

# Nonlinear Dynamic Soft Sensor Development with a Supervised Hybrid CNN-LSTM Network for Industrial Processes

Jiaqi Zheng, Lianwei Ma, Yi Wu, Lingjian Ye, and Feifan Shen\*

Cite This: *ACS Omega* 2022, 7, 16653–16664

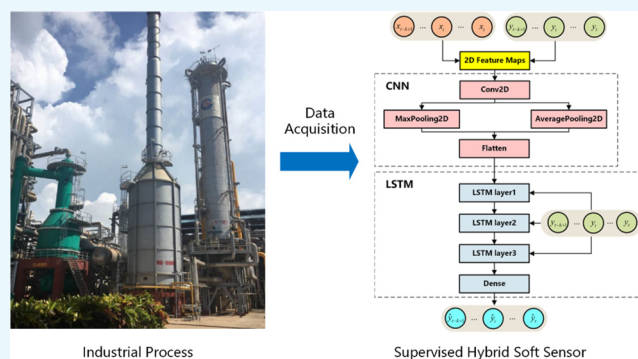
Read Online

ACCESS |

Metrics &amp; More

Article Recommendations

**ABSTRACT:** A soft sensor is a key component when a real-time measurement is unavailable for industrial processes. Recently, soft sensor models based on deep-learning techniques have been successfully applied to complex industrial processes with nonlinear and dynamic characteristics. However, the conventional deep-learning-based methods cannot guarantee that the quality-relevant features are included in the hidden states when the modeling samples are limited. To address this issue, a supervised hybrid network based on a dynamic convolutional neural network (CNN) and a long short-term memory (LSTM) network is designed by constructing multilayer dynamic CNN-LSTM with improved structures. In each time instant, data augmentation is implemented by dynamic expansion of the original samples. Moreover, multiple supervised hidden units are trained by adding quality variables as part of the layer input to acquire a better quality-related feature learning performance. The effectiveness of the proposed soft sensor development is validated through two industrial applications, including a penicillin fermentation process and a debutanizer column.



## 1. INTRODUCTION

In practical industry, it is necessary and important to explore the real-time indices of product quality to ensure the safety and efficiency of industrial processes. However, key variables indicating the quality are often difficult to measure, while other ordinary process variables can be routinely collected.<sup>1</sup> These quality variables need to be inspected by laboratory analyzers, which are costly and are unable to meet the demand of real-time measurement. In addition, the first-principles model is becoming extremely difficult to obtain due to the complexity of modern industries.<sup>2–4</sup> To alleviate the aforementioned problems, soft sensors have been developed to implement the online measurement of quality variables by constructing inherent data-driven models based on those easy-to-measure process variables.

In the past decades, virtual sensing techniques have been successfully applied to many industrial processes, bringing about accurate real-time prediction results of quality variables.<sup>5,6</sup> Principal component analysis (PCA) and partial least squares (PLS) are typical data-driven methods to establish soft sensor models.<sup>7–11</sup> In order to cope with the nonlinear feature of industrial processes, several developments have been made on the basis of PCA and PLS for soft sensor modeling. A typical strategy is constructing a kernel function to establish the nonlinear mapping between the original data space and a high-dimensional space.<sup>12,13</sup> Therefore, a linear regression model can be constructed in the high-dimensional

space, where nonlinear optimization can be avoided and the data characteristic can be captured. Other methods beyond the framework of PCA and PLS are also designed to meet the demand of quality prediction for nonlinear processes such as support vector regression (SVR),<sup>14,15</sup> Gaussian process regression (GPR),<sup>16–18</sup> and an artificial neural network (ANN).<sup>19,20</sup> ANN is one of the most widely virtual sensing modeling methods and can give a promising performance of nonlinear approximation and adaptive learning. In terms of the advantages, the quality prediction results by ANN for nonlinear processes with a large data scale are fairly good. Unfortunately, ANN-based soft sensor models often confront the gradient vanishing and exploding problem during the training procedure when the network structure is complicated.

To resolve this limitation and improve the robustness of quality prediction for nonlinear processes, soft sensors based on deep-learning techniques have been developed in recent years.<sup>21</sup> A deep belief network (DBN),<sup>22–24</sup> a stacked autoencoder (SAE),<sup>25–27</sup> and a convolutional neural network

Received: February 24, 2022

Accepted: April 18, 2022

Published: May 2, 2022



(CNN)<sup>28–31</sup> are practical tools to construct soft sensor models for nonlinear processes. In comparison to conventional soft sensor models, the deep-learning-based models can provide a better quality prediction performance for nonlinear processes. Further improvements have been made by scholars on the basis of the original deep-network structures to handle other process characteristics. For example, Sun and Ge developed a novel soft sensor model based on a gated stacked target-related autoencoder (GSTAE) by adding prediction errors of target values into the loss function when the pretraining procedure was executed, which improved the prediction performance in comparison to the conventional SAE-based soft sensor models.<sup>32</sup> Yuan et al. proposed a soft sensor framework with a multichannel CNN model to capture local correlations of distant process variables.<sup>33</sup> In addition, the dynamic nature of a process is another common and important issue for soft sensing modeling. Generally, variable trajectories of practical processes present correlations along a time index, which results in the coexistence of process nonlinearity and dynamics. To address this problem, dynamic soft sensor models have been designed to obtain an accurate prediction performance for complex processes. For example, He et al. developed a soft sensor model using a dynamic extreme learning machine (ELM) by adding a special linear hidden layer node based on the traditional ELM.<sup>34</sup> Lately, a novel soft sensor development using an echo state network (ESN) integrated with a singular value decomposition was proposed and applied to complex chemical processes.<sup>35</sup> In addition, a recurrent neural network (RNN) has also been introduced to construct nonlinear dynamic soft sensors for quality prediction.<sup>36</sup> Although RNN is a mainstream deep-learning model, it still suffers from the problem of gradient vanishing and exploding due to the “tanh” activation function. For an improvement of the network structure, a long short-term memory (LSTM) neural network has been developed to overcome the deficiency of RNN.<sup>37</sup> The long-term memory is taken into consideration for LSTM, which is able to describe the time-series model more accurately with more parameters in comparison to RNN. So far, LSTM-based soft sensors have been successfully designed and applied to different industrial processes with both nonlinear and dynamic properties.<sup>38,39</sup> However, soft sensor models based on the conventional RNN and LSTM structures are unsupervised, which means that the quality information may not be exploited in the hidden units. To make full use of the quality data, a soft sensor model based on a dynamic neural network named nonlinear autoregression with exogenous input (NARX) was designed.<sup>40</sup> Under the supervised framework, the correlations between ordinary process variables and quality variables can be extracted properly by hidden layers. When a multilayer perceptron is implemented with NARX, the quality variables are utilized as part of the model input. However, the quality variables are not directly used for the intermediate hidden layers that are not connected to the input layer. Further development can be conducted under the supervised framework. By stacking multiple layers in a hierarchical way and adding quality variables to each hidden layer through hierarchical learning, the stacked network is able to enhance the extraction of deep quality-relevant characteristics that are beneficial for quality prediction.<sup>41</sup> To make full use of the quality-relevant information, improved supervised soft sensor models based on deep learning have been developed.<sup>42</sup> Quality variables are employed as part of the layer input, where the model parameters of each hidden layer are determined by both

the quality variables and process variables. The supervised soft sensor framework has been proved to be effective to deal with the quality prediction problem for nonlinear dynamic processes. Although the deep-learning-based supervised soft sensors can provide acceptable prediction performance for nonlinear dynamic processes, there are still some limitations when they face complicated practical processes. First, most dynamic soft sensor models focus on the temporal correlations of process data, where the feature of local correlations is not extracted adequately. As a consequence, effective information and potential relationships of discontinuous data may be ignored and the accuracy of soft sensor models will be influenced. Meanwhile, the sampling interval varies between different processes and the scale of modeling samples can be small. Thus, data augmentation is a necessary and important strategy to describe process characteristics thoroughly.

In light of the aforementioned problems, a supervised dynamic CNN-LSTM (SDCNN-LSTM) network has been designed to construct the soft sensor model for complex industrial processes with nonlinear and dynamic features. The major procedure and contributions of the proposed method are demonstrated as follows. First, quality variables are prepared for the original unsupervised layers, where the quality-relevant features can be better captured from each hidden layer. Second, a data augmentation strategy is designed after the input layer by expanding the original one-dimensional (1D) samples into two-dimensional (2D) feature maps. Hence, the scale of modeling data is enlarged and the temporal correlations remain, which is adopted to solve the problem of data deficiency. Finally, the hybrid dynamic CNN-LSTM network is constructed on the basis of the supervised framework with data augmentation. In summary, the advantages of both CNN and LSTM networks can be used for nonlinear dynamic processes, where the data augmentation strategy and the full utilization of quality information will help to improve the accuracy of the soft sensor model.

