE and SDAE, it can be concluded that with the denoising ability, the SDAE is more suitable for soft sensors. Thirdly, compared with the benchmark method, the RNN and CNN take the dynamics of process data into consideration and their RMSE is improved by 7.78% and 21.90%, and R^2 is improved by 2.63% and 6.2%, respectively. Moreover, by combining the SDAE and CNN, the SDAE-CNN obtains a higher improvement than the two alone. This is because the SDAE cannot capture the dynamic characteristics, and the CNN directly extracts features without a special denoising operation. More importantly, the proposed DMRDN improves the most: RMSE and R^2 are improved by 45.39% and 12.22%, respectively. The DMRDN benefits from the superiority of the denoising and multiscale residual structure and will be further analyzed in subsequent sections.

4.2.2. Ablation study. To verify the contribution of each component in the proposed DMRDN, ablation experiments were conducted. For comparative experiments with the proposed method, SDAE-MRCNN-SA, SDAE-LA-MRCNN, SDAE-LA-RCNN-SA, and SDAE-LA-MCNN-SA methods were established, which removed the level-aware attention, the scale-aware attention, the multiscale kernels, and the residual connection, respectively. The mean and standard deviation (SD) of the ablation experimental results are shown in table 5.

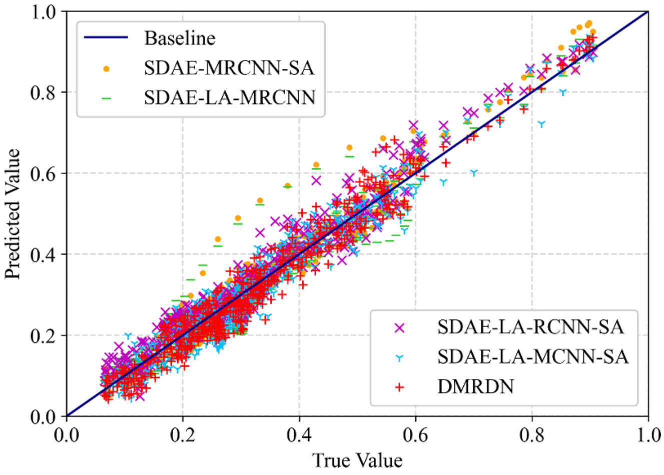

Figure 7. The scatterplot of predicted value and ground truth of the ablation study results.

The RMSE and R^2 values in table 5 demonstrate that the removal of any component results in a decrease in the overall performance of the proposed method. Specially, the accuracy of the SDAE-LA-RCNN-SA method that removes the multiscale kernels decreases the most, as its RMSE and R^2 decrease by 25.86% and 2.71%, respectively. Moreover, to further compare their fitting ability, the fitting scatter diagram of the above models is shown in figure 7.

The results from figure 7 confirm that the DMRDN performs the best in fitting the true value compared with other methods that eliminate a certain part as its scatterplot data points are distributed closest to the baseline. In conclusion, each component of the proposed method is indispensable in the entire architecture and in achieving the overall performance.

4.2.3. Evaluation of denoising effects. Since industrial process data often contain noise, the proposed DMRDN utilizes the SDAE-LA module to denoise the process data. To evaluate the denoising ability of the proposed method in different noise environments, we compared the prediction accuracy of the SAE, SDAE, RNN, CNN, MRCNN-SA (without special denoising operation), SDAE-MRCNN-SA, and the proposed DMRDN under three different noise distributions. In this section, the three different distributions of noise added to the testing data set are Gaussian noise with a mean value of 0 and standard deviation of 0.2 ($N1$), Gaussian noise with a mean value of 0.1 and standard deviation of 0.1 ($N2$), and noise with simultaneous addition of $N1$ and $N2$ ($N3$). The prediction results of different models are given in figure 8.