The rest of the paper is organized as follows. Section 2 illustrates some basics of CNN and LSTM networks. Then, the detailed framework of the SDCNN-LSTM soft sensor is demonstrated in Section 3. Two applications are introduced to evaluate the performance of the proposed soft sensor development in Section 4, including a penicillin fermentation process and a debutanizer column. Finally, conclusions are made in Section 5.

## 2. BACKGROUND

**2.1. Convolutional Neural Network.** CNN is a typical feed-forward neural network, as well as a multilayer representative deep-learning algorithm. The core idea of CNN is the scheme of local connection, weight sharing, and pooling. By modeling strategies based on CNN, significant features of the original data can be extracted spontaneously to implement target identification, classification, and recognition. For different CNN frameworks such as LeNet-5 and its improved form AlexNet, the particular network structures vary from one network to another. A common feature of these network structures is that both networks consist of four major layers, which are the input layer, the convolutional layer, the subsampling layer, and the output layer.

For the input layer, usually a 2D data matrix is collected from the raw image or sequential data set. For the convolutional layer, the layer input is the output of the previous layer. Then, the layer input is operated by

convolution kernels to form several feature maps, while the number of feature maps is equal to the convolutional kernels. The size of a convolutional kernel for 2D input can be 1D or 2D with fixed kernel weights. A simple and specific case of the convolutional operation is illustrated in Figure 1. It can be

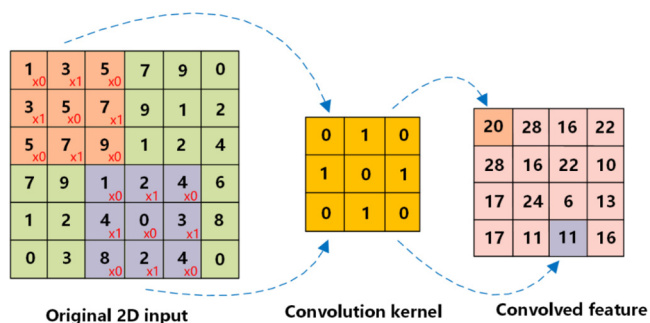


Figure 1. Convolutional operation with a 6 × 6 input map, 3 × 3 convolution kernel, 1 stride size, and no padding.

inferred from Figure 1 that the dimension of the original input matrix is reduced and the convolved feature is extracted after the operation through a convolution kernel. For the convolutional operation, the height  $h_o$  and width  $w_o$  of the output can be denoted as

$$h_o = (h_i - h_k + 2p) / s + 1 \tag{1}$$

$$w_o = (w_i - w_k + 2p) / s + 1 \tag{2}$$

where  $h_i$  and  $w_i$  are the height and width of the input data, respectively,  $h_k$  and  $w_k$  are the height and width of the convolutional kernel, respectively,  $p$  is the padding size, and  $s$  is the stride size.

After the convolutional operation, the rectified liner unit (ReLU) function defined in eq 1 is adopted as the activation function for the feature maps, which is able to remarkably improve the learning efficiency and nonlinear representation.

$$f(x) = x^+ = \max(0, x) \tag{3}$$

For the subsampling layer, the pooling strategy is implemented and often works after the convolutional layer. Similarly to the convolutional layer, the feature of the local connection is extracted in the subsampling layer. Differently, the pooling rule is predefined and no extra parameters are required in the model training procedure. Figure 2 shows two types of pooling approaches as the max pooling strategy and

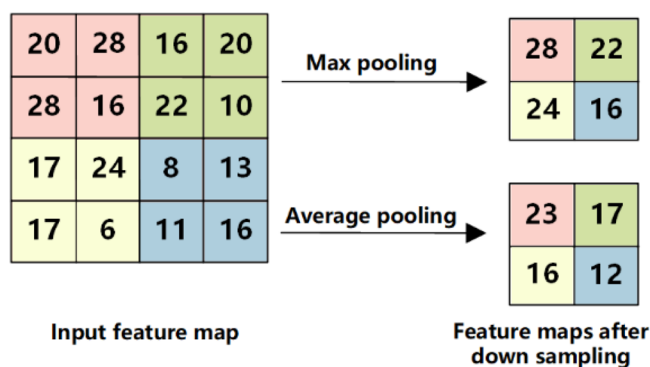


Figure 2. Two pooling strategies with a 4 × 4 input map, 2 × 2 pooling size, and 2 stride size.

the average pooling strategy, which are widely used to construct the subsampling layer. Hence, the scale of the feature maps is reduced while the representative data features can be preserved.

For the output layer, the feature extraction results from different channels should be concatenated into a single vector. By this means, the output layer is also called as connection layer, which establishes connections among different feature maps as well as the final model output. As a result, an activation function is also required to achieve a specific purpose such as classification and regression.

**2.2. Long Short-Term Memory.** A long short-term memory (LSTM) network is developed on the basis of a recurrent neural network (RNN). Although RNN has an advantage in handling dynamic processes, the gradient vanishing problem of RNN often influences the accuracy of modeling. In comparison with RNN, LSTM is able to avoid the aforementioned issue by designing the cell and gate structure. The structure of the single-layer LSTM network is demonstrated in Figure 3.

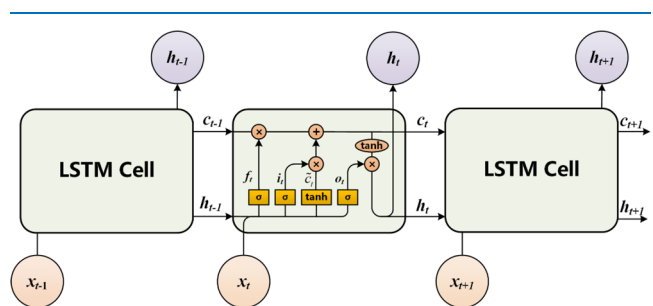


Figure 3. Network structure of the single-layer LSTM.

Three gate structures, including the input gate  $i_t$ , the forget gate  $f_t$ , and the output gate  $o_t$ , are defined on the basis of the LSTM cell, which are described as

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{4}$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{5}$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{6}$$

where  $\sigma(\cdot)$  denotes the sigmoid activation function that  $\sigma(x) = \frac{1}{1 + e^{-x}}$ ,  $W_{**}$  are the weighting parameters and  $b_*$  are the bias parameters of different gate structures,  $x_t$  is the model input at time index  $t$ , and  $h_{t-1}$  is the LSTM hidden state at time instant  $t - 1$ .

Then,  $\tilde{c}_t$  defines what features of the cell input should be kept using the tanh function as

$$\tilde{c}_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{7}$$

Hence, the cell state  $c_t$  can be determined with the aforementioned network structures as

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{8}$$

where  $\odot$  denotes the pointwise multiplication, which indicates that the current cell state is the weighted combination of the previous cell state and the current cell input.

On the basis of the current cell state, the hidden state at time instant  $t$  can be calculated as

$$h_t = o_t \odot \tanh c_t \tag{9}$$

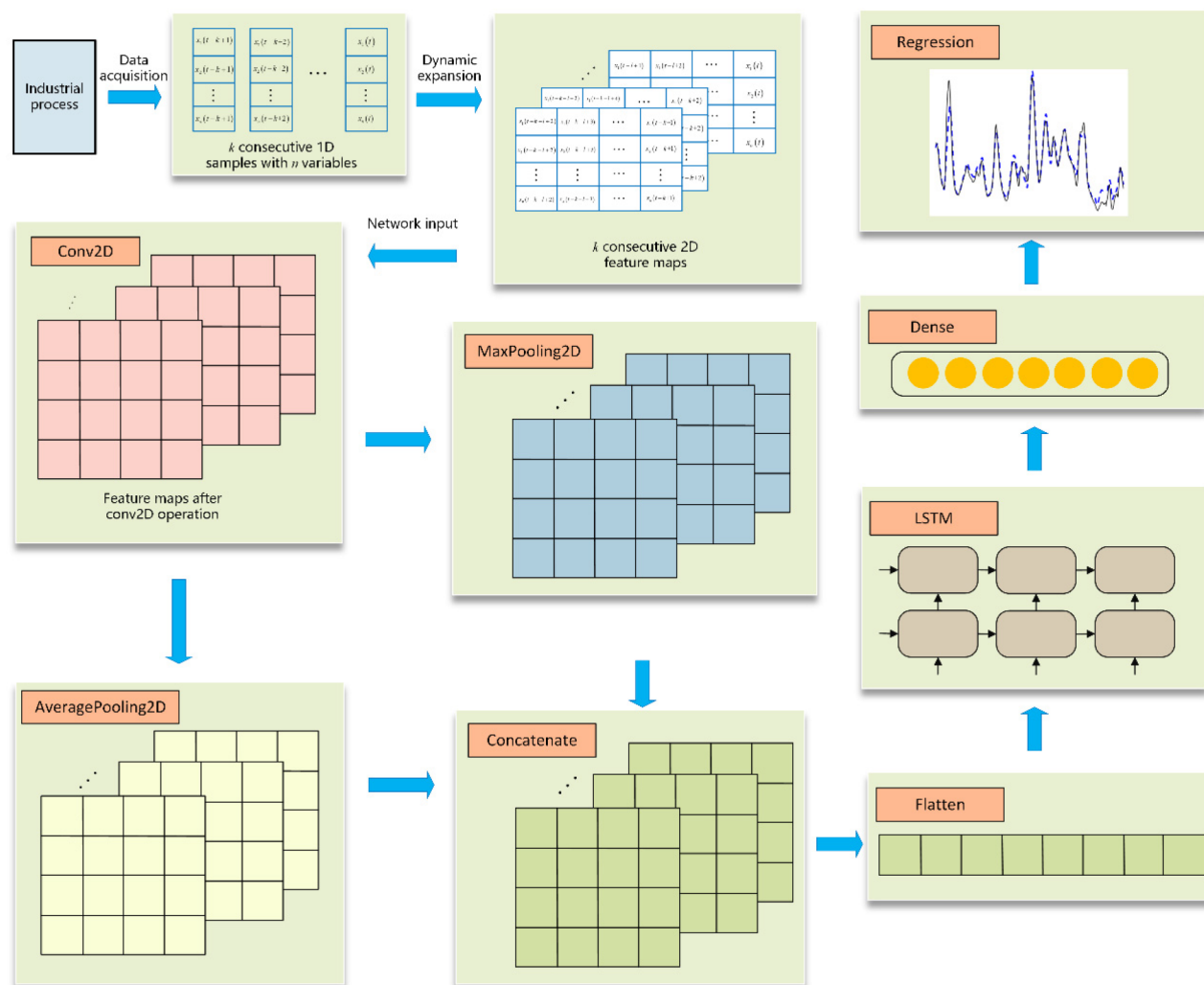


Figure 4. Design procedure of the hybrid DCNN-LSTM network.

When multiple LSTM layers are constructed to form a deep neural network, the hidden state will become the cell input of the next layer. To implement the regression task, the estimated model output  $\hat{y}_t$  is usually connected with the hidden state  $h_t$  by the sigmoid activation function as

$$\hat{y}_t = \sigma(W_y h_t + b_y) \quad (10)$$

On consideration of the particular characteristic of the sequential data, the back-propagation through time (BPTT) algorithm is usually used to train the LSTM-based network, as presented in the Appendix.

### 3. SOFT SENSOR DEVELOPMENT BASED ON SUPERVISED DCNN-LSTM NETWORK

**3.1. Hybrid Dynamic CNN-LSTM Network.** Although CNN is a useful technique to extract the latent features of the original data and reduce the data complexity, it is usually applied to image processing such as image classification and recognition. As a widely used deep-learning algorithm, CNN is able to deal with process nonlinearity effectively. Due to this advantage, the problem of the nonlinear property of industrial processes can be automatically resolved by the CNN modeling strategy. However, a single CNN soft sensor model may neglect the time-series correlations of sequential data while the

prediction task is conducted. To address the problem, the LSTM network is concatenated to the CNN layer. Different from the existing literature, which takes the 1D sequential samples as the model input of the hybrid CNN-LSTM network, a deep DCNN-LSTM structure is developed in this work.

The proposed DCNN-LSTM network structure is illustrated in Figure 4. It can be inferred that there are two parts involved in the hybrid network. The DCNN layers are designed to extract the features of sequential data, while the LSTM layers are developed to predict the process quality. The original 1D samples are expanded to 2D feature maps through a data augmentation step. To improve the reliability of the down-sampling stage, both the max pooling and average pooling strategies are adopted with a concatenating operation. After the pooling layer, a flatten layer is connected to the network to unfold the data into 1D form. Then, two LSTM layers are added after the DCNN network. In addition, both LSTM layers are followed by a dropout layer to avoid the overfitting problem during the training procedure. After the LSTM structures, a fully connected layer is designed as the weighted sum of the previous network output. Finally, a regression layer is generated as the model output.

The “dynamic” characteristic of the proposed network is reflected in two aspects. First of all, a deep neural network with

the LSTM structure has the capacity to extract the dynamic feature of processes, which means that a CNN-LSTM-based network is a dynamic model in essence. Meanwhile, it is noted that the model input of this network in each time instant is a 2D feature map instead of the conventional 1D sample vector for sequential data modeling. The moving window strategy is used to expand the original 1D vector  $\mathbf{x}(t) = [x_1(t), x_2(t), \dots, x_n(t)]^T$  to the 2D dynamic matrix as

$$\mathbf{X}(t) = [\mathbf{x}(t-l+1), \mathbf{x}(t-l), \dots, \mathbf{x}(t)]$$

$$= \begin{bmatrix} x_1(t-l+1), x_1(t-l), \dots, x_1(t) \\ x_2(t-l+1), x_2(t-l), \dots, x_2(t) \\ \vdots \\ x_n(t-l+1), x_n(t-l), \dots, x_n(t) \end{bmatrix} \quad (11)$$

where  $n$  is the number of process variables and  $l$  is the expanding length of the original vectors. Thus, the model input of the DCNN-LSTM network can be denoted as  $\{\mathbf{X}(t-k+1), \mathbf{X}(t-k), \dots, \mathbf{X}(t)\}$ , where  $k$  is the modeling length of the sequential data.

### 3.2. Soft Sensor with Supervised DCNN-LSTM Model.

The LSTM network has been proven to be an excellent framework for the soft sensor modeling of nonlinear dynamic processes. To implement the quality prediction scheme, the values of process variables are usually collected as the model input, while the key variable that is difficult to directly measure is regarded as the model output. However, the feature of the quality variable is often ignored during the prediction process, since most of the soft sensor models are unsupervised. To overcome the aforementioned deficiency and make full use of the obtained quality data, it is necessary to construct a supervised soft sensor model, where the state of the quality variable should be exploited as part of the model input.

A case of the overall network structure of the proposed SDCNN-LSTM model with one CNN layer and three LSTM layers is presented in Figure 5. To implement the supervised framework, the quality variables are first introduced to form the 2D feature maps with other input variables during the data augmentation procedure. Thus, the quality information is preserved in the CNN layer when local correlations of variables are extracted. Moreover, in the LSTM structure, the quality variables are utilized as part of the cell input in each LSTM unit. The LSTM network structure of this model can be modified as

$$i_t = \sigma(W_{hi}h_{t-1} + W_{yi}y_t + b_i) \quad (12)$$

$$f_t = \sigma(W_{hf}h_{t-1} + W_{yf}y_t + b_f) \quad (13)$$

$$o_t = \sigma(W_{ho}h_{t-1} + W_{yo}y_t + b_o) \quad (14)$$

$$\tilde{c}_t = \tanh(W_{hc}h_{t-1} + W_{yc}y_t + b_c) \quad (15)$$

Therefore, the quality information is involved in the training procedure of LSTM layers, where the feature extraction performance of quality-relevant features for nonlinear dynamic processes can be improved significantly.

It is worth mentioning that the current state of the quality variable is unable to be obtained during the online prediction procedure, since it is definitely the model output at the same time. To conduct the supervised modeling and real-time prediction process, an initialization step has been carried out to

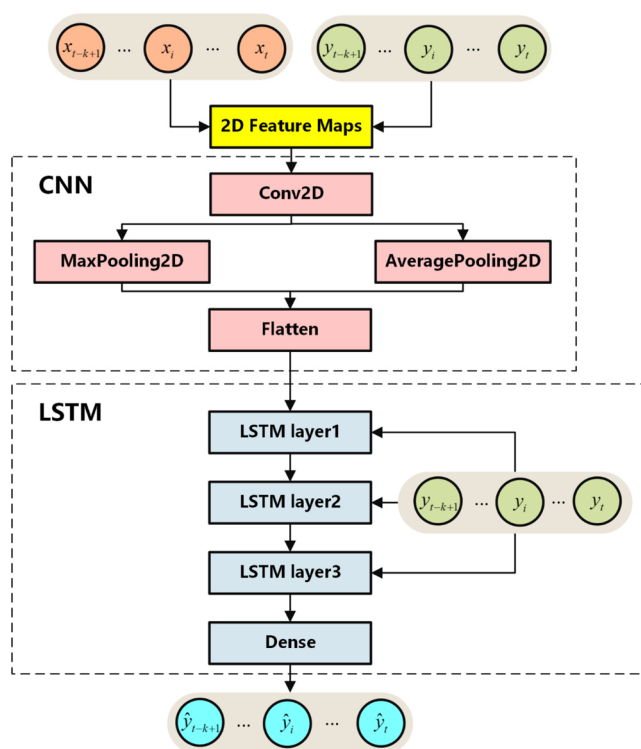


Figure 5. Network structure of the SDCNN-LSTM soft sensor.

estimate the current output as  $\tilde{y}(t) = y(t-1)$ . Therefore, the complete procedure of the proposed supervised DCNN-LSTM-based soft sensor framework can be summarized as Figure 6.