It is obvious from figure 8 that the proposed DMRDN achieves the best prediction performance for each noise distribution, even if the noise distributions are different from that in our DMRDN training. Firstly, since the noise $N3$ is the addition of $N1$ and $N2$, its noise distribution is more complex. The performance of each method with $N3$ is inferior to that with $N1$ and $N2$. For example, compared with the RMSE of the SAE with $N1$ and $N2$, the RMSE of the SAE with $N3$ decreases by 18.99% and 17.37%, respectively. This indicates that the noise affects the feature extraction of the model and decreases the accuracy. Secondly, the performance of the SDAE is more accurate than the SAE under each noise distribution, which shows that the features extracted by the SDAE are more robust when encountering noise due to its feature extraction ability from noise-added data. Finally, it can be seen that the prediction results of the proposed DMRDN are better than the SDAE-MRCNN-SA method in terms of the RMSE and R^2, where DMRDN obtains improvements of 21.30% and 3.23% with $N3$, respectively. This verifies the effectiveness of utilizing multilevel denoised features to predict the target values. The SDAE-MRCNN-SA denoises the process data via ordinary SDAE, which only utilizes the features of the last hidden layer. In contrast, considering the denoised features of different layers, all contribute to the prediction and the proposed method takes advantage of all hidden layers and adaptively integrates different levels of the denoised features by the developed level-aware attention, which effectively improves the performance.

Additionally, to intuitively show the importance of different denoising layers, the level-aware attention values that represent their contributions were visualized. We randomly selected six samples to calculate the average attention weights from multiple heads, and their weights are visualized in figure 9. The x-axis and y-axis of figure 9 represent the three layers used and the six samples, respectively. Each pixel represents the attention weight of the lth layer for the yth sample, and a lighter pixel indicates a larger weight.

As shown in figure 9, it can be concluded that the different layers have different weights for each prediction, and the weight of the higher layer is not always larger. For example, for the $y3$ sample, the $L1$ layer has the maximal weight while the $L3$ layer has the minimal weight, which indicates that the $L1$ layer is relatively more useful for quality prediction. For the $y6$ sample, we can observe that the $L2$ layer plays a more important role than the $L1$ layer and the $L3$ layer. Hence, integrating different denoising layers is a more effective way to predict the quality variables than just utilizing the last layer.

4.2.4. Evaluation of multiscale. Since process states generally show great variations due to nonstationary industrial conditions, it is necessary to consider the multiscale information. In the DMRDN proposed in this paper, multiscale convolutional kernels are employed to extract multiscale features from process data. To analyze the effects of different scale numbers in the DMRDN on the prediction performance, different scales ranging from two to six were considered. The experimental results of the proposed method with the different numbers of scales are shown in figure 10.

On the one hand, the results from figure 10 show that the proposed DMRDN from two to six scales is always superior to the SDAE-LA-RCNN with single scale in section 4.2.2. The single scale convolution kernel can only learn the feature representations on a certain scale, while multiscale convolution