According to Figure 6, the training stage of the proposed method can be summarized as

- (1) collect the training data set  $\{\mathbf{x}(t), y(t)\}$  and conduct variable-wise normalization
- (2) augment the original 1D training samples to the 2D dynamic matrices
- (3) determine the network structures and model hyperparameters
- (4) train the SDCNN-LSTM soft sensor model with the predefined hyperparameters and the training data set

Then, the prediction stage can be implemented on the basis of the trained soft sensor model as

- (1) collect the testing data set  $\{\mathbf{x}(t), \tilde{y}(t)\}$ , where  $\tilde{y}(T) = y(T-1)$  is the estimated output at time instant  $T$
- (2) conduct the data normalization step on the basis of the result of the training samples
- (3) expand the original 1D testing samples to the 2D dynamic matrices
- (4) predict the current quality variable  $\hat{y}(T)$  on the basis of the trained SDCNN-LSTM soft sensor model
- (5) move to the next online prediction stage with  $T = T + 1$

To train the proposed soft sensor model, the mean squared error (MSE) is utilized as the cost function

$$\text{MSE} = \sum_{i=1}^n (\hat{y}(i) - y(i))^2 / n \quad (16)$$

where  $n$  is the number of the training samples,  $i$  is the index of the training samples,  $y(i)$  is the actual value of the key variable, and  $\hat{y}(i)$  is the prediction result of the key variable. In this

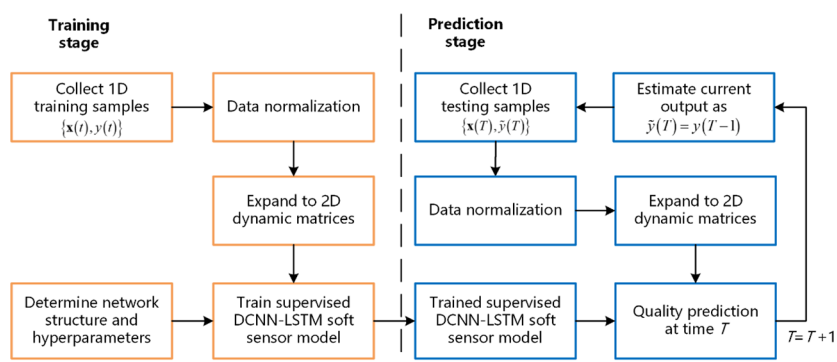


Figure 6. Flow diagram of the SDCNN-LSTM soft sensor.

work, the Adam algorithm is used to minimize the cost function during the training stage.

To evaluate the performance of the soft sensor model, the root-mean-squared error (RMSE) is calculated for the testing data set

$$\text{RMSE} = \sqrt{\frac{\sum_{j=1}^k (\hat{y}(j) - y(j))^2}{k}} \quad (17)$$

where  $k$  is the number of the testing samples,  $j$  is the time index, and  $y(j)$  and  $\hat{y}(j)$  are the actual and predicted values of the testing quality variables, respectively. A smaller value of RMSE will indicate that the general error of prediction is simultaneously less.

In addition, the coefficient of determination  $R^2$  is also calculated for the testing data set

$$R^2 = 1 - \frac{\sum_{j=1}^k (\hat{y}(j) - y(j))^2}{\sum_{j=1}^k (y(j) - \bar{y}(j))^2} \quad (18)$$

where the statistical analysis of the residual space is carried out and a larger value of  $R^2$  will indicate a more accurate prediction performance. The calculation result of  $R^2$  is able to reveal the total variance in the residual space and the related information carried in the testing output.

From the advantages of the original CNN-LSTM network and other dynamic soft sensor models, the proposed supervised network provides two main developments. The first improvement is the data augmentation strategy expanding the original 1D samples to the 2D feature maps, by which the problem of data deficiency can be resolved.<sup>31</sup> By construction of the 2D feature maps, two types of correlations are involved. One is the variablewise correlations between variables. The other is the temporal autocorrelations of variables along the time index. Thus, both local nonlinear spatial and dynamic feature hierarchies can be learned from the massive unlabeled data using local patches with convolution and pooling operators layer by layer. Therefore, the scale of the modeling data is enlarged, where both the variablewise and temporal correlations that are difficult to learn for the 1D-data-based model can now be extracted properly. Another contribution is the design of the supervised network, where the quality variables are fully used as the input of each hidden layer. In comparison to the traditional supervised dynamic networks such as NARX, the quality information is permeated into the entire network structure by the proposed model, which is able to extract more abundant quality-related information within

the hidden units. The determination of the hyperparameters is conducted by trial and error. The limitation of the current work is that the selected hyperparameters may not reach the optimal values.

## 4. RESULTS AND DISCUSSION

**4.1. Penicillin Fermentation Process.** The fed-batch penicillin fermentation process is a typical biochemical process

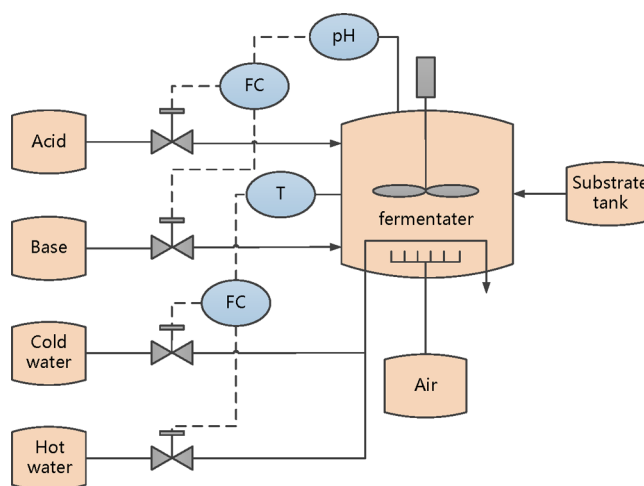


Figure 7. Flowchart of the penicillin fermentation process.

Table 1. Process Variables of the Penicillin Process

variable	description
$x_1$	aeration rate
$x_2$	agitator power
$x_3$	substrate feed rate
$x_4$	substrate temperature
$x_5$	substrate concentration
$x_6$	dissolved oxygen concentration
$x_7$	biomass concentration
$x_8$	culture volume
$x_9$	CO <sub>2</sub> concentration
$x_{10}$	pH
$x_{11}$	generated heat
$x_{12}$	cooling water flow rate

with both nonlinear and dynamic characteristics, which is widely used as a benchmark platform for research on soft sensor modeling, fault diagnosis, real-time control, and production optimization of industrial processes. The flowchart

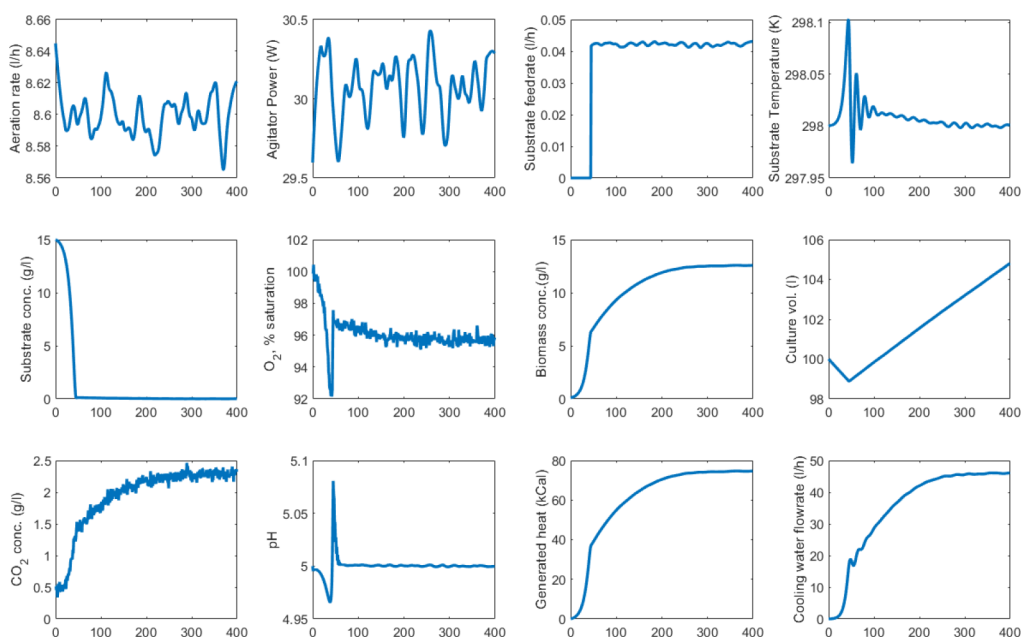


Figure 8. Variable trajectories of the penicillin process.

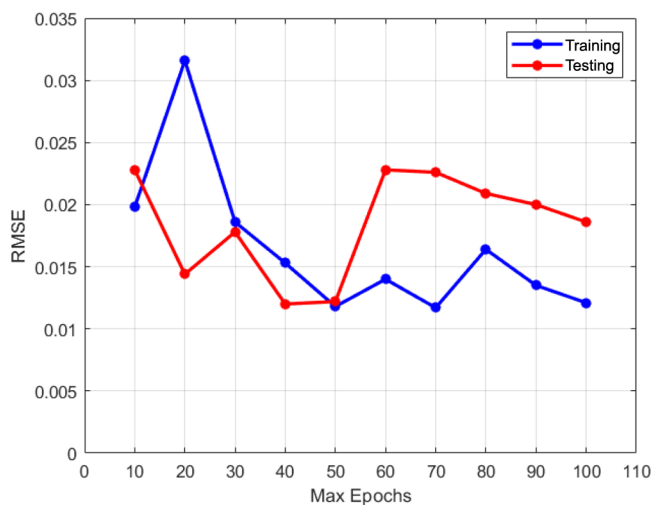


Figure 9. RMSEs of the penicillin process with different maximum epochs.

of the penicillin fermentation process is presented in Figure 7. The penicillin fermentation process consists of three operating phases. During the preculture phase, the biomass reactants are growing for the preparation of the reaction up to the critical concentration. Then, the penicillin concentration begins to increase rapidly at the second phase, where the penicillin

Table 3. RMSEs of the Penicillin Process by Different Methods

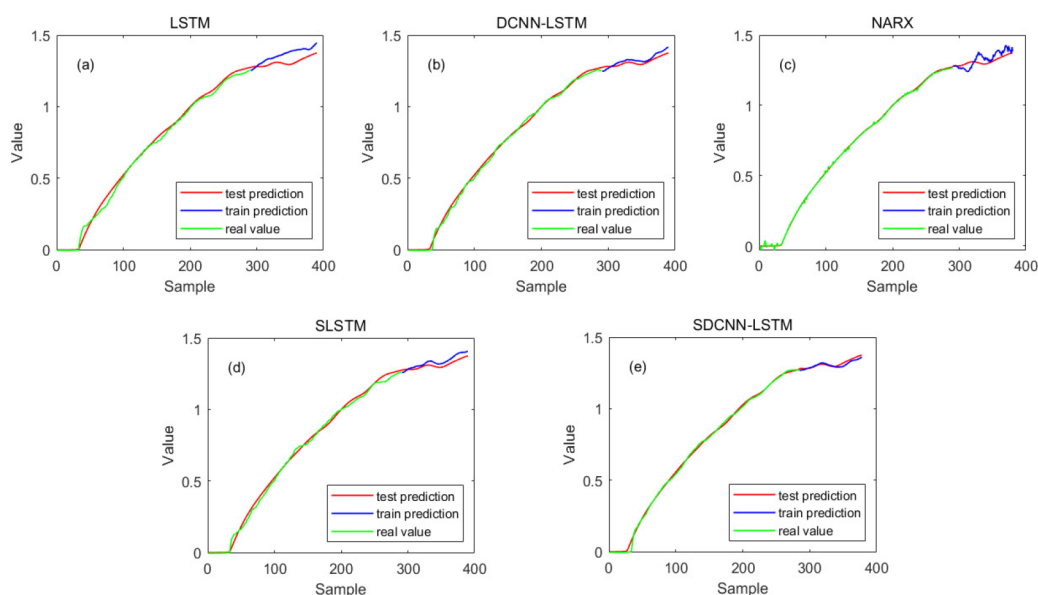
	LSTM	DCNN-LSTM	NARX	SLSTM	SDCNN-LSTM
RMSE	0.0573	0.0219	0.0390	0.0263	0.0122
R <sup>2</sup>	0.6772	0.8790	0.8273	0.9055	0.9761

production rate reaches its peak. At the final stage, the production rate of penicillin decreases due to the consumption of the biomass reactants until the end of a batch.

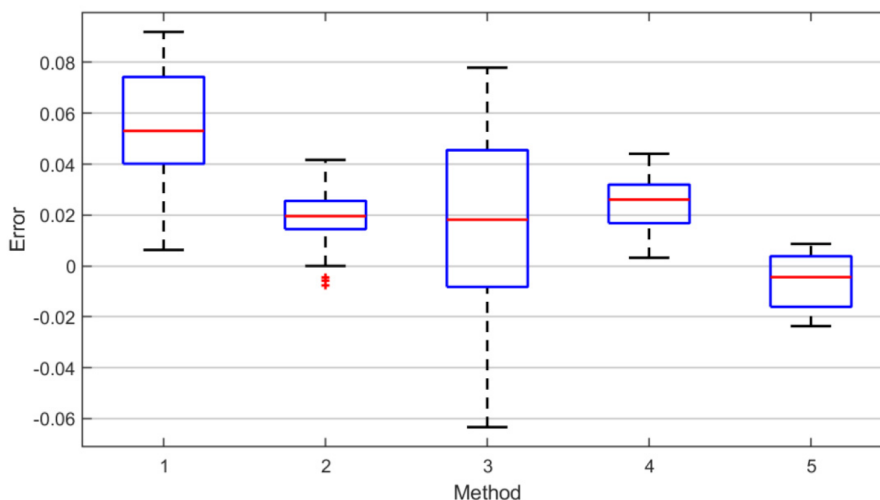
The PenSim v2.0 simulator developed by the research group of the Illinois Institute of Technology is widely used in many studies for performance evaluation. On the basis of the PenSim benchmark, our research group redeveloped the simulator in MATLAB/Simulink with the same kinetic model. The improved simulator allows users to customize the trajectories of manipulated variables freely, which brings about adequate flexibility. In this simulator, the penicillin concentration is regarded as the process quality and the key variable. Twelve other process variables as given in Table 1, including manipulated variables and state variables, are collected as the input of soft sensor models. In practical industry, it is important to predict the penicillin concentration according to these easy-to-measure variables to ensure the production safety and quality.

Table 2. Hyperparameters of the Penicillin Process by Different Methods

	LSTM	DCNN-LSTM	NARX	SLSTM	SDCNN-LSTM
maximum epochs	50	70	100	60	50
gradient threshold	6	6		6	6
sequence length	30	30		30	30
input delays			10		
minimum batch size	24	24		24	24
hidden layers	[50 20 100]	[50 20 100]	[60 30]	[50 20 100]	[50 20 100]



**Figure 10.** Penicillin concentration prediction results: (a) LSTM; (b) DCNN-LSTM; (c) NARX; (d) SLSTM; (e) SDCNN-LSTM.



**Figure 11.** Boxplot of the testing penicillin prediction errors by different methods: (1) LSTM; (2) DCNN-LSTM; (3) NARX; (4) SLSTM; (5) SDCNN-LSTM.

The total operation time of the penicillin fermentation process is 400 h, where the sampling interval of process variables is 1 h. Thus, 400 samples can be collected for one batch. The first 300 samples are collected as the training samples, while the remaining 100 samples are regarded as the testing samples. The trajectories of process variables are presented in Figure 8, where strong dynamic characteristics are involved in the process.

According to the process variables in Table 1, the dimensions of  $x(t)$  and  $y(t)$  of the SDCNN-LSTM network are 12 and 1, where 1 convolutional layer and 2 LSTM layers are constructed. In the convolutional layer, the filter size is set as [2 1] and the filter number is 15. The pool sizes of the max pooling layer and the average pooling layer are both set as [3 1]. The numbers of hidden units in each LSTM layer is set as [50 20 100]. The sequence length for training and prediction is set as 10. The prediction performance of the soft sensor can vary due to different training algorithms. In this work, the Adam algorithm is adopted to train the proposed network.