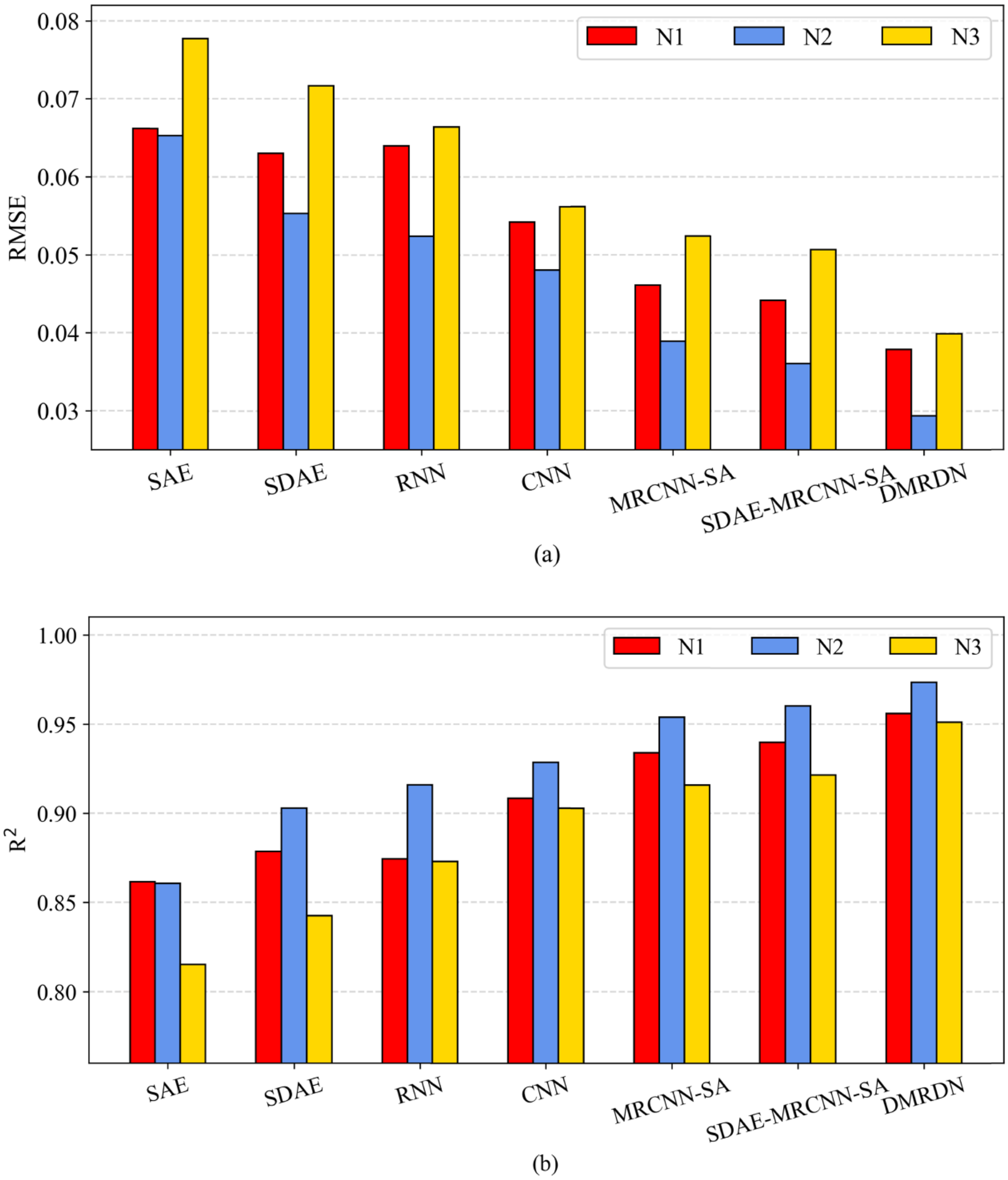

Figure 8. The (a) RMSE and (b) R^2 of different models for the three noise distributions.

kernels capture multiscale information from different time scales. On the other hand, we can observe that from twoto five scales, the RMSE and R^2 of the DMRDN increase from 0.0421 and 0.9458 to 0.0379 and 0.9561, respectively, and as more scales are adopted, the proposed method shows better prediction performance. Since the increase of in the number of scales used results in more model parameters, the proposed method accomplishes peak performance with five scales, and the accuracy decreases slightly with six scales, which may result from overfitting caused by model structures which are too complex.

Moreover, scale-aware attention is embedded in the proposed DMRDN to intuitively inspect the relationship between features of different scales and the prediction task. To visualize the importance of different scales, six samples were randomly selected to calculate the average attention values from multiple heads, which are shown in figure 11. The x-axis and y-axis of figure 11 represent the five scales used and the six samples, respectively, and each pixel represents the attention weight of the sth scale for the yth sample.