During the training procedure with the Adam algorithm, the value of the gradient threshold is 6 and the minimum batch size is 24. In addition, the number of maximum epochs has a great influence on the prediction result as well. The number of maximum epochs is selected as [10 20 30 40 50 60 70 80 90 100] for both the training data and testing data. The detailed prediction results of the proposed soft sensor under each epoch number are presented in Figure 9. It can be inferred that the best performance of quality prediction occurs under the circumstance of 50 maximum epochs, since the RMSEs of both the training data and testing data reach a low level. Hence, the number of the maximum epochs is determined to be 50 in this case.

Although the proposed soft sensor framework provides a promising result, it is still insufficient to prove its effectiveness. Therefore, prediction results of the penicillin concentration based on the LSTM, DCNN-LSTM, NARX, and SLSTM soft sensors were carried out as comparisons, where the hyper-



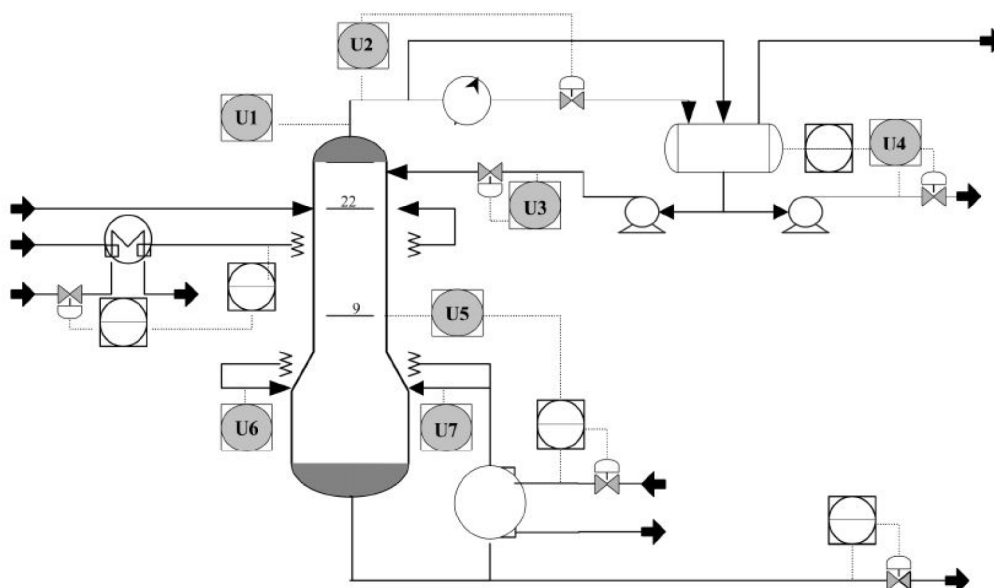


Figure 12. Block scheme of the debutanizer column.

Table 4. Process Variables of the Debutanizer Column

variable	description
$u_1$	top temperature
$u_2$	top pressure
$u_3$	reflux flow
$u_4$	flow to next process
$u_5$	sixth tray temperature
$u_6$	bottom temperature A
$u_7$	bottom temperature B

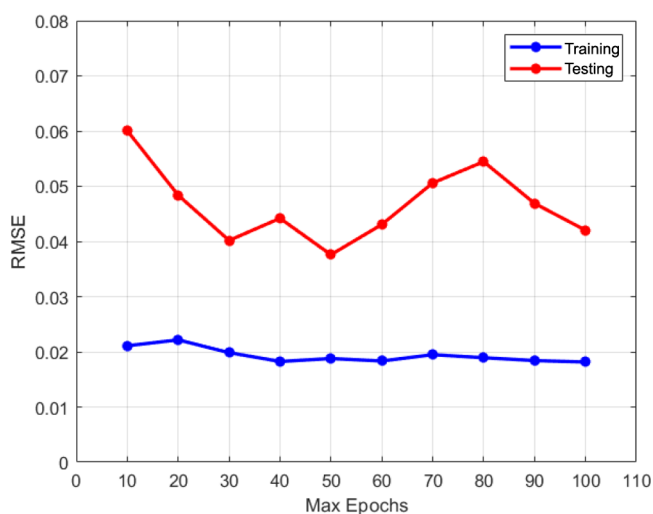


Figure 13. RMSEs of the debutanizer column with different maximum epochs.

parameters were determined by the trial-and-error technique as given in Table 2.

Table 3 shows the RMSEs of these methods with the same training data and testing data. Comparatively, the SDCNN-LSTM soft sensor provides smaller prediction errors and larger  $R^2$  values among these methods, which demonstrates the advantage of the proposed method over the existing methods. As complements, the detailed penicillin concentration

Table 5. Hyperparameters of the Penicillin Process by Different Methods

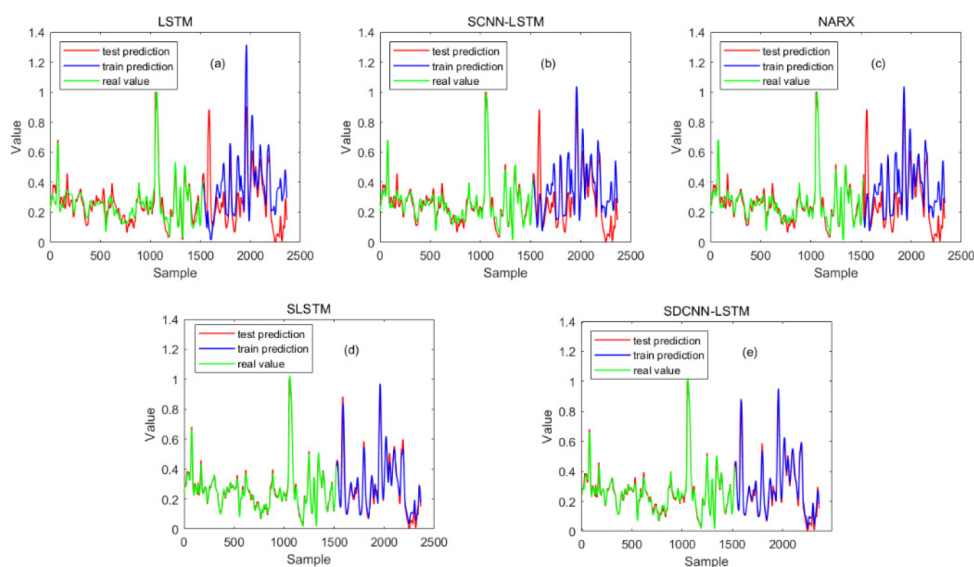
	LSTM	DCNN-LSTM	NARX	SLSTM	SDCNN-LSTM
maximum epochs	50	60	100	50	50
gradient threshold	2	2		2	2
sequence length	30	30		30	30
input delays			15		
minimum batch size	32	32		32	32
hidden layers	[80 50]	[80 50]	[40 30]	[80 50]	[80 50]

Table 6. RMSEs of the Debutanizer Column by Different Methods

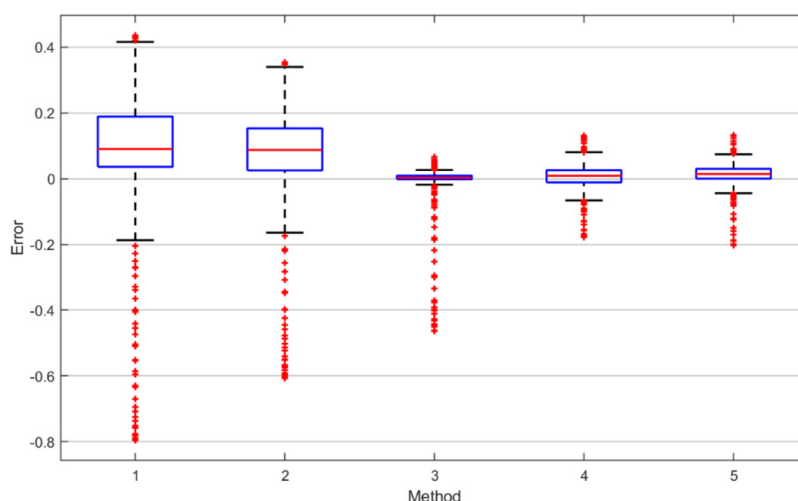
	LSTM	DCNN-LSTM	NARX	SLSTM	SDCNN-LSTM
RMSE	0.1907	0.1554	0.0620	0.0435	0.0376
$R^2$	0.4770	0.5337	0.8943	0.9556	0.9610

prediction results of these methods are presented in Figure 10 and a boxplot of the testing prediction error distributions by different methods is shown in Figure 11. It can be inferred from Figure 10 that the prediction trajectory of the SDCNN-LSTM soft sensor is able to track the real trajectory more precisely in comparison to the rest of methods. Furthermore, Figure 11 illustrates that the prediction errors of the proposed soft sensor (method 5) are much smaller since the median value is closer to zero. In addition, no exception value that exceeds the maximum or minimum threshold is found in the boxplot of the proposed method.