It can be seen in figure 11 that the features of different scales all play a certain role in the prediction, and the importance of the scales is distinct in predicting the quality of different samples. For example, for the $y4$ sample, the scales that have the minimal and maximal weights are $S4$ and $S5$, and their values are around 0.15 and 0.26, respectively. For the $y5$ sample and the $y6$ sample, the $S4$ and $S1$ scales are the most important, respectively. For the $y3$ sample, the weights of the five scales are similar. Therefore, it is necessary to employ multiscale kernels for multiscale dynamic feature extraction and to integrate the complementary features of different scales according to their importance.

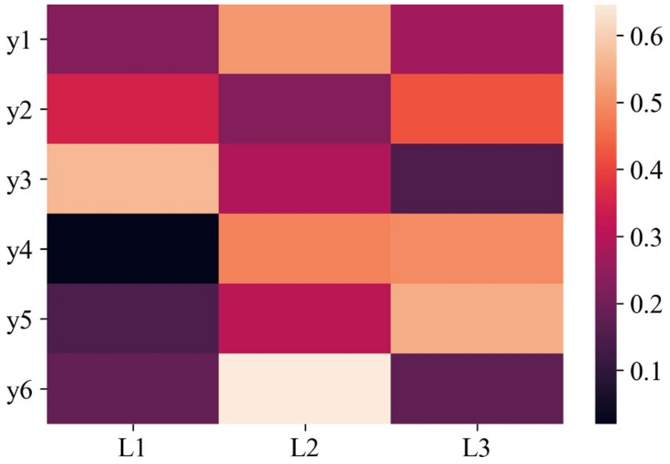

Figure 9. The level-aware attention values of six samples found by the DMRDN.

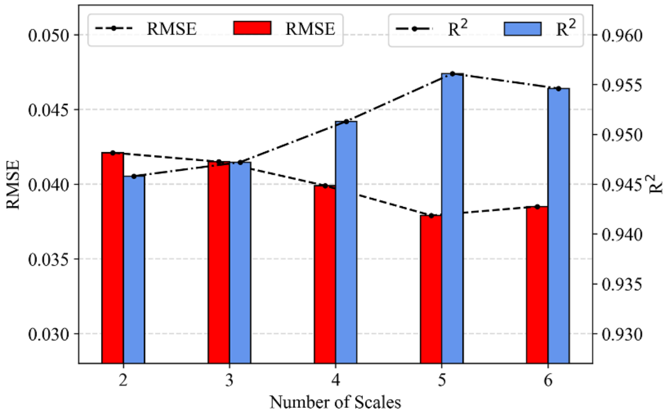

Figure 10. The RMSE and R^2 of the proposed DMRDN with different scales from two to six.

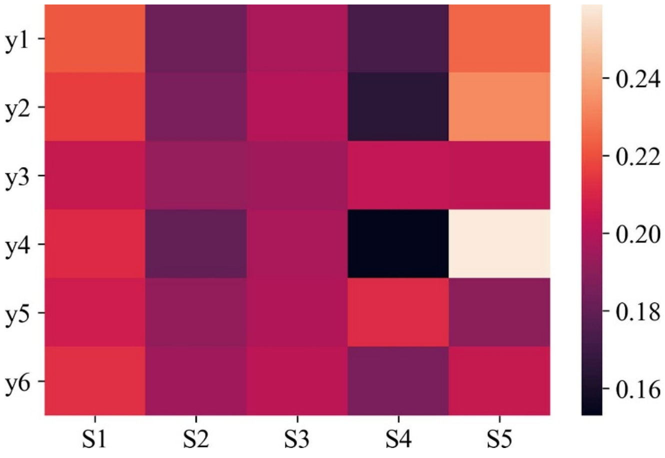

Figure 11. The scale-aware attention values of six samples found by the DMRDN.

4.2.5. Evaluation of residual connection. Residual connection is used in the proposed DMRDN to alleviate the possible degradation problem caused by the increase of CNN layers. To verify the effect of residual connection, the proposed method and the compared SDAE-LA-MCNN-SA without residual connection were tested with different depths of convolution layers. The results are shown in figure 12.