**4.2. Debutanizer Column.** The debutanizer column is an important part of the desulfuring and naphtha splitter plant, as shown in Figure 12. Propane and butane are removed in the debutanizer column from the top, while stabilized gasoline is separated in the bottom as well as the remaining part of butane. To obtain a good separation effect, the butane



**Figure 14.** Butane concentration prediction results: (a) LSTM; (b) DCNN-LSTM; (c) NARX; (d) SLSTM; (e) SDCNN-LSTM.



**Figure 15.** Boxplot of the testing butane prediction errors by different methods: (1) LSTM; (2) DCNN-LSTM; (3) NARX; (4) SLSTM; (5) SDCNN-LSTM.

concentration is required to be minimized in the bottom of the debutanizer column. Several sensors are installed around the debutanizer column, as shown by gray circles in Figure 12. Hence, it is necessary and feasible to predict the butane concentration with those easy-to-measure process variables for a further process optimization and control scheme.

The process variables collected by the physical sensors are given in Table 4, which are utilized as  $x(t)$  values of the virtual sensor. As was already mentioned, the butane concentration is regarded as the quality variable  $y(t)$ . Therefore, the dimensions of  $x(t)$  and  $y(t)$  are 7 and 1, respectively.

In total, 2394 samples are collected during the entire process, where the numbers of training samples and testing samples are 1556 and 838, respectively. One convolutional layer and 2 LSTM layers are constructed in this case. In the convolutional layer, the filter size is set as [2 2] and the filter number is 30. The pool sizes of the max pooling layer and the average pooling layer are both set as [3 3]. The numbers of hidden units in each LSTM layer is set as [80 50]. The sequence length for modeling is set as 30. The value of the

gradient threshold is 6, and the minimum batch size is 32 for the Adam algorithm. Similar to the first case, the number of maximum epochs is selected between 10 and 100 with an interval of 10 for both the training data and testing data. The prediction results of the SDCNN-LSTM soft sensor with diverse epoch numbers are provided in Figure 13. With reference to the curve, 50 maximum epochs are adopted in the debutanizer column case due to the smallest predicted RMSE of the butane concentration.

For comparison, quality prediction was carried out on the basis of LSTM, DCNN-LSTM, SLSTM, and the proposed SDCNN-LSTM soft sensors for the debutanizer column. The hyperparameters were also determined by the trial-and-error technique, as given in Table 5.

Table 6 displays the prediction results of each method. With respect to the prediction RMSEs of the training and testing data, the proposed method shows its merit with the smallest prediction error among all of the methods. Meanwhile, the detailed butane concentration prediction results are presented in Figure 14 with curves of prediction trajectories and the real

values. Intuitively, the prediction curve of the proposed method is more accurate from the perspective of the tight trajectory tracking. The results of the  $R^2$  calculation also indicate that the soft sensing modeling with the proposed method can describe better correlations in the residual space. Furthermore, a boxplot of the error distributions of the testing data by different methods is presented in Figure 15, which illustrates that the prediction result of the proposed SDCNN-LSTM soft sensor (method 5) has the ability to contribute great effort to the accurate prediction of the key variable with fewer large errors that exceed the boundary of the boxplot. In conclusion, the additional DCNN layer is able to extract the dynamic feature of the process more effectively in comparison to the original LSTM network. In addition, the supervised modeling framework significantly improves the prediction accuracy of the key variable.

## 5. CONCLUSION

In this paper, a hybrid supervised dynamic CNN-LSTM network is proposed to construct a soft sensor model for complex industrial processes with nonlinear and dynamic characteristics. In comparison to the traditional stacked LSTM network, the hybrid dynamic CNN-LSTM network is designed to implement data augmentation by expanding the original 1D samples into 2D feature maps, which makes the virtual sensor more efficient to cope with strong process dynamics. Furthermore, the quality variable is utilized as the labeled data to meet the demand of supervised modeling and prediction. The well-established supervised dynamic CNN-LSTM network is able to provide an accurate and reliable prediction result for nonlinear dynamic processes. Two applications, including a penicillin fermentation process and a debutanizer column case, were tested to evaluate the performance of the SDCNN-LSTM-based soft sensor. The experimental results in comparison with other soft sensor methods provide solid evidence of the effectiveness of the SDCNN-LSTM model. It is also noted that the determination of model parameters is crucial to the prediction performance of deep-learning-based soft sensor models. Therefore, future work will focus on the development of a general parameter optimization approach with the proposed soft sensor model to further improve the prediction performance.

## APPENDIX

The BPTT algorithm is described as

$$L = \sum_{t=1}^T (\hat{y}_t - y_t)^2 / N \quad (19)$$

$$\delta h_t = \frac{\partial L}{\partial h_t} + \delta i_{t+1} W_{hi}^T + \delta i_{f+1} W_{hf}^T + \delta o_{t+1} W_{ho}^T + \delta \tilde{c}_{t+1} W_{hc}^T \quad (20)$$

$$\delta i_t = (\delta h_t)^T \odot o_t \odot (1 - \tanh^2 c_t) \odot c_{t-1} \odot (i_t \cdot (1 - i_t)) \quad (21)$$

$$\delta f_t = (\delta h_t)^T \odot o_t \odot (1 - \tanh^2(c_t)) \odot \tilde{c}_t \odot (f_t \cdot (1 - f_t)) \quad (22)$$

$$\delta o_t = (\delta h_t)^T \odot \tanh c_t \odot (o_t \cdot (1 - o_t)) \quad (23)$$

$$\delta \tilde{c}_t = (\delta h_t)^T \odot o_t \odot (1 - \tanh^2 c_t) \odot i_t \tilde{c}_t (1 - \tanh^2 \tilde{c}_t) \quad (24)$$

$$\frac{\partial E}{\partial W_{x^*}} = \sum_{t=1}^T \langle \delta^*_t, x_t \rangle \quad (25)$$

$$\frac{\partial E}{\partial W_{h^*}} = \sum_{t=1}^T \langle \delta^*_t, h_{t-1} \rangle \quad (26)$$

$$\frac{\partial E}{\partial b_*} = \sum_{t=1}^T \delta^*_t \quad (27)$$

where  $L$  denotes the loss function,  $N$  is the number of the training samples, and  $*$  can be  $i$ ,  $f$ ,  $o$ , and  $\tilde{c}$ . Then, the model parameters can be estimated by gradient descent algorithms such as the Adam algorithm.

## AUTHOR INFORMATION

### Corresponding Author

Feifan Shen – School of Information Science and Engineering, NingboTech University, Ningbo 315100, People's Republic of China; [orcid.org/0000-0002-7086-6710](https://orcid.org/0000-0002-7086-6710); Email: [fshen@nbt.edu.cn](mailto:fshen@nbt.edu.cn)

### Authors

Jiaqi Zheng – College of Science & Technology, Ningbo University, Ningbo 315300, People's Republic of China

Lianwei Ma – School of Information Science and Engineering, NingboTech University, Ningbo 315100, People's Republic of China

Yi Wu – School of Information Science and Engineering, NingboTech University, Ningbo 315100, People's Republic of China

Lingjian Ye – School of Engineering, Huzhou University, Huzhou 313000, People's Republic of China; [orcid.org/0000-0001-8732-593X](https://orcid.org/0000-0001-8732-593X)

Complete contact information is available at:

<https://pubs.acs.org/10.1021/acsomega.2c01108>

### Notes

The authors declare no competing financial interest.

## ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (62103360, 61673349), the Natural Science Foundation of Zhejiang Province (LQ19F030004), the Teaching Reform Research Project of Zhejiang Province (jg20190607), and the K.C. Wong Magna Fund in Ningbo University.