As can be seen from figure 12, for both the DMRDN and SDAE-LA-MCNN-SA, the prediction performance increases with the depth of convolution layers at the beginning. The main reason is that more convolution layers can learn more complex and abstract features that are helpful in quality prediction. However, when the depth of convolution layers of SDAE-LA-MCNN-SA exceeds four, its performance decreases, which indicates the possible degradation problem. Significantly, the performance of the proposed DMRDN continues to improve with the increase of depth from one to five, and the accuracy of the DMRDN with residual connection is always higher than SDAE-LA-MCNN-SA. Compared with the RMSE of SDAE-LA-MCNNSA with convolution layers from one to five, the RMSE of DMRDN with convolution layers from one to five increases by 0.88%, 6.18%, 9.03%, 10.51%, and 12.87%, respectively. This is because the residual network utilizes the features of previous layers and is expected to learn the residuals, which are easier to learn than the original values. Therefore, it can be confirmed that the residual connection enhances the feature extraction ability and learning efficiency of the proposed method.

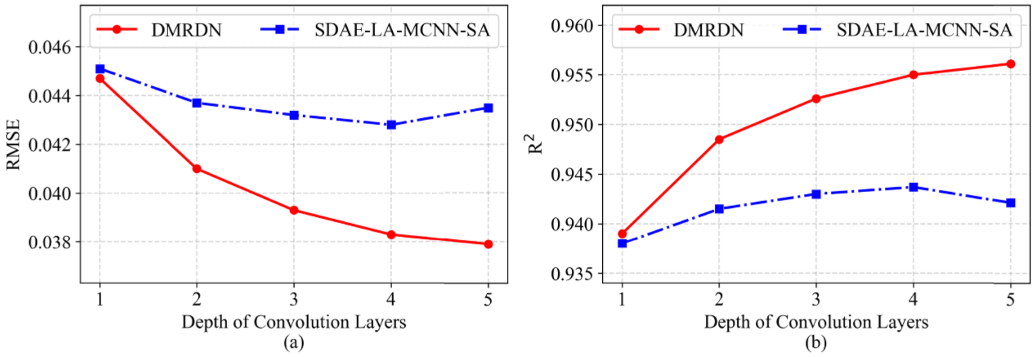

Figure 12. The (a) RMSE and (b) R^2 of DMRDN and SDAE-LA-MCNNSA with convolution layers from one to five.

5. Conclusions and future directions

In this paper, considering the noisy and nonstationary industrial conditions, a novel DMRDN was proposed for soft sensor modeling of industrial processes. Firstly, the SDAE-LA was introduced to denoise the process data, which can fuse denoised features of different levels and effectively eliminate the interference of noise. Secondly, the MRCNN-SA was designed to automatically learn and fuse deep multiscale features, which can capture complementary features from multiple timescales and enable deeper network layers for more complex features, so as to accurately predict the quality variable. Finally, a real-world data set collected from a debutanizer column was used to verify the effectiveness of the proposed method. The experimental results demonstrated that the proposed method was superior to other conventional machine learning methods and deep learning methods. In general, the proposed soft sensor modeling method is of significance in practical industrial processes for quality prediction.

Although the proposed DMRDN has achieved remarkable performance in our experiments, there are still several future research directions for this study. Firstly, as more and more complex factors appear in industrial processes, it is worthwhile to verify the proposed research framework on larger and more diverse industrial process data sets. Secondly, since the denoising operation in the proposed method is unsupervised learning, the semi-supervised learning strategy needs to be further exploited to deeply mine the information contained in abundant unlabeled samples. Thirdly, to implement better process control, soft sensors that can achieve multistep prediction are worth exploring.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China (72071062), the Science Fund for Distinguished Young Scholars of Anhui Province (2208085J12), the Anhui Provincial Key Research and Development Program (202104a05020038), and the Fundamental Research Funds for the Central Universities (PA2021KCPY0032).

ORCID iDs

Gang Wang https://orcid.org/0000-0002-6395-9409
Zhangjun Wu https://orcid.org/0000-0003-2414-5768

References

[1] Shen B and Ge Z 2020 Supervised nonlinear dynamic system for soft sensor application aided by variational auto-encoder *IEEE Trans. Instrum. Meas.* **69** 6132–42

[2] Yuan X, Feng L, Wang K, Wang Y and Ye L 2021 Deep learning for data modeling of multirate quality variables in industrial processes *IEEE Trans. Instrum. Meas.* **70** 2509611

[3] Yan W, Xu R, Wang K, Di T and Jiang Z 2020 Soft sensor modeling method based on semisupervised deep learning and its application to wastewater treatment plant *Ind. Eng. Chem. Res.* **59** 4589–601

[4] Yin X, Niu Z, He Z, Li Z and Lee D-H 2020 Ensemble deep learning based semi-supervised soft sensor modeling method and its application on quality prediction for coal preparation process *Adv. Eng. Inform.* **46** 101136

[5] Liu Y, Yang C, Gao Z and Yao Y 2018 Ensemble deep kernel learning with application to quality prediction in industrial polymerization processes *Chemometr. Intell. Lab. Syst.* **174** 15–21

[6] Guo F, Xie R and Huang B 2020 A deep learning just-in-time modeling approach for soft sensor based on variational autoencoder *Chemometr. Intell. Lab. Syst.* **197** 103922

[7] Yuan Z, Yang Z, Ling Y, Wu C and Li C 2021 Spatiotemporal attention mechanism-based deep network for critical parameters prediction in chemical process *Process Saf. Environ. Prot.* **155** 401–14

[8] Zhao Y, Ding B, Zhang Y, Yang L and Hao X 2021 Online cement clinker quality monitoring: a soft sensor model based on multivariate time series analysis and CNN *ISA Trans.* **117** 180–95

[9] Gao S Z, Li X Y, Zhang Y M and Wang J 2021 A soft-sensor model of VCM rectification concentration based on an improved WOA-RBFNN *Meas. Sci. Technol.* **32** 085104

[10] Yuan X, Qi S, Wang Y, Wang K, Yang C and Ye L 2021 Quality variable prediction for nonlinear dynamic industrial processes based on temporal convolutional networks *IEEE Sens. J.* **21** 20493–503

[11] Wang Y, Sun K, Yuan X, Cao Y, Li L and Koivo H N 2018 A novel sliding window PCA-IPF based steady-state detection framework and its industrial application *IEEE Access* **6** 20995–1004

[12] Tang Q, Li D and Xi Y 2018 A new active learning strategy for soft sensor modeling based on feature reconstruction and uncertainty evaluation *Chemometr. Intell. Lab. Syst.* **172** 43–51

[13] Li W, Zhuo Y, Bao J and Shen Y 2021 A data-based soft-sensor approach to estimating raceway depth in ironmaking blast furnaces *Powder Technol.* **390** 529–38

[14] Xie X, Sun W and Cheung K C 2015 An advanced PLS approach for key performance indicator related prediction and diagnosis in case of outliers *IEEE Trans. Ind. Electron.* **63** 2587–94

[15] Galicia H J, He Q P and Wang J 2011 A reduced order soft sensor approach and its application to a continuous digester *J. Process Control* **21** 489–500

[16] Zheng J and Song Z 2019 Mixture modeling for industrial soft sensor application based on semi-supervised probabilistic PLS *J. Process Control* **84** 46–55

[17] Gonzaga J C B, Meleiro L A C, Kiang C and Maciel Filho R 2009 ANN-based soft-sensor for real-time process monitoring and control of an industrial polymerization process *Comput. Chem. Eng.* **33** 43–49

[18] Pani A K, Amin K G and Mohanta H K 2016 Soft sensing of product quality in the debutanizer column with principal component analysis and feed-forward artificial neural network *Alex. Eng. J.* **55** 1667–74