## REFERENCES

- Jiang, Q.; Yan, X.; Huang, B. Review and Perspectives of Data-Driven Distributed Monitoring for Industrial Plant-Wide Processes. *Ind. Eng. Chem. Res.* **2019**, *58*, 12899–12912.
- Zhang, X.; Kano, M.; Li, Y. Locally weighted kernel partial least squares regression based on sparse nonlinear features for virtual sensing of nonlinear time-varying processes. *Comput. Chem. Eng.* **2017**, *104*, 164–171.
- Kong, X.; Ge, Z. Deep Learning of Latent Variable Models for Industrial Process Monitoring. *IEEE Trans. Ind. Informat.* **2021**, *1*.
- Sun, Q.; Ge, Z. Deep Learning for Industrial KPI Prediction: When Ensemble Learning Meets Semi-Supervised Data. *IEEE Trans. Ind. Informat.* **2021**, *17*, 260–269.
- Yuan, X.; Zhou, J.; Huang, B.; Wang, Y.; Yang, C.; Gui, W. Hierarchical Quality-Relevant Feature Representation for Soft Sensor

- Modeling: A Novel Deep Learning Strategy. *IEEE Trans. Ind. Informat.* **2020**, *16*, 3721–3730.
- (6) Sun, Q.; Ge, Z. A Survey on Deep Learning for Data-Driven Soft Sensors. *IEEE Trans. Ind. Informat.* **2021**, *17*, 5853–5866.
- (7) Cheng, T.; Harrou, F.; Sun, Y.; Leiknes, T. Monitoring influent measurements at water resource recovery facility using data-driven soft sensor approach. *IEEE Sens. J.* **2019**, *19*, 342–352.
- (8) Kim, S.; Okajima, R.; Kano, M.; Hasebe, S. Development of soft-sensor using locally weighted PLS with adaptive similarity measure. *Chemom. Intell. Lab. Syst.* **2013**, *124*, 43–49.
- (9) Wang, Z. X.; He, Q. P.; Wang, J. Comparison of variable selection methods for PLS-based soft sensor modeling. *J. Process Control* **2015**, *26*, 56–72.
- (10) Li, X.; Wu, F.; Zhang, R.; Gao, F. Nonlinear Multivariate Quality Prediction Based on OSC-SVM-PLS. *Ind. Eng. Chem. Res.* **2019**, *58*, 8154–8161.
- (11) Shen, F.; Ge, Z.; Song, Z. Multivariate trajectory-based local monitoring method for multiphase batch processes. *Ind. Eng. Chem. Res.* **2015**, *54*, 1313–1325.
- (12) Shah, D.; Wang, J.; He, Q. P. A feature-based soft sensor for spectroscopic data analysis. *J. Process Control* **2019**, *78*, 98–107.
- (13) Jin, H.; Chen, X.; Yang, J.; Wu, L. Adaptive soft sensor modeling framework based on just-in-time learning and kernel partial least squares regression for nonlinear multiphase batch processes. *Comput. Chem. Eng.* **2014**, *71*, 77–93.
- (14) Ge, Z.; Song, Z. A comparative study of just-in-time-learning based methods for online soft sensor modeling. *Chemom. Intell. Lab. Syst.* **2010**, *104*, 306–317.
- (15) Yu, J. A Bayesian inference based two-stage support vector regression framework for soft sensor development in batch bioprocesses. *Comput. Chem. Eng.* **2012**, *41*, 134–144.
- (16) Grbić, R.; Slišković, D.; Kadlec, P. Adaptive soft sensor for online prediction and process monitoring based on a mixture of Gaussian process models. *Comput. Chem. Eng.* **2013**, *58*, 84–97.
- (17) Jin, H.; Chen, X.; Wang, L.; Yang, K.; Wu, L. Adaptive soft sensor development based on online ensemble Gaussian process regression for nonlinear time-varying batch processes. *Ind. Eng. Chem. Res.* **2015**, *54*, 7320–7345.
- (18) Zhou, L.; Chen, J.; Song, Z. Recursive Gaussian process regression model for adaptive quality monitoring in batch processes. *Math. Probl. Eng.* **2015**, *2015*, 1.
- (19) Rani, A.; Singh, V.; Gupta, J. Development of soft sensor for neural network based control of distillation column. *ISA Trans.* **2013**, *52*, 438–449.
- (20) Souza, F. A.; Araújo, R.; Mendes, J. Review of soft sensor methods for regression applications. *Chemom. Intell. Lab. Syst.* **2016**, *152*, 69–79.
- (21) Shang, C.; Yang, F.; Huang, D.; Lyu, W. Data-driven soft sensor development based on deep learning technique. *J. Process Control* **2014**, *24*, 223–233.
- (22) Zhu, C.-H.; Zhang, J. Developing soft sensors for polymer melt index in an industrial polymerization process using deep belief networks. *International Journal of Automation and Computing* **2020**, *17*, 44–54.
- (23) Wang, Y.; Pan, Z.; Yuan, X.; Yang, C.; Gui, W. A novel deep learning based fault diagnosis approach for chemical process with extended deep belief network. *ISA Trans.* **2020**, *96*, 457–467.
- (24) Wang, G.; Jia, Q.-S.; Zhou, M.; Bi, J.; Qiao, J. Soft-sensing of wastewater treatment process via deep belief network with event-triggered learning. *Neurocomputing* **2021**, *436*, 103–113.
- (25) Yan, W.; Tang, D.; Lin, Y. A Data-Driven Soft Sensor Modeling Method Based on Deep Learning and its Application. *IEEE Trans. Ind. Electron.* **2017**, *64*, 4237–4245.
- (26) Yuan, X.; Huang, B.; Wang, Y.; Yang, C.; Gui, W. Deep Learning-Based Feature Representation and Its Application for Soft Sensor Modeling With Variable-Wise Weighted SAE. *IEEE Trans. Ind. Informat.* **2018**, *14*, 3235–3243.
- (27) Yuan, X.; Ou, C.; Wang, Y.; Yang, C.; Gui, W. A Layer-Wise Data Augmentation Strategy for Deep Learning Networks and Its Soft Sensor Application in an Industrial Hydrocracking Process. *IEEE Transactions on Neural Networks and Learning Systems* **2021**, *32*, 3296–3305.
- (28) Wang, K.; Shang, C.; Liu, L.; Jiang, Y.; Huang, D.; Yang, F. Dynamic soft sensor development based on convolutional neural networks. *Ind. Eng. Chem. Res.* **2019**, *58*, 11521–11531.
- (29) Jiang, X.; Ge, Z. Augmented multidimensional convolutional neural network for industrial soft sensing. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1.
- (30) Geng, Z.; Chen, Z.; Meng, Q.; Han, Y. Novel transformer based on gated convolutional neural network for dynamic soft sensor modeling of industrial processes. *IEEE Trans. Ind. Informat.* **2022**, *18*, 1521–1529.
- (31) Yuan, X.; Qi, S.; Wang, Y.; Xia, H. A dynamic CNN for nonlinear dynamic feature learning in soft sensor modeling of industrial process data. *Control Engineering Practice* **2020**, *104*, 104614.
- (32) Sun, Q.; Ge, Z. Gated stacked target-related autoencoder: A novel deep feature extraction and layerwise ensemble method for industrial soft sensor application. *IEEE Transactions on Cybernetics* **2020**, 1–12 Early Access.
- (33) Yuan, X.; Qi, S.; Shardt, Y. A.; Wang, Y.; Yang, C.; Gui, W. Soft sensor model for dynamic processes based on multichannel convolutional neural network. *Chemom. Intell. Lab. Syst.* **2020**, *203*, 104050.
- (34) He, Y.; Xu, Y.; Zhu, Q. Soft-sensing model development using PLSR-based dynamic extreme learning machine with an enhanced hidden layer. *Chemom. Intell. Lab. Syst.* **2016**, *154*, 101–111.
- (35) He, Y.; Tian, Y.; Xu, Y.; Zhu, Q. Novel soft sensor development using echo state network integrated with singular value decomposition: Application to complex chemical processes. *Chemom. Intell. Lab. Syst.* **2020**, *200*, 103981.
- (36) Kataria, G.; Singh, K. Recurrent neural network based soft sensor for monitoring and controlling a reactive distillation column. *Chem. Prod. Process Model.* **2018**, *13*, 13.
- (37) Ballesteros, M.; Dyer, C.; Smith, N. A. Improved Transition-Based Parsing by Modeling Characters instead of Words with LSTMs. *Computer Science* **2015**, *8*, No. e74515.
- (38) Yuan, X.; Li, L.; Wang, Y.; Yang, C.; Gui, W. Deep learning for quality prediction of nonlinear dynamic processes with variable attention-based long short-term memory network. *Can. J. Chem. Eng.* **2020**, *98*, 1377–1389.
- (39) Shen, F.; Zheng, J.; Ye, L.; Ma, X. LSTM Soft Sensor Development of Batch Processes with Multivariate Trajectory Based Ensemble Just-in-time Learning. *IEEE Access* **2020**, *8*, 73855–73864.
- (40) Fontes, D. O. L.; Vasconcelos, L. G. S.; Brito, R. P. Blast furnace hot metal temperature and silicon content prediction using soft sensor based on fuzzy C-means and exogenous nonlinear autoregressive models. *Comput. Chem. Eng.* **2020**, *141*, 107028.
- (41) Yuan, X.; Gu, Y.; Wang, Y. Supervised deep belief network for quality prediction in industrial processes. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1.
- (42) Yuan, X.; Li, L.; Wang, Y. Nonlinear dynamic soft sensor modeling with supervised long short-term memory network. *IEEE Trans. Ind. Informat.* **2020**, *16*, 3168–3176.