[19] Bispo V, Scheid C M, Calçada L A and Meleiro L 2017 Development of an ANN-based soft-sensor to estimate the apparent viscosity of water-based drilling fluids *J. Pet. Sci. Eng.* **150** 69–73

[20] Lian P, Liu H, Wang X and Guo R 2020 Soft sensor based on DBN-IPSO-SVR approach for rotor thermal deformation prediction of rotary air-preheater *Measurement* **165** 108109

[21] Desai K, Badhe Y, Tambe S S and Kulkarni B D 2006 Soft-sensor development for fed-batch bioreactors using support vector regression *Biochem. Eng. J.* **27** 225–39

[22] Zhang M and Liu X 2013 A soft sensor based on adaptive fuzzy neural network and support vector regression for industrial melt index prediction *Chemometr. Intell. Lab. Syst.* **126** 83–90

[23] Yuan X F, Ou C and Wang Y L 2021 Development of NVW-SAEs with nonlinear correlation metrics for quality-relevant feature learning in process data modeling *Meas. Sci. Technol.* **32** 015006

[24] Yan X, Wang J and Jiang Q 2020 Deep relevant representation learning for soft sensing *Inf. Sci.* **514** 263–74

[25] Liu C, Wang K, Ye L, Wang Y and Yuan X 2021 Deep learning with neighborhood preserving embedding regularization and its application for soft sensor in an industrial hydrocracking process *Inf. Sci.* **567** 42–57

[26] Yan W, Tang D and Lin Y 2017 A data-driven soft sensor modeling method based on deep learning and its application *IEEE Trans. Ind. Electron.* **64** 4237–45

[27] Ba-Alawi A H, Vilela P, Loy-Benitez J, Heo S and Yoo C 2021 Intelligent sensor validation for sustainable influent quality monitoring in wastewater treatment plants using stacked denoising autoencoders *J. Water Process Eng.* **43** 102206

[28] Zhang Z, Jiang T, Li S and Yang Y 2018 Automated feature learning for nonlinear process monitoring—an approach using stacked denoising autoencoder and k-nearest neighbor rule *J. Process Control* **64** 49–61

[29] Geng Z, Chen Z, Meng Q and Han Y 2022 Novel transformer based on gated convolutional neural network for dynamic soft sensor modeling of industrial processes *IEEE Trans. Ind. Inf.* **18** 1521–9

[30] Yuan X, Li L, Shardt Y A W, Wang Y and Yang C 2021 Deep learning with spatiotemporal attention-based LSTM for industrial soft sensor model development *IEEE Trans. Ind. Electron.* **68** 4404–14

[31] Lee M, Bae J and Kim S B 2021 Uncertainty-aware soft sensor using Bayesian recurrent neural networks *Adv. Eng. Inform.* **50** 101434

[32] Wang G, Huang J and Zhang F 2021 Ensemble clustering-based fault diagnosis method incorporating traditional and deep representation features *Meas. Sci. Technol.* **32** 095110

[33] Li X, Zhang F, Wang G and Fang F 2020 Joint optimization of statistical and deep representation features for bearing fault diagnosis based on random subspace with coupled LASSO *Meas. Sci. Technol.* **32** 025115

[34] Xie W, Wang J, Xing C, Guo S, Guo M and Zhu L 2021 Variational autoencoder bidirectional long and short-term memory neural network soft-sensor model based on batch training strategy *IEEE Trans. Ind. Inf.* **17** 5325–34

[35] Guo R and Liu H 2021 Semisupervised dynamic soft sensor based on complementary ensemble empirical mode decomposition and deep learning *Measurement* **183** 109788

[36] Han Y, Fan C, Xu M, Geng Z and Zhong Y 2019 Production capacity analysis and energy saving of complex chemical processes using LSTM based on attention mechanism *Appl. Therm. Eng.* **160** 114072

[37] Zhang X and Ge Z 2020 Automatic deep extraction of robust dynamic features for industrial big data modeling and soft sensor application *IEEE Trans. Ind. Inf.* **16** 4456–67

[38] Jiang X and Ge Z 2021 Augmented multidimensional convolutional neural network for industrial soft sensing *IEEE Trans. Instrum. Meas.* **70** 2508410

[39] Wang G and Zhang F 2021 A sequence-to-sequence model with attention and monotonicity loss for tool wear monitoring and prediction *IEEE Trans. Instrum. Meas.* **70** 1–11

[40] Wang K, Shang C, Liu L, Jiang Y, Huang D and Yang F 2019 Dynamic soft sensor development based on convolutional neural networks *Ind. Eng. Chem. Res.* **58** 11521–31

[41] Yuan X, Qi S, Shardt Y A W, Wang Y, Yang C and Gui W 2020 Soft sensor model for dynamic processes based on multichannel convolutional neural network *Chemometr. Intell. Lab. Syst.* **203** 104050

[42] Guo R, Liu H, Wang W, Xie G and Zhang Y A hybrid-driven soft sensor with complex process data based on DAE and mechanism-introduced GRU *2021 IEEE 10th Data Driven Control and Learning Systems Conf. (DDCLS)2021* pp 553–8

[43] Lu C, Wang Z-Y, Qin W-L and Ma J 2017 Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification *Signal Process.* **130** 377–88

[44] Ge Z and Song Z 2010 A comparative study of just-in-time-learning based methods for online soft sensor modeling *Chemometr. Intell. Lab. Syst.* **104** 306–17

[45] Xu Y, Wang Y, Yan T, He Y, Wang J, Gu D, Du H and Li W 2021 Quality-related locally weighted soft sensing for non-stationary processes by a supervised Bayesian network with latent variables *Front. Inf. Technol. Electron. Eng.* **22** 1234–46

[46] Liu T, Chen S, Liang S, Du D and Harris C J 2020 Fast tunable gradient RBF networks for online modeling of nonlinear and nonstationary dynamic processes *J. Process Control* **93** 53–65

[47] Srivastava R K, Greff K and Schmidhuber J 2015 Highway networks (arXiv:1505.00387)

[48] He K M, Zhang X Y, Ren S Q and Sun J and IEEE (eds) 2016 Deep residual learning for image recognition *2016 IEEE*

Conf. on Computer Vision and Pattern Recognition (CVPR) (*Seattle*, *27–30 June*) p WA2016

[49] Vincent P, Larochelle H, Bengio Y and Manzagol P-A 2008 Extracting and composing robust features with denoising autoencoders *Proc. 25th Int. Conf. on Machine learning—ICML '082008* pp 1096–103

[50] Sun Q Q and Ge Z Q 2021 Deep learning for industrial KPI prediction: when ensemble learning meets semi-supervised data *IEEE Trans. Ind. Inf.* **17** 260–9

[51] Feng L, Zhao C and Sun Y 2021 Dual attention-based encoder-decoder: a customized sequence-to-sequence learning for soft sensor development *IEEE Trans. Neural Netw. Learn. Syst.* **32** 3306–17

[52] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A N, Kaiser L and Polosukhin I (eds) 2017 Attention is all you need *31st Annual Conf. on Neural Information Processing Systems (NIPS)* (*Long Beach*, 4–9 *December* 2017) p CA2017

[53] Yuan X, Qi S, Wang Y and Xia H 2020 A dynamic CNN for nonlinear dynamic feature learning in soft sensor modeling of industrial process data *Control Eng. Pract.* **104** 104614

[54] Fortuna L, Salvatore G P D, Alessandro R P D and Maria G X P D 2007 *Soft Sensors for Monitoring and Control of Industrial Processes* 1st edn vol 22 Advances in Industrial Control (AIC) (London: Springer) pp XVIII, 271

[55] Fortuna L, Graziani S and Xibilia M G 2005 Soft sensors for product quality monitoring in debutanizer distillation columns *Control Eng. Pract.* **13** 499–